



移动开发最有价值专家
原创作品

清华

开发者书库



The Practical Developing Guide for Cocos2d-x

Cocos2d-x游戏 实战指南

李宁◎著

Li Ning

触控科技副总裁 Jane

微软开发体验和合作事业部开发技术顾问 梅颖广

51CTO学院运营总监 曹亚莉

哈尔滨工业大学 王峥

推|荐|阅|读

源代码
视频课程

清华大学出版社



清华

开发者书库



The Practical Developing Guide for Cocos2d-x

Cocos2d-x游戏 实战指南

李宁◎著

Li Ning

清华大学出版社
北京

内 容 简 介

本书深入系统地讲解了 Cocos2d-x 3.10 游戏开发的理论与实践,书中内容涵盖了 Cocos2d-x 3.x 各方面的知识点与示例分析。这些内容包括如何搭建 Cocos2d-x 的开发环境(iOS 和 Android 平台)、Cocos2d-x 的工程结构、核心类和相关的 API、标签、菜单、控件、本地化、事件处理机制、网络技术、动作、调度、绘图 API、动画、存储技术、Sprite3D、瓦片地图、粒子系统、物理引擎、骨骼动画、Objective-C、Swift、C++ 和 Java 交互的方式。本书的第 18 章提供了一个完整的案例——星空大战,这个案例是一款完整的射击类游戏,类似于雷电游戏。该游戏使用了本书介绍的基本知识点,通过这个案例,读者可以更进一步巩固基本理论知识。

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 游戏实战指南/李宁著. —北京:清华大学出版社,2016

(清华开发者书库)

ISBN 978-7-302-44784-9

I. ①C… II. ①李… III. ①手机软件—游戏程序—指南 IV. ①TP317.67-62

中国版本图书馆 CIP 数据核字(2016)第 190017 号

责任编辑:盛东亮

封面设计:李召霞

责任校对:焦丽丽

责任印制:王静怡

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

印装者:北京密云胶印厂

经 销:全国新华书店

开 本:186mm×240mm

印 张:33.75

字 数:759 千字

版 次:2016 年 11 月第 1 版

印 次:2016 年 11 月第 1 次印刷

印 数:1~2500

定 价:79.00 元

产品编号:068048-01

前言

PREFACE

Cocos2d-x 是目前最热门的 2D/3D 开源游戏引擎,国内外已经有很多游戏开发厂商使用 Cocos2d-x 开发自己的游戏产品。在写作本书时,Cocos2d-x 的最新版本是 Cocos2d-x 3.10,可能在读者拿到本书时,Cocos2d-x 的版本会再次升级,不过这并不影响您使用本书,因为 Cocos2d-x 3.x 的 API 是比较稳定的,基本能向下兼容。

从理论上来说,Cocos2d-x 可以同时实现桌面游戏(Windows、Mac OS X 和 Linux)和移动游戏(iOS 和 Android),不过考虑到 Cocos2d-x 的主要应用场景是手机,所以本书着重介绍了 Cocos2d-x 在 Android 和 iOS 设备上的开发和实现。这里推荐使用 Mac OS X 系统,因为 Cocos2d-x 与 XCode 的兼容性特别好,而且很容易调试;此外,Mac OS X 可以同时开发 iOS 和 Android 版的游戏,而 Windows 只能开发 Android 版本的游戏。

本书的编写历经 9 个月,于 2016 年 4 月完成初稿。期间经历了 Cocos2d-x 的多次版本升级,为力求让本书采用的技术保持最新,在完成初稿的最后时刻,将 Cocos2d-x 升级到 3.10。而且,笔者特意为本书精心做了一个 Demo,通过 Demo 中的菜单,可以进入每一章和每一节的代码演示界面,这也大大方便了读者观看本书的代码演示。

本书的源代码和勘误都将通过微信公众号提供,读者可以扫描下方二维码加公众号。

本书作者的技术博客网址为 <http://geekori.com>。QQ 交流群号为 264268059。

本书配套视频教程: <http://edu.51cto.com/pack/view/id-16.html>。

李 宁

2016 年 9 月



目 录

CONTENTS

第 1 章 初识 Cocos2d-x	1
1.1 跨平台的由来	1
1.2 跨平台开发模式的种类和优缺点	2
1.2.1 虚拟机与跨平台技术	3
1.2.2 Web 与跨平台技术	5
1.2.3 跨平台编译技术	7
1.2.4 为每个平台定制本地应用	8
1.3 跨平台游戏引擎	11
1.3.1 什么是跨平台游戏引擎	11
1.3.2 有哪些游戏引擎可以跨平台	13
1.4 Cocos2d-x 横空出世	14
1.4.1 Cocos2d-x 的前世今生	14
1.4.2 Cocos2d-x 的特点	15
1.5 小结	15
第 2 章 搭建和使用跨平台 Cocos2d-x 开发环境	16
2.1 使用 Cocos2d-x 开发游戏需要准备些什么	16
2.2 Cocos2d-x 3.10 开发环境的最低要求	17
2.3 让 Eclipse 支持 C++ 11	18
2.4 Cocos2d-x 3.10 开发环境的安装、配置和使用	20
2.4.1 Cocos2d-x 3.10 的目录结构	20
2.4.2 使用 cocos 建立跨平台工程	21
2.4.3 编译和运行 Cocos2d-x 3.10 for iOS/Mac 程序	24
2.4.4 编译和运行 Cocos2d-x 3.10 for Android 程序	28
2.4.5 编译和运行 Cocos2d-x 3.10 for Win32 程序	32
2.5 Cocos2d-x 3.10 例子代码的使用	33
2.6 小结	33

第 3 章 Cocos2d-x 默认工程模板的架构和源代码分析	34
3.1 Cocos2d-x 实现跨平台的原理	34
3.2 默认场景类 HelloWorldScene 的实现原理分析	38
3.3 控制横竖屏切换	43
3.4 小结	45
第 4 章 Cocos2d-x 中的核心类	46
4.1 导演类(Director)	46
4.1.1 Director 类与初始化 Cocos2d-x	47
4.1.2 结束、暂停与恢复 Cocos2d-x 程序	49
4.1.3 获取窗口的尺寸和位置	51
4.1.4 Director 类中的其他成员方法	57
4.2 节点类(Node)	57
4.2.1 节点类的功能	58
4.2.2 节点类的成员方法	59
4.2.3 Cocos2d-x 对象的创建、自动释放(Autorelease)与 ARC	64
4.2.4 CREATE_FUNC 宏与 create 方法	65
4.2.5 Cocos2d-x 中的坐标系	68
4.2.6 节点的移动、缩放和旋转	74
4.3 场景类(Scene)	79
4.3.1 创建场景	80
4.3.2 运行和切换场景	80
4.3.3 场景切换的各种特效	81
4.4 图层类(Layer)	89
4.4.1 Layer 类的基本应用	89
4.4.2 Layer 的子类	91
4.4.3 颜色图层类(LayerColor)	91
4.4.4 颜色渐变图层类(LayerGradient)	96
4.4.5 多图层管理类(LayerMultiplex)	98
4.5 精灵类(Sprite)	100
4.5.1 Sprite 类	100
4.5.2 贴图类(Texture2D)和贴图缓冲类(TextureCache)	102
4.5.3 精灵批处理类(SpriteBatchNode)	104
4.5.4 精灵框架类(SpriteFrame)	106
4.5.5 精灵框架缓冲类(SpriteFrameCache)	108

4.5.6 九宫格缩放精灵类(Scale9Sprite)	111
4.6 集合类(Vector 和 Map)	114
4.7 小结	117
第5章 标签、菜单与控件	118
5.1 标签类	118
5.1.1 LabelTTF 标签(使用 TTF 字体文件)	119
5.1.2 LabelAtlas 标签(需要截取图像)	122
5.1.3 LabelBMFont 标签(使用 FNT 字体文件)	123
5.1.4 生成 FNT 字体文件	125
5.2 菜单(Menu)与菜单项(MenuItem)	127
5.2.1 Menu 类	127
5.2.2 标签菜单项(MenuItemLabel)	128
5.2.3 字体菜单项(MenuItemFont)	131
5.2.4 精灵菜单项(MenuItemSprite)	133
5.2.5 图像菜单项(MenuItemImage)	136
5.2.6 开关菜单项(MenuItemToggle)	139
5.3 控件	142
5.3.1 UI 布局(Layout)	142
5.3.2 文本控件(Text、TextAtlas 和 TextBMFont)	145
5.3.3 按钮控件(Button)	147
5.3.4 文本输入框控件(TextField)	149
5.3.5 高级文本输入框控件(EditText)	152
5.3.6 复选框控件(CheckBox)	158
5.3.7 开关控件(ControlSwitch)	162
5.3.8 图像框控件(ImageView)	164
5.3.9 进度条控件>LoadingBar)	165
5.3.10 滑杆控件(Slider)	167
5.3.11 高级滑杆控件(ControlSlider)	169
5.3.12 步进控件(ControlStepper)	172
5.3.13 列表控件(ListView)	174
5.3.14 表格控件(Tableview)	178
5.3.15 滚动视图控件(ScrollView)	182
5.3.16 分页控件(PageView)	183
5.3.17 颜色选择控件(ControlColourPicker)	185
5.4 小结	186

第 6 章 本地化	187
6.1 自适应屏幕分辨率	187
6.1.1 尺寸类型及屏幕适配原理	188
6.1.2 Cocos2d-x 中的各种尺寸深度详解	189
6.1.3 分辨率策略(ResolutionPolicy)	193
6.1.4 通过设计尺寸进行屏幕适配	195
6.1.5 通过内容缩放因子进行屏幕适配	199
6.1.6 为每一个 Node 单独调整尺寸	201
6.2 Cocos2d-x 多语言适配	201
6.3 小结	205
第 7 章 Cocos2d-x 中的事件处理机制	206
7.1 触摸事件与移动节点	206
7.2 多点触摸	213
7.3 键盘响应事件	219
7.4 鼠标响应事件	220
7.5 加速度传感器事件	222
7.6 自定义事件	225
7.7 小结	227
第 8 章 网络技术	228
8.1 HttpClient 类	228
8.1.1 发送 HTTP Get 请求	228
8.1.2 发送 Http Post 请求	231
8.2 WebSocket 类	234
8.3 SocketIO 类	239
8.4 小结	242
第 9 章 Cocos2d-x 中的动作类	243
9.1 所有动作的基类(Action)	244
9.2 位置(Position)动作	245
9.2.1 沿直线匀速移动动作(MoveBy/MoveTo)	245
9.2.2 跳跃动作(JumpBy/JumpTo)	247
9.2.3 贝塞尔曲线动作(BezierBy/BezierTo)	248
9.2.4 固定张力的样条曲线动作(CatmullRomBy/CatmullRomTo)	251

9.2.5	可变张力的样条曲线动作(CardinalSplineBy/CardinalSplineTo) ...	254
9.3	旋转与缩放动作	254
9.3.1	旋转动作(RotateBy/RotateTo)	255
9.3.2	缩放动作(ScaleBy/ScaleTo)	256
9.4	执行规则动作	257
9.4.1	顺序执行动作(Sequence)	258
9.4.2	并行动作(Spawn)	259
9.4.3	重复执行动作(Repeat)	260
9.4.4	无限循环动作(RepeatForever)	261
9.4.5	反序动作(ReverseTime)	262
9.5	控制节点显示和隐藏的动作	263
9.5.1	闪烁动作(Blink)	263
9.5.2	淡入淡出动作(FadeIn/FadeOut)	263
9.5.3	透明度渐变动作(FadeTo)	265
9.5.4	颜色渐变动作(TintBy/TintTo)	265
9.6	可立即执行的动作	266
9.7	回调函数动作(CallFunc/CallFuncN)	269
9.8	变速动作(Ease Action)	270
9.8.1	幂加速动作(EaseIn)	270
9.8.2	幂减速动作(EaseOut)	272
9.8.3	幂加速减速动作(EaseInOut)	273
9.8.4	指数加速动作(EaseExponentialIn)	274
9.8.5	指数减速动作(EaseExponentialOut)	274
9.8.6	指数加速减速动作(EaseExponentialInOut)	275
9.8.7	正弦加速动作(EaseSineIn)	275
9.8.8	正弦减速动作(EaseSineOut)	276
9.8.9	正弦加速减速动作(EaseSineInOut)	276
9.8.10	弹性加速动作(EaseElasticIn)	277
9.8.11	弹性减速动作(EaseElasticOut)	278
9.8.12	弹性加速减速动作(EaseElasticInOut)	279
9.8.13	弹跳加速动作(EaseBounceIn)	280
9.8.14	弹跳减速动作(EaseBounceOut)	281
9.8.15	弹跳加速减速动作(EaseBounceInOut)	281
9.8.16	回退加速动作(EaseBackIn)	282
9.8.17	回退减速动作(EaseBackOut)	282
9.8.18	回退加速减速动作(EaseBackInOut)	283

9.9 网格动作(Grid Action)	284
9.9.1 实现测试网格动作的基类	284
9.9.2 2D 波浪动作(Waves)	285
9.9.3 3D 波浪动作(Waves3D)	286
9.9.4 水平 3D 翻转动作(FlipX3D)	287
9.9.5 垂直 3D 翻转动作(FlipY3D)	288
9.9.6 3D 震动动作(Shaky3D)	289
9.9.7 3D 透镜动作(Lens3D)	290
9.9.8 3D 波纹动作(Ripple3D)	290
9.9.9 流体动作(Liquid)	291
9.9.10 旋转变形动作(Twirl)	292
9.9.11 3D 瓦片震动动作(ShakeTiles3D)	293
9.9.12 3D 瓦片破碎动作(ShatteredTiles3D)	294
9.9.13 3D 瓦片洗牌动作(ShuffleTiles)	294
9.9.14 瓦片从左下角到右上角淡出动作(FadeOutTRTiles)	295
9.9.15 瓦片从右上角到左下角淡出动作(FadeOutBLTiles)	296
9.9.16 瓦片向上淡出动作(FadeOutUpTiles)	297
9.9.17 瓦片向下淡出动作(FadeOutDownTiles)	297
9.9.18 关闭瓦片动作(TurnOffTiles)	298
9.9.19 3D 波浪瓦片动作(WavesTiles3D)	299
9.9.20 3D 跳跃瓦片动作(JumpTiles3D)	300
9.9.21 拆分行动作(SplitRows)	301
9.9.22 拆分列动作(SplitCols)	301
9.9.23 3D 翻页动作(PageTurn3D)	302
9.10 其他动作	303
9.10.1 绑定节点和动作(TargetedAction)	303
9.10.2 扭曲动作(SkewBy 和 SkewTo)	304
9.10.3 轨道照相机动作(OrbitCamera)	305
9.10.4 延迟运行动作(DelayTime)	307
9.11 小结	307
第 10 章 调度器、绘图 API 与动画	308
10.1 调度器(Scheduler)	308
10.1.1 update 定时器调度	308
10.1.2 设置 update 定时器的优先级	311
10.1.3 自定义调度器	313

10.1.4 停止调度器	316
10.2 绘图 API	317
10.3 动画	320
10.3.1 基于绘图 API 与调度器的动画	320
10.3.2 帧(Frame)动画	323
10.4 小结	327
第 11 章 数据存储技术	328
11.1 使用 UserDefaults 读写 Key-Value 类型的数据	328
11.2 读写流文件	332
11.3 操作 XML 文件	336
11.3.1 使用 tinyxml2 生成 XML 文件	336
11.3.2 使用 tinyxml2 以 DOM 方式读取 XML 文件	338
11.3.3 使用 SAXParser 以 SAX 方式读取 XML 文件	340
11.4 使用 rapidjson 读写 JSON 文件	343
11.5 使用 SQLite 数据库	349
11.6 小结	353
第 12 章 Sprite3D 技术	354
12.1 什么是 Sprite3D	354
12.2 Cocos2d-x 支持哪些 3D 模型格式	355
12.3 装载 obj 格式的 3D 模型文件	355
12.4 装载和使用 c3t 和 c3b 格式的 3D 模型文件	358
12.4.1 使用 fbx-conv 将 fbx 转换为 c3t 和 c3b 格式	358
12.4.2 旋转的 3D 怪物	358
12.4.3 游泳的海龟	360
12.5 小结	362
第 13 章 Cocos2d-x 中的瓦片和地图	363
13.1 瓦片和地图编辑器	364
13.1.1 瓦片地图的种类和用途	364
13.1.2 瓦片地图编辑器(Tiled Map Editor)	366
13.1.3 用 Tiled 编辑地图	367
13.2 在 Cocos2d-x 中使用地图	372
13.2.1 在程序中显示地图	372
13.2.2 隐藏和显示图层	374

13.2.3	编辑图块	375
13.2.4	获取图块的属性值	377
13.2.5	在地图中添加精灵	378
13.2.6	让精灵只在道路上移动	379
13.2.7	改变遮挡关系(zOrder 和 PositionZ)	385
13.2.8	拖动地图	388
13.2.9	在 45 度角地图上种树	389
13.2.10	使用对象层	394
13.3	小结	400
第 14 章 粒子系统		401
14.1	Cocos2d-x 内置的标准粒子特效	401
14.1.1	群星闪烁	401
14.1.2	幽灵鬼火	403
14.1.3	群星爆炸	404
14.2	粒子特效属性详解	405
14.3	自定义粒子特效	407
14.3.1	五彩旋转戒指	407
14.3.2	放大版的群星闪烁	408
14.4	可视化粒子特效设计器	410
14.4.1	Particle Designer(仅适用于 OS X)	410
14.4.2	Particle Editor(只适用于 Windows 系统)	414
14.5	通过 plist 文件实现粒子特效	415
14.5.1	彗星特效	415
14.5.2	用 plist 文件实现放大版的群星闪烁	416
14.6	燃烧的圣火	418
14.7	小结	421
第 15 章 物理引擎		422
15.1	物理引擎概述	422
15.1.1	什么是物理引擎	423
15.1.2	物理引擎在游戏中的地位	423
15.1.3	Box2D 和 Chipmunk 的对比	424
15.2	Cocos2d-x 3.x 中抽象封装的物理引擎	424
15.2.1	抽象物理引擎概述	424
15.2.2	使用抽象物理引擎创建 Body(刚体)	427

15.3	Android 平台如何使用物理引擎	430
15.4	改变 Android 和 iOS 平台默认的物理引擎	431
15.5	碰撞检测	432
15.6	只撞击一侧边缘的碰撞检测	440
15.7	足球撞击特效	442
15.8	骷髅堆中的足球	444
15.9	用重力感应控制的足球	445
15.10	小结	448
第 16 章 骨骼动画		449
16.1	骨骼动画概述	449
16.2	骨骼动画的优势	450
16.3	Spine 骨骼动画	450
16.3.1	Spine 骨骼动画简介	450
16.3.2	在 Cocos2d-x 中使用 Spine 骨骼动画	451
16.3.3	Spine 简介	454
16.3.4	用 Spine 导入骨骼动画图像资源	455
16.3.5	完成图像的摆放	457
16.3.6	绑定骨骼	457
16.3.7	在骨骼上旋转图像	458
16.3.8	设计骨骼动画	459
16.4	Armature 骨骼动画	460
16.4.1	安装和运行 Cocos Studio	460
16.4.2	将图像导入工程	461
16.4.3	使用图像资源	462
16.4.4	创建骨骼	463
16.4.5	绑定骨骼和图像	463
16.4.6	设计骨骼动画	464
16.4.7	导出骨骼动画	465
16.4.8	在 Cocos2d-x 中使用骨骼动画	465
16.5	小结	467
第 17 章 Objective-C、Swift、C++ 和 Java 交互		468
17.1	C++ 与 Objective-C 的交互	468
17.1.1	C++ 调用 Objective-C API	468
17.1.2	通过 Objective-C 获取 Web 数据	472

17.1.3	Objective-C 调用 C++ API	474
17.1.4	.mm 文件到底起什么作用	478
17.2	C++ 调用 Swift API	480
17.2.1	创建 Swift 文件	480
17.2.2	编写 Swift 类	481
17.2.3	自动生成的头文件的位置和命名规则	482
17.2.4	自动生成的头文件的内容	484
17.2.5	自动生成头文件非得用@objc 吗	484
17.2.6	用 Objective-C 调用 Swift API	485
17.2.7	C++ 调用 Swift API	487
17.3	Swift 调用 Objective-C API	488
17.3.1	在桥接头文件中引用 Objective-C 的头文件	488
17.3.2	在 Swift 源代码文件中调用 Objective API	489
17.3.3	Objective-C 调用 Swift API	489
17.3.4	声明和实现 C++ 风格的方法	490
17.3.5	C++ 调用 Swift API	491
17.4	Android 平台多语言交互	492
17.4.1	C++ 调用 Java 类的无参数静态方法	492
17.4.2	C++ 调用有参数的静态 Java 方法	495
17.4.3	C++ 调用 Java 类的非静态方法	497
17.4.4	C++ 调用 Java API 向 SQLite 数据库中插入信息 (获取 Context 对象)	500
17.4.5	C++ 调用 Java API 查询 SQLite 数据库中的记录	502
17.4.6	Java 调用 C++ API	505
17.5	小结	508
第 18 章	项目实战：星空大战	509
18.1	游戏概述	509
18.2	滚动背景	510
18.2.1	滚动游戏背景	510
18.2.2	让云彩在背景上移动	511
18.3	创建战机	512
18.3.1	创建战机类	512
18.3.2	创建战机图层	513
18.3.3	让战机移动	513
18.3.4	为战机添加光子鱼雷武器	514

18.3.5	为战机添加激光束武器	515
18.3.6	为战机添加带有自动跟踪功能的导弹	517
18.3.7	让战机发射武器	518
18.4	创建敌机	520
18.4.1	创建小敌机	520
18.4.2	创建大敌机	522
18.4.3	敌机发送炮弹	522
18.5	小结	524

第 1 章



初识 Cocos2d-x

已经从事移动游戏开发或即将进入该领域的读者对 Cocos2d-x 一定不陌生。Cocos2d-x 是近年来比较流行的 2D 跨平台游戏引擎^①。尤其是在国内的 IT 领域, Cocos2d-x 几乎是无人不知、无人不晓。Cocos2d-x 在国内之所以如此火爆,除了使用 C++ 开发带来的卓越的性能和用户体验外,还因为该游戏引擎的开发团队主要成员都是中国人。这一点也许改变了长期以来 IT 核心技术一直无缘中国的窘境。

尽管很多人已经对 Cocos2d-x 的基本情况有所了解,但本章作为 Cocos2d-x 的开篇,仍然需要为那些对 Cocos2d-x 还一知半解的读者解开心中的疑惑。不仅如此,本章还对 Cocos2d-x 的突出优点——跨平台进行深入的探讨。希望对那些喜欢跨平台技术的读者有所启发。

如果读者对跨平台、游戏引擎和 Cocos2d-x 的概念已经非常了解了,可以跳过本章,继续学习下一章的内容。

本章要点

- ❑ 跨平台开发模式的种类和优缺点;
- ❑ 分析各种类型的跨平台游戏引擎的优缺点;
- ❑ Cocos2d-x 的历史;
- ❑ Cocos2d-x 的特点。

1.1 跨平台的由来

不管读者是否开发过游戏,相信大家都一定对“游戏引擎”这个词不陌生。游戏引擎在很多年前就存在了,不过最近几年突然在前面加上了“跨平台”三个字。尽管给游戏引擎带了个“头套”,不过还是可以认为:这仍然是原来的游戏引擎,只是时代变了。

也许十多年前的游戏开发人员不会考虑那么多,因为那时是 Windows 一统天下。游戏

^① 尽管 Cocos2d-x 是以 2D 起家,但现在 Cocos2d-x 已经支持 3D 了。不过本书仍然沿用最初的习惯,将 Cocos2d-x 称为 2D 游戏引擎。

引擎大多数时候是指基于微软 DirectX 的游戏引擎,这样的游戏引擎,毫无疑问是运行在 Windows 操作系统下的。尽管在 Linux、Mac OS^① 等操作系统下已有与 DirectX 类似的 OpenGL。不过,由于当时 Linux 和 Mac OS 操作系统的用户数相对于 Windows 操作系统来说很少(其他系统的用户数就更少了),所以当时很少有人做 Linux 和 Mac OS 操作系统的游戏。而在当年,尽管第一代智能手机(以 Nokia 为老大的时代)可以运行手机游戏,但那时的智能手机远不能和现在的 iPhone 或 Android 相提并论。所以,十多年前的移动游戏仍属于发展的初期,远未达到一定的规模,所以当时专门开发移动游戏的公司也没有现在多。基于这些原因,十多年前的游戏市场主要指基于 Windows 的 PC 游戏市场。而当年的程序员要比现在的程序员容易得多,因为在 Windows 下开发游戏,大多数人都会使用 Visual C++ 来开发,而且程序员只需要在 Windows 下测试游戏即可。

然而自从苹果公司在乔帮主^②的带领下于 2007 年发布第一款 iPhone 以来,一切都改变了。开发游戏不再单单指 Windows 平台,而又多了一个 iOS 平台,随着后来的 Android、Windows Phone 等更多的移动系统的加入,我们所开发的游戏就遇到了需要在多个平台上部署的窘境。当然,最直接的方法是根据所部署的平台,使用不同的开发工具、语言和相关技术。不过这将会使开发成本骤然增加。例如,只为 Windows、Android 和 iOS 三种平台开发游戏,就需要使用 C++,Java 和 Objective-C 三种语言以及不同的 IDE。从理论上说,为这三个平台开发同一款游戏的成本会是只为单一平台开发该款游戏的成本的三倍,如果考虑到有些资源可以共享的情况,至少也在两倍以上。

尽管发布的平台越多,盈利就可能越多;但盈利并不是成倍增长的,而开发成本一般却成倍增长,而且由于各个团队的进度很难统一,所以极难保证各个平台的版本在同一时间发布。

为了解决这些棘手的问题,很多技术专家开发出了尽可能少编码,但却可以在多个平台上运行的技术,这就是跨平台技术。Cocos2d-x 就是其中之一。不过 Cocos2d-x 的跨平台模式只是众多跨平台模式中的一个,那么还有哪些跨平台模式呢?这些跨平台开发模式有什么优点值得我们去使用呢?“金无足赤、人无完人。”这些跨平台开发模式也会有不足,那么这些不足是否可以通过技术或人为的手段弥补呢?要回答这些问题,就请接着阅读下一节的内容,后面的内容更精彩!

1.2 跨平台开发模式的种类和优缺点

随着近几年各种操作系统平台的增多,跨平台的讨论也越演越烈。因为广大程序员不想同时开发功能几乎完全相同的多平台版本;对于那些创业公司或实力不够的中小型公

① 现在使用的 Mac OS 是 Mac OS X,这里的 X 是罗马数字,也就是阿拉伯数字中的 10,目前最新的是 10.11.4。以前还有 Mac OS 9、Mac OS 8 等。不过在 iOS 出现之前,Mac OS 的用户数相对于 Windows 来说,微不足道。

② 国人给前苹果 CEO 乔布斯起的绰号。乔帮主,金庸名著《天龙八部》的男主角乔峰。乔峰曾任丐帮帮主,在得知自己是契丹后裔后,被迫退位。苹果公司创始人史蒂夫·乔布斯曾被苹果放逐,二人身世沉浮,有相似之处,传奇不凡。