

卞 安 (@火云红孩儿)

李华明 (@Himi)

廖宇雷 (@廖宇雷\_cocoslua\_cn)

岳 阳 (@做游戏的老G)

屈光辉 (@子龙山人)

联/袂/推/荐

清華

开发者书库



Cocos2d-x Engine: the

gramming, Second Edition

# Cocos2d-x之Lua 核心编程

(第2版)

刘克男 ◎ 著

Liu Kenan

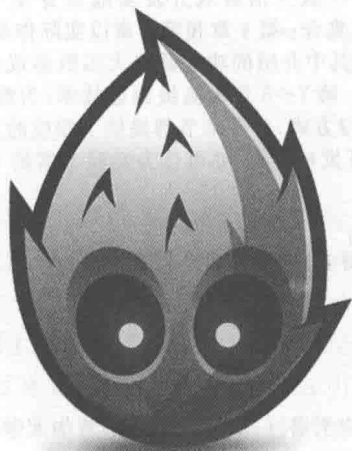
王哲 | 林顺

作 | 序 | 推 | 荐

清华大学出版社

清華

开发者书库



Cocos2d-x Engine: the Definitive Guide of Lua Programming, Second Edition

# Cocos2d-x之Lua 核心编程

【第2版】

刘克男 著

Liu Kenan

清华大学出版社

北京

## 内 容 简 介

本书在《Cocos2d-x 之 Lua 核心编程》(第1版)的基础上进行了内容扩充,增加了 Spine 骨骼动画、拆剪节点、渲染纹理、Lua Binding 实战、OpenGL ES 2.0 与 shader 编程和 SDK 接入等内容,全面而又系统地论述了 Cocos2d-x Lua 游戏开发的核心技术。本书从 Lua 语言基础开始,全面介绍 Cocos2d-x Lua 的基础知识、进阶应用和高级编程技术,并以一款三消游戏开发实战贯穿全书。第1~3章为基础知识,是使用 Cocos2d-x Lua 开发游戏的必备知识集合;第4章和第6章以实际作品“Fruit Fest”展示了完整的游戏开发流程;第5章为游戏开发进阶知识,其中介绍的功能是绝大多数游戏都需要的,熟知这些知识可以在开发游戏功能模块的时候更加得心应手;第7~8章为高级编程技术,为游戏产品发布阶段提供有力支撑。全书采用理论结合实践的最佳编程学习方式,各个章节都提供了相应的实例。

本书适合 Cocos2d-x Lua 游戏开发初学者,亦可作为经验丰富的 Cocos2d-x Lua 游戏前端开发工程师的参考读物。

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 之 Lua 核心编程/刘克男著. —2版. —北京:清华大学出版社,2017

(清华开发者书库)

ISBN 978-7-302-47314-5

I. ①C… II. ①刘… III. ①移动电话机—游戏程序—程序设计 IV. ①TN929.53

中国版本图书馆 CIP 数据核字(2017)第 124460 号

责任编辑:盛东亮

封面设计:李召霞

责任校对:徐俊伟

责任印制:沈 露

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](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

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

印装者:北京密云胶印厂

经 销:全国新华书店

开 本:186mm×240mm

印 张:18.25

字 数:432千字

版 次:2015年11月第1版

2017年8月第2版

印 次:2017年8月第1次印刷

印 数:1~2500

定 价:49.00元

产品编号:073889-01

# 推荐序(一)

## FOREWORD

最初得知刘克男要编写 Cocos 引擎的 Lua 教程的时候,我第一反应就是——太及时了!因为在目前 Cocos Lua 游戏盛行的市场环境下,相关的书籍却太少了;而另一方面,这种书由他们来写,很让人放心。

廖宇雷在 Cocos 引擎的基础上重写了 Lua 绑定和接口层代码,很大程度上解决了易用性问题,成为了 Quick-x 品牌。后来,廖宇雷加入了 Cocos 引擎团队,继续改良和维护 Lua 绑定,让有需要的开发者可以使用 Lua 这种相对简单易懂的脚本语言来开发游戏,从而提高开发效率。而本书的作者刘克男又和廖宇雷领导的 Cocos Lua 团队在触控科技公司成都办公室一起工作,一起讨论需求和技术问题,完善引擎文档,对外技术支持,参加各种技术沙龙,以及组织 Cocos 引擎的培训课程,因此没有比刘克男更合适写这本书了。

目前,市面上高质量的 Cocos C++ 与 JavaScript 的书籍已有不少,特别是 JavaScript 又新出版了几种。但是相比之下,介绍 Cocos Lua 的书籍实在太少,新书基本没有。然而,与书籍的缺乏相比,Cocos 社区的 Lua 游戏却长年持续繁荣。以 2015 年 8 月 8 日为例,AppStore 畅销榜 Top10 游戏中有 8 款游戏采用 Cocos 引擎,而其中竟然有 5 款都采用 Lua 方案!

除了《我是 MT》、《大掌门》等老牌游戏,就 2015 年而言雄踞榜单前列的多款 S 级游戏,例如《梦幻西游》、《刀塔传奇》、《开心消消乐》、《拳皇 98 终极之战 OL》、《放开那三国》、《奇迹暖暖》、《少年三国志》等,都是采用 Lua 方案在 Cocos 引擎上开发的。所以,你与“土豪”在技术上只有一个“Cocos 引擎+Lua 语言”的距离。

与传统语言相比,Lua 开发游戏讲求一个“快”字,因此在“天下武功唯快不破”的现在,开始得到越来越多开发者的重视。

本书委托清华大学出版社隆重出版,同时也是“触控教育游戏开发技术丛书”排头兵,可以说是扛鼎之作。全书内容涵盖了 Lua 基础、Lua 核心到 Cocos Lua 方案的系统论述,拆解得很合理,内容讲解与案例分析也很详细,非常值得一读。

欢迎大家跟着刘克男一起,用正确的方法学习 Cocos2d-x lua。

祝大家开发出属于自己的游戏,收获玩家与粉丝,向成功游戏大作的道路一步步迈进。

王 哲

# 推荐序(二)

## FOREWORD

2015 年 11 月,经历了一年多的精心准备,《Cocos2d-x 之 Lua 核心编程》第 1 版作为触控教育的第一本书籍正式出版,填补了当时 Cocos 社区 Lua 游戏开发方面书籍的空白,在 Cocos 社区里掀起了一波学习 Lua 开发的高潮。

该书第 1 版上市以来,Cocos 社区 Lua 游戏开发者的数量保持稳步增长,用 Lua 语言开发的 S 级 Cocos 游戏更是层出不穷,成为各个游戏畅销榜的常客。本书作为在 Lua 游戏开发品类名列前茅的高质量图书,深受开发者好评,成为 Cocos 社区里的畅销图书和触控公司合作高校的指定教材。

随着引擎技术的不断迭代,更多的新技术应用和工作流持续完善。作为社区的技术类畅销书籍作者,作者也感受到开发者希望本书同步更新的迫切愿望。

《Cocos2d-x 之 Lua 核心编程(第 2 版)》在原书的基础上进行了内容扩充,增加了 Spine 骨骼动画、裁剪节点、渲染纹理、Lua Binding 实战、OpenGL ES 2.0 与 Shader 编程、SDK 接入等内容。该书仍然保持第 1 版的风格,循序渐进,论述全面,每一章节配套有实例,帮助读者理解 Cocos2d-x 的核心知识点,并用实战案例带领读者畅游开发过程。该书适合作为新手的入门指导书,也可作为有经验的开发者的实践方法参考读物。

目前,Cocos 社区官方编辑器 Cocos Creator 已经支持 Lua 方案游戏开发的场景编辑器,可导出场景在 Lua 游戏项目中加载使用,进一步提高游戏开发效率。Lua 分支在 Cocos 社区中会持续发展壮大,这对于 Lua 开发者是一个重大的利好消息,本书的再版和其他 Lua 图书也会受到开发者的热烈欢迎。

很高兴有机会为本书第 2 版作序,希望新版图书可以更好地帮助各位即将开始进行游戏开发的读者快速入门;给正在使用 Lua 语言进行 Cocos2d-x 游戏开发的读者提供最佳实践参考,少走弯路,实现自己游戏开发的梦想。

林 顺

# 前言

## PREFACE

本书分 8 章,知识由浅入深、步步推进,建议按照顺序阅读。

第 1 章介绍 Cocos2d-Lua 背景,Cocos2d 各版本之间的关系。

第 2 章旨在引导读者快速入门,已有 Lua 编程基础的读者,可跳过这一部分。

第 3 章介绍 Cocos2d-Lua 基础,本章是开发游戏必须掌握的知识结构。

第 4 章介绍 Fruit Fest 消除游戏的第一部分,实现了游戏的核心玩法。本章应用 Cocos2d-Lua 基础知识,展示了游戏开发的主要流程。

第 5 章介绍 Cocos2d-Lua 进阶,涵盖 UI 控件、瓦片地图、内存管理以及数据存储等内容。

第 6 章介绍 Fruit Fest 游戏的第二部分,为游戏添加了音乐与粒子特效。

第 7 章介绍 Cocos2d-Lua 高级内容,虽然其中的网络通信和物理引擎不是每个游戏都会用到,但它们是引擎必不可少的一部分。自定义事件和状态机为游戏系统架构提供基础支撑,而 Lua Binding 要求熟练掌握 C 语言开发,它们都是为高级工程师准备的。

第 8 章介绍打包与发布。之所以把该部分放在最后,是因为 Cocos2d-Lua 提供的 Player 模拟器已足够应对游戏的前期开发,通过 Player 测试游戏将为开发者节省大量时间。

## 适用版本

本书基于 Quick-Cocos2dx-Community 3.6.4 版本撰写,由于 Quick 框架对接口稳定性所做的贡献,本书与引擎功能模块相关的章节适用于 Cocos2d-Lua v3.3 Final 之后的所有版本。可以在 <http://cocos2d-lua.org/download/index.md> 下载到最新的 Cocos2d-Lua 引擎。

---

注:本书不适用于 Cocos2d-Lua 2.x 分支版本。

---

## 书中实例

本书以理论结合实践,各章均配备了测试用例。其中第 4 章和第 6 章以 Fruit Fest 展示了完整游戏的开发流程,每一节各介绍一个独立的功能并配备实例代码,让初学者充分体

验到游戏开发的细节。

读者可以从本书的主页(<http://cocos2d-lua.org/book/index.md>)免费获取所有章节配套的最新实例代码。

## 读者对象

本书作为 Cocos2d-Lua 的权威书籍,从 Lua 语言基础开始,全面覆盖 Cocos2d-Lua 基础、进阶和高级编程,并指导读者逐步完成一款消除游戏的开发。

对于 Cocos2d 新手而言,本书可以作为手机游戏开发的入门书籍。

对于有经验的手游开发者而言,本书依然是进行 Cocos2d-Lua 开发必不可少的参考手册,大量的实例代码可以帮助读者节省宝贵的时间。

对于院校师生而言,本书的编排完全按照学习顺序系统展开,是教材的不二之选。

由于笔者水平有限,书中内容难免会有不妥或者疏漏,欢迎读者批评指正。

作者

2017年5月



# 目录

## CONTENTS

|                                    |    |
|------------------------------------|----|
| 第 1 章 Cocos2d-Lua .....            | 1  |
| 1.1 概述 .....                       | 1  |
| 1.1.1 Cocos2d 的起源 .....            | 1  |
| 1.1.2 Cocos2d-x 的诞生 .....          | 1  |
| 1.1.3 Cocos2d-Lua 的发展 .....        | 3  |
| 1.2 版本介绍 .....                     | 3  |
| 1.2.1 Cocos2d-x 版本介绍(C++) .....    | 3  |
| 1.2.2 Cocos2d-Lua 版本介绍 .....       | 4  |
| 第 2 章 Lua 编程 .....                 | 5  |
| 2.1 Lua 在 Windows 下的运行环境搭建 .....   | 5  |
| 2.1.1 安装 .....                     | 5  |
| 2.1.2 运行 .....                     | 5  |
| 2.2 Lua 基础 .....                   | 7  |
| 2.2.1 Lua 介绍 .....                 | 7  |
| 2.2.2 Lua 语法 .....                 | 8  |
| 2.3 Lua 进阶 .....                   | 18 |
| 2.3.1 Metatable(元表) .....          | 18 |
| 2.3.2 重要元方法介绍 .....                | 20 |
| 2.3.3 封装 .....                     | 21 |
| 2.3.4 继承 .....                     | 22 |
| 2.3.5 多态 .....                     | 22 |
| 第 3 章 Cocos2d-Lua 基础 .....         | 23 |
| 3.1 Cocos2d-Lua 开发环境配置 .....       | 23 |
| 3.1.1 安装 Cocos2d-Lua .....         | 23 |
| 3.1.2 安装 Sublime 与 QuickXDev ..... | 24 |
| 3.1.3 测试开发环境 .....                 | 28 |



|       |                            |    |
|-------|----------------------------|----|
| 3.2   | 引擎架构与目录结构                  | 29 |
| 3.2.1 | 引擎架构                       | 29 |
| 3.2.2 | 引擎文件结构                     | 30 |
| 3.2.3 | 项目文件结构                     | 32 |
| 3.3   | MVC 框架                     | 33 |
| 3.3.1 | 什么是 MVC                    | 33 |
| 3.3.2 | Cocos2d-Lua 中的类实现          | 34 |
| 3.3.3 | Cocos2d-Lua 的 MVC 实现       | 38 |
| 3.4   | 基础概念                       | 40 |
| 3.4.1 | 导演                         | 40 |
| 3.4.2 | 节点                         | 41 |
| 3.4.3 | 场景                         | 42 |
| 3.4.4 | 层                          | 43 |
| 3.4.5 | 精灵                         | 43 |
| 3.5   | 坐标系                        | 44 |
| 3.5.1 | 笛卡儿坐标系                     | 45 |
| 3.5.2 | 屏幕坐标系和 Cocos2d-Lua 坐标系     | 45 |
| 3.5.3 | 世界坐标系 VS 本地坐标系             | 45 |
| 3.5.4 | 锚点                         | 46 |
| 3.5.5 | 忽略锚点                       | 48 |
| 3.5.6 | ZOrder 与渲染顺序               | 48 |
| 3.6   | 文本标签                       | 50 |
| 3.6.1 | TTF 文本标签                   | 50 |
| 3.6.2 | BMFont 文本标签                | 51 |
| 3.6.3 | UILabel 控件                 | 52 |
| 3.6.4 | Mac 下使用 GlyphDesigner 制作字体 | 53 |
| 3.6.5 | Windows 下使用 BMFont 制作字体    | 54 |
| 3.7   | 按钮                         | 55 |
| 3.7.1 | UIButton 按钮                | 56 |
| 3.7.2 | UIPushButton 按钮            | 56 |
| 3.7.3 | UICheckBoxButton 按钮        | 58 |
| 3.7.4 | UICheckBoxButtonGroup 按钮   | 59 |
| 3.8   | 场景转换                       | 61 |
| 3.8.1 | 概念                         | 61 |
| 3.8.2 | 带转场特效的场景                   | 61 |
| 3.8.3 | 场景转换示例                     | 63 |
| 3.9   | 动作                         | 64 |

|        |                                   |     |
|--------|-----------------------------------|-----|
| 3.9.1  | 瞬时动作 .....                        | 64  |
| 3.9.2  | 有限时间动作 .....                      | 65  |
| 3.9.3  | 复合动作 .....                        | 67  |
| 3.9.4  | 变速动作 .....                        | 69  |
| 3.9.5  | 节点与动作相关的接口 .....                  | 70  |
| 3.10   | 序列帧动画 .....                       | 71  |
| 3.10.1 | 精灵表单 .....                        | 72  |
| 3.10.2 | 播放序列帧动画 .....                     | 73  |
| 3.10.3 | 动画缓存 .....                        | 75  |
| 3.11   | 调度器 .....                         | 76  |
| 3.11.1 | 全局调度器 .....                       | 76  |
| 3.11.2 | 节点调度器 .....                       | 78  |
| 3.12   | 事件分发机制 .....                      | 78  |
| 3.12.1 | 节点事件 .....                        | 79  |
| 3.12.2 | 帧事件 .....                         | 80  |
| 3.12.3 | 键盘事件 .....                        | 81  |
| 3.12.4 | 加速计事件 .....                       | 82  |
| 3.12.5 | 触摸事件 .....                        | 82  |
| 3.13   | 多分辨率适配 .....                      | 86  |
| 3.13.1 | Cocos2d-x 中的多分辨率适配 .....          | 86  |
| 3.13.2 | Cocos2d-Lua 中的多分辨率适配 .....        | 92  |
| 第 4 章  | 消除游戏实战(一) .....                   | 94  |
| 4.1    | Fruit Fest(1): 主场景 .....          | 94  |
| 4.1.1  | 游戏玩法 .....                        | 94  |
| 4.1.2  | 美术设计 .....                        | 94  |
| 4.1.3  | 分辨率适配策略 .....                     | 95  |
| 4.1.4  | 主场景 .....                         | 96  |
| 4.2    | Fruit Fest(2): 创建 PlayScene ..... | 97  |
| 4.2.1  | 准备 BMFont .....                   | 97  |
| 4.2.2  | 创建 PlayScene .....                | 97  |
| 4.2.3  | 添加转场代码 .....                      | 98  |
| 4.2.4  | 添加 UI .....                       | 98  |
| 4.3    | Fruit Fest(3): 初始化水果矩阵 .....      | 99  |
| 4.3.1  | 水果类 .....                         | 100 |
| 4.3.2  | 矩阵算法 .....                        | 102 |
| 4.3.3  | 掉落算法 .....                        | 102 |

|                           |                       |            |
|---------------------------|-----------------------|------------|
| 4.3.4                     | 矩阵初始化的代码实现            | 103        |
| 4.4                       | Fruit Fest(4): 选中水果   | 105        |
| 4.4.1                     | 绑定触摸事件                | 106        |
| 4.4.2                     | 清除已高亮区域               | 106        |
| 4.4.3                     | 高亮算法                  | 107        |
| 4.4.4                     | 分数算法                  | 108        |
| 4.5                       | Fruit Fest(5): 消除与掉落  | 109        |
| 4.5.1                     | 消除高亮水果                | 110        |
| 4.5.2                     | 掉落与补全                 | 111        |
| <b>第5章 Cocos2d-Lua 进阶</b> |                       | <b>113</b> |
| 5.1                       | UI 控件                 | 113        |
| 5.1.1                     | 输入控件                  | 113        |
| 5.1.2                     | 进度条控件                 | 115        |
| 5.1.3                     | 滑动条控件                 | 116        |
| 5.1.4                     | 滚动视图控件                | 119        |
| 5.1.5                     | 列表视图控件                | 121        |
| 5.1.6                     | 分页视图控件                | 125        |
| 5.2                       | 瓦片地图                  | 127        |
| 5.2.1                     | 用 Tiled 制作瓦片地图        | 128        |
| 5.2.2                     | 地图视角                  | 130        |
| 5.2.3                     | 在 Cocos2d-Lua 中使用 TMX | 131        |
| 5.3                       | 精灵批处理                 | 132        |
| 5.3.1                     | 手动批处理                 | 132        |
| 5.3.2                     | 自动批处理                 | 133        |
| 5.3.3                     | 手动批处理 VS 自动批处理        | 135        |
| 5.4                       | 碰撞检测                  | 135        |
| 5.4.1                     | 点与点的碰撞                | 135        |
| 5.4.2                     | 点与矩形的碰撞               | 135        |
| 5.4.3                     | 圆与圆之间的碰撞              | 136        |
| 5.4.4                     | 轴对齐矩形之间的碰撞            | 137        |
| 5.4.5                     | 非轴对齐矩形之间的碰撞           | 137        |
| 5.5                       | 内存管理                  | 138        |
| 5.5.1                     | Lua 内存管理              | 138        |
| 5.5.2                     | Cocos2d-Lua 内存管理      | 139        |
| 5.5.3                     | 纹理缓存                  | 141        |
| 5.5.4                     | 精灵帧缓存                 | 144        |

|       |                              |     |
|-------|------------------------------|-----|
| 5.6   | 数据与存储 .....                  | 145 |
| 5.6.1 | JSON 数据 .....                | 145 |
| 5.6.2 | crypto 数据编码 .....            | 146 |
| 5.6.3 | UserDefault 数据存储 .....       | 148 |
| 5.6.4 | GameState 数据存储 .....         | 149 |
| 5.6.5 | 文件读写 .....                   | 151 |
| 5.7   | 背景音乐与音效 .....                | 152 |
| 5.8   | 粒子系统 .....                   | 154 |
| 5.8.1 | 什么是粒子系统 .....                | 154 |
| 5.8.2 | Cocos2d-Lua 中的粒子系统 .....     | 154 |
| 5.8.3 | 粒子系统批处理节点 .....              | 155 |
| 5.8.4 | 粒子属性 .....                   | 156 |
| 5.8.5 | 粒子编辑器介绍 .....                | 158 |
| 5.8.6 | 使用粒子系统 .....                 | 161 |
| 5.9   | Spine 骨骼动画 .....             | 163 |
| 5.9.1 | Spine 运行库 .....              | 164 |
| 5.9.2 | 用法示例 .....                   | 166 |
| 5.10  | 裁剪节点 .....                   | 167 |
| 5.11  | 渲染纹理 .....                   | 169 |
| 第 6 章 | 消除游戏实战(二) .....              | 172 |
| 6.1   | Fruit Fest(6): 过关与信息存储 ..... | 172 |
| 6.1.1 | 添加进度条 .....                  | 172 |
| 6.1.2 | 过关与数据持久化 .....               | 173 |
| 6.2   | Fruit Fest(7): 爆炸特效与声音 ..... | 175 |
| 6.2.1 | 爆炸特效 .....                   | 175 |
| 6.2.2 | 游戏音乐 .....                   | 176 |
| 6.2.3 | 后记 .....                     | 178 |
| 第 7 章 | Cocos2d-Lua 高级 .....         | 179 |
| 7.1   | 网络通信 .....                   | 179 |
| 7.1.1 | network .....                | 179 |
| 7.1.2 | SocketTCP .....              | 182 |
| 7.1.3 | WebSocket .....              | 184 |
| 7.2   | 物理引擎 .....                   | 187 |
| 7.2.1 | 创建带物理世界的场景 .....             | 188 |
| 7.2.2 | 创建物理边界 .....                 | 188 |

|                    |                                |            |
|--------------------|--------------------------------|------------|
| 7.2.3              | 创建受重力作用的精灵                     | 189        |
| 7.2.4              | 碰撞检测                           | 189        |
| 7.2.5              | 完整示例                           | 191        |
| 7.3                | 摄像机                            | 193        |
| 7.3.1              | OpenGL 视口                      | 193        |
| 7.3.2              | cc.Camera                      | 194        |
| 7.3.3              | 场景的默认摄像机                       | 196        |
| 7.3.4              | 自定义摄像机                         | 197        |
| 7.4                | 自定义事件                          | 198        |
| 7.4.1              | 加载 EventProtocol 组件            | 198        |
| 7.4.2              | 主要接口                           | 201        |
| 7.4.3              | 完整示例                           | 201        |
| 7.5                | 有限状态机                          | 203        |
| 7.5.1              | 基本用法                           | 203        |
| 7.5.2              | 完整示例                           | 205        |
| 7.6                | 扩展 Lua 接口                      | 208        |
| 7.6.1              | Lua C API                      | 208        |
| 7.6.2              | Lua 栈                          | 209        |
| 7.6.3              | Lua Binding 实战(1): tolua++手动绑定 | 210        |
| 7.6.4              | Lua Binding 实战(2): 扩展 Spine 接口 | 217        |
| 7.7                | OpenGL ES 2.0 与 shader 编程      | 224        |
| 7.7.1              | OpenGL ES 概述                   | 224        |
| 7.7.2              | OpenGL ES 绘制几何图形               | 227        |
| 7.7.3              | 矩阵变换                           | 235        |
| 7.7.4              | 纹理贴图                           | 239        |
| 7.7.5              | Cocos2d-Lua 中使用自定义 shader      | 248        |
| <b>第 8 章 打包与发布</b> |                                | <b>254</b> |
| 8.1                | Android 项目的编译与打包               | 254        |
| 8.1.1              | 安装 JDK                         | 254        |
| 8.1.2              | Build Native                   | 255        |
| 8.1.3              | Android Studio 打包              | 258        |
| 8.2                | Mac 下编译 iOS 版本                 | 260        |
| 8.2.1              | 先决条件                           | 260        |
| 8.2.2              | Debug 真机调试                     | 260        |
| 8.2.3              | Release 打包                     | 262        |
| 8.3                | 发布时加密 Lua 源文件                  | 263        |

|       |                              |     |
|-------|------------------------------|-----|
| 8.3.1 | compile_scripts 脚本 .....     | 263 |
| 8.3.2 | 去掉源代码引用 .....                | 264 |
| 8.3.3 | 修改 Lua 启动入口 .....            | 265 |
| 8.4   | 发布时加密图片资源文件 .....            | 265 |
| 8.4.1 | encrypt_res 脚本 .....         | 265 |
| 8.4.2 | 资源加密示例 .....                 | 266 |
| 8.5   | SDK 接入 .....                 | 266 |
| 8.5.1 | 使用 luaj 接入 Android SDK ..... | 266 |
| 8.5.2 | 使用 luaoc 接入 iOS SDK .....    | 269 |

## 第 1 章



# Cocos2d-Lua

## 1.1 概述

Cocos2d-x 是一个跨平台开源游戏引擎,用它可以构建基于图形交互的跨平台游戏和应用。Cocos2d 作为 Cocos2d-x 以及其他分支的鼻祖,有其特别的历史故事。

### 1.1.1 Cocos2d 的起源

2005 年在阿根廷的一个叫 Los Cocos 的镇上,一个叫 Ricardo 的开发者(后来 Cocos2d-iPhone 的作者)和一群朋友计划每一个星期使用一种编程语言完成一个小游戏。在开发这些小游戏的过程中,他们发现做一个游戏引擎可以节省不少开发时间。2008 年他们借鉴之前做小游戏的经验,发现使用 python 作为开发语言需要的时间最少,所以用 python 开发了第一个版本引擎,并用 Ricardo 的家乡地名命名该引擎为 Los Cocos。一个月后(2008 年 3 月),团队发布了 0.1 版并将引擎更名为 Cocos2d。

在 Cocos2d 发布不久后,Apple 公司正式发布 AppStore 以及相关 SDK。此后大量使用 Objective-C 开发的 iOS 应用和游戏流行起来。这一年,Ricardo 使用 Objective-C 重写了 Cocos2d 引擎,并发布了 Cocos2d-iPhone 的第一个版本。

### 1.1.2 Cocos2d-x 的诞生

最开始,Cocos2d 没有跨平台的版本,推动 Cocos2d-x 诞生的重要因素在于 Android 的普及和国内对跨平台(iOS 和 Android)游戏开发的强烈需求。Android 的开放性使得国内产生一大批的 Android 智能手机,这时候急需一款简单易用的跨平台游戏引擎来解决游戏内容提供商的疾苦。Cocos2d-x 的作者王哲抓住了这个机遇,他发邮件给 Ricardo 表达了想衍生一款跨平台 Cocos2d 引擎的想法,出乎意料地,这个想法得到了 Ricardo 的大力支持并提供了很多技术支援,于是 Cocos2d-x 在王哲及其团队的努力下诞生了。

Cocos2d-x 1.x 和 2.x 与 Cocos2d-iPhone 的 1.x 和 2.x 在设计上没有区别,Cocos2d-x 用 C++ 把 Cocos2d-iPhone 重写了一遍,并加入了一些有用的跨平台库和接口,很多用法都很像 Objective-C 风格,一些初学 Cocos2d-x 的用户会觉得不适应。这时候的 Cocos2d-x 的



发展还跟不上 Cocos2d-iPhone,然而在 2013 年的苹果开发者大会上,这一切有了变化。苹果公布了自家的 SpriteKit 游戏框架,设计概念完全抄袭了 Cocos2d-iPhone,这种赤裸裸的剽窃激怒了 Ricardo。Ricardo 当即在社区宣布他将停止 Cocos2d-iPhone 的开发。随后 Ricardo 从 Zenga 公司跳槽到触控,正式转入 Cocos2d-x 的开发,于是全新的 Cocos2d-x 3.x 出现。

伴随着 Cocos2d 的成长,很多分支版本随之崛起,包括 Cocos2d-x。这些分支版本罗列如图 1-1 所示,不同的分支支持不同的开发语言和平台,如表 1-1 所示。

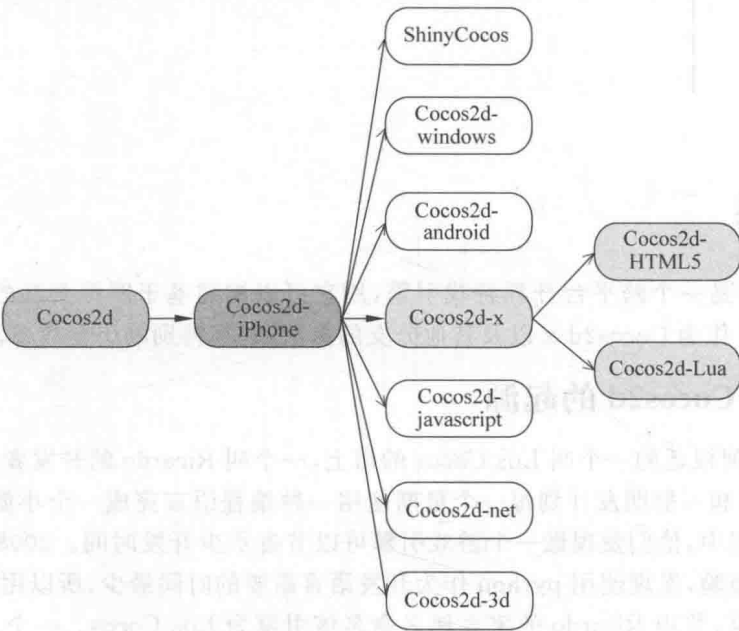


图 1-1 Cocos 家族

表 1-1 不同分支对应开发语言和平台

| 分支引擎               | 编程语言                   | 平台                                             |
|--------------------|------------------------|------------------------------------------------|
| cocos2d            | Python                 | 桌面操作系统                                         |
| Cocos2d-iPhone     | Objective-C,JavaScript | iOS                                            |
| Cocos2d-swift      | Swift                  | iOS                                            |
| Cocos2d-x          | C++、Lua、JavaScript     | 移动平台、桌面平台、浏览器                                  |
| Cocos2d-html5      | JavaScript             | 支持 HTML5 的浏览器                                  |
| ShinyCocos         | Ruby                   | iOS                                            |
| Cocos2d-windows    | C++                    | Windows XP/7                                   |
| Cocos2d-android    | Java                   | Android                                        |
| Cocos2d-javascript | JavaScript             | 支持 HTML5 的浏览器                                  |
| cocos-net          | C#                     | Mono-supported-Mono-Novell 公司开发的跨平台, .NET 运行环境 |
| cocos3d            | Objective-c            | iOS                                            |
| Cocos2d-Lua        | Lua                    | Android,iOS、桌面平台                               |

### 1.1.3 Cocos2d-Lua 的发展

2012年,由于Cocos2d-x中使用的C++对开发人员要求较高,所以网龙科技利用tolua++,将Cocos2d-x的C++接口转为Lua接口(这层C++与Lua之间的胶水代码叫Lua Binding)。开发者可以使用Lua这种简单易懂的脚本语言来编写游戏,从而极大地提高开发效率。

2012年上半年,廖宇雷的公司开始使用Cocos2d-x+Lua来开发游戏。但是发现当时Cocos2d-x对Lua的支持还存在相当多不完善的地方。所以他重写了整个Lua Binding代码,解决了内存泄漏和只能使用全局函数做回调等问题。

在Cocos2d-x 2.0发布后,Lua Binding又进行了不少改进和完善,截止到Cocos2d-x 2.1.4,整个Lua Binding已经可以说是相当稳定了。所以《我是MT》(月流水3000万人民币)、《大掌门》(月流水4000万人民币)这些赚钱像印钱的游戏,纷纷采用Cocos2d-x+Lua的解决方案。

自从2012年后,Cocos2d-x团队开发把重心放在了JavaScript脚本方案上,于是廖宇雷继续改进Lua代码并以Quick-Cocos2d-x为代号开发了一个新的开源引擎。2013年底,Quick-Cocos2d-x以一款千万级游戏《唐门世界》证明了引擎的实力,最终Quick-Cocos2d-x被Cocos2d-x并入主线,成为新的引擎版本Cocos2d-Lua,而廖宇雷也成为这个版本的核心开发者。2015年Cocos2d-x团队再次偏移开发重心到以JavaScript为主的Cocos Creator工具上,Quick-Cocos2d-x的后续版本由社区志愿者进行维护,并建设<http://cocos2d-lua.org/>网站,作为Quick-Cocos2dx-Community的社区网站。

Cocos2d-Lua的发展始终贯彻了三个目标:

- (1) 降低学习曲线;
- (2) 提高易用性;
- (3) 创建一个精简、但更容易扩展的架构。

## 1.2 版本介绍

### 1.2.1 Cocos2d-x 版本介绍(C++)

#### 1. Cocos2d-x 1.x

这个版本沿袭了Cocos2d-iPhone 1.x版本的架构与风格接口。引擎很多模块使用了iOS中的功能模块,不同的是这些模块都是由C++模拟实现的。此外,创建Xcode工程需要安装Cocos2d-x的项目模板。底层图形引擎使用的是OpenGL ES 1.0。

#### 2. Cocos2d-x 2.x

随着iPhone对OpenGL ES 2.0的支持,Cocos2d-iPhone发布了2.x版本,以支持OpenGL ES 2.0,并且Cocos2d-iPhone开始支持Lua和JS脚本开发游戏。

Cocos2d-x紧随其后发布2.x版本,但是它已不再只是局限于简单的Cocos2d-iPhone