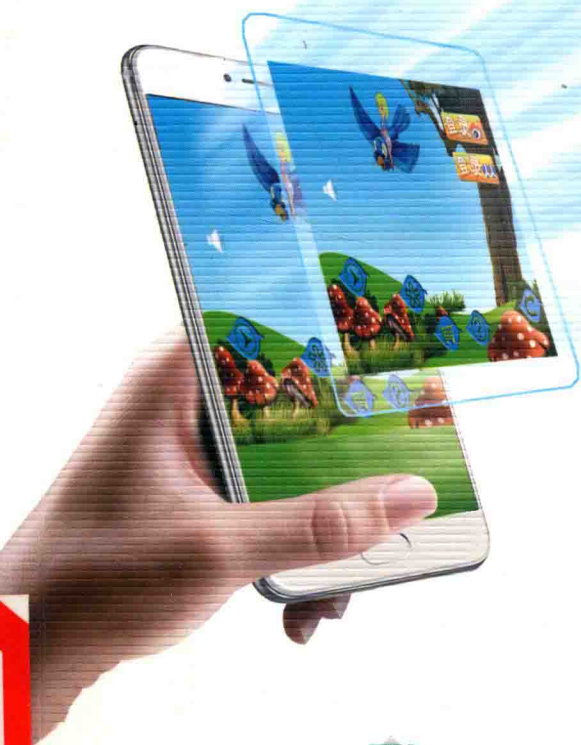




Cocos 实战: 2d-x 3.x

卡牌手游 开发指南

朱先忠 高嗣慧 编著



173

Cocos2d - x 3. x 实战： 卡牌手游开发指南

朱先忠 高嗣慧 编著

電子工業出版社·

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书使用 Cocos2d-x 引擎 (3.9 版本) 技术, 带领读者从零开始逐步进行“拇指接龙”手游 (手机游戏) 的开发。此游戏基于 Windows 平台知名扑克牌游戏“空当接龙”开发而成, 其中融入了世界闻名童话故事《拇指姑娘》一书的主要情节。读者可以通过学习这个游戏的开发过程, 全面掌握 Cocos2d-x C++ 游戏跨平台实战开发的主要方法和技巧。

本书遵循理论和实践相结合的编写原则, 并尽量按照软件工程指导思想, 循序渐进地介绍“拇指接龙”游戏项目的开发过程。通过一款完整的跨平台 (Android & iOS) 实战型手游的开发学习, 读者能够从中领悟 Cocos2d-x 引擎的神奇魅力, 从而更加深入地理解和掌握该引擎。

本书共分为 3 篇, 17 章。在第 1 篇中, 我们简要介绍进行 Cocos2d-x 编程所必需的基础知识。在第 2 篇中, 我们正式转入“拇指接龙”游戏的实战开发, 在这一部分知识的编写中, 我们尽量遵循软件开发思路展开介绍, 适当穿插 Cocos2d-x 实战技巧的讲解。在第 3 篇中, 我们转而关注基于 Cocos2d-x 实战手游跨平台开发、移植和第三方市场 SDK 嵌入乃至弱联网相关技术的探讨。通过这 3 篇内容的渐进性学习, 力图使读者以最快的速度掌握基于 Cocos2d-x 引擎进行跨平台手游开发的关键技术。

本书适合具有一定 C/C++ 开发经验的着眼于 Cocos2d-x 引擎的游戏开发初学者、游戏设计人员阅读, 也适合作为高等院校和培训学校相关专业师生的教学参考用书。

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 实战: 卡牌手游开发指南/朱先忠, 高嗣慧编著. —北京: 电子工业出版社, 2016.8

ISBN 978-7-121-29272-9

I. ①C… II. ①朱… ②高… III. ①移动电话机—游戏程序—程序设计—指南 IV. ①TN929.53-62
②TP311.5-62

中国版本图书馆 CIP 数据核字 (2016) 第 150559 号

策划编辑: 张 迪

责任编辑: 底 波

印 刷: 三河市华成印务有限公司

装 订: 三河市华成印务有限公司

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: 31.25 字数: 796 千字

版 次: 2016 年 8 月第 1 版

印 次: 2016 年 8 月第 1 次印刷

印 数: 3 000 册 定价: 79.00 元

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, 88258888。

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

本书咨询联系方式: zhang@phei.com.cn。

前言

本书特色

第一，本书案例“拇指接龙”游戏源码基于目前 Cocos2d-x 引擎最新版本 3.9 及相应的最新 UI 编辑器 Cocos Studio 2.3.3。与 2.x 版本相比，Cocos2d-x 3.x 版本主要实现了三大优化：一是通过引入新一代渲染器，可创建渲染队列，从而简化渲染过程并更容易实现跨平台；二是经全新修改过的 API 将语言用法改为 C++ 程序员更习惯的使用方式，进一步提升源码的稳定性和可读性；三是大幅度提升、优化系统性能，充分利用自动裁剪、自动批处理技术。因此，尽快适应和掌握 Cocos2d-x 3.x 版本引擎，并熟练运用对应的 UI 编辑器 Cocos Studio 2.3.3，既是每一位游戏程序员的内心渴求，也是形势所迫。

第二，本书给出的是一个相对完整的近乎于企业级的跨平台手游案例剖析，从游戏策划、美工设计到后台编码完全靠本人自己利用业余时间独立开发完成。因此，无论从编码风格一致性方面，还是从一名全栈式手游程序员发展角度来看，本书都值得一读。

创作技术背景

“空当接龙”游戏，本人十多年前就对 Windows 的这款自带单机版应用爱不释手，曾经尝试使用 WiEngine 引擎开发，并从数据结构和软件工程开发角度对该游戏做过多次更改。之后，该游戏于 2014 年使用 Cocos2d-x 2.2.3 和 Cocos Studio 2.4.0.1 二次开发，如今升级到当前最新的 Cocos2d-x 3.9 及 Cocos Studio 2.3.3，并进一步修整（加入弱联网和跨 iOS 与 Android 双平台支持）。

另外，本人自五年前进入 Android 游戏开发领域，经过几番筛选，于 2013 年年末，定位于 Cocos2d-x 引擎。2015 年上半年本人集中学习了网游服务器端开发技术，主要针对基于 Python 的 Firefly 和基于 Node 的 Pomelo 两款开源引擎，后来尝试使用 LeanCloud 云引擎技术简化弱联网游戏开发。

主要内容与编写思路

为便于阅读，本书共分为 3 篇：第 1 篇是基础篇（第 1~3 章），相对短小，是 Cocos2d-x 3.9 + Cocos Studio 2.3.3 的基础内容，主要介绍引擎在 Windows 和 Mac 双平台的开发环境搭建，并给出各自的入门级实例（部署到 Android 和 iPhone 手机上）；第 2 篇是实践篇（第 4~11 章），篇幅最长，从最初的游戏策划开始，并遵循软件工程开发指导理念，按顺序详细介绍整款“拇指接龙”游戏的实现过程；第 3 篇属于高级扩展篇（第 12~17 章），给出开发真正跨平台、弱联网功能的手游必须了解甚至要掌握的一些技术，如必备的 Android 平台 Java 开发基本功和 iOS 平台必需的 Objective-C++ 基本功，NDK 开发基本功（全面剖析工具 EasyNDK），作为独立手游戏开发者必需的 Photoshop 和 Flash 基本功，服务器端开发技术

基本功，等等。

读者对象定位

尽管从横向来说，本书适合于所有具有一定 C/C++ 开发经验的读者阅读，但读者如果具有一定的 Android 开发基础知识或基本 Mac 平台编程经验则更适合。归纳来看，本书适合于：

- 希望尽快通过 Cocos2d-x 开发出跨 Android 平台和 iOS 平台游戏的在校大学生；
- 希望全面学习 Cocos2d-x 跨平台游戏开发技术的程序员；
- 立志成为独立手游创业者的业余计算机软件开发爱好者；
- 对于跨平台嵌入式应用开发技术表现出浓厚兴趣的其他研究人员。

系统需求与源码

本书案例“拇指接龙”游戏几经修整和版本升级，先在 Windows 7 + Vmware 虚拟机 + Eclipse 下调试好 Win32 版本、Android 真机版本和 iOS 模拟器版本，后在 Mac OS 10.11 + Android Studio 1.4 + Cocos2d-x 3.9（结合 Cocos Studio 2.3.3）环境下调试通过 Android 和 iOS 真机版本。

本书所有案例源码可从如下一些途径下载。

邮箱：thumbelina_2015@163.com，密码：134137138_ok。

另外，也可以通过本人电子邮件与我联系索取。

勘误及联系作者

尽管在编写本书的过程中，本人努力做到严谨无误，但限于个人水平以及时间的原因，书中难免有所遗漏或错误，真诚期望各位读者不吝指正。

读者可以通过本人的 51CTO 博客（<http://zhuxianzhong.blog.51cto.com>）或电子邮件（sdmyzxz@163.com）与我取得联系。

致谢

真诚感谢电子工业出版社张迪编辑以及所有编辑人员在本书的策划、撰写及校对的全程中所给予本人的耐心细致的指导及帮助。

感谢山东畜牧兽医职业学院经贸系主任高嗣慧参与了本书的成书指导，并参与了书中第 1 章和最后 3 章的编写。

最后要感谢的是我的所有亲人和朋友，是你们的鼓励与支持让我一步步坚持向前并逐步走向成熟与成功。

朱先忠

sdmyzxz@163.com

目 录

基 础 篇

第1章	Cocos2d - x 开发环境搭建	2
1.1	选择 Cocos2d - x 3. x 的理由	2
1.2	C++ 11 主要新特性	2
1.2.1	nullptr	2
1.2.2	auto	3
1.2.3	lambda 函数	3
1.2.4	std::function 类型	4
1.2.5	C++11 多线程 std::thread	4
1.3	Windows 7 平台开发环境搭建	6
1.3.1	Win32 版本开发环境搭建	6
1.3.2	安装 Cocos Studio 2.3.3	9
1.3.3	Android 版本开发环境搭建	11
1.4	Mac 平台开发环境搭建	13
1.4.1	安装 Cocos2d - x 3.9 开发环境与创建示例工程	13
1.4.2	运行官方示例工程 cpp - tests	15
1.4.3	安装 Mac 版本的 Cocos Studio	16
1.4.4	Android 开发环境搭建	17
1.5	小结	19
第2章	Photoshop 与 Flash——独立手游开发者的必修课	21
2.1	Photoshop CS 6 鼠绘技巧与示例	21
2.2	与手游 UI 制作密切相关的 Photoshop 技术	22
2.2.1	GIF 帧动画制作	22
2.2.2	由 GIF 动画文件导出一组 PNG 图像文件	27
2.2.3	GIF 时间轴动画	28
2.3	Flash CS 6 鼠绘技巧与示例	34
2.4	与手游 UI 制作密切相关的 Flash 技术	37
2.4.1	生成 Sprite 表单	38
2.4.2	尽量不把舞台元件实例集合导出到精灵表单	38
2.4.3	导出库元件集到精灵表单时的注意事项	39
2.4.4	Flash 动画简介	40

2.4.5	Flash + DragonBone 联合制作 heart_ani 动画	42
2.5	小结	53
第3章	手游辅助开发利器——TexturePacker 和 ParticleEditor	55
3.1	图片切片工具 TexturePacker	55
3.1.1	为什么要用 TexturePacker	55
3.1.2	TexturePacker 的基本用法	56
3.1.3	Retina 支持——AutoSD	58
3.1.4	TextureSettings 面板其他参数说明	60
3.1.5	空当接龙游戏使用的碎图打包技术	61
3.1.6	TexturePacker 与 XCode 集成简介	62
3.2	Windows 平台免费开源的粒子编辑器 ParticleEditor	63
3.2.1	粒子系统属性讲解	65
3.2.2	官方示例工程 cpp - tests 中粒子系统使用小结	68
3.2.3	让粒子系统沿着指定轨迹运动	74
3.2.4	SplashScreen 场景粒子系统应用回顾	76
3.3	小结	76

实 践 篇

第4章	拇指接龙手游整体策划与设计	80
4.1	拇指接龙游戏创意构思及快照欣赏	80
4.1.1	游戏构思	80
4.1.2	传统拇指接龙游戏算法（玩法）回顾	80
4.1.3	传统空当接龙游戏特点分析	82
4.1.4	目前市场同类游戏简单分析	82
4.1.5	旧瓶装新酿的可能性	83
4.1.6	快照预览	83
4.2	拇指接龙游戏整体数据策划与存储方案	84
4.2.1	数据策划与总体算法分析	84
4.2.2	数据存储方案分析	88
4.3	模块及功能图设计	89
4.3.1	系统总体模块及关系图	89
4.3.2	游戏主场景 MainScene 流程图	90
4.4	小结	92
第5章	Splash 场景设计与多分辨率屏幕适配方案	93
5.1	Splash UI 设计	93
5.1.1	设计场景图层文件	93
5.1.2	设计场景中的动画文件	94

5.2	后台代码分析	96
5.2.1	SplashScene 头文件定义	96
5.2.2	SplashScene 实现文件分析	97
5.3	多分辨率适配方案	102
5.3.1	单例类 SmartRes 头文件定义	102
5.3.2	单例类 SmartRes 实现代码	104
5.4	小结	107
第6章	拇指接龙游戏导航场景设计	108
6.1	使用场景编辑器构建场景 FirstScene	108
6.1.1	用 Cocos Studio 2.3.3 创建帧动画	110
6.1.2	用 Cocos Studio 2.3.3 创建骨骼动画	117
6.1.3	Cocos Studio 2.3.3 导出资源应用于编程	120
6.2	游戏导航场景编码设计	122
6.2.1	游戏导航场景功能分析	122
6.2.2	游戏导航场景编码设计	123
6.3	全局变量及 SQLite 开发技术分析	132
6.3.1	系统中的全局变量	132
6.3.2	两种本地文件存储方案——UserDefaults 和 SQLite 数据库	133
6.4	小结	138
第7章	游戏帮助系统设计	139
7.1	FaqScene 场景设计	139
7.1.1	FaqScene 场景 UI 设计	139
7.1.2	FaqScene 场景编码分析	140
7.1.3	多语言内容显示方案	147
7.2	TutorialScene 场景设计	147
7.2.1	概述	148
7.2.2	TutorialScene 场景 UI 设计	150
7.2.3	TutorialScene 场景头文件设计	151
7.2.4	TutorialScene 场景动画设计	154
7.2.5	TutorialScene 场景的双击式移动扑克牌编程	160
7.2.6	TutorialScene 场景中拖放式移动扑克牌	166
7.2.7	扑克牌自动飞动的硬编码方式	183
7.2.8	无须扑克牌匹配判定与恢复栈创建	184
7.3	小结	185
第8章	拇指接龙游戏关卡及回合设计	186
8.1	大关卡设计的模拟愤怒的小鸟界面	186
8.1.1	大关卡选择场景 UI 设计	187

8.1.2	图层内置动画与 ImageView_Title 控件	188
8.1.3	关于艺术数字控件 TextAtlas	189
8.1.4	大关卡选择编码的头文件设计	191
8.1.5	大关卡选择编码的实现文件编程	192
8.2	小关卡设计的模拟“RuneMasterPuzzle”界面	198
8.2.1	小关卡选择场景 UI 设计	198
8.2.2	小关卡选择设计的编码实现	199
8.3	游戏回合选择设计	204
8.3.1	UI 设计	204
8.3.2	后台编码	205
8.4	小结	211
第9章	创建通用场景切换类	212
9.1	LoadingScene.h 定义	212
9.2	LoadingScene 成员函数分析	213
9.3	小结	225
第10章	拇指接龙游戏道具商店设计	227
10.1	了解游戏道具	227
10.1.1	道具定义	227
10.1.2	游戏道具获取方式	228
10.1.3	游戏道具数据库设计	228
10.2	拇指接龙游戏道具数据策划	229
10.2.1	操作撤销道具——undo	229
10.2.2	列重洗牌道具——fan	229
10.2.3	列移动道具——worm	230
10.2.4	wildcard 道具——wand	231
10.2.5	Joker 道具——net	232
10.2.6	道具——owl	233
10.3	模拟“进击的小怪物”设计道具商店	233
10.3.1	道具商店主界面设计	233
10.3.2	道具购买场景设计	235
10.3.3	诱导玩家安装应用赚积分场景设计	236
10.4	游戏道具商店编码实现	236
10.4.1	由游戏导航场景切换到道具商店系统	236
10.4.2	道具商店系统 StoreScene.h 分析	237
10.4.3	道具数值定义	239
10.4.4	加载场景资源文件	240
10.4.5	触发道具购买行为编程	242
10.4.6	进行购买道具活动相关编程	244

10.4.7	积分墙相关编程	248
10.5	小结	248
第11章	拇指接龙游戏主场景开发	249
11.1	游戏主场景开发的功能设计	249
11.1.1	道具子系统	250
11.1.2	发牌子系统	250
11.1.3	玩牌子系统	250
11.1.4	奖励子系统	251
11.2	游戏主场景开发的界面设计	251
11.2.1	游戏主场景界面设计	251
11.2.2	主场景工具栏（即游戏场景 HUD）设计	253
11.2.3	奖励信息提示图层设计	254
11.2.4	道具使用说明图层设计	255
11.2.5	游戏暂停图层设计	257
11.2.6	游戏继续操作图层设计	258
11.2.7	游戏回合结束小结图层设计	259
11.3	游戏主场景开发的数据结构设计	260
11.3.1	创建扑克牌类——Card	264
11.3.2	定义双击或拖放扑克牌的有关变量	271
11.3.3	定义用于存储扑克牌的数据结构	271
11.4	游戏主场景开发的总体流程设计	272
11.5	Undo 道具开发与 STL 函数	272
11.5.1	Undo 道具功能回顾	272
11.5.2	Undo 道具有关数据结构与操作	273
11.5.3	使用与定义 stackTidying 函数	275
11.5.4	使用 Undo 道具编程	279
11.6	Fan 和 Worm 道具编程	283
11.6.1	Fan 道具	283
11.6.2	Worm 道具	287
11.7	Wand、Net 和 Owl 道具编程	291
11.7.1	Wand 道具	292
11.7.2	Net 道具	298
11.7.3	Owl 道具	303
11.8	奖励机制编程	307
11.8.1	拇指接龙游戏奖励子系统功能模块	308
11.8.2	奖励定义	308
11.8.3	奖励相关编程分析	310
11.8.4	奖励系统监控与执行	318

11.9	扑克牌拖放与单击事件编程	322
11.9.1	onTouchBegan 函数总体执行流程示意图	323
11.9.2	屏幕单击事件编程	325
11.9.3	onTouchMoved 函数总体执行流程示意图	336
11.9.4	onTouchEnded 函数总体执行流程示意图	339
11.10	游戏进程监控编程	341
11.10.1	游戏系统时间监控	341
11.10.2	奖品监控	341
11.10.3	游戏全程监控机制	343
11.10.4	缺憾	348
11.11	游戏暂停编程	348
11.11.1	功能模块图	348
11.11.2	暂停模块编程	349
11.12	双击式移动扑克牌编程	353
11.13	拖放单张扑克牌编程	358
11.14	拖放多张扑克牌编程	371
11.14.1	onTouchBegan() 函数主要功能总结	372
11.14.2	onTouchMoved() 函数主要功能总结	372
11.14.3	onTouchEnded 函数内系列扑克牌执行流程示意图	374
11.15	小结	378

高级扩展篇

第12章	iOS 移植的 Objective - C 编程知识必备	382
12.1	必须了解的 Objective - C 2.0 编程知识	382
12.1.1	Objective - C 语言特性的 Block	382
12.1.2	委托和协议	386
12.1.3	Objective - C = 编译器 + 运行时	388
12.2	iOS 应用程序文件夹架构	388
12.2.1	文件 main.m 的作用	388
12.2.2	UIApplicationMain 函数	389
12.2.3	应用程序委托	390
12.2.4	视图控制器	390
12.2.5	Info.plist 文件	390
12.2.6	图标和默认图像	391
12.2.7	沙盒机制	391
12.2.8	MVC 编程规范	391
12.3	iOS 应用程序生命周期分析	391
12.3.1	应用程序的状态	392

12.3.2	各个程序运行状态时代理的回调	392
12.3.3	加载应用程序进入前台	395
12.3.4	加载应用程序进入后台	396
12.3.5	基于警告式响应中断	396
12.3.6	进入后台运行	397
12.3.7	返回前台运行	399
12.3.8	程序终止	399
12.3.9	The Main Run Loop 主运行循环	400
12.4	iOS 应用程序 MVC 架构	400
12.4.1	MVC 框架结构	400
12.4.2	深入理解 UIViewController 类	401
12.5	Cocos2d-x 框架下 iOS 版本的 AppController	404
12.6	Objective-C 反射技术回顾	405
12.6.1	获取 Class	406
12.6.2	确定继承关系	406
12.6.3	方法的动态调用	407
12.7	C++ 与 Objective-C 混合编程	409
12.7.1	Objective-C 和 C++ 混编支持	409
12.7.2	结构成员与类成员	411
12.7.3	关于命名空间	412
12.7.4	关于 C++ 模板	412
12.7.5	关于异常处理	412
12.8	EasyNdk 的 iOS 版深入剖析	412
12.8.1	实现原理图展示	413
12.8.2	EasyNdk-for-cocos2dx-master 下载 Demo 应用全面剖析	413
12.8.3	总结 EasyNdk 的使用套路	419
12.9	参考资料	419

第13章	Android 移植的 Java&Ndk 编程知识必备	420
13.1	Android 应用程序体系架构	420
13.2	Android 应用程序基本组件	421
13.2.1	Activity 与 View	421
13.2.2	Service 与 Activity	422
13.2.3	Intent 类和 IntentFilter 类	424
13.2.4	Android 应用程序生命周期	425
13.3	了解 Android 应用 UI 设计	425
13.3.1	丰富的可视化控件	426
13.3.2	关联后台代码与前台 UI	427
13.4	Android 事件处理机制	428

13.4.1	组件间消息传递——Intent	428
13.4.2	线程间消息传递——Message	429
13.5	EasyNDK 的 Android 版深入剖析	430
13.5.1	实现原理图	431
13.5.2	EasyNDK - for - cocos2dx - master 下载 Demo 应用全面分析	431
13.5.3	Classes 文件夹下子文件夹 jansson 和 NDKHelper	432
13.5.4	EasyNDK\proj.android\src\com\easyndk 文件夹下内容解析	433
13.5.5	Classes 下文件 HelloWorldScene.cpp 分析	434
13.5.6	总结 EasyNDK 的使用套路	434
13.6	小结	435
第14章	拇指接龙集成社交分享 ShareSDK for iOS	436
14.1	下载 ShareSDK iOS 版本	436
14.2	下载 ShareSDK iOS 版本的 Cocos2d - x 插件	437
14.3	添加必需的依赖库	438
14.4	Cocos2d - x 插件部分有关编程	439
14.4.1	选择需要的平台 SDK 和 Cocos2d - x 环境	439
14.4.2	修改 AppDelegate.cpp 文件——初始化	440
14.5	ShareSDK 接口的调用	441
14.6	截图欣赏	443
14.7	小结	444
第15章	拇指接龙集成社交分享 ShareSDK for Android	445
15.1	通用部分集成步骤	445
15.1.1	下载 ShareSDK For COCOS2D - X	445
15.1.2	初始化 ShareSDK 并配置社交平台信息	446
15.1.3	分享内容、用户授权及获取用户资料	447
15.2	Android 部分集成步骤	452
15.3	其他说明	453
15.3.1	ShareSDK.xml 配置文件说明	453
15.3.2	java 层代码解释	454
15.4	效果图展示	454
15.5	小结	455
第16章	拇指接龙游戏集成有米积分墙的 Android 版	457
16.1	积分墙概述	457
16.1.1	计费方式	457
16.1.2	展现方式	457
16.2	有米积分墙的基本配置	458
16.2.1	导入 SDK	458

16.2.2	权限配置	458
16.2.3	广告组件配置	458
16.2.4	初始化	459
16.2.5	混淆配置	459
16.2.6	打包	459
16.2.7	设置渠道号（推荐）	459
16.3	集成有米积分墙代码	460
16.3.1	初始化	460
16.3.2	展示全屏积分墙	460
16.3.3	展示悬浮半屏积分墙	460
16.3.4	积分管理接口	461
16.4	积分墙高级功能	462
16.4.1	积分余额变动通知	462
16.4.2	客户端 SDK 获取订单信息	463
16.4.3	服务器获取订单信息	464
16.4.4	验证积分墙配置是否正确	464
16.4.5	配置积分墙 SDK 的积分到账提醒	465
16.5	使用 EasyNDK 简化有米积分墙集成	465
16.5.1	C++ 端调用 Java 端	465
16.5.2	Java 端调用 C++ 端	466
16.5.3	下载并解压 EasyNDK for Cocos2d-x	468
16.5.4	Classes 文件夹下的复制	468
16.5.5	proj.android 中 src 文件夹下的复制	469
16.5.6	ThumbelinaCell.java 代码修改	469
16.5.7	修改 Android.mk 文件	470
16.5.8	修改 StoreScene.cpp 文件	470
16.6	软件截图欣赏	471
16.6.1	提示玩家积分交换	471
16.6.2	积分墙界面	472
16.7	游戏中积分墙相关数据算法	472
16.7.1	集成有米积分墙时的典型问题	472
16.7.2	有米广告过滤列表中显示的数据形式	473
16.7.3	由道具价格推算积分墙显示数据	474
16.7.4	我们的算法推算	475
16.8	小结	476
第 17 章	弱联网手游开发的云存储技术	477
17.1	Firebase（国外）	477
17.2	AppWarp（国外）	478

17.3	野狗（国内）	479
17.4	LeanCloud（国内）	480
17.4.1	数据存储服务	480
17.4.2	消息推送服务	481
17.4.3	实时通信服务	481
17.4.4	iOS/OSX 数据存储开发	481
17.5	关于 Pomelo 和 Firefly 等	483
17.6	小结	484

基 础 篇

第1章 Cocos2d-x 开发环境搭建

作为本书的开篇，本章将介绍一些 Cocos2d-x 手游开发必需的基础知识与环境搭建。首先要介绍的是 C++ 11 关键技术，因为 3.x 版本的 Cocos2d-x C++（本书游戏基于 Cocos2d-x 3.9）中利用 C++ 11 技术提高系统开发效率，而且其官方示例中也广泛使用这种新技术。然后，依次介绍在 Windows 7 平台与 Mac 平台上搭建 Cocos2d-x 开发环境的关键技术、步骤与注意事项。

1.1 选择 Cocos2d-x 3.x 的理由

Cocos2d-x 3.x 版本，比 2.x 版本主要实现了三大优化：一是新的渲染器，可创建渲染队列，可简单地实现渲染，跨平台也非常容易；二是新修改过的 API，将语言用法改为 C++ 程序员更习惯的使用方式，稳定性和可读性得到提升；三是性能再次提升优化，利用自动裁剪、自动批处理技术，性能得到很大提升。

在版本升级过程中，Cocos2d-x 3.x 大量地引入了 C++ 11 的新特性，这正是选择 Cocos2d-x 3.x C++ 的程序员必须学习 C++ 11 的最大理由。

1.2 C++ 11 主要新特性

Cocos2d-x 引擎自 3.0 版本开始逐步引入 C++ 11 特性，其根本目的是为了最终取代此前的 Objective-C 风格。摒弃 Objective-C 风格将有助于 Cocos2d-x 开发者更加自然地使用 C++ 11 特性，如使用 Lambda 表达式、强类型枚举、更易懂的属性命名风格等。如今，各种主流 C++ 编译器都争先引入 C++ 11 特性，以便从根本上扩展原有功能，这也是我们需要学习 C++ 11 的重要原因。

由于篇幅所限，本节我们仅对 Cocos2d-x 3.x C++ 编程中常用的和主要的 C++ 11 功能进行归纳总结。

1.2.1 nullptr

nullptr 表示空指针，其“==”运算符是经过重载的，主要目的是为了消除“NULL”的歧义性。

注意：两个不同类型的指针不能判断是否相等，即使它们都是 nullptr。例如：