

国内首本系统论述开源跨平台移动开发引擎CrossApp的原理、架构与开发实战的入门经典著作！

龙图游戏李翀 作序



官方推荐用书

清华

开发者书库



Getting Started with CrossApp

CrossApp

入门指南

沈大海 张磊◎编著

Shen Dahai Zhang Lei

清华大学出版社



清华

开发者书库



Getting Started with CrossApp

CrossApp

入门指南

沈大海 张磊◎编著

Shen Dahai Zhang Lei

清华大学出版社
北京

内 容 简 介

本书系统论述了 CrossApp 跨平台 App 开源引擎的开发理论与实践。全书内容涵盖了 CrossApp 引擎特点,开发环境设置,核心类 CAView、CAViewController 和 CAWindow 的实现原理和使用,CrossApp 内存管理机制,CrossApp 核心控件使用,设备功能调用及网络通信功能等。本书共 8 章,分为如下三大部分。

第一部分为开发基础,即第 1~4 章,内容包括 CrossApp 开发环境搭建,引擎原理介绍,创建项目和核心 UI 组件类的使用。通过该部分内容的学习,读者可以创建一个简单的跨平台应用。

第二部分为开发进阶,即第 5~7 章,内容包括在 CrossApp 项目中使用多媒体功能,使用文件存储功能,实现网络功能,实现 UI 组件的动画效果。通过该部分内容的学习,读者可以实现一款功能强大的跨平台应用。

第三部分为项目实战,即第 8 章,通过一款手机电商 App 的源码解析,可以让读者具备架构一款大型跨平台联网 App 的能力。

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

CrossApp 入门指南/沈大海,张磊编著.--北京:清华大学出版社,2016

(清华开发者书库)

ISBN 978-7-302-42134-4

I. ①C… II. ①沈… ②张… III. ①移动终端—应用程序—程序设计 IV. ①TN929.53

中国版本图书馆 CIP 数据核字(2015)第 267383 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

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:北京嘉实印刷有限公司

经 销:全国新华书店

开 本:186mm×240mm 印 张:9.75

字 数:247 千字

版 次:2016 年 1 月第 1 版

印 次:2016 年 1 月第 1 次印刷

印 数:1~2000

定 价:39.00 元

产品编号:065985-01

目录

CONTENTS

第 1 章 CrossApp 简介及开发环境搭建	1
1.1 CrossApp 简介	1
1.1.1 CrossApp 功能与特色	1
1.1.2 CrossApp 的优势	2
1.2 CrossApp 开发环境搭建	3
1.2.1 Windows 开发环境搭建	3
1.2.2 在 Windows 系统创建 CrossApp 工程	5
1.2.3 CrossApp 项目目录说明	6
1.2.4 通过 Visual Studio 2013 启动一个项目	7
1.2.5 Windows 环境下 Android 配置	9
1.2.6 Mac OS X 开发环境搭建	12
1.2.7 Mac 下配置 Android 开发环境	13
1.2.8 第一个 CrossApp 项目解析	14
第 2 章 CrossApp 基础概念	19
2.1 核心类	19
2.1.1 CAView	19
2.1.2 CAViewController	21
2.1.3 CAWindow	22
2.2 内存管理	22
2.2.1 对象内存引用记数	22
2.2.2 手工对象内存管理	24
2.2.3 自动对象内存管理	24
2.3 坐标系	26
2.4 适配方案	29
2.5 深入理解 CAViewController 和 MVC	30
2.5.1 CAViewController 的职责	30

2.6	CAViewController 类的使用	32
2.6.1	CAViewController 生命周期	33
2.6.2	CAViewController 使用	33
2.7	CANavigationController 导航视图控制器	37
2.8	CATabBarController 切换视图控制器	39
2.9	CADrawerController 侧边抽屉式导航控制器	42
第3章	CrossApp 核心控件与视图	44
3.1	文本 CALabel	44
3.2	按钮 CAButton	46
3.3	图片 CAImageView	52
3.4	九宫格图片 CAScale9ImageView	54
3.5	单行输入框 CATextField	56
3.6	多行输入框 CATextView	60
3.7	开关 CASwitch	63
3.8	提示框 CAAlertView	65
3.9	进度条 CAProgress	68
3.10	滚动条 CASlider	70
3.11	步进控件 CASTepper	72
3.12	滚动视图 CAScrollView	75
3.13	列表视图 CAListView	79
3.14	表格视图 CATableView	86
3.15	容器 CACollectionView	96
3.16	切换页面 CAPageView	102
第4章	CrossApp 数据存储与解析	105
4.1	CAUserDefault 简单存储	105
4.2	SQLite 的使用	107
4.3	JSON 解析	109
4.4	XML 解析	111
第5章	CrossApp 设备功能调用	115
5.1	相机	117
5.2	相册	119
5.3	通讯录	121
5.4	本章小结	127

第 6 章 CrossApp 多媒体	128
6.1 CAViewAnimation 动画	128
6.2 SimpleAudioEngine 音效	130
第 7 章 CrossApp 网络通信	133
7.1 HTTP 基础使用	133
7.2 HTTP 加载网络图片	135
第 8 章 CrossApp 项目实战	137
8.1 折 800 开源项目介绍	137
8.2 项目架构设计	138
8.3 核心模块说明	140
8.4 本章小结	146

1.1 CrossApp 简介

CrossApp 是一款完全开源,免费和跨平台的移动应用开发引擎,基于最宽松的 MIT 开源协议,开发者可以根据自身情况使用 CrossApp 开发任何商业项目。CrossApp 以 C++ 作为开发语言,图形渲染基于 OpenGL ES 2.0,采用 MVC 框架模式。使用 CrossApp 开发的应用程序支持导出到各大主流移动平台,真正实现一次编码,多处运行的跨平台开发技术。

CrossApp 主要由 9 秒社团(www.9miao.com)自研,官方制定了 CrossApp 的基本架构,确立了 CrossApp 的基本雏形,其后的版本也将由来自各方的开发精英自愿组成的 9 秒社团常务贡献委员会进行更新和维护。

CrossApp 的官方网站为 <http://crossapp.9miao.com/>。

最新版本源码的下载地址: Oschina: <http://git.oschina.net/9miao/CrossApp> 和 Git Hub: <https://github.com/9miao/CrossApp>。

1.1.1 CrossApp 功能与特色

1) 跨平台

如图 1-1 所示,CrossApp 主要支持目前最主流的移动平台 iOS 和 Android,后续更新版本将完善更多平台的支持。

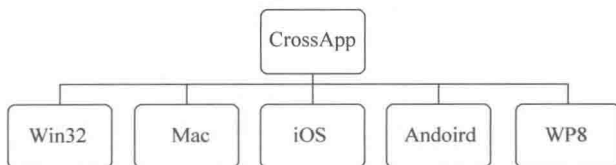


图 1-1 CrossApp 所支持的平台

2) 整合第三方库

CrossApp 整合一些第三方库,例如常见的数据解析库 JsonCpp、TinyXML 和 HTTP 等。

3) 基于 OpenGL ES 2.0

CrossApp 的图形渲染使用的是 Open GL ES 2.0,渲染效率高,可以使移动设备的 GPU 发挥到最佳效果。

4) 丰富的 UI 控件

CrossApp 的设计宗旨在于为移动应用开发者提供快速、高效的开发解决方案。在此基础上,CrossApp 封装了大量的 UI 控件,各类控件的功能十分丰富,开发者可以直接使用这些控件进行应用的开发。这些控件基本满足应用开发需求中的大部分功能,也免去了封装 UI 控件花费的大量时间,能进一步提升开发的效率,节省开发的成本。同时 CrossApp 也可以整合部分由第三方开发者提供的优秀控件资源,更大地丰富 CrossApp 的 UI 控件。

5) CrossApp 耗电优化

由于之前 CrossApp 抽取了 Cocos2d-x 的渲染驱动模式,在程序生命周期中不断重绘,渲染驱动模式的缺点就是 CPU 占用高,因此耗电量大。这种不断的重绘方式对于游戏可能影响不大,但对于 App 来说太浪费了。因此 CrossApp 针对应用程序的特性,将渲染机制改为事件驱动模式,这种模式的渲染要有外界触发才会重绘,在没有外界触发的时候画面静止,渲染停止,以达到节能的效果。

1.1.2 CrossApp 的优势

通过表 1-1 分析可以得出如下结论。

1) Mobile Web

以 HTML5 和 JQuery 为代表的 Web 开发技术是以纯浏览器为基础的,所以没有离线能力,更无法充分发挥各平台的功能和特性。

2) Hybrid App

Hybrid App 即混合模式应用程序,是介于 Web 和 Native 之间的 App,具备一定 Native 原生 App 的优秀用户体验和 Web App 跨平台的优势。但由于 Hybrid 仍旧以前端技术为基础,无法对内存和系统资源进行有效的管理。

3) Interpreted App

以前端技术为基础,同样不能有效地对内存和系统资源进行管理。

4) Native App

原生开发以 iOS 和 Android 为代表,虽然原生开发能够实现最佳的用户体验和高优化,但开发的成本较高。而且因为原生开发几乎不支持跨平台特性,相对于跨平台技术来说,实际效益更低。

综上所述,Native App 和 Cross Compiling 的综合效率更高,开发者需要根据实际情况,考虑各方面因素来选择合适的开发解决方案。

表 1-1 Cross App 与其他跨平台引擎对比

说明 \ 技术	Mobile Web	Hybrid App	Interpreted App	Cross Compiling	Native App
代表产品	HTML5/JQuery Mobile	PhoneGap	Titanium	CrossApp	iOS/Android
跨平台性能	强	强	中	强	低
离线能力	无	有	有	有	有
入门要求	低	低	低	低	高
功能	弱	中	中	强	强

1.2 CrossApp 开发环境搭建

CrossApp 可以分别在 Windows 平台和 Mac OS 平台上配置开发环境,展示良好的跨平台性,我们推荐大家使用 Mac OS 平台做 CrossApp 的开发工作,这样能更方便地在 iOS 平台和 Android 平台进行测试,得到及时的 bug 反馈。

当然也可以先在 Windows 平台开发后,再转到 Mac OS 发布 iOS 版本,但需要注意转码问题。

1.2.1 Windows 开发环境搭建

Windows 环境开发 CrossApp 系统需求如下。

- (1) 操作系统: Windows 7 以上版本。
- (2) 开发工具: Visual Studio 2012 以上版本(推荐 Visual Studio 2013)。
- (3) Python: 推荐 Python 2.7 版本。

首先下载 CrossApp 引擎源码:

```
github 地址: https://github.com/9miao/CrossApp
oschina 地址: http://git.oschina.net/9miao/CrossApp
```

1. Python 脚本创建

- 1) 双击安装 Python
- 2) 配置环境 Path

右击“计算器(我的电脑)”→“属性”→“高级系统设置”→“环境变量”→“系统变量”,双击 Path,在变量值结尾添加“;C:/Python27”(引号内容为 Python 的安装路径,注意前面有一个分号)如图 1-2 所示。

- 3) 测试 Python 环境

如图 1-3 所示在命令窗口输入 python -version,如果出现 Python 2.7.3 说明 Python 安装成功。

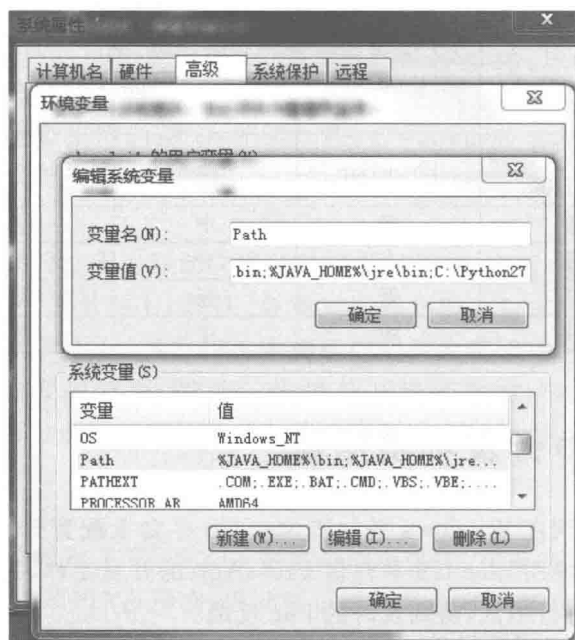


图 1-2 设置 Python 环境变量

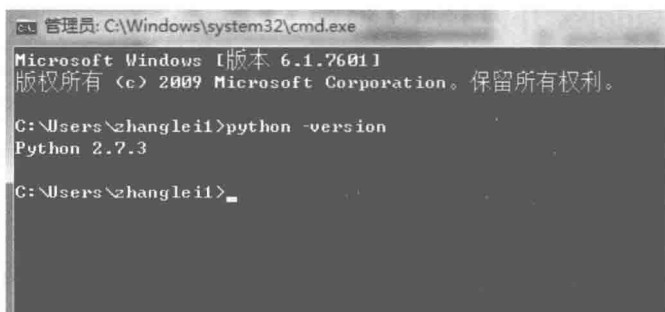


图 1-3 Python 环境配置

2. 解压缩 CrossApp 项目源码

将下载的 CrossApp 项目源码解压缩,目录结果如图 1-4 所示。

- (1) CocosDenshion: 包含了 CrossApp 多媒体 API 的源码。
- (2) CrossApp: 包含了 CrossApp 引擎的核心源码。
- (3) document: 引擎相关文档。
- (4) extensions: 引擎扩展库 JSON、网络和 SQLite 等的源码。
- (5) licenses: 引擎的相关开源协议说明。
- (6) project-creator.app: 在 Mac 平台创建项目的应用程序。

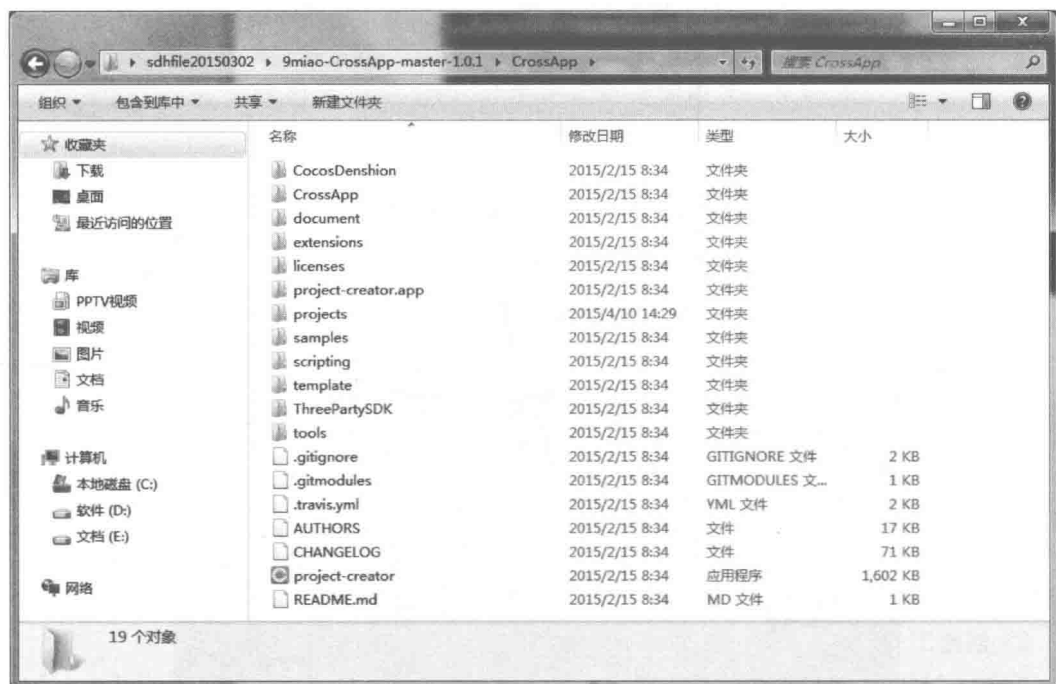


图 1-4 CrossApp 项目源码目录

(7) projects: 项目文件夹(该文件夹会在创建第一个项目后自动创建,源码包里并不包含)。

(8) samples: CrossApp 官方 Demo。

(9) scripting: 对 JS 脚本的支持源码。

(10) template: CrossApp 项目模板。

(11) ThreePartSDK: 第三方库 ShareSDK 和 UMSDK。

(12) tools: 引擎相关工具

1.2.2 在 Windows 系统创建 CrossApp 工程

在 Windows 系统创建 CrossApp 工程有如下两种方法。

1. 使用工具一键创建(无须 Python 环境)

在 CrossApp 的引擎目录下,双击 project-creator.exe,弹出如图 1-5 所示窗口。

第一栏填写工程名,第二栏填写 Android 包名,然后单击 Create Project Now,完成 CrossApp 项目的创建。

创建工程会在引擎的根目录多出一个名为“projects”的文件夹,创建的工程也就在这个目录下。



图 1-5 创建第一个 CrossApp 项目

2. Python 脚本创建

1) 配置环境 Path

右击“计算器(我的电脑)”→“属性”→“高级系统设置”→“环境变量”→“系统变量”，然后双击 Path，在变量值结尾添加：“;引擎根目录\tools\”。

2) 创建工程

打开 CMD 命令提示符，输入：python create_project.py -project HelloCrossApp -package com.9miao.crossapp -language cpp，如图 1-6 所示。

```
D:\CrossApp\tools\project-creator>python create_project.py -project HelloCrossApp
-p -package com.9miao.crossapp -language cpp
proj.ios : Done!
proj.android : Done!
proj.win32 : Done!
proj.mac : Done!
proj.linux : Done!
proj.wp8-xaml : Done!
New project has been created in this path: D:\CrossApp\tools\project-creator\..\
..\projects\HelloCrossApp
Have Fun!

D:\CrossApp\tools\project-creator>
```

图 1-6 使用命令行创建 CrossApp 项目

看到提示成功的信息，就说明工程创建成功了。

其中命令行结构如下：

Python create_project.py -project 项目名称-package 包名-language 开发语言。

1.2.3 CrossApp 项目目录说明

新建完项目之后，projects 的文件夹目录结构如下：

- (1) Classes: 包含新项目的源码。
- (2) Resources: 新项目的资源文件，包括图片、声音和文本等。

- (3) Proj. android: 新项目的 Android 工程, 可以使用 ADT 打开。
- (4) Proj. ios: 新项目的 iOS 工程, 可在 Mac 平台使用 Xcode 打开。
- (5) Proj. mac: 新项目的 Mac 工程, 可在 Mac 平台使用 Xcode 打开。
- (6) Proj. win32: 新项目的 Win32 工程, 可在 Windows 平台使用。

1.2.4 通过 Visual Studio 2013 启动一个项目

打开项目目录下: \proj.win32\HelloCrossApp.sln(创建的工程名)。

(1) 启动 Visual Studio 2013 后, 右击 HelloCrossApp, 设置为启动项目, 如图 1-7 所示。

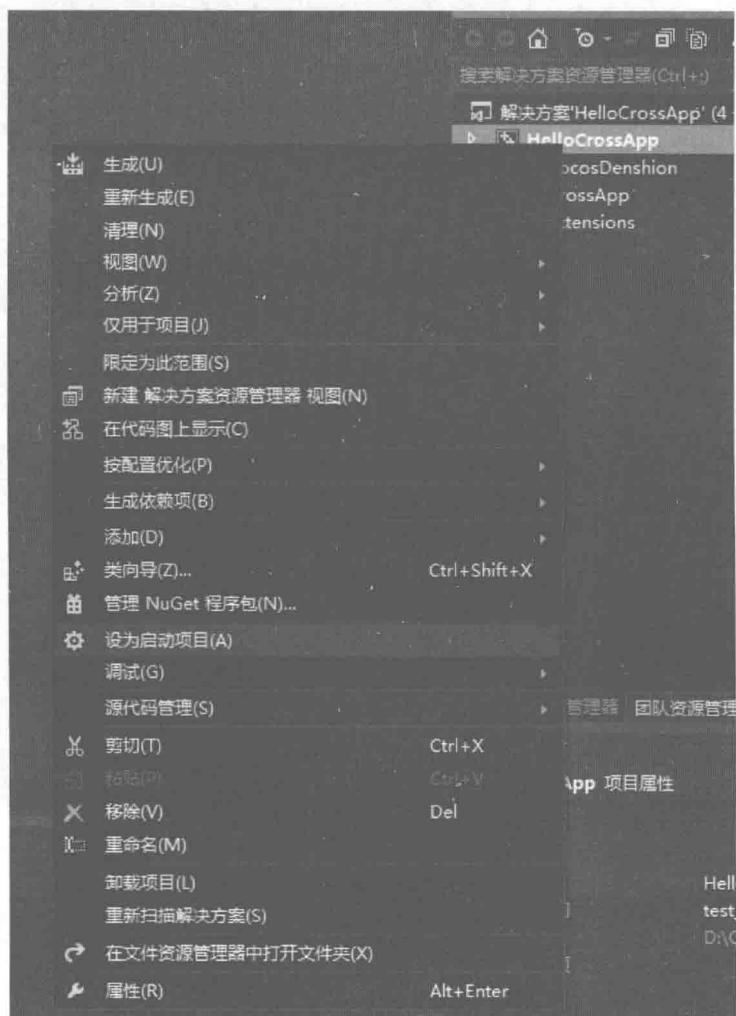


图 1-7 在 Visual Studio 2013 中打开项目

(2) 右击 HelloCrossApp→“调试”→“启动新实例”,如图 1-8 所示。

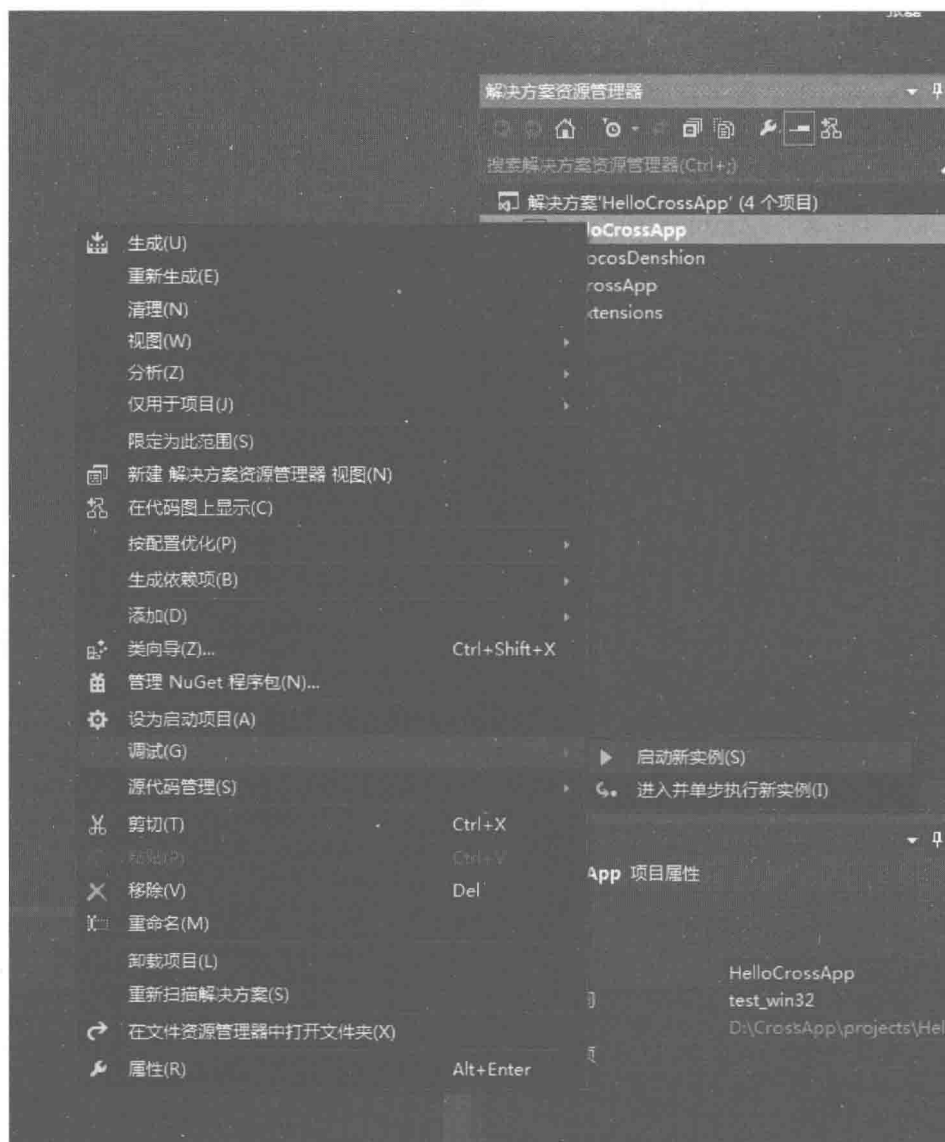


图 1-8 启动新实例

(3) 第一次启动等待时间较长,如果没有错误信息,将会弹出 CrossApp 项目窗口,如图 1-9 所示。

本书以在 Windows 环境下使用 Visual Studio 2013 为开发环境介绍 CrossApp 的开发技术。

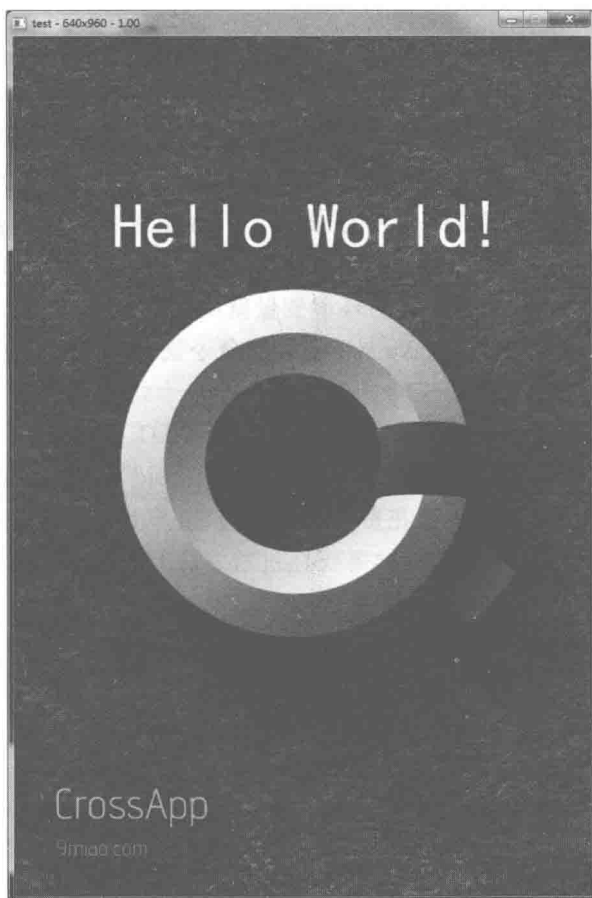


图 1-9 第一个 CrossApp 项目

1.2.5 Windows 环境下 Android 配置

移植 Android 的环境配置所需的工具如下：

(1) adt-bundle-windows-x86_64-20130917

下载地址：<http://developer.android.com/sdk/index.html>

(2) android-ndk-r9c

下载地址：<http://developer.android.com/tools/sdk/ndk/index.html>

(3) jdk-7u45-windows-x64

下载地址：<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

以上列出的工具也可以使用其他的版本，NDK 的 r8 以上版本，配置方法基本一样，请根据操作系统型号选择配置环境所需的工具包。

1. 安装 Java 环境

配置 Android 环境必须要先安装 Java 环境,安装过程并没有特别要求。

安装完成后需要配置环境变量,在 Windows 系统变量中新建一个变量名为 JAVA_HOME 的变量,变量值为 JDK 的安装目录,例如: C:\Program Files\Java\jdk1.7.0_45。然后新建一个名为 CLASSPATH 的变量,变量的值为:;%JAVA_HOME%\lib;%JAVA_HOME%\lib\tools.jar(最前面的“.”一定要加上),在系统变量中找到 Path 变量,点击编辑,在变量值最前面添加一个分号,然后在分号的前面添加%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin。这样 Java 的环境变量就配置好了。

打开 DOS,输入 java、javac 和 java -version 等命令就可以看见相关的信息。

2. 更新 ADT 以及配置 NDK

将下载好的 ADT 和 NDK 解压在指定目录,运行 ADT 目录下的 eclipse。如果提示要求更新,在 eclipse 的菜单栏中单击 Windows,选择 Android SDK Manager 在线更新 SDK,有更新的话尽量全选,更新的过程会花费较多的时间,直至 SDK 更新完成(后面的在 Xcode 下配置 Android 环境,如果有需要,同样需要经过这一步)。

3. 移植 Android

将工程导入 eclipse,File→New→Other→Android Project from Existing Code,单击 Browse 找到我们创建的工程目录下的 proj.android,如图 1-10~图 1-12 所示。

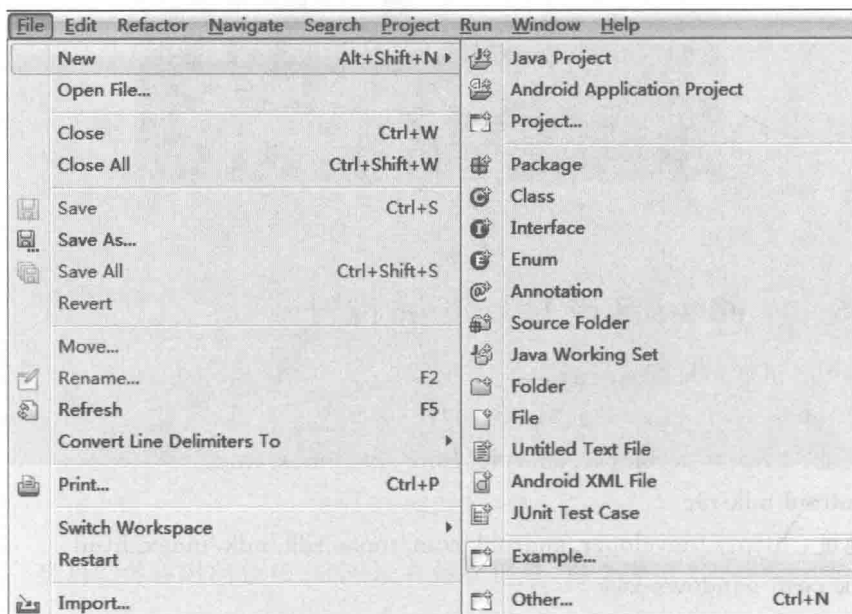


图 1-10 在 ADT 中导入 CrossApp 项目(一)

工程导入 eclipse 后,编译运行过程中所出现的异常和错误以及对应的解决办法如表 1-2 所示。

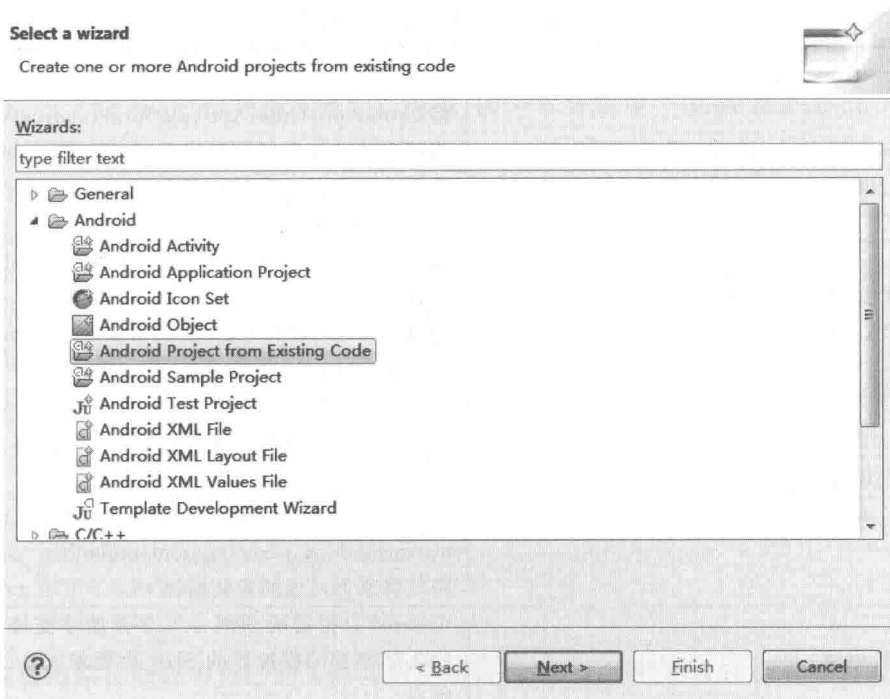


图 1-11 在 ADT 中导入 CrossApp 项目(二)

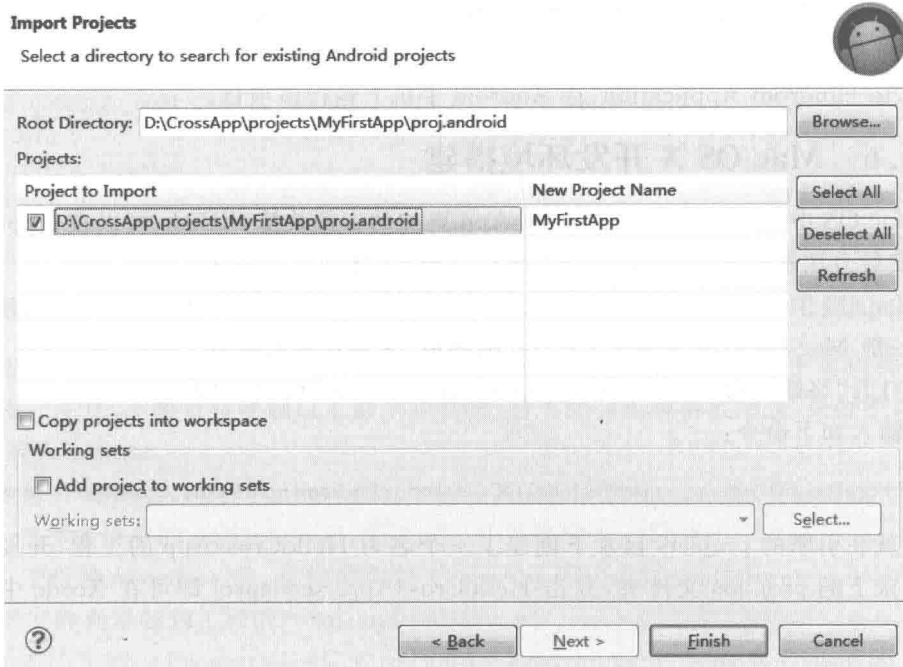


图 1-12 在 ADT 中导入 CrossApp 项目(三)