

卞安 (@ 火云红孩儿)

李华明 (@ 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!**

**王哲先生作序!**



触控教育官方策划出版作品!

Cocos2d-x



之 Lua 核心编程

Cocos2d-x Engine: the Definitive Guide of Lua Programming

刘克男 杨雍 著

Liu Kenan Yang Yong

清华大学出版社



刘克男 杨雍 著  
Liu Kenan Yang Yong

Cocos2d-x



之 Lua 核心编程

Cocos2d-x Engine:

of Lua Programming

清华大学出版社

北京

## 内 容 简 介

本书系统论述了 Cocos2d-x Lua 开发的核心技术。本书从 Lua 语言基础开始,全面介绍了 Cocos2d-x Lua 的基础知识、基本应用和高级编程技术,并最终介绍了游戏的实战开发。全书贯穿理论结合实践的编写方式,各个章节都提供了配套实例。其中第 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。

本书适合作为从事移动开发、手机游戏开发,尤其是基于 Cocos2d-x 进行游戏开发的广大读者的案头工具图书。

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 之 Lua 核心编程/刘克男,杨雍著.--北京:清华大学出版社,2015

触控教育游戏开发技术丛书

ISBN 978-7-302-41471-1

I. ①C… II. ①刘… ②杨… III. ①移动电话机—游戏程序—程序设计 IV. ①TN929.53  
②TP311.5

中国版本图书馆 CIP 数据核字(2015)第 212280 号

责任编辑:盛东亮

封面设计:李召霞

责任校对:胡伟民

责任印制:杨 艳

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:北京嘉实印刷有限公司

经 销:全国新华书店

开 本:186mm×240mm

印 张:15.5

字 数:380 千字

版 次:2015 年 11 月第 1 版

印 次:2015 年 11 月第 1 次印刷

印 数:1~2500

定 价:39.00 元

---

产品编号:065033-01

# 序

## FOREWORD

最初得知杨雍和刘克男要编写 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,这本书由他们来写,很让人放心。

三年前,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。而两位作者杨雍、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,因此没有比杨雍和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 今年新出版了好几本。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。以今年 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,今年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。

本次这本《Cocos2d-x 之 Lua 核心编程》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。

欢迎大家跟着杨雍、刘克男一起,用正确的方法学习 Cocos2d-x lua。

也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

2015 年 9 月

# 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 背景及 Cocos2d 各版本之间的关系。

第 2 章:Lua 快速入门。已有 Lua 编程基础的读者,可跳过这一部分。

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

第 4 章:Fruit Fest 游戏第一部分,实现了游戏的核心玩法。本章包含了 Cocos2d-Lua 基础知识,展示了游戏开发的主要流程。

第 5 章:Cocos2d-Lua 进阶。涵盖 UI 控件、瓦片地图、内存管理及数据存储等内容。

第 6 章:Fruit Fest 游戏第二部分,为游戏添加了音乐与粒子特效。

第 7 章:Cocos2d-Lua 高级。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;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:打包与发布。之所以把该部分放在最后,是因为 Cocos2d-Lua 为我们提供的 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 Cocos2d-Lua v3.3 Final 版本撰写,由于 Quick 框架对接口稳定所作的贡献,本书与引擎功能模块相关的章节适用于 Cocos2d-Lua 3.x 所有版本;目录结构与环境搭建的相关章节适用于 Cocos2d-Lua v3.3 Final。

注:本书不适用于 Cocos2d-Lua 2.x 分支版本。

### 书中实例

本书理论结合实践,各章节均配备了测试用例,其中第 4 章和第 6 章以《Fruit Fest》展示了完整的游戏开发流程,每一节都介绍一个独立的功能并配备实例代码,可以让初学者充分体验到游戏开发的细节。

读者可以从泰然网本书主页(<http://www.tairan.com/cocos2d-lua-book/>)免费获取所有实例代码以及所使用到的资源。

### 读者对象

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

对于 Cocos2d 新手而言,本书可作为手机游戏开发的入门书籍。

对于有经验的开发者而言,本书依然是 Cocos2d-Lua 开发必不可少的参考手册,大量的实例代码可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。

刘克男为本书的主要作者,完成了书中绝大部分内容的编写。杨雍负责整体设计及编写思路。

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

2015 年 9 月

# 致谢

## ACKNOWLEDGEMENTS

---

首先,感谢 Cocos2d-x 的引擎团队,感谢他们为游戏开发者带来了如此优秀的引擎。

其次,感谢以下朋友对本书做出的贡献,他们让本书更加精彩:

感谢王帅,作为前 Cocos2d-lua 团队成员,他为本书提供了更加完整的基础知识体系,以及贡献了基础知识章节的初稿,让本书的知识体系更加完善。

感谢杨田波(博客地址: <http://www.zeroyang.com>),他贡献了本书 UI 控件等章节的初稿,他的努力使得本书的 UI 控件等章节显得与众不同。

感谢罗言,她作为本书唯一的美术人员,工作之余花费了大量的时间来制作《Fruit Fest》游戏资源,以及书中用到的其他精美图片。

感谢任珊(博客地址: <http://www.shannn.com/>),她对本书进行了文字校对工作,让本书更加严谨。

感谢查鑫(博客地址: <http://gameeer.com/>),他在本书定稿之后,对书中的每一行代码都进行了严格的测试,确保书中每一行代码正确性。

感谢摩多(厦门)科技有限公司授权使用《小小风暴》美术资源,让本书的封面看起来更加炫酷。

最后,感谢泰然网(<http://tairan.com/>)以及泰然教程组的朋友们,是他们的支持让我们一起快速成长,让本来枯燥的工作、学习变得如此生动有趣。

# 目 录

## CONTENTS

|                                         |     |
|-----------------------------------------|-----|
| 序 .....                                 | III |
| 前言 .....                                | V   |
| 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 Cocos-Lua 基础 .....                | 23  |
| 3.1 Windows 下的 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 .....       | 26 |
| 3.1.3 | 测试开发环境 .....                       | 28 |
| 3.2   | 引擎架构与目录结构 .....                    | 30 |
| 3.2.1 | 引擎架构 .....                         | 30 |
| 3.2.2 | 引擎文件结构 .....                       | 31 |
| 3.2.3 | 项目文件结构 .....                       | 33 |
| 3.3   | MVC 框架 .....                       | 34 |
| 3.3.1 | 什么是 MVC .....                      | 34 |
| 3.3.2 | Quick 中的类实现 .....                  | 36 |
| 3.3.3 | 父类为 function 或者 C++Object 的类 ..... | 36 |
| 3.3.4 | 父类为 Lua object .....               | 38 |
| 3.3.5 | 两种继承的异同 .....                      | 38 |
| 3.3.6 | Quick 的 MVC 实现 .....               | 39 |
| 3.4   | 基础概念 .....                         | 41 |
| 3.4.1 | 导演 .....                           | 42 |
| 3.4.2 | 节点 .....                           | 42 |
| 3.4.3 | 场景 .....                           | 43 |
| 3.4.4 | 层 .....                            | 44 |
| 3.4.5 | 精灵 .....                           | 44 |
| 3.5   | 坐标系 .....                          | 46 |
| 3.5.1 | 笛卡尔坐标系 .....                       | 46 |
| 3.5.2 | 屏幕坐标系 和 Cocos2d-Lua 坐标系 .....      | 46 |
| 3.5.3 | 世界坐标系 VS 本地坐标系 .....               | 47 |
| 3.5.4 | 锚点 .....                           | 47 |
| 3.5.5 | 忽略锚点 .....                         | 49 |
| 3.5.6 | ZOrder 与渲染顺序 .....                 | 50 |
| 3.6   | 文本标签 .....                         | 51 |
| 3.6.1 | TTF 文本标签 .....                     | 51 |
| 3.6.2 | BMFont 文本标签 .....                  | 52 |
| 3.6.3 | UILabel 控件 .....                   | 53 |
| 3.6.4 | Mac 下使用 Glyph Designer 制作字体 .....  | 54 |
| 3.6.5 | Windows 下使用 BMFont 制作字体 .....      | 55 |
| 3.7   | 按钮 .....                           | 57 |
| 3.7.1 | UIButton 按钮事件 .....                | 58 |
| 3.7.2 | UIPushButton 按钮 .....              | 58 |
| 3.7.3 | UICheckBoxButton 按钮 .....          | 59 |

|        |                                   |    |
|--------|-----------------------------------|----|
| 3.7.4  | UICheckBoxButtonGroup 按钮 .....    | 60 |
| 3.8    | 场景转换 .....                        | 62 |
| 3.8.1  | 概念 .....                          | 62 |
| 3.8.2  | 带转场特效的场景 .....                    | 63 |
| 3.8.3  | 场景转换示例 .....                      | 64 |
| 3.9    | 动作 .....                          | 65 |
| 3.9.1  | 瞬时动作 .....                        | 66 |
| 3.9.2  | 有限时间动作 .....                      | 67 |
| 3.9.3  | 复合动作 .....                        | 69 |
| 3.9.4  | 变速动作 .....                        | 71 |
| 3.9.5  | 节点与动作相关的接口 .....                  | 72 |
| 3.10   | 序列帧动画 .....                       | 73 |
| 3.10.1 | 精灵表单 .....                        | 73 |
| 3.10.2 | 播放序列帧动画 .....                     | 74 |
| 3.10.3 | 动画缓存 .....                        | 76 |
| 3.11   | 调度器 .....                         | 77 |
| 3.11.1 | 全局调度器 .....                       | 78 |
| 3.11.2 | 节点调度器 .....                       | 79 |
| 3.12   | 事件分发机制 .....                      | 80 |
| 3.12.1 | 节点事件 .....                        | 80 |
| 3.12.2 | 帧事件 .....                         | 82 |
| 3.12.3 | 键盘事件 .....                        | 83 |
| 3.12.4 | 加速计事件 .....                       | 83 |
| 3.12.5 | 触摸事件 .....                        | 83 |
| 3.13   | 多分辨率适配 .....                      | 87 |
| 3.13.1 | Cocos2d-x 多分辨率适配 .....            | 88 |
| 3.13.2 | Cocos2d-Lua 中的多分辨率适配 .....        | 93 |
| 第 4 章  | 消除游戏实战(一) .....                   | 96 |
| 4.1    | Fruit Fest(1): 主场景 .....          | 96 |
| 4.1.1  | 游戏玩法 .....                        | 96 |
| 4.1.2  | 美术设计 .....                        | 96 |
| 4.1.3  | 分辨率适配策略 .....                     | 97 |
| 4.1.4  | 主场景 .....                         | 98 |
| 4.2    | Fruit Fest(2): 创建 PlayScene ..... | 99 |
| 4.2.1  | 准备 BMFont .....                   | 99 |

|       |                              |     |
|-------|------------------------------|-----|
| 4.2.2 | 创建 PlayScene .....           | 100 |
| 4.2.3 | 添加转场代码 .....                 | 100 |
| 4.2.4 | 添加 UI .....                  | 100 |
| 4.3   | Fruit Fest(3): 初始化水果矩阵 ..... | 101 |
| 4.3.1 | 水果类 .....                    | 101 |
| 4.3.2 | 矩阵算法 .....                   | 104 |
| 4.3.3 | 掉落算法 .....                   | 105 |
| 4.3.4 | 矩阵初始化的代码实现 .....             | 105 |
| 4.4   | Fruit Fest(4): 选中水果 .....    | 107 |
| 4.4.1 | 绑定触摸事件 .....                 | 108 |
| 4.4.2 | 清除已高亮区域 .....                | 108 |
| 4.4.3 | 高亮算法 .....                   | 109 |
| 4.4.4 | 分数算法 .....                   | 110 |
| 4.5   | Fruit Fest(5): 消除与掉落 .....   | 111 |
| 4.5.1 | 消除高亮水果 .....                 | 111 |
| 4.5.2 | 掉落与补全 .....                  | 112 |
| 第 5 章 | Cocos2d-Lua 进阶 .....         | 115 |
| 5.1   | UI 控件 .....                  | 115 |
| 5.1.1 | 输入控件 .....                   | 115 |
| 5.1.2 | 进度条控件 .....                  | 117 |
| 5.1.3 | 滑动条控件 .....                  | 118 |
| 5.1.4 | 滚动视图控件 .....                 | 121 |
| 5.1.5 | 列表视图控件 .....                 | 123 |
| 5.1.6 | 分页视图控件 .....                 | 127 |
| 5.2   | 瓦片地图 .....                   | 130 |
| 5.2.1 | 用 Tiled 制作瓦片地图 .....         | 130 |
| 5.2.2 | 地图视角 .....                   | 133 |
| 5.2.3 | 在 Cocos2d-Lua 中使用 TMX .....  | 134 |
| 5.3   | 精灵批处理 .....                  | 135 |
| 5.3.1 | 手动批处理 .....                  | 135 |
| 5.3.2 | 自动批处理 .....                  | 136 |
| 5.3.3 | 手动批处理 VS 自动批处理 .....         | 137 |
| 5.4   | 碰撞检测 .....                   | 137 |
| 5.4.1 | 点与点的碰撞 .....                 | 138 |
| 5.4.2 | 点与矩形的碰撞 .....                | 138 |

|       |                        |     |
|-------|------------------------|-----|
| 5.4.3 | 圆与圆之间的碰撞               | 139 |
| 5.4.4 | 轴对齐矩形之间的碰撞             | 139 |
| 5.4.5 | 非轴对齐矩形之间的碰撞            | 140 |
| 5.5   | 内存管理                   | 140 |
| 5.5.1 | Lua 内存管理               | 140 |
| 5.5.2 | Cocos2d-Lua 内存管理       | 142 |
| 5.5.3 | 纹理缓存                   | 144 |
| 5.5.4 | 精灵帧缓存                  | 146 |
| 5.6   | 数据与存储                  | 148 |
| 5.6.1 | JSON 数据                | 148 |
| 5.6.2 | crypto 数据编码            | 149 |
| 5.6.3 | UserDefault 数据存储       | 150 |
| 5.6.4 | GameState 数据存储         | 152 |
| 5.6.5 | 文件读写                   | 154 |
| 5.7   | 背景音乐与音效                | 155 |
| 5.7.1 | 支持的平台与格式               | 155 |
| 5.7.2 | Quick 框架声音模块           | 155 |
| 5.8   | 粒子系统                   | 156 |
| 5.8.1 | 什么是粒子系统                | 156 |
| 5.8.2 | Cocos2d-Lua 中的粒子系统     | 157 |
| 5.8.3 | 粒子系统批处理节点              | 158 |
| 5.8.4 | 粒子属性                   | 158 |
| 5.8.5 | 粒子编辑器介绍                | 160 |
| 5.8.6 | 使用粒子系统                 | 164 |
| 第 6 章 | 消除游戏实战(二)              | 167 |
| 6.1   | Fruit Fest(6): 过关与信息存储 | 167 |
| 6.1.1 | 添加进度条                  | 167 |
| 6.1.2 | 过关与数据持久化               | 168 |
| 6.2   | Fruit Fest(7): 爆炸特效与声音 | 170 |
| 6.2.1 | 爆炸特效                   | 170 |
| 6.2.2 | 游戏音乐                   | 171 |
| 6.2.3 | 后记                     | 172 |
| 第 7 章 | Cocos2d-Lua 高级         | 173 |
| 7.1   | 网络通信                   | 173 |

|       |                                  |     |
|-------|----------------------------------|-----|
| 7.1.1 | network .....                    | 173 |
| 7.1.2 | SocketTCP .....                  | 176 |
| 7.1.3 | WebSocket .....                  | 178 |
| 7.2   | 物理引擎 .....                       | 181 |
| 7.2.1 | 创建带物理世界的 scene .....             | 181 |
| 7.2.2 | 创建物理边界 .....                     | 182 |
| 7.2.3 | 创建受重力作用的 sprite .....            | 183 |
| 7.2.4 | 碰撞检测 .....                       | 183 |
| 7.2.5 | 完整示例 .....                       | 185 |
| 7.3   | 摄像机 .....                        | 187 |
| 7.3.1 | OpenGL 视口 .....                  | 187 |
| 7.3.2 | cc.Camera .....                  | 189 |
| 7.3.3 | 场景的默认摄像机 .....                   | 191 |
| 7.3.4 | 自定义摄像机 .....                     | 192 |
| 7.4   | 自定义事件 .....                      | 193 |
| 7.4.1 | 加载 EventProtocol 组件 .....        | 193 |
| 7.4.2 | 主要接口 .....                       | 195 |
| 7.4.3 | 完整示例 .....                       | 196 |
| 7.5   | 有限状态机 .....                      | 197 |
| 7.5.1 | 基本用法 .....                       | 198 |
| 7.5.2 | 完整示例 .....                       | 200 |
| 7.6   | 扩展 Lua 接口 .....                  | 203 |
| 7.6.1 | Lua Binding .....                | 203 |
| 7.6.2 | Lua 栈 .....                      | 204 |
| 7.6.3 | tolua++ .....                    | 205 |
| 7.6.4 | Lua Binding 实战:扩展 spine 接口 ..... | 205 |
| 第 8 章 | 打包与发布 .....                      | 213 |
| 8.1   | Mac 下编译 Andriod 版本 .....         | 213 |
| 8.1.1 | 安装 JDK .....                     | 213 |
| 8.1.2 | 安装 ADT .....                     | 214 |
| 8.1.3 | 安装 NDK .....                     | 215 |
| 8.1.4 | 安装 Apache Ant .....              | 216 |
| 8.1.5 | 编译项目 .....                       | 216 |
| 8.2   | Mac 下编译 iOS 版本 .....             | 218 |
| 8.2.1 | 先决条件 .....                       | 218 |

|       |                              |     |
|-------|------------------------------|-----|
| 8.2.2 | Debug 真机调试.....              | 219 |
| 8.2.3 | Release 打包 .....             | 221 |
| 8.3   | Windows 下编译 Android 版本 ..... | 222 |
| 8.3.1 | 安装 JDK .....                 | 222 |
| 8.3.2 | 安装 ADT .....                 | 224 |
| 8.3.3 | 安装 NDK .....                 | 224 |
| 8.3.4 | 安装 Apache Ant .....          | 225 |
| 8.3.5 | 编译项目.....                    | 225 |

## 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,一名叫 Ricardo 的开发者(后来 Cocos2d-iPhone 的作者)和一群朋友进行了一件有趣的事情。他们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 年借鉴之前做小游戏的经验,他们发现使用 python 作为开发语言使用的时间最少,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 版本并将引擎更名为 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,国内对跨平台(iOS 和 Android)游戏开发的强烈需求。Android 的开放性,使得国内催生了一大批的 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 的作者王哲看到了这个机遇,他发邮件给 Ricardo 表达了想衍生一款跨平台的 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 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 在发展上还不如 Cocos2d-iPhone,然而就在 2013 年的苹果开发者大会上,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 到来了。

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。

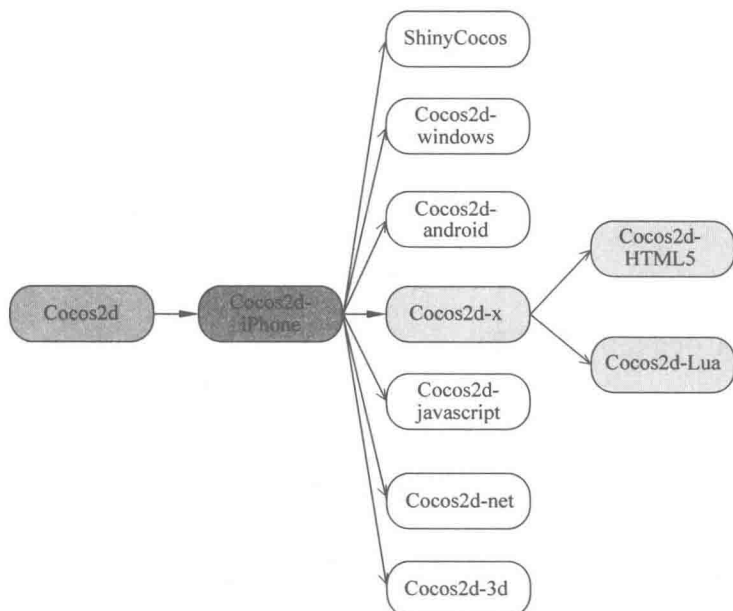


图 1-1 Cocos family

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++ 接口转为了 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,从而极大提高开发效率。

大概在 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,解决了内存泄露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 已经相当稳定,所以《我是 MT》(月流水 3000 万人民币)和《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 团队把开发重心放在了 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 fork 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。

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 的 C++ 版本,因为它有着更大的市场目标——Android。为了创建跨平台的项目,在没有脚