

Cocos2d-x 3.X.

游戏开发实战

肖文吉 编著

资深手游开发高手基于Cocos2d-x最新版本编写
适用Android、iOS两大平台
详细介绍Cocos Studio开发工具和最新UI控件
《萌仙》《保卫公主》等商业级游戏案例大揭秘



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

Cocos2d-x 3.X

游戏开发实战

电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书是一本介绍 Cocos2d-x 游戏引擎的实用图书,全面介绍了最新的 Cocos2d-x 3.2 游戏引擎各方面的知识。

本书从内容层次上可分为四个部分。第一部分介绍了游戏开发的基础知识、游戏引擎概念、Cocos2d-x 的下载与安装,以及跨平台开发环境的搭建。第二部分介绍了 Cocos2d-x 中的核心类、动作、动画、3D 特效、文字、字体、菜单、事件处理、UI 控件、声音和数据存储等知识。第三部分介绍了 Cocos2d-x 的粒子系统、Tile 地图、最新的物理引擎、内存管理机制。最后一部分详细介绍了 Cocos2d-x 官方提供的游戏开发工具 Cocos Studio。

本书针对每一个知识点都通过相应的程序给出了示范,并结合 Cocos2d-x 自带的 cpp-tests 实例进行讲解,在总结一些知识点之后还开发了一些小游戏,目的是让读者全面掌握 Cocos2d-x 的基础理论和基本使用。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

Cocos2d-x 3.x 游戏开发实战 / 肖文吉编著. — 北京: 电子工业出版社, 2015.1
ISBN 978-7-121-24689-0

I. ①C... II. ①肖... 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(2014)第 256589 号

责任编辑: 张月萍

特约编辑: 梁卫红

印 刷: 北京中新伟业印刷有限公司

装 订: 河北省三河市路通装订厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

开 本: 787×1092 1/16 印张: 35.25 字数: 925 千字 彩插: 2

版 次: 2015 年 1 月第 1 版

印 次: 2015 年 1 月第 1 次印刷

定 价: 88.00 元(含光盘 1 张)

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前言

移动互联网热潮在全世界引起了巨大反响，移动互联网正在改变着传统互联网的格局，全世界的 IT 公司争相将业务中心向移动互联网转移，移动互联网业务也成为业内最大的利润增长点。移动互联网时代的到来，使得移动应用成为人们生活中的一部分，而游戏，在移动应用中更是占有举足轻重的位置。

随着苹果 iOS 系统和 iPhone、iPad 等一系列产品的发布，随着 Android 系统和相应设备的诞生，移动平台从单一的手机发展到平板电脑等多种手持设备上。开发者们需要针对不同平台进行游戏的开发。Cocos2d-x 具有跨平台的特性，只需要编写一次代码，就可以无缝地部署在包括 iOS、Android、Windows Phone 等在内的主流游戏平台上。在移动终端日趋多样化的今天，把游戏部署到多种平台上是游戏开发的大趋势，Cocos2d-x 的跨平台特性无疑能够最大限度地减少游戏开发者重复开发的工作，从而大大提高开发者的开发效率。

Cocos2d-x 不仅具有跨平台的特点，它还有众多的社区支持，而且 Cocos2d-x 免费开源，简单易学，非常容易快速上手。现如今，Cocos2d-x 已发展为全球占有率第一的手机游戏引擎，全球份额超过 30%，中国份额超过 70%。《保卫萝卜》《找你妹》《捕鱼达人》《我叫 MT》《时空猎人》《全民英雄》《大掌门》等 60 余款月入千万的国内手游，以及国外的《Monster Strike》《勇者斗恶龙：超级光芒》《勇者前线》《Make it rain》《Badland》等多款精品手游，都是采用 Cocos2d-x 引擎开发的。

掌握了 Cocos2d-x 开发技术将带来更多的就业机会与创业机会，这都值得当下的开发者好好把握。

本书特点

本书是一本介绍 Cocos2d-x 游戏引擎的实用图书，其全面介绍了最新的 Cocos2d-x 3.2 游戏引擎各方面的知识。

本书从内容层次上可分为四个部分。

第一部分（第 1~2 章）为准备篇，为读者介绍了游戏开发的基础知识、游戏引擎概念、Cocos2d-x 的下载与安装以及跨平台开发环境的搭建。

第二部分（第3~9章）为基础篇，全面、系统地介绍了 Cocos2d-x 中的核心类、动作、动画、3D 特效、文字、字体、菜单、事件处理、UI 控件、声音和数据存储等知识。

第三部分（第10~15章）为高级篇，在前面章节的基础上，介绍了 Cocos2d-x 的粒子系统、Tile 地图以及最新的物理引擎。第14章重点分析了 Cocos2d-x 的内存管理机制，帮助读者在游戏开发中更好地管理内存。

第四部分（第16章）为游戏开发工具篇，详细介绍了 Cocos2d-x 官方提供的游戏开发工具 Cocos Studio，重点介绍了 UI 编辑器和动画编辑器的用法，并给出了示例。

本书针对每一个知识点都给出了示范程序，并结合 Cocos2d-x 自带的 cpp-tests 实例进行讲解，在总结了一些知识点之后还开发了一些小游戏，目的是让读者全面掌握 Cocos2d-x 的基础理论和基本使用。

本书第9章、第12章、第15章为游戏实战项目。第9章是一个射击类游戏《疯狂打飞机》；第12章是正在运营的真实手游《萌仙》中的一个模块；第15章是一个塔防类游戏《保卫公主》。这些游戏实战项目都具有极高的参考价值，希望读者充分利用本书的实战案例，真正步入 Cocos2d-x 游戏开发的殿堂。

本书并不是一本关于所谓“思想”的书，也没有一堆“深奥”的新名词和“高深”的思想，旨在让你学会实际的 Cocos2d-x 游戏开发。本书的特点是操作步骤详细，编程思路清晰，语言平易近人。只要读者认真阅读本书，并掌握书中知识，就完全可以胜任 Cocos2d-x 手机游戏开发的工作。

阅读本书需要具备一定的计算机知识以及编程功底。Cocos2d-x 采用 C++ 编写，熟练掌握 C++ 语言对于学习本书是很有必要的。限于篇幅，本书并没有涉及 C++ 语言的知识，请读者参考相关的书籍和文档。

本书是针对 Cocos2d-x 游戏开发者撰写的，无论是游戏开发爱好者、初学者，还是有一定经验的游戏开发者，都十分适合阅读本书。

- 对于初学者，如果你完全没有学过 C++，建议遵从学习规律，循序渐进，先阅读 C++ 相关书籍，学习 C++ 语法，再从第1章开始阅读。
- 对于有 C++ 基础但是没有接触过 Cocos2d-x 引擎的读者，请从第1章开始阅读。
- 对于从事过游戏开发，使用过 Cocos2d-x，包括使用过其他 Cocos2d 引擎的读者，可以从第3章开始阅读。

本书适合的读者

- Cocos2d-x 初级及中级开发者
- 了解 C++ 和游戏开发的读者
- 没有接触过 Cocos2d-x，但有过 Cocos2d 其他版本开发经验的开发者
- 大专院校相关专业的师生
- 喜欢游戏并怀揣梦想的有志青年

- 智能手机设备的用户，想制作一款自娱自乐的游戏产品
- 具备其他平台游戏开发经验的人

光盘说明

1. 光盘内容

光盘中的代码按章节存放，即第 2 章所使用的代码放在 `codes\02` 文件夹下，依此类推。

另外，书中每份源代码也给出了与光盘源文件的对应关系，方便读者查找。

本光盘中有的 `codes` 文件夹下有 15 个文件夹，其内容和含义说明如下：

(1) 文件夹名 02~16 对应于书中的章号，即第 2 章所使用的代码放在 02 文件夹下，依此类推。

(2) Cocos2d-x 3.0 之后创建新项目的同时会将 Cocos2d-x 所需的所有库文件都复制到新项目中，这样做的好处是简化项目配置环境，也就是说不管谁拿到这个项目都可以跑起来，不需要重新去改项目的配置。弊端就是新建一个项目需要更多的磁盘空间，最少 200MB。

由于本书的实例过多，出于光盘大小的考虑，所以针对每个项目只提供了 `Classes` 和 `Resources` 文件夹，`Classes` 中是项目的代码，`Resources` 中是项目所需的资源文件，包括图片、声音和配置文件等。读者需要运行光盘程序时，可先新建一个 Cocos2d-x 的项目（创建项目的具体步骤在第 2 章当中有详细介绍），然后将光盘中对应的 `Classes` 和 `Resources` 中的文件复制到项目中就可以运行了。

2. 运行环境

本书中的全部代码可运行在 Xcode 5.x、Visual Studio 2012 和 Eclipse 中，并通过 iPhone 4、iPhone 4S、iPhone 5、iPhone 5S 和部分 Android 4.x 系统真机测试。

衷心感谢

衷心感谢李刚老师，他是一位非常好的朋友，在本书的创作过程中，他提供了大量切实、有用的帮助。同时也感谢疯狂软件教育中心技术团队一贯的支持。

一直以来，很想感谢我的妻子王丹娜女士，她给了我生活上、精神上无微不至的照顾，并承担几乎所有的日常家务，使我能够心无旁骛地专注于程序的研究和书籍的撰写。

感谢所有参加疯狂软件实训的学生，他们是本书的“试验品”，他们所付出的努力证明了本书的价值，他们的反馈让本书更加实用。

肖文吉

2014 年 9 月

目 录

第 1 章 Cocos2d-x 游戏引擎介绍	1
1.1 什么是游戏引擎	2
1.2 Cocos2d 游戏引擎	2
1.2.1 Cocos2d 的发展	2
1.2.2 Cocos2d 的特点	4
1.2.3 Cocos2d 的现在和未来	4
1.3 Cocos2d-x 游戏引擎	4
1.3.1 选择 Cocos2d-x 的理由	5
1.3.2 Cocos2d-x 的主要功能	6
1.3.3 Cocos2d-x 3.0 版本的各语言优化	7
1.3.4 Cocos2d-x 3.0 的新特性	8
1.3.5 使用了 Cocos2d-x 的游戏	10
1.4 本章小结	11
第 2 章 跨平台开发环境的搭建	12
2.1 搭建 iOS 下的开发环境	13
2.1.1 Cocos2d-x 的安装与配置	14
2.1.2 Cocos2d-x 项目结构和代码分析	15
2.2 搭建 Windows 下的开发环境	25
2.2.1 Cocos2d-x 的安装与配置	25
2.2.2 Cocos2d-x 工程目录和代码分析	27
2.3 搭建 Android 下的开发环境	29
2.3.1 软件的安装与环境变量配置	30
2.3.2 构建 Cocos2d-x 工程	31
2.3.3 在 Eclipse 下运行程序	32
2.4 本章小结	34
第 3 章 Cocos2d-x 的核心类	35
3.1 节点类 (Node)	36
3.1.1 Node 类的成员变量	36
3.1.2 Node 类的函数	38
3.1.3 坐标系	41

示例：测试锚点——AnchorPointTest	43
示例：测试 Node 类的常用函数——NodeTest	45
3.2 场景类 (Scene)	48
3.2.1 TransitionScene 场景切换	48
3.2.2 Cocos2d-x 支持的场景过渡效果	48
3.3 层类 (Layer)	50
3.3.1 Layer 类的常用成员变量	50
3.3.2 Layer 类的常用函数	50
3.4 导演类 (Director)	51
3.4.1 Director 导演类常用的成员变量	51
3.4.2 Director 导演类常用的函数	52
3.4.3 Director 导演类在游戏初始化时的使用分析	53
3.5 精灵类 (Sprite)	54
3.5.1 纹理的相关概念	54
3.5.2 Sprite 精灵类的常用成员变量	55
3.5.3 Sprite 精灵类的常用函数	55
示例：测试 Sprite 类的常用函数——SpriteTest	57
3.5.4 SpriteFrame 精灵帧类	59
3.5.5 SpriteFrameCache 精灵帧缓存类	60
3.5.6 SpriteBatchNode 精灵表单类	60
示例：测试 SpriteFrameCacheTest 精灵帧缓存类——SpriteFrameCacheTest	62
3.5.7 制作精灵表单	64
示例：测试自己制作的精灵表单——SpriteSheetTest	67
3.6 Cocos2d-x 3.0 新的数据结构	68
3.6.1 cocos2d::Vector<T>	69
示例：测试 cocos2d::Vector<T>类的常用函数——VectorTest	70
3.6.2 cocos2d::Map<K,V>	73
示例：测试 cocos2d::Map<K,V>类的常用函数——MapTest	74
3.6.3 cocos2d::Value	76
示例：测试 cocos2d::Value 类的常用函数——ValueTest	77
3.7 本章小结	79
第 4 章 Cocos2d-x 的动作和动画	80
4.1 动作基类 Action	81
4.1.1 Action 类的常用成员变量	82
4.1.2 Action 类的常用函数	82
4.1.3 Action 类的子类	82
4.2 示例：测试基本动作——ActionBasicTest	84
4.2.1 显示、隐藏、闪烁动作	85
4.2.2 淡入淡出动作	87
4.2.3 色彩值渐变动作	89
4.2.4 移动动作	92
4.2.5 跳跃动作	95
4.2.6 缩放动作	98
4.2.7 旋转动作	101
4.3 示例：测试组合动作——ActionCombinationTest	104
4.4 示例：测试可变速度动作——ActionEaseTest	108
4.4.1 EaseIn、EaseOut 和 EaseInOut	109

4.4.2	EaseSineIn、EaseSineOut 和 EaseSineInOut.....	111
4.4.3	EaseExponentialIn、EaseExponentialOut 和 EaseExponentialInOut.....	111
4.4.4	EaseBounceIn、EaseBounceOut 和 EaseBounceInOut.....	112
4.4.5	EaseElasticIn、EaseElasticOut 和 EaseElasticInOut.....	112
4.5	示例: 测试速度动作——ActionSpeedTest.....	112
4.6	示例: 测试跟随动作——ActionFollowTest.....	114
4.7	示例: 测试延迟动作——ActionDelayTest.....	116
4.8	示例: 测试函数回调动作——ActionCallFuncTest.....	117
4.9	Cocos2d-x 中与动画相关的类.....	122
4.10	示例: 英雄舞剑动画.....	124
4.11	示例: 僵尸走路动画.....	125
4.12	示例: 使用动画帧缓冲实现动画.....	127
4.13	示例: 制作精灵表单实现动画.....	129
4.14	Cocos2d-x 新增的 3D 特性 fbv-conv 的用法.....	132
4.15	示例: Sprite3D 创建 3D 精灵.....	133
4.16	示例: Animation3D 创建 3D 动画.....	135
4.17	本章小结.....	137
第 5 章 Cocos2d-x 的文本渲染系统和菜单.....		138
5.1	文本渲染相关的类.....	138
5.2	TTF 类型标签 LabelTTF.....	139
	示例: 测试 LabelTTF 类——LabelTTFTest.....	139
5.3	BMFont 标签类 LabelBMFont.....	141
	示例: 测试 LabelBMFont 类——LabelBMFontTest.....	143
5.4	Atlas 标签类 LabelAtlas.....	144
	示例: 测试 LabelAtlas 类——LabelAtlasTest.....	145
5.5	Cocos2d-x 新增的标签类 Label.....	147
	示例: 测试 Label 类——LabelTest.....	148
5.6	菜单类 Menu.....	150
5.7	菜单项 MenuItem 和常用子类.....	151
5.7.1	标签菜单项 (MenuItemLabel).....	151
5.7.2	自定义字体菜单项 (MenuItemAtlasFont).....	152
5.7.3	字体菜单项 (MenuItemFont).....	152
5.7.4	精灵菜单项 (MenuItemSprite).....	152
5.7.5	图片菜单项 (MenuItemImage).....	152
5.7.6	触发器菜单项 (MenuItemToggle).....	153
	示例: 测试菜单——MenuTest.....	153
5.8	本章小结.....	157
第 6 章 Cocos2d-x 的事件处理机制.....		158
6.1	事件处理机制.....	158
6.1.1	接收数据 (Touch).....	159
6.1.2	分发事件 (EventDispatcher).....	160
6.1.3	处理响应 (EventListener).....	161
6.2	触摸事件.....	162
6.2.1	单点触摸事件 (EventListenerTouchOneByOne).....	162

示例：场景优先事件监听——TouchOneByOneTest	162
示例：滑动屏幕选择菜单——TouchOneByOneTest	167
示例：固定值优先事件监听——TouchPriorityTest	172
6.2.2 多点触摸事件（EventListenerTouchAllAtOnce）	176
示例：多点触摸（缩放功能的实现）——TouchAllAtOnceTest	176
6.3 加速度计事件（EventListenerAcceleration）	180
示例：滚动的小球——AccelerometerTest	180
6.4 按键事件（EventListenerKeyboard）	184
6.5 鼠标事件（EventListenerMouse）	184
6.6 自定义事件（EventListenerCustom）	185
示例：自定义事件处理——CustomTest	186
6.7 本章小结	188
第 7 章 Cocos2d-x 的常用控件	189
7.1 Cocos2d-x 的控件	189
7.1.1 按钮类 ControlButton	191
示例：测试按钮——ControlButtonTest	192
7.1.2 拖动滑块类 ControlSlider	196
示例：测试拖动滑块——ControlSliderTest	196
7.1.3 开关按钮类 ControlSwitch	198
示例：测试开关按钮——ControlSwitchTest	199
7.2 Cocos Studio 中的 UI 控件	201
7.2.1 Widget（所有 UI 控件的父类）	202
7.2.2 Button（按钮）	205
示例：测试 UI 控件——UITest	205
7.2.3 Text（文本）	211
7.2.4 TextField（文本框）	213
7.2.5 CheckBox（复选框）	217
7.2.6 ImageView（图片）	219
7.2.7 LoadingBar（进度条）	221
7.2.8 Slider（滑动条）	223
7.2.9 Layout（布局容器）	225
7.2.10 ScrollView（滚动层容器）	231
7.2.11 ListView（列表层容器）	235
7.2.12 PageView（分页层容器）	238
7.3 本章小结	241
第 8 章 Cocos2d-x 的声音系统和数据存储	242
8.1 声音系统	242
8.1.1 背景音乐和声音特效	242
8.1.2 CocosDenshion 库	243
8.1.3 SimpleAudioEngine 类	243
示例：背景音乐和音效控制——SimpleAudioEngineTest	245
8.2 数据存储	255
8.2.1 UserDefault 类	255
示例：用户数据存储测试——UserDefaultTest	257
8.3 本章小结	260

第 9 章 射击类游戏：疯狂打飞机	261
9.1 游戏简介	261
9.2 添加游戏菜单	262
9.3 预加载游戏资源	268
9.4 玩家飞机飞行效果	273
9.5 玩家飞机 Touch 实现	276
9.6 背景滚动效果	278
9.7 添加敌机	280
9.8 玩家飞机发射子弹	284
9.9 背景音乐和积分统计	288
9.10 游戏大 Boss	290
9.11 本章小结	296
第 10 章 粒子系统	297
10.1 粒子系统相关的类	297
10.1.1 ParticleSystem 类	297
10.1.2 ParticleSystemQuad 类	303
10.1.3 ParticleBatchNode 类	303
10.2 粒子系统的使用	303
示例：使用内置的粒子系统——ParticleEffectTest	305
示例：自己定制粒子系统——MyParticleTest	306
10.3 使用 Particle Designer 工具制作粒子	310
示例：使用 plist 文件创建粒子系统——ParticleDesignerTest	312
示例：休闲类游戏——疯狂打地鼠	313
10.4 本章小结	337
第 11 章 瓦片地图	338
11.1 下载和安装 Tiled 地图编辑器	338
11.2 使用 Tiled 绘制地图	339
11.3 Tiled 地图相关的类	345
11.3.1 TMXTiledMap 类	345
11.3.2 TMXLayer 类	346
11.3.3 TMXObjectGroup 类	347
示例：使用 Tiled 地图——TileMapTest	348
11.4 本章小结	353
第 12 章 真实手游：萌仙	354
12.1 设计地图场景	354
12.2 载入地图	355
12.3 地图的移动	358
12.4 遮挡效果和设置障碍物	362
12.5 寻路算法	365
12.6 随机出现怪物	375
12.7 玩家和怪物进行战斗	380

12.8 本章小结	388
第 13 章 Cocos2d-x 的物理引擎	389
13.1 物理引擎概述	389
13.2 Box2D 的核心概念	390
13.3 Box2D 中常用的属性和函数	393
13.4 Box2D 示例项目分析	394
13.5 Cocos2d-x 和 Box2D 的整合开发	402
13.6 Box2D 开发步骤	404
13.7 Cocos2d-x 3.0 中的物理引擎	405
13.7.1 PhysicsWorld 类	405
13.7.2 PhysicsBody 类	406
13.7.3 PhysicsShape 类	408
13.8 使用 3.0 的物理引擎开发项目	409
示例: 测试 3.0 新增的物理引擎——PhysicsTest	409
13.9 益智类游戏: 愤怒的小鸟	412
13.9.1 开始前的准备工作	412
13.9.2 进度条制作	412
13.9.3 游戏中的粒子效果	415
13.9.4 游戏选关操作	420
13.9.5 设计关卡数据	425
13.9.6 设计精灵类	428
13.9.7 游戏主界面	431
13.9.8 弹弓发射小鸟	435
13.9.9 物理引擎	441
13.9.10 物理碰撞	445
13.9.11 游戏过关设计	448
13.10 本章小结	452
第 14 章 Cocos2d-x 的内存管理	454
14.1 内存管理概述	454
14.2 手动内存管理	455
14.2.1 对象的引用计数	455
14.2.2 对象所属权	458
14.2.3 函数中的保留和释放	461
14.2.4 使用自动释放池	462
14.2.5 手动内存管理总结	465
14.3 自动释放池	465
14.3.1 Ref 的引用计数	465
14.3.2 AutoreleasePool 自动释放池	468
14.3.3 PoolManager 自动释放池管理器	471
14.3.4 create 函数	474
14.4 本章小结	476
第 15 章 塔防类游戏: 保卫公主	477
15.1 设计关卡地图场景	477

15.2	游戏菜单	479
15.3	选择关卡	482
15.4	加载关卡数据	489
15.5	准备进入游戏	497
15.6	随机出现怪物	503
15.7	建造炮塔	508
15.8	旋转炮塔	511
15.9	检测碰撞	515
15.10	游戏结束	517
15.11	本章小结	520
第 16 章 Cocos Studio		521
16.1	Cocos Studio 概述	521
16.2	Cocos Studio 安装	521
16.3	Cocos Studio 特性	523
16.4	Cocos Studio 界面	523
16.4.1	界面布局	523
16.4.2	起始页面	524
16.4.3	渲染面板	524
16.4.4	动画帧面板	525
16.4.5	属性面板	526
16.4.6	资源面板	526
16.4.7	控件面板	527
16.4.8	快捷工具面板	527
16.5	UI 编辑器的使用	528
示例: 测试 UI 编辑器——CocosStudioTest		528
16.5.1	按钮和文本框控件	528
16.5.2	进度条控件	535
16.5.3	滑动条控件	537
16.5.4	层容器控件	539
16.5.5	滚动层容器控件	540
16.5.6	分页层容器控件	543
16.5.7	UI 动画制作	545
16.6	本章小结	549
附录 Cocos2d-x 中常用的宏		550

第 1 章

Cocos2d-x 游戏引擎介绍

本章要点

游戏引擎的介绍

Cocos2d 的发展

Cocos2d-x 的主要功能

Cocos2d-x 3.0 版本新增功能

移动互联网热潮在全世界引起了巨大反响，移动互联网正在改变着传统互联网的格局，全世界的 IT 公司争相将业务中心向移动互联网转移，移动互联网业务也成为业内最大的利润增长点。移动互联网时代的到来，使得移动应用成为人们生活中的一部分，而游戏，在移动应用中更是占有着举足轻重的位置。

随着科技的发展，移动设备的功能也越来越多，越来越强大。而手机游戏也远远不是我们印象中的什么“俄罗斯方块”、“贪吃蛇”之类画面简陋、规则简单的游戏，其已发展到了可以和掌上游戏机相媲美，具有很强的娱乐性和交互性。一旦某款手游产品反响火爆，立即就会引来众多效仿者。

随着智能手机和 3G、4G 网络普及的进一步深化，移动互联网已经迎来爆发式增长。业界普遍认为，移动互联网是一个比互联网大数倍的万亿级产业，手机等移动设备将作为未来人类线上生活娱乐的最主要载体而存在。

在此大背景下，手机游戏也普遍被业界所看好。如今，移动设备的游戏市场规模已经发展到几十甚至上百亿元，仅次于客户端网游和网页游戏。盛大游戏、腾讯、阿里巴巴等业界巨头已经高调进驻手机游戏行业。在各家投入巨资和布下重兵之后，这个行业还将会有一轮大跃进式的发展。

因而，如何能够更快更好地开发出游戏产品，显得更加重要。而在游戏开发中，游戏引擎可以起到提高开发效率、提高游戏品质的关键性作用。

1.1 什么是游戏引擎

游戏引擎是指一些已经编写好的可编辑电脑游戏系统或者一些交互式实时图像应用程序的核心组件。这些系统为游戏设计者提供各种编写游戏所需的各种工具，其目的在于让游戏设计者能容易和快速地做出游戏程序而不用由零开始。游戏引擎包含以下系统：渲染引擎（即“渲染器”，含二维图像引擎和三维图像引擎）、物理引擎、碰撞检测系统、音效、脚本引擎、电脑动画、人工智能、网络引擎以及场景管理等。

我们可以把游戏的引擎比作汽车的引擎，大家知道，引擎是汽车的心脏，决定着汽车的性能和稳定性，汽车的速度、操纵感这些指标都是建立在引擎的基础之上的。游戏引擎和汽车引擎在概念上是一样的，都是驱动整体运转的核心部件。游戏中玩家所体验到的剧情、关卡、美工、音乐、操作等内容都是由游戏的引擎直接控制的，它扮演着中场发动机的角色，把游戏中的所有元素捆绑在一起，在后台指挥它们同时、有序地工作。简单地说，引擎就是，“用于控制所有游戏功能的主程序，从计算碰撞、物理系统和物体的相对位置，到接受玩家的输入，以及按照正确的音量输出声音，等等。”由此可见，游戏产品的核心就是引擎，它是每款游戏的运行基础。引擎的好坏直接影响着游戏的品质。

游戏引擎相当于游戏的框架，框架搭好后，只要往里面填充内容就可以了。因此，在游戏的过程中，引擎的制作往往会占用非常多的时间。例如，马克思·佩恩的 MAX-FX 引擎从最初的雏形 Final Reality 到最终的成品共花了四年多时间，LithTech 引擎的开发共花了整整五年时间，耗资 700 万美元。Monolith 公司（LithTech 引擎的开发者）的老板詹森·霍尔甚至不无懊悔地说：“如果当初意识到制作自己的引擎要付出这么大的代价的话，我们根本就不可能去做这种傻事。没有人会预料得到 5 年后的市场究竟是怎样的。”正是出于节约成本、缩短周期和降低风险这三方面的考虑，越来越多的开发者倾向于使用第三方的现成引擎制作自己的游戏。

1.2 Cocos2d 游戏引擎

Cocos2d 是一个基于 MIT 协议（一种开源软件许可协议）的开源框架，用于构建游戏、应用程序和其他图形界面交互应用。它可以让你在创建自己的多平台游戏时节省很多的时间。Cocos2d 可以说是现在移动设备中使用最多、功能最强大的游戏引擎之一。

1.2.1 Cocos2d 的发展

2008 年 3 月，Ricardo 发布了 Cocos2d 游戏引擎的第一个版本，最初的版本定位于平面游戏框架，使用的开发语言为 Python 语言。因为引擎原创团队当时在阿根廷 Córdoba 市附近的 Los Cocos，所以游戏引擎命名为 Los Cocos。一个月后 Los Cocos 引擎发布了 0.1 版，其名称从 Los Cocos 更名为 Cocos2d。

2008 年，苹果公司在乔布斯的带领下进军手机移动市场，开始发展 iPhone 市场。Cocos2d 团队抓住这个机会，在 2008 年 6 月宣布与 iPhone 平台进行接轨，并在当月就公布了用 Objective-C 编写的 Cocos2d for iPhone 0.1 版，它与 Python 版的 Cocos2d 拥有相同的设计思路。随着 iOS 设备市场占有率不断上升，iPhone 游戏产品也成为用户最喜爱的应用。截止到 2008 年 12 月，App Store

上已有超过 40 个用 Cocos2d 引擎开发的游戏。

到了 2010 年, Cocos2d 游戏引擎在游戏开发者中大受欢迎。许多开发者第一次接触 Cocos2d 引擎, 都是 iPhone 平台的版本。之后, 使用 Cocos2d 开发的游戏《StickWars》在 App Store 美国区付费榜夺得冠军, Cocos2d 进入了可商用阶段, 这宣告 Cocos2d 游戏引擎的时代已经到来。与此同时, 英国设计大师 Michael Heald 为 Cocos2d 设计了新的标志, 如图 1.1 所示 (Cocos2d 最早的标志左图是一个奔跑的椰子, 右图是 Michael Heald 之后设计的 Cocos2d-x 的标志)。

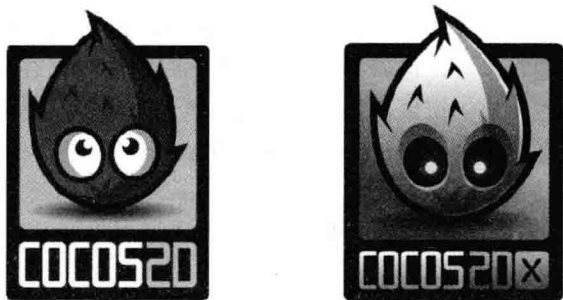


图 1.1 Cocos2d 的标志和 Cocos2d-x 的标志

凭借用户对 iPhone 设备的热衷, Cocos2d-iPhone 也变成开发者的首选, 可以说 Cocos2d-iPhone 版本的诞生是 Cocos2d 引擎发展的一个标志性事件, 它是引擎体系中的第一个明星。随后不久, 更具历史意义的 Cocos2d-x 诞生了, “x”意味着 Cross, 表示“交叉”的意思。该版本开启了跨平台游戏开发的时代。开发者只需要使用 C++语言编写一个游戏逻辑, 就可以在 iOS、Android 和其他移动平台上运行。

提示: 令人感到骄傲和自豪的是, Cocos2d-x 游戏引擎是由中国国内的开发者王哲和他的开发团队制作完成的。

同一时期, 美国人用 C#改写出了 CocosNet, 新西兰人 Ryan Williams 用 JavaScript 改写出了 HTML5 的版本, Cocos2d 社区也出现了 Texture Packer、Glyph Designer 等丰富的工具和编辑器, 以及大量的书籍教程、示例代码、培训视频等供开发者学习和使用。图 2.1 所示为 Cocos2d 引擎的发展及其分支。

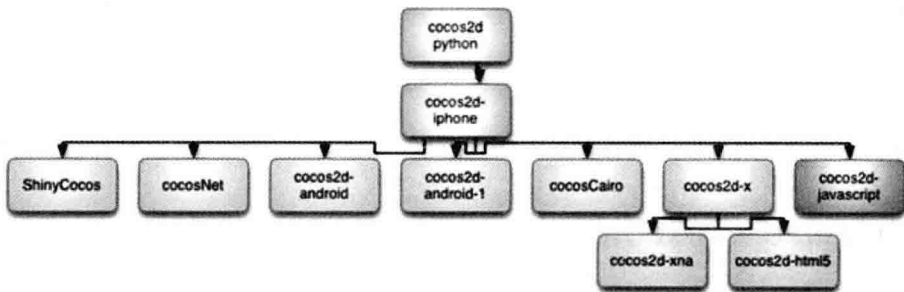


图 1.2 Cocos2d 引擎的发展及其分支

1.2.2 Cocos2d 的特点

Cocos2d 之所以被游戏开发者广泛使用是因为，它在游戏开发方面具有很多实用的功能和特点。

- **易于使用**: Cocos2d 是基于 OpenGL (Open Graphics Library, 开放图形库) 和 OpenES (OpenGL for Embedded Systems, 嵌入式系统开放图形库) 的, 因此游戏开发者使用 Cocos2d 完成诸如图片缩放、旋转和动画等游戏功能时, 无须关心底层是如何实现的。同时由于它是开源的, 开发者也可以使用 OpenGL 的知识来修改或生成新的游戏对象。也就是说, 对于初学者, 可以不需要花很多的时间去研究 OpenGL 的相关知识。而对于资深开发者, 也可以对引擎进行修改, 使之更适合自己的使用。Cocos2d 中还集成了目前比较流行的两款物理引擎 (Box2D 和 Chipmunk), 这大大方便了游戏开发者对物理引擎的使用, 提高了开发效率。一句话, Cocos2d 使游戏开发者可以只把关注焦点放在游戏设置本身, 而不必消耗大量时间学习晦涩难懂的 OpenGL 和 OpenES。此外, Cocos2D 还提供了大量的规范。
- **高效**: Cocos2d 基于 OpenGL 和 OpenES 进行图形渲染, 从而让移动设备的 GPU (Graphic Processing Unit, 图形处理器) 性能发挥到极致。
- **灵活**: 方便扩展, 易于集成第三方库。
- **免费**: 基于 MIT 协议的免费开源框架, 用户可以放心使用, 不用担心商业授权的问题。
- **社区支持**: Cocos2d 的开发者自发建立了多个社区组织, 用户可以方便地查阅各类技术资料, 而用户在开发过程中遇到的问题, 也会很快得到解决。

1.2.3 Cocos2d 的现在和未来

目前, Cocos2d 也拥有了更加稳定的引擎, 包括 Cocos2d-iPhone、Cocos2d-x, 以及被社区普遍看好的 Cocos2d-HTML5 和 Cocos2d-JavaScript。同时也拥有了非常优秀的编辑器(独立编辑器), 例如 SpriteSheet Editors (精灵表单编辑器)、Particle Editors (粒子效果编辑器)、Font Editors (字体编辑器) 和 Tilemap Editors (地图编辑器), Cocos Studio 集成编辑器已更新至 v1.6 版本, 其在优化体验的同时增加了大量新功能、新亮点, 包括 UI 编辑器、动画编辑器、场景编辑器和数据编辑器。UI 编辑器和动画编辑器主要面向美术, 而场景编辑器和数据编辑器则面向游戏策划, 这四个工具合在一起构成了一套完整的游戏开发体系, 帮助开发者进一步降低开发难度, 提高开发效率, 减少开发成本。

在各个平台的应用商店里, 到处可见大量基于 Cocos2d 引擎开发的游戏。当前, Cocos2d 绝对是移动平台领域应用最广、使用最多的游戏引擎, 相信将来 Cocos2d 引擎也会有更好的发展。

1.3 Cocos2d-x 游戏引擎

Cocos2d-x 是 Cocos2d 家族系列的一个重要分支, 它是由中国国内的开发者王哲和他的开发团队使用 C++ 语言将 Cocos2d 进行改写而完成的, 是一个在 MIT 许可证下发布的、支持多平台的开源游戏引擎, 基于 OpenGL ES 和 Cocos2d-iPhone, 游戏开发快速、简易、功能强大。Cocos2d-x