

Cocos 2d-x 游戏编程

—— C++篇

徐飞 著



基于引擎3.5版本

除基本2D编程外，还涉及3D编程

使用Shader编程实现部分计算机图形学和图像学方面的理论知识

在手机游戏中实现加解密、自动更新等功能



 中国工信出版集团

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

Cocos2d-x 游戏编程——C++篇

徐 飞 著

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

本书基于 Cocos2d-x 引擎,使用 C++作为开发语言,内容涉及从手机游戏开发到发布各方面的知识。本书分为 4 部分:基础篇、3D 篇、网络篇和高级篇。其中,基础篇包括 Cocos2d-x 基础知识、基础类 etc 知识;3D 篇包括引擎的 3D 知识、引擎中涉及的数学知识、可编程管道着色;网络篇包括 HTTP 编程和 Socket 编程;高级篇包括游戏中的适配知识、自动更新、跨平台编译和 iOS、Android 平台功能的开发。

本书适合 Cocos2d-x 初学者,同时也适合从事移动开发和有手游开发经验的广大程序员。

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 游戏编程. C++篇 / 徐飞著. —北京:电子工业出版社, 2015.9

ISBN 978-7-121-27029-1

I. ①C… II. ①徐… III. ①游戏程序—C 语言—程序设计 IV. ①TP311.5

中国版本图书馆 CIP 数据核字 (2015) 第 201103 号



策划编辑: 张 迪

责任编辑: 张 迪

印 刷: 北京京科印刷有限公司

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: 23.25 字数: 595.2 千字

版 次: 2015 年 9 月第 1 版

印 次: 2015 年 9 月第 1 次印刷

印 数: 3 000 册 定价: 69.80 元 (含 CD 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

随着智能手机的普及，相关的应用犹如雨后春笋般涌现。其中，游戏扮演着一个重要的角色，和端游、页游形成三足鼎立之势。例如，卡牌类游戏我叫 MT、大掌门、放开那三国、名将无双和萌江湖等；RPG 类游戏天龙八部 3D、仙剑奇侠传、全民奇迹和秦时明月等；跑酷类游戏神庙逃亡、天天酷跑、熊大快跑 2015、爸爸去哪儿 2、奔跑吧兄弟、天天风之旅等；塔防类游戏植物大战僵尸 2、保卫萝卜等；三消类游戏 PopStar、天天星连萌、天天爱消除和开心消消乐等；音乐类游戏节奏大师、乐动达人、全民炫舞和乐动魔方手机版(jukebeat)；街机类游戏捕鱼达人系列、拳皇 2014 和狂野飙车 8：急速凌云等不乏精品。

目前市面上两款比较火的手机游戏开发引擎要属 Unity3D 和 Cocos2d-x，前者多用于 3D 游戏的开发，后者在 3.0 版本之前是一款 2D 游戏开发引擎，在 3.0 后引入了对 3D 开发的支持，随着版本的升级，3D 开发功能也越来越完善。上面介绍的游戏，有很多是使用 Cocos2d-x 引擎开发的。例如，我叫 MT、大掌门、秦时明月、保卫萝卜、开心消消乐、放开那三国、仙剑奇侠传和捕鱼达人系列等。

本书以 Cocos2d-x 引擎为基础，以 C++ 作为开发语言，内容涉及从手机游戏开发到发布各个方面的知识，主要可以分为 4 篇：基础篇、3D 篇、网络篇和高级篇。其中，各个篇的内容如下。

1. 基础篇

基础篇共 6 章，各章的具体情况如下：

第 1 章介绍引擎的基础知识，内容包括 Cocos2d-x 引擎的组织结构、Hello World 项目的创建、版本变化和 C++ 11 标准中的一些知识。

第 2 章介绍引擎中的一些基础类和常用类，分为 6 个部分讲解：Director 类、高级节点类（如 Scene、Layer、Sprite 等）、文字相关类（如 LabelTTF、Label 和 Text 类等）、菜单类 Menu、列表类和一些其他常用类（如 SpriteBatchNode、Scale9Sprite 和 EditBox 等）。

第 3 章介绍动画，动画有多种类型，如瞬时动画、延时动画、缓冲动画、Speed 动画、Follow 动画、ActionTimeLine 动画和一些特效动画。

第 4 章介绍引擎中的定时器和事件响应机制。其中，定时器有全局和 Node 之分，事件响应负责处理界面单击事件等操作。

第 5 章介绍粒子系统，分为普通的 2D 粒子特效和 3D 粒子特效。

第 6 章介绍引擎中的声音，有两种方式：SimpleAudioEngine 和 AudioEngine。

2. 3D 篇

3D 篇有 3 章，各章的具体情况如下：

第 7 章介绍引擎中的 3D 知识，内容涉及模型、动画、相机、光照和碰撞检测等。

第 8 章讲解引擎中的数学知识，内容有向量、矩阵、四元素、Size 类和 Rect 类。

第 9 章从 GLSL 语言基础知识、Cocos2d-x 着色相关类、2D 和 3D 特效 4 个方面介绍可



编程管道着色。

3. 网络篇

网络篇有 2 章，各章的具体情况如下：

第 10 章介绍 HTTP 编程，内容涉及 libcurl 和 pthread 库的基础知识、游戏中基本的数据通信、文件的上传和下载，以及 C++ 11 标准中的线程库 std::thread。

第 11 章介绍 Socket 编程，内容有 select 函数介绍、Socket 编程中的一些常用函数和数据通信的实现。

4. 高级篇

高级篇有 5 章，各章的具体情况如下：

第 12 章介绍游戏中的适配知识，从资源、设计和屏幕分辨率三个方面来讲解。

第 13 章介绍在 Android 平台下怎样实现代码和资源的自动更新，分为 3 方面的知识：怎么实现代码和资源的管理、代码文件更新和资源文件更新。

第 14 章介绍游戏中资源的加密和解密操作，有 2.x 版本和 3.5 版本之分。

第 15 章介绍跨平台编译到 iOS 平台；iOS 平台下一些功能的开发，如经纬度、照相和应用内购买；真机调试和游戏的发布。

第 16 章介绍跨平台编译到 Android 平台；C++ 和 Java 语言的互调；该平台下一些功能的开发，如经纬度、照相等。

本书中涉及作者编写的类以“XJ”开头，和引擎 2.x 版本中的“CC”功能一样，只是一种命名的方式，前 9 章的知识都是基于引擎 3.5 版本编写的，在后几章知识的讲解中以引擎 2.2.3 版本为基础，但这部分涉及引擎的知识不多，大部分在光盘中都提供 2.2.3 和 3.5 两个版本的代码，读者在阅读的过程中可以对照参考，如果读者仔细阅读了第 1 章的知识，理解这两个版本的代码应该不是什么难事。

感谢我的研究生导师，北京理工大学的王全玉老师，是他把我带进虚拟现实的世界，让我毕业后就能从事游戏开发的工作；感谢电子工业出版社的王敬栋和张迪两位老师，没有二位老师的帮助，该书就不会顺利出版；感谢我的爷爷奶奶和父母，是他们把我抚育成人；感谢我亲爱的妻子和可爱的儿子，在我创业期间给予我无微不至的关怀和理解。

本书通过 16 章讲解游戏的编写，但由于作者水平有限，书中难免会存在缺点和不足，殷切希望广大读者通过电子邮件（cocos2dx_net@163.com）批评指正。

徐 飞

2015 年 5 月 3 日于深圳

目 录

第 1 章 Cocos2d-x 基础知识	1
1.1 引擎组织结构	1
1.2 从 2.x 到 3.x 的变化	3
1.2.1 类名变化	3
1.2.2 单件类变化	4
1.2.3 宏定义与结构的变化	4
1.2.4 回调函数变化	5
1.2.5 事件响应机制的变化	6
1.3 Cocos2d-x 中的 C++ 11 知识	7
1.3.1 Lambda 表达式	7
1.3.2 bind 函数	10
1.3.3 function 函数	11
1.3.4 auto 和 nullptr	12
1.3.5 override 和 final	13
1.4 小结	13
第 2 章 基础类	14
2.1 Director 类	15
2.1.1 场景函数	15
2.1.2 视图信息函数	16
2.1.3 管理器函数	17
2.1.4 控制函数	17
2.2 高级节点类	18
2.2.1 Node 类	18
2.2.2 Scene 类	24
2.2.3 Layer 类	26
2.2.4 Sprite 类	27
2.3 文字类	31
2.3.1 LabelTTF 方式	32
2.3.2 LabelAtlas 方式	32
2.3.3 LabelBMFont 方式	33
2.4 菜单类	33
2.4.1 文字按钮	34
2.4.2 图片按钮	35



2.4.3	切换按钮	35
2.4.4	XJMenu 按钮	37
2.5	列表类	40
2.5.1	ScrollView 类	40
2.5.2	TableView 类	42
2.6	其他类	45
2.6.1	SpriteBatchNode 类	45
2.6.2	DrawNode 类	46
2.6.3	Scale9Sprite 类	48
2.6.4	EditText 类	48
2.6.5	RenderTexture 类	50
2.7	小结	51
第3章	动画	52
3.1	ActionManager 类	52
3.2	瞬时动画	54
3.2.1	可见性动画	54
3.2.2	翻转动画	54
3.2.3	回调动画	55
3.2.4	Place 动画	55
3.2.5	RemoveSelf 动画	56
3.3	延时动画	56
3.3.1	移动动画	57
3.3.2	旋转动画	57
3.3.3	缩放动画	58
3.3.4	扭曲动画	58
3.3.5	跳跃动画	59
3.3.6	淡入淡出动画	59
3.3.7	Tint 动画	60
3.3.8	Blink 动画	61
3.3.9	Bezier 动画	61
3.3.10	DelayTime 动画	62
3.3.11	重复动画	62
3.3.12	Sequence 动画	62
3.3.13	Spawn 动画	63
3.3.14	序列帧动画	64
3.3.15	进度动画	65
3.4	缓冲动画	67
3.4.1	弹性动画	69
3.4.2	跳跃动画	69



3.4.3	指数动画	70
3.4.4	Sine 动画	70
3.4.5	回震动画	71
3.4.6	Ease 动画	71
3.5	Speed 动画	72
3.6	Follow 动画	73
3.7	ActionTimeline 动画	74
3.8	动画特效	75
3.8.1	Grid3DAction 动画	75
3.8.2	TiledGrid3DAction 动画	78
3.9	小结	80
第 4 章	定时器与事件响应机制	81
4.1	定时器	81
4.1.1	Scheduler 类	81
4.1.2	Node 定时器	83
4.2	事件响应机制	85
4.2.1	EventDispatcher 类	85
4.2.2	EventListener 类	87
4.2.3	触摸事件	88
4.2.4	键盘事件	89
4.2.5	鼠标事件	90
4.2.6	加速器事件	90
4.2.7	Event 类	91
4.2.8	EventTouch 类	91
4.2.9	EventKeyboard 类	92
4.2.10	EventMouse 类	93
4.2.11	EventAcceleration 类	93
4.3	小结	94
第 5 章	粒子系统	95
5.1	普通粒子系统	95
5.1.1	粒子结构	96
5.1.2	ParticleSystem 类	97
5.1.3	ParticleSystemQuad 类	98
5.1.4	实例	99
5.1.5	粒子编辑器	101
5.2	3D 粒子	102
5.2.1	ParticleSystem3D 类	103
5.2.2	PUParticleSystem3D 类	104
5.2.3	材质文件	105



5.2.4	粒子文件	105
5.2.5	粒子使用	106
5.3	小结	107
第 6 章	声音	108
6.1	SimpleAudioEngine 类	108
6.1.1	背景音乐	108
6.1.2	音效	109
6.2	AudioEngine 类	110
6.3	小结	112
第 7 章	Cocos2d-x 中的 3D 技术	113
7.1	模型	113
7.1.1	模型格式	113
7.1.2	Sprite3D 类	116
7.1.3	动画相关类	117
7.1.4	坐标系	118
7.2	模型挂接与换装	119
7.2.1	挂接	119
7.2.2	换装	121
7.3	相机	123
7.3.1	相机类	123
7.3.2	相机类型	124
7.3.3	遮罩	127
7.3.4	视锥体裁剪	128
7.3.5	自由相机	128
7.3.6	弧度相机	130
7.3.7	第一人称相机	132
7.3.8	第三人称相机	132
7.4	光照	133
7.4.1	灯光基础	133
7.4.2	环境光	134
7.4.3	方向光	136
7.4.4	点光源	136
7.4.5	聚光灯	137
7.5	碰撞检测	138
7.5.1	2D 包围盒检测	139
7.5.2	距离检测	139
7.5.3	AABB 检测	139
7.5.4	OBB 检测	141
7.5.5	Ray 检测	141



7.6	Billboard	143
7.7	多线程资源加载	144
7.8	2D 与 3D 混合	146
7.9	DrawNode3D 类	147
7.10	展望	148
第 8 章	数学库	149
8.1	向量	149
8.2	矩阵	153
8.2.1	矩阵的创建	155
8.2.2	向量和值的提取	156
8.2.3	零矩阵和单位矩阵	156
8.2.4	负矩阵	156
8.2.5	矩阵算术运算	157
8.2.6	平移、旋转和缩放	158
8.2.7	矩阵转置	159
8.2.8	方阵	159
8.2.9	逆矩阵	160
8.3	四元素	160
8.3.1	四元素的创建	160
8.3.2	零四元素和单位四元素	161
8.3.3	负四元素	161
8.3.4	四元素的模	161
8.3.5	四元素共轭	162
8.3.6	逆四元素	162
8.3.7	四元素点乘	162
8.3.8	四元素叉乘	162
8.3.9	线性插值	163
8.3.10	slerp 插值	163
8.3.11	squad 插值	164
8.3.12	四元素的运用	164
8.4	Size 类	165
8.5	Rect 类	165
8.6	小结	166
第 9 章	Shader 编程	168
9.1	GLSL 基础	168
9.1.1	Hello World	168
9.1.2	数据类型	169
9.1.3	数组和结构	171
9.1.4	类型限定	172



9.1.5	参数限定	173
9.1.6	操作符	173
9.1.7	流程控制	174
9.1.8	函数	175
9.1.9	顶点着色	175
9.1.10	片段着色	176
9.1.11	内置变量	176
9.1.12	内置函数	176
9.2	Cocos2d-x 对 Shader 的处理	178
9.2.1	GLProgram	179
9.2.2	GLProgramCache	180
9.2.3	GLProgramState	182
9.2.4	GLProgramStateCache	182
9.3	2D 特效	183
9.3.1	水平翻转	183
9.3.2	垂直翻转	184
9.3.3	图像变灰	185
9.3.4	图像反色	185
9.3.5	图像噪点	186
9.3.6	图像模糊	187
9.3.7	马赛克	188
9.3.8	浮雕效果	188
9.3.9	边缘检测	189
9.4	3D 特效	190
9.4.1	环境光	191
9.4.2	散射光	192
9.4.3	镜面光	194
9.4.4	雾	195
9.4.5	波光	197
9.4.6	UV 动画	200
9.4.7	伪阴影	201
9.4.8	卡通着色	203
9.4.9	模型轮廓	204
9.5	小结	210
第 10 章	HTTP 编程	211
10.1	libcurl 基础知识	211
10.1.1	global 函数	212
10.1.2	init 和 cleanup 函数	212
10.1.3	setopt 函数	212



10.1.4	表单函数	214
10.1.5	perform 函数	217
10.2	pthread 基础知识	218
10.2.1	线程管理	218
10.2.2	互斥变量	221
10.2.3	条件变量	223
10.3	基本数据通信	225
10.4	文件下载	236
10.5	文件上传	241
10.6	std::thread	242
10.6.1	线程创建	242
10.6.2	阻塞	242
10.6.3	互斥	242
10.6.4	条件变量	243
10.7	小结	243
第 11 章	Socket 编程	244
11.1	select 介绍	244
11.2	Socket 常用函数	246
11.2.1	socket	246
11.2.2	bind	247
11.2.3	listen	248
11.2.4	accept	248
11.2.5	connect	249
11.2.6	send	249
11.2.7	recv	250
11.2.8	sendto	250
11.2.9	recvfrom	250
11.2.10	closesocket	251
11.2.11	shutdown	251
11.2.12	gethostname	251
11.2.13	gethostbyname	251
11.2.14	getpeername	252
11.2.15	setsockopt	252
11.2.16	getsockopt	252
11.3	通信实现	253
11.3.1	XJSocket 类	253
11.3.2	XJSocketIO 类	259
11.3.3	XJSocketIOMgr 类	277
11.3.4	XJSocketData 类	279



11.4	小结	282
第 12 章	适配	283
12.1	资源分辨率	283
12.2	设计分辨率	284
12.2.1	kResolutionExactFit 适配	285
12.2.2	kResolutionNoBorder 适配	289
12.2.3	固定宽或高适配	296
12.3	屏幕分辨率	297
12.4	小结	298
第 13 章	自动更新	299
13.1	文件版本管理	299
13.2	so 文件更新	302
13.3	资源文件更新	309
13.3.1	XJFileUtil 类	310
13.3.2	XJVersionMgr 类	311
13.3.3	UpdateMgr 类	313
13.4	小结	315
第 14 章	加解密	316
14.1	AES 算法	316
14.2	加密	319
14.3	解密	319
14.3.1	2.x 解密	320
14.3.2	3.5 版本解密	320
14.4	小结	322
第 15 章	iOS	323
15.1	平台编译	323
15.2	基本设置	324
15.2.1	info.plist 文件	324
15.2.2	Icon 和 Launchimage 设置	324
15.2.3	横竖屏	325
15.3	真机调试	326
15.3.1	证书	326
15.3.2	App ID	327
15.3.3	设备	327
15.3.4	配置文件	328
15.4	In-App Purchase	328
15.4.1	Store Kit 框架类	329
15.4.2	创建发布程序	330
15.4.3	获取产品列表	330



15.4.4 购买	332
15.4.5 验证	334
15.5 经纬度	334
15.6 照相	337
15.7 发布	342
第 16 章 Android	344
16.1 Win32 平台下开发环境搭建	344
16.1.1 安装配置 python	344
16.1.2 Cocos2d-x 2.x 项目创建	344
16.1.3 Cocos2d-x 3.x 项目创建	344
16.2 Android 开发环境搭建	345
16.2.1 JDK	345
16.2.2 Android SDK	346
16.2.3 NDK	346
16.2.4 Eclipse 编译	346
16.3 基本知识	347
16.3.1 AndroidManifest 文件	347
16.3.2 Android.mk 文件	347
16.3.3 启动方式	348
16.3.4 签名	348
16.4 C++与 Java 互调	349
16.4.1 C++部分编写	349
16.4.2 Java 部分编写	350
16.5 经纬度	350
16.5.1 getLastKnownLocation 函数	351
16.5.2 条件查找	352
16.6 获取图片	353
16.7 应用关闭	357
16.8 小结	358

Cocos2d-x 基础知识

Cocos2d-x 是一个跨平台且开源的游戏引擎，诞生于 2010 年。它允许开发人员使用 Windows 和 Mac 桌面系统进行开发和调试，可选择 C++、Lua 或 JavaScript 作为开发语言，游戏开发完成后，可部署到 iOS、Android 和 Windows Phone 等平台，实现了一套代码，多个平台运行，大大节约了开发时间和成本。

引擎本身使用 C++ 语言开发，基于 OpenGL ES 渲染引擎，除了渲染部分外，还封装了 2D 物理引擎、网络通信、声音引擎等，是一个比较完整的游戏开发引擎。它提供了丰富的 API 集合，供开发者使用，且 API 文档浅显易懂，开发人员很容易入手。

引擎自从发布后，经历了多个版本的变化。其中，最大的是从 2.x 到 3.x 版本的变化。在 3.x 版本中，加入了 3D 编程的支持、新的声音引擎、增加了 3D 粒子特效（3.5 版本中加入）和引入了 C++ 11 标准等，也遗弃了 2.x 版本中一些内容，如 pthread 线程库、CCString 类等。本章将先介绍引擎最新版本关于框架的一些知识和怎么创建一个 HelloWorld 程序，然后再介绍引擎从 2.x 到 3.x 版本发生了哪些变化，最后介绍 C++ 11 标准的一些知识。

1.1 引擎组织结构



读者可以到 Cocos2d-x 的官网（www.cocos2d-x.org）获取源代码，它提供了最近几个版本源文件的下载地址。在写本书时，引擎的最新版本是 3.5，所以本书大部分知识都是基于这个版本讲解的，下载完成后是一个 zip 压缩文件，解压出来后，可以看到如图 1-1 所示的目录结构。

从图 1-1 中可以看出，共 11 个目录，下面介绍其中一些重要的目录：

- **build**：Win32 平台下的工程文件，用 VS2012 打开 cocos2d-win32.vc2012.sln 文件，可直接对引擎源码和例子进行编译。
- **cocos**：该目录下包括了引擎的核心代码文件，如 2D、3D、声音（audio）、网络（network）、物理（physics）、数学库（math）、渲染（renderer）及与平台相关的一些代码文件等。
- **extensions**：资源更新管理、UI 控件和 3D 粒子特效（Particle3D）相关类文件存放的地方，其中控件有 ScrollView、TableView 等。
- **external**：包含一些外部库，如物理库（Box2D 和 chipmunk）、curl 库、json 文件解析库、XML 文件解析库（tinyxml2）、sqlite3 和 unzip 等。
- **templates**：存放着创建 C++ 和 lua 项目的模板类和资源。

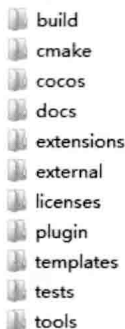


图 1-1 Cocos2d-x 3.5 目录结构



- tests: 包含 C++ 和 Lua 项目的 Samples。
- tools: 项目创建、lua bindings 等相关工具。

对引擎的目录结构大致了解后,进入 build 目录下,用 VS2012 打开 cocos2d-win32.vc2012.sln 这个文件,在解决方案标签下可以看到如图 1-2 所示引擎的工程结构。

从图 1-2 中可以看出,在解决方案下,一共有 8 个工程,它们各自的功能如下:

- libbox2d: 2D 物理库。
- libSpine: 角色骨骼动画库。
- libcocos2d: Cocos2d-x 引擎库,引擎的核心代码都在该工程下。
- libluacocos2d: 引擎对 lua 支持库。
- cpp-tests: C++ 语言编写的 Samples 工程。
- cpp-empty-test: C++ 语言空工程。
- lua-tests: lua 语言 Samples 工程。
- lua-empty-test: lua 语言空工程。

对整个解决方案进行了解后,重新编译它,成功后,把 cpp-tests 工程设为启动项,运行该项目,可以看到 C++ 实例工程项目的界面,如图 1-3 所示。

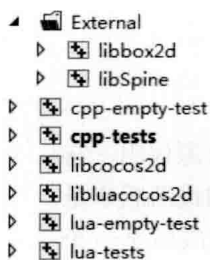


图 1-2 Cocos2d-x 项目结构图

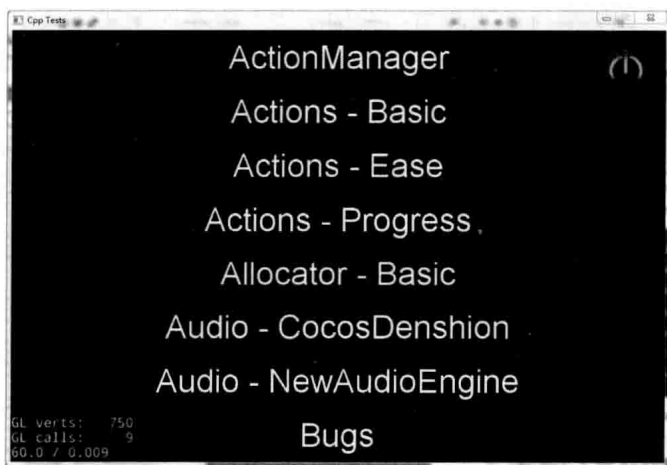


图 1-3 cpp-tests 工程运行界面

cpp-tests 工程是引擎提供的 Samples,几乎囊括了它各个方面知识的运用,这对于初学者是一份很好的学习材料。引擎提供的源代码项目编译成功后,是生成属于我们自己游戏项目的时候了。前面讲过,引擎根目录的 tools 目录下有创建项目的工具,在创建项目之前要用到 Python 这个脚本工具,所以到 Python 的官网下载一个,使用 2.x 版本的,最新版本是 2.7.9,下载完成后安装即可,然后使用 Windows 的命令行工具,进入“tools\cocos2d-console\bin”目录下,输入如下命令:

```
python cocos.py new mygame -l cpp -p xjs.mygame.com
```

对于这条命令的解释,将有专门的章节讲述,在这里我们只要了解,执行完这条命令后,将会在 bin 目录下创建一个名为 mygame 的项目,这是一个 HelloWorld 的项目,该项目的目录结构如图 1-4 所示。



该工程的目录结构解释如下：

- **Classes**: 项目的类文件，以后我们自己写的 C++ 文件都放到该目录下。
- **Resources**: 项目的资源文件，图片、声音、字体等都放在该目录下。
- **cocos2d**: 引擎相关文件。
- **proj** 目录: 平台相关目录。

了解完 mygame 的目录结构后，在 proj.win32 目录下找到解决方案 mygame.sln，双击打开它，编译执行后，可以看到如图 1-5 所示的界面。



图 1-4 mygame 目录结构



图 1-5 mygame 项目的 HelloWorld 界面

到此，本节分析了引擎的组织结构和怎么去创建一个属于我们自己的 HelloWorld 项目，在本书其他章节的代码讲解中，将以这个项目为基础，大部分知识的讲解都建立在这个项目上。

1.2 从 2.x 到 3.x 的变化



引擎在 3.0 后，在保持引擎整个框架结构不变的前提下，较前一个版本还是发生了比较大的变化，其中包括类名、回调函数、事件响应机制等，并且增加了 C++ 11 标准的一些知识。在 2.x 中以宏出现的功能函数和结构，在 3.x 版本中都通过类的封装来实现，这样使得程序更加简洁，使初学者更加容易理解。在本小节中，将介绍主要的一些变化，使 2.x 编程者很快过渡到 3.x 版本。



1.2.1 类名变化

引擎中，各个类对应的头文件（.h）和执行文件（.cpp）虽然在命名上没有变化，但是里面定义的类在命名上发生了变化，去掉了之前的 CC 前缀。例如，要在界面上显示一张图片，实现程序如程序片段 1-1 所示。

程序片段 1-1 显示图片代码

```
//2.x 显示一张图片  
CCSprite* pSprite = CCSprite::create("test.png");
```