

第2章 搭建跨平台的开发环境

所谓“工欲善其事，必先利其器”。学习 Cocos2D-x 的开发技巧，首先从搭建跨平台的开发环境开始。

目前，Cocos2D-x 2.0 以上版本支持 Win32、Android、iOS 三个平台的开发，所以本章分别介绍在这三个环境下的开发环境搭建。一个完整的开发环境包括开发、编译、链接、运行和调试等几部分，所以需要相应的开发环境，比如 Win32 环境下的 VS（Visual Studio，微软公司集成开发环境），Android 的开发环境 Eclipse 和 Android SDK 等，iOS 则需要 Xcode 环境，然后使用 Cocos2D-x 的模板编译出相应环境的软件包。

一般情况下，使用 Cocos2D-x 的开发流程是：首先在 VS 环境中进行开发调试，之后分别在另外两个平台的开发环境中编译生成相应的软件包。当然，如果是首先使用 Cocos2D-iPhone 进行开发，并使用 Cocos2D-x 进行 Android 版本移植的，一般先在 Xcode 环境中进行开发，然后使用 Android 的开发环境进行开发。

下面我们就开始搭建跨平台的开发环境。

2.1 Windows 下的开发环境配置

搭建 Windows 下的 Cocos2D-x 开发环境，首先需要安装 VS，VS 是微软公司推出的集成开发环境，是目前 Windows 平台上最流行的开发环境。它可以用来创建 Windows 平台下的 Windows 应用程序、网络应用以及网络服务等，支持的语言包括 C++、Basic、C# 等。Cocos2D-x 框架支持的 VS 版本包括 VS2008、VS2008 Express 以及 VS2010。本书选择的开发环境是 VS2008。

首先，从微软官方网站下载 VS（VS 是一款收费软件），然后双击安装文件，便可以开始安装，默认安装即可。如果是自定义安装，应选中 Visual C++ 组件。

VS 安装好了之后，Cocos2D-x 的 Windows 开发环境就算搭建完成了。如果需要提高开发效率，可以安装 Visual Assist X。这是一款非常不错的 VS 插件，具有强大的编辑特色，提高了生产效率，可以完全集成到你的 Microsoft 开发环境中，起到升级集成开发环境的效果，在不改变编程习惯的同时你可以感受到 Visual Assist X 带来的好处。

简而言之，Visual Assist X 可以在编程时给我们提示，比如某个类别都有什么方法，还可以进行语法检查等。注意，Visual Assist X 也是收费的，同样需要注册。

2.1.1 Cocos2D-x 软件安装与配置

在 Windows 的 VS 环境下安装 Cocos2D-x 框架，需要如下几个步骤：

12 ❖ 第一部分 准备篇

1) 从 Cocos2D-x 的官方网站 (<http://www.cocos2d-x.org/>) 下载最新版的 Cocos2D-x。首页上就有下载链接。

2) 将下载的压缩文件解压至某目录下。

3) 双击运行目录下的 install-templates-msvc.bat 文件进行安装。

注意 早期的 Cocos2D-x 版本没有 install-templates-msvc.bat 文件。可以在 \template\CCXAppWiz.vs 文件夹下找到 setup_vs2008.js 脚本并运行, 也可以安装该模板。不过早期版本只有 VS2008 版本的模板, 建议使用最新版的引擎。

4) 双击对应开发工具的项目组 SLN 文件。VS2008 对应的是 cocos2d-win32.vc2008.sln 文件, VS2010 对应的是 cocos2d-win32.vc2010.sln 文件。打开文件的同时也启动了相应的 VS 工具。

5) 编译并运行项目自带的 HelloWorld 项目来验证 Cocos2D-x 的安装完成正确。首先单击鼠标右键选中项目, 选择如图 2-1 所示的选项, 将项目设置为启动项。

6) 这时可以单击构建项目, 即 Build 按钮, 构建整个 Cocos2D-x 的项目, 也可以直接单击调试项目, 即 Debug 按钮, 运行 HelloWorld 项目, 效果如图 2-2 所示。

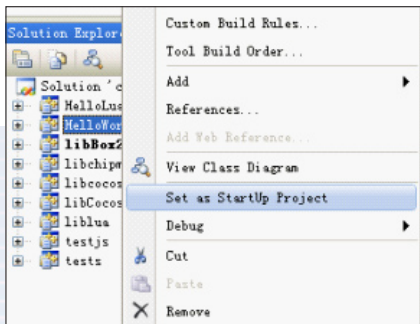


图 2-1 在 VS 中将项目设置为启动项



图 2-2 HelloWorld 运行效果

注意 早期的 Cocos2D-x 版本需要将 .\Debug.win32 目录下的所有 DLL 文件复制至 Windows\system32 目录下, 如果放置的位置有问题, 会报 “error PRJ0019: 某个工具从以下位置返

回了错误代码”。在 2.0 及更高版本中，由于支持的 OpenGL 版本的变化（从 2.01 版本的 Windows 版本支持 OpenGL 4.1 版本），所以如果新版本报错的话，请更新显卡驱动。

2.1.2 Cocos2D-x 工程目录介绍

Cocos2D-x 每一个版本的目录都会随着功能的增减有所变化。Cocos2D-x 的 1.0.1-x-0.11.0 版本和 2.0-rc2-x-2.0.1 版本的目录如图 2-3 和图 2-4 所示。

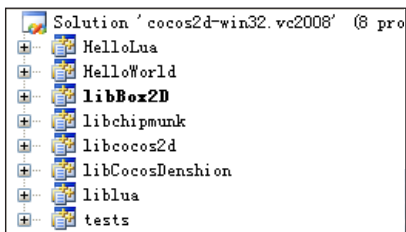


图 2-3 Cocos2D-x 的 1.0.1-x-0.11.0 版本的目录

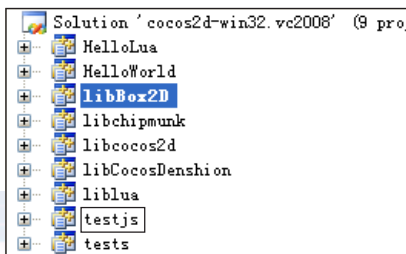


图 2-4 Cocos2D-x 的 2.0-rc2-x-2.0.1 版本的目录

可以看出在 Cocos2D-x 的 2.0 版本以后增加了 testjs 的 JavaScript 脚本的解析测试项目。各个项目的功能如下：

- ☐ HelloLua：Lua 脚本语言的测试项目。
- ☐ HelloWorld：开始的测试项目。
- ☐ libBox2D：物理引擎 Box2D 解决方案项目。
- ☐ libchipmunk：物理引擎 Chipmunk 解决方案项目。
- ☐ libCocos2d：引擎的核心项目。
- ☐ libCocosDenshion：声音解决方案项目。
- ☐ liblua：Lua 库解决项目。
- ☐ testjs：JavaScript 脚本的解析测试项目。
- ☐ tests：引擎 API 示例项目，建议初学者从这里看起。

在 VS 中一个 Cocos2D-x 项目应该包含哪些部分呢？Cocos2D-x 2.0 以前的项目目录结构如图 2-5 所示。

图 2-5 为 Cocos2D-x 的 1.0.1-x-0.11.0 版本的目录在 VS 中的目录结构，具体内容如下：

- ☐ Classes：Cocos2D-x 的 C++ 类文件，游戏的入口类为 AppDelegate.cpp，其余文件根据项目情况安排。
- ☐ Debug.win32：包括编译生成的 OBJ 文件、编译日志文件等。
- ☐ Resources：资源文件夹，放置声音、图片等文件。
- ☐ win32：Win32 平台的入口文件 main.cpp 等。

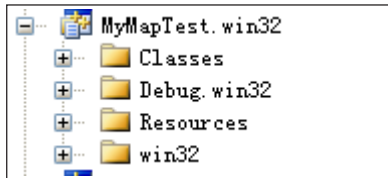


图 2-5 Cocos2D-x 的 1.0.1-x-0.11.0 版本的项目在 VS 中的目录结构

14 ◆ 第一部分 准备篇

注意 在 2.0 版本之前的 Cocos2D-x 中, 可以发现 tests 项目的资源文件放在了 Res 目录下而不是 Resources 目录下。这是为什么呢? 在 VS 中资源文件是可以设置的, 单击右键选择属性项目, 然后在 Debugging 项目中的 Working Directory 中设置项目的资源目录, 可以看见 tests 项目设置的是 Res 资源文件夹。

Cocos2D-x 的 2.0-rc2-x-2.0.1 版本的项目在 VS 中的目录结构如图 2-6 所示。

具体内容如下:

- ☐ include: 头文件 (文件扩展名为 h) 文件夹。
- ☐ resource: 资源文件夹, 放置声音、图片等文件。
- ☐ source: 源文件 (文件扩展名为 cpp) 文件夹。

至此, 我们已经了解了 Cocos2D-x 在 Windows 环境下的目录结构等。下一小节, 我们将开始新建一个 Cocos2D-x 的 Windows 项目。

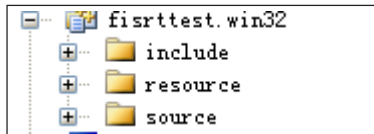


图 2-6 Cocos2D-x 的 2.0-rc2-x-2.0.1 版本项目在 VS 中的目录结构

2.1.3 新建项目及交叉编译

到目前, 我们已经介绍了 Cocos2D-x 项目在 Windows 环境下的搭建和配置, 并且介绍了 Cocos2D-x 的引擎目录和 Cocos2D-x 在 Windows 的 VS 环境下的目录。下面介绍如何在 Windows 环境中新建项目, 并且编译运行我们的项目。

1) 首先在解决方案的目录级别单击右键选择新建项目, 如图 2-7 所示。

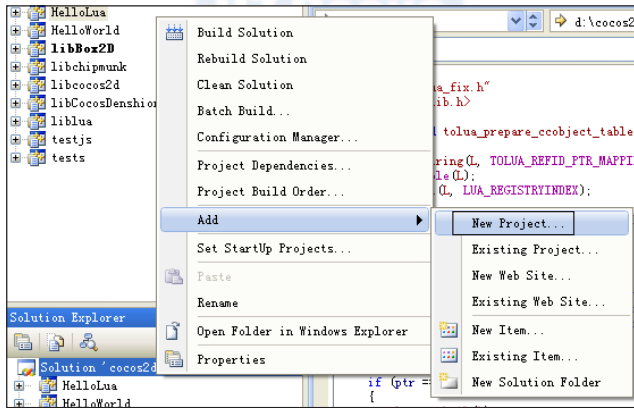


图 2-7 选择新建项目

2) 在图 2-8 所示的对话框中选择 Visual C++ 项目, 选择 Cocos2D-x 项目类型, 并且选择 Cocos2D-x Win32 项目 (在 Cocos2D-x 2.0 版本以后, 不提供 Cocos2D-x Wophone 项目), 并输入相应的项目名称, 这里输入 fisrttest 作为项目名称。

3) 对应的项目概况和功能选择对话框如图 2-9 所示, 可以选择使用的物理引擎、是否使用 Cocos Denshion 音乐模块、是否支持 Lua 脚本等, 这里只选择我们需要的项目, 单击

Finish 按钮，建立项目完毕。

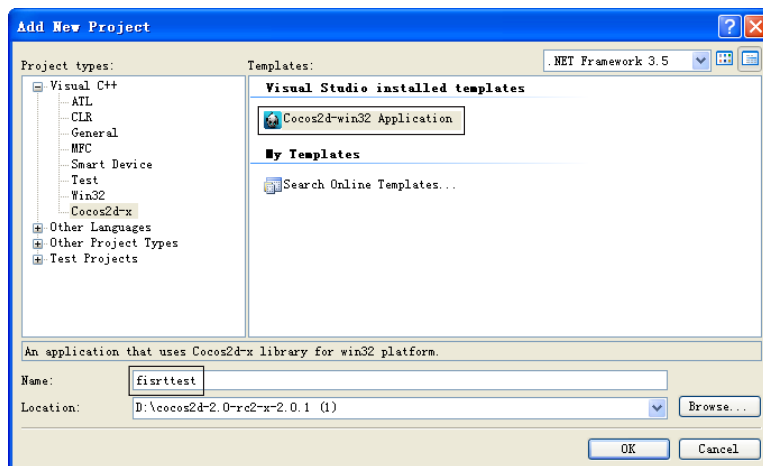


图 2-8 新建项目对话框

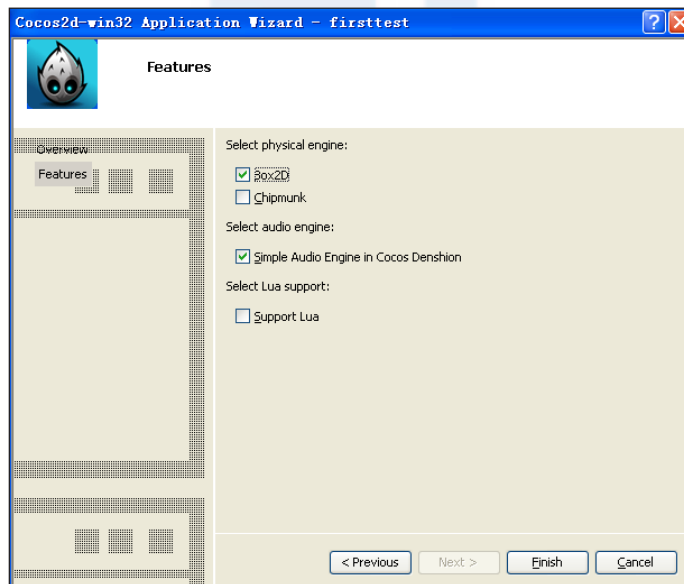


图 2-9 项目属性选择界面

4) 单击 Debug 按钮运行，效果如图 2-10 所示。

下面是在 Windows 环境下编译有可能遇到的问题及解决方法。

❑ 某库文件没找到。

解决方式：重新 build 整个项目。

❑ “0xC0000005: Access violation reading location 0x00000000”

解决方式：当前的 OpenGL 版本比较低，升级显卡驱动。

❑ “LNK1000: Internal error during IncrBuildImage”

解决方式：旧版本的 Cocos2D-x 会有这个错误，可以在项目名处单击右键，依次选择：属性→配置属性→链接器→常规→启用增量链接选择否，或者更新系统或引擎。

❑ “error PRJ0019: 某个工具从以下位置返回了错误代码”

解决方式：旧版本的 Cocos2D-x 会有这个问题，因为旧版本的引擎应该把 DLL 库文件放在 Windows\system32\ 目录下，如果放置错误，就会发生类似的问题。



图 2-10 新建项目运行效果

2.2 Android 下的开发环境配置

Android 本意是“机器人”，是 Google 公司推出的开源手机操作系统。Android 基于 Linux 操作系统，由操作系统、中间件、用户界面和应用软件组成，号称首个为移动终端打造的真正开放和完整的移动软件。

在 Android 最早发布时，Google 公司官方将 Java 语言作为第三方应用的开发语言，但是也没有完全拒绝 C 语言的开发人员使用自己的语言进行开发，因为在 Android 发布初期，Google 就表明其虚拟机支持 JNI（Java Native Interface，Java 本地调用），也就是第三方可以通过 JNI 调用自己的 C 动态库。但是最早 Google 并未为这种方式提供相应的工具支持这种开发方式。直到 2009 年 6 月，Google Android 方面发布了 NDK（Native Develop Kit，原生态本地开发包），支持开发者使用 C/C++ 语言开发 Android 程序。

Android NDK 作为 Android SDK 的一个附加组件提供给开发者，也就是说，开发者必须同时安装 SDK 和 NDK。NDK 只是作为 SDK 的一个补充，增加了代码的重用性和程序的运行效率，并且使 C/C++ 程序员也可以加入 Android 的开发中。本书成书之时 NDK 的最新版本是 android-ndk-r7 版本，本书也是使用这个版本进行开发的。

正因为 Android 支持了 C/C++ 的开发，Cocos2D-x 才选择了 C++ 作为主要的开发语言。因为 Java 语言是 iOS 开发不支持的，所以跨平台的 Cocos2D-x 选择了支持两个平台的 C++ 作为主要的开发语言。因此配置 Cocos2D-x 环境需要如下的开发工具。

❑ Eclipse：支持 Java 和 Android 开发的开源集成开发工具。

❑ JDK：JDK（Java Development Kit）是 Java 开发工具。

❑ ADT：ADT 是（Android Development Tools，Android 开发插件）Android 开发的 Eclipse 插件。

❑ CDT：CDT 是（C++ Development Tools，C++ 开发插件）C++ 开发的 Eclipse 插件。

❑ Android SDK：Android 开发工具，包括模拟器等。

❑ Android NDK：Android 原生态本地开发包，辅助 SDK 进行编译开发，支持 C/C++。

□ Cygwin: 在 Windows 平台上运行的 UNIX 模拟环境, 是 Cygnus Solutions 公司开发的自由软件。

使用 Cocos2D-x 进行 Android 开发需要上述工具。本节就介绍如何使用这些工具来搭建 Cocos2D-x 的 Android 开发环境。

2.2.1 Cocos2D-x 软件安装与配置

与 Cocos2D-x 的 Win 32 环境下的软件开发与配置相比, Android 的环境配置要相对麻烦, 包括 Android 开发环境的搭建、Android NDK 开发环境的搭建、Cygwin 的安装与配置及 Cocos2D-x 的编译运行等。

1) 下载并安装 JDK。

由于运行 Android 的模拟器和 Eclipse 等需要 Java 的 JDK 环境, JDK 是 Oracle 公司的 Java 开发工具, 下载地址为 <http://www.oracle.com/technetwork/java/javase/downloads/index.html>, 下载的文件直接运行安装即可。

2) 下载 Java 的开发工具 Eclipse。

Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台。就其本身而言, 它只是一个框架和一组服务, 用于通过插件组件构建开发环境。幸运的是, Eclipse 附带了一个标准的插件集, 包括 Java 开发工具等。Eclipse 的下载地址是 <http://www.eclipse.org/downloads/>。Eclipse 无须安装, 解压完成便可直接使用。

这里需要说明的是, 一般的 Java 开发和 Android 开发需要的只是 Eclipse 的 Java 版开发工具, 但是因为我们这里需要使用 C++ 进行开发, 所以建议下载 Eclipse 的 C++ 版本, 这样会更加方便些。本书成书之时最新的 Eclipse 的 C++ 版本的下载地址是 <http://www.eclipse.org/downloads/packages/eclipse-cc-developers/junior>。

3) 为 Eclipse 安装相应的插件, 首先是 ADT 插件。

ADT 是 Android 开发的 Eclipse 插件, 为我们提供了 Android 的 DDMS (Dalvik Debug Monitor Service, Android 虚拟机调试监视服务) 等 Android 开发的功能。一般使用 Eclipse 在线安装方式安装 ADT。

首先打开 Eclipse, 选择 Help → Install New Software (可能会由于 Eclipse 版本不同而有所不同), 弹出如图 2-11 所示的界面, 输入网址 <http://dl-ssl.google.com/android/eclipse>, 单击 Add 按钮便可出现如图 2-11 所示的结果。

选择安装全部, 同时安装开发工具和 NDK 插件。安装过程中会首先下载所需的软件, 所以时间可能会长些, 请耐心等待。

4) 安装 Eclipse 的 C++ 插件。

如果直接安装 Eclipse 的 C++ 版本, 可以跳过这一步。下载地址为 <http://wiki.eclipse.org/CDT/User/DownloadCDTLatest>。下载压缩包以后, 同样选择 Help-Install New Software, 弹出相应的界面后新增本地 CDT 的地址, 即可安装。第三步和第四步这两种方式同时适用于 CDT 和 ADT 的安装, 这里分别介绍两种安装方式, 大家可以根据自己的情况选择。

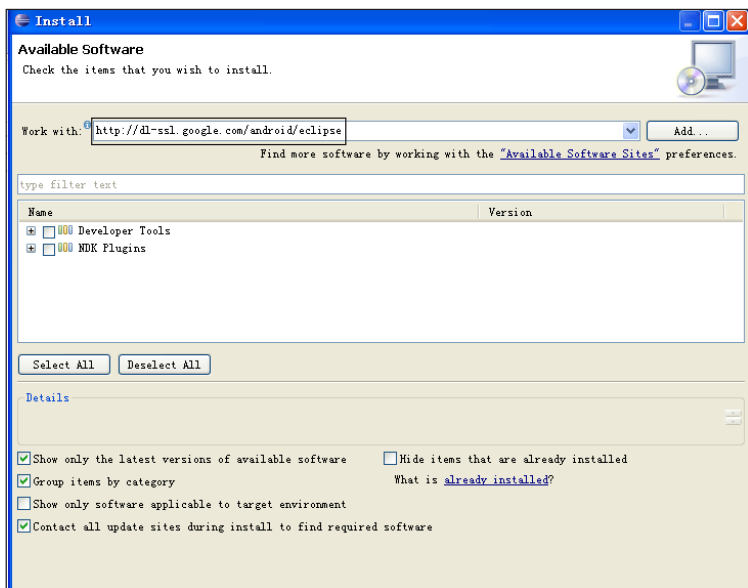


图 2-11 安装 ADT 的界面

5) 下载安装 Android SDK。

下载地址为: <http://developer.android.com/sdk/index.html>。下载完成后解压到某个目录下, 然后选择 Window → Preferences → Android, 弹出如图 2-12 所示的对话框, 在 SDK Location 中输入 Android SDK 的地址, 单击 Apply 按钮并确定, Android SDK 的配置就完成了。

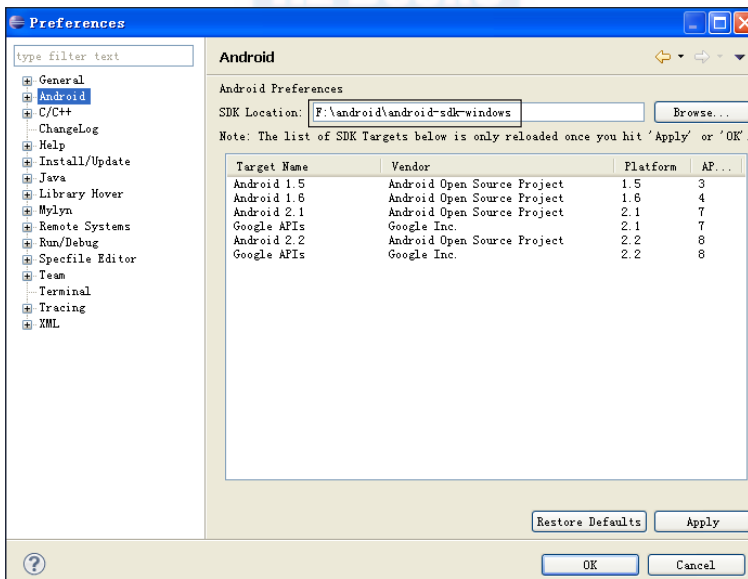


图 2-12 Android SDK 的 Eclipse 配置

选择 Window → Android SDK Manager, 弹出如图 2-13 所示的界面, 可以安装 Android 的其他版本和相应工具的补充安装。

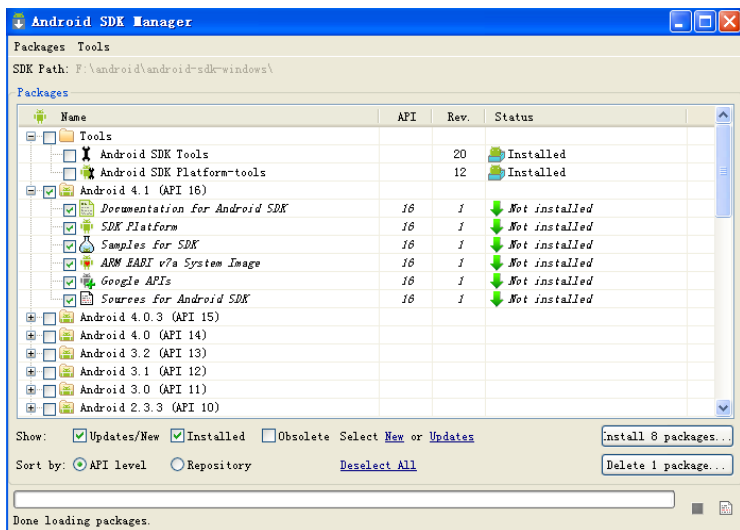


图 2-13 Android SDK 工具的安装界面

6) 安装和配置 Android 的 NDK 工具。

Android NDK 的最新版本为 r7, 可以在编译时省去 Cgywin, 提高效率, 下载地址为 <http://dl.google.com/android/ndk/android-ndk-r7-windows.zip>。将 NDK 解压到某一目录下, 然后选择 Window → Preferences → Android → NDK, 弹出如图 2-14 所示的对话框, 在 NDK Location 中输入 NDK 的地址, 单击应用按钮并确定, 完成 NDK 的配置。

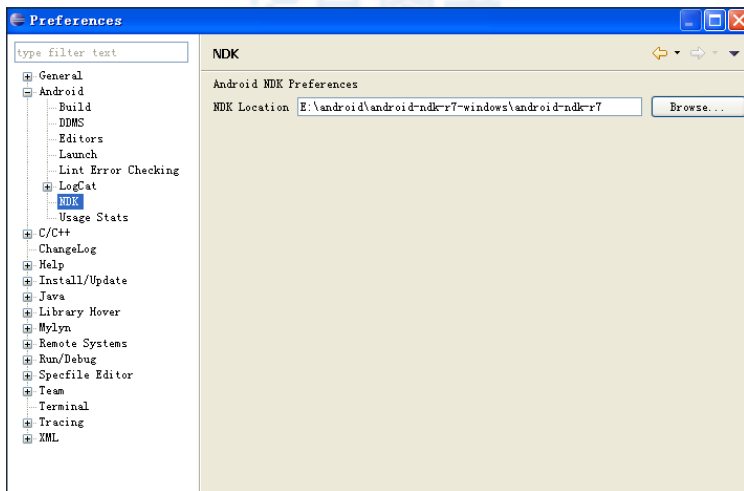


图 2-14 NDK 的配置界面

20 ❖ 第一部分 准 备 篇

7) 编译并运行 HelloWorld 项目。

选择 File-New-project, 在弹出的界面中选择 Android Project from Existing Code, 如图 2-15 所示。

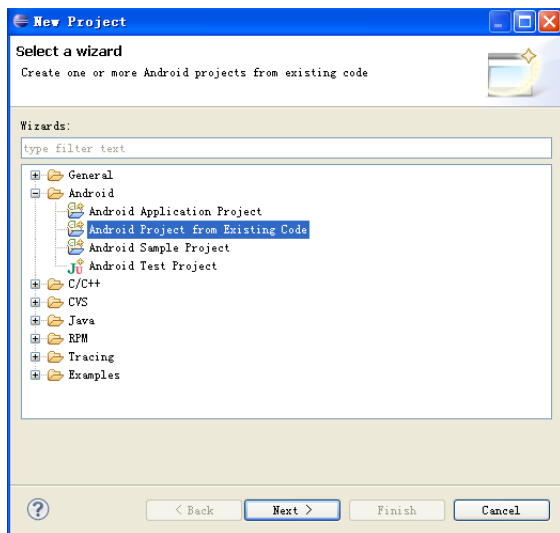


图 2-15 新建 Android 项目选择 Android Project from Existing Code

选择项目路径下的 proj.android 路径 (之前的版本是 android 文件夹), 选中并新建文件, 如图 2-16 所示。

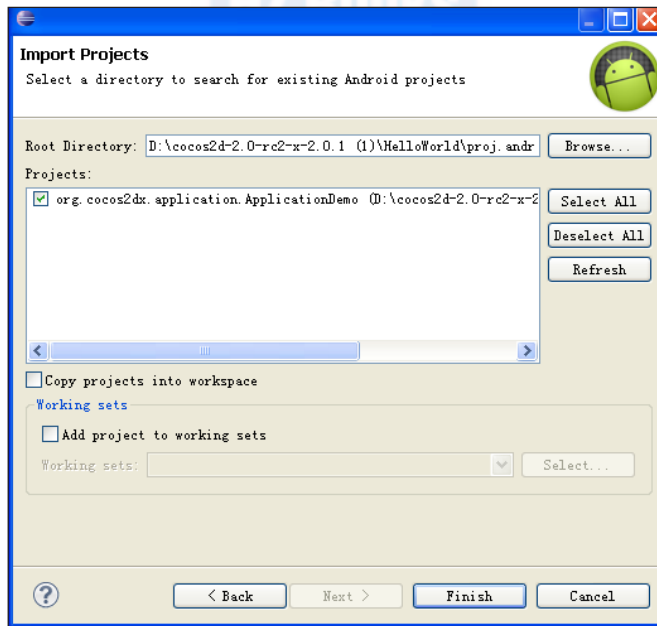


图 2-16 新建 Android 项目

8) 进行编译前的配置。

单击鼠标右键选中项目后, 选择 properties-Builder-New, 添加 NDK 路径, 如图 2-17 所示。

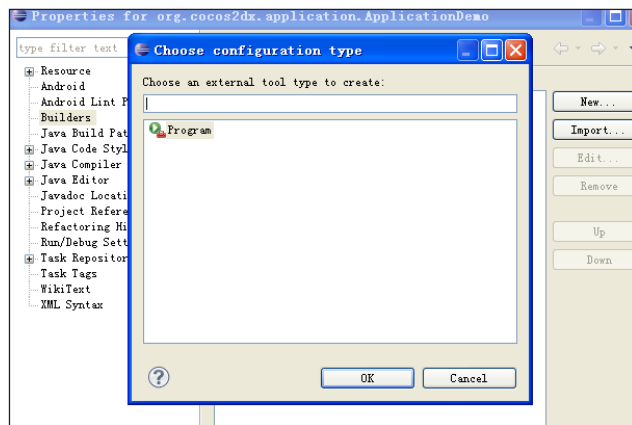


图 2-17 添加新编译路径

在如图 2-18 所示的界面中分别选择 Browse File System 和 Browse Workspace, 选择 ndk-build.cmd 和项目的路径。

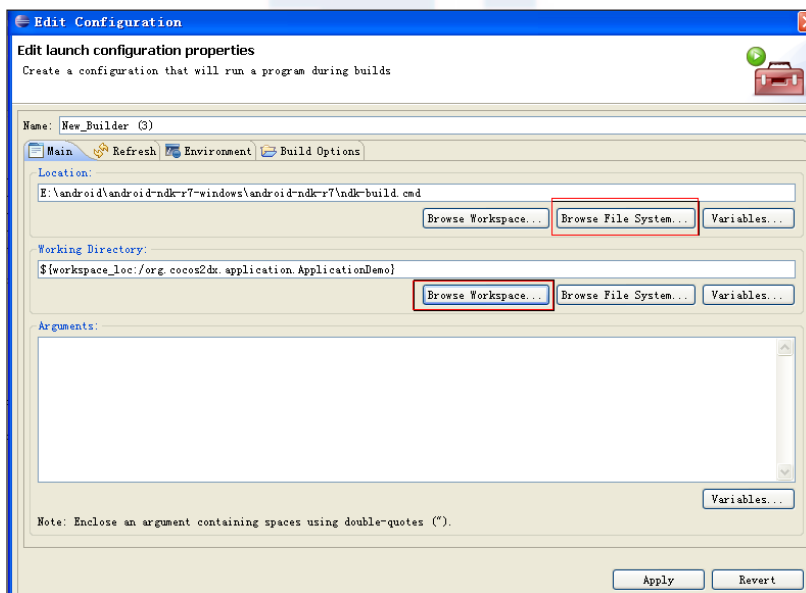


图 2-18 画框为相应的选项

Cocos2D-x 2.0 以上版本还要设置 NDK_MODULE_PATH 环境变量, 如图 2-19 所示, 填写 Cocos2D-x 的库的路径和第三方库的路径, 中间用分号相隔, 如例: \cocos2d-2.0-rc0a-x-2.0; \cocos2d-2.0-rc0a-x-2.0\cocos2dx\platform\third_party\android\prebuilt。

22 ❖ 第一部分 准 备 篇

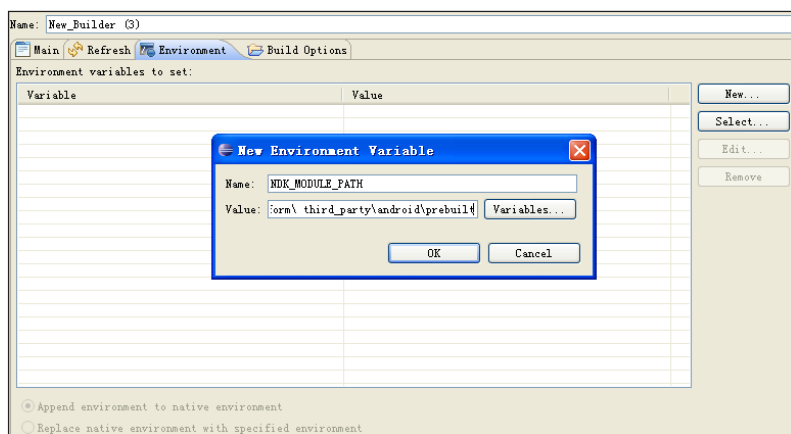


图 2-19 设置环境变量

9) 运行之前将资源文件复制到 assest 文件夹下。

将 NDK 目录下的 /sources/cxx-stl/gnu-libstdc++/libs/armeabi/ 目录下的 libgnustl_static.a 文件复制到项目文件夹下的 obj/local/armeabi/libgnustl_static.a 目录下, 之后右键项目选择 runas-Android Application 运行, 效果如图 2-20 所示。

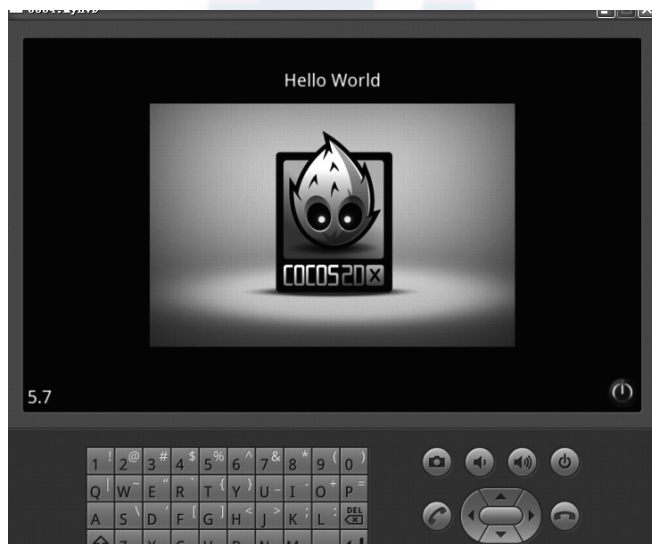


图 2-20 HelloWorld 项目在 Android 上的运行效果

注意 从 NDK 的 r7 版本开始, 编译部分不需要使用 Cygwin 的辅助了。如果使用之前的版本, 建议升级, 因为从 2.0 版本起 Cocos2D-x 只适用 NDK 的 r7 版本。否则需要打开 Cygwin, 输入工程路径下的 Android 文件夹的完整路径, 然后运行 build_native.sh 文件, 具体输入为 ./build_native.sh。使用命令行模式输入命令很麻烦, 最好将命令以记事本形式打开 cygwin 根目录下的 .\etc\bash.bashrc 修改启动的目录。

2.2.2 Cocos2D-x 工程目录介绍

Cocos2D-x 在 Android 平台上的工程项目与 Android 的 Application 一致，其目录变化比较小，如图 2-21 所示。

Cocos2D-x 的 Android 项目目录内容如下：

- src：Java 源文件的目录，由于 Android 项目的入口类都是 Activity，所以这里包括 Activity、事件捕捉在内的 Java 实现的程序。Java 文件目录包括入口 Activity 的 ApplicationDemo 和它的父类 Cocos2dxActivity，以及处理声音、加速度传感器的类，如图 2-22 所示。
- gen：自动生成的 R.java 文件、BuildConfig.java 文件等。
- assets：资源文件。
- bin：APK 包文件等配置文件。
- jni：包括编译时使用的 MK 文件等。
- libs：顾名思义，库文件，包括 SO 文件等。
- obj：包括部分 SO 文件和 O 文件等。
- res：XML（Extensible Markup Language，可扩展标记语言）等配置文件。

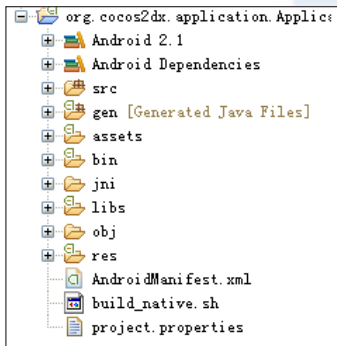


图 2-21 Cocos2D-x 的 Android 项目目录

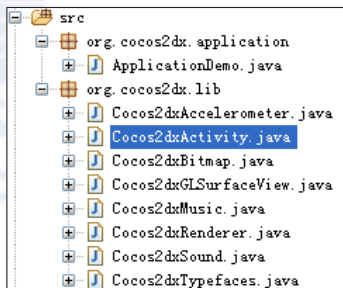


图 2-22 Java 文件目录

2.2.3 新建项目及交叉编译

Android 下的 Cocos2D-x 编译在 2.0 版本以后已经因为使用 NDK 的 r7 版本，而不需要使用 Cygwin 的辅助，但是新建项目还是需要 Cygwin 的。

1) 下载 Cygwin。

从官方地址 <http://cygwin.com/install.html> 下载 Cygwin 的在线安装文件，双击下载安装 Cygwin。

2) 使用任意文本编辑工具打开 Cocos2D-x 目录下的 create-android-project.bat 文件。

如图 2-23 所示，方框中的部分为我们需要修改 Cygwin、Android SDK 和 Android NDK 目录的位置。

24 ❖ 第一部分 准备篇

```
1 @echo off
2 :: This script is used to create an android project.
3 :: You should modify _ANDROIDTOOLS_CYGBIN_NDKROOT to work under your environment.
4 :: Don't change it until you know what you do.
5
6 setlocal
7
8 :: Check if it was run under cocos2d-x root
9 if not exist "%cd%\create-android-project.bat" echo Error!!! You should run it under cocos2d-x root & pause & exit 2
10
11 if not exist "%-dpm0.sh" echo Script "%-dpm0.sh" not found & pause & exit 3
12
13 :: modify it to work under your environment
14 set _CYGBIN=E:\android\bin
15 if not exist "%_CYGBIN%" echo Couldn't find Cygwin at "%_CYGBIN%" & pause & exit 4
16
17 :: modify it to work under your environment
18 set _ANDROIDTOOLS=F:\android\android-sdk-windows\tools
19 if not exist "%_ANDROIDTOOLS%" echo Couldn't find android sdk tools at "%_ANDROIDTOOLS%" & pause & exit 5
20
21 :: modify it to work under your environment
22 set _NDKROOT=E:\android\android-ndk-r7
23 if not exist "%_NDKROOT%" echo Couldn't find ndk at "%_NDKROOT%" & pause & exit 6
24
25 :: create android project
26 set /P _PACKAGEPATH=Please enter your package path. For example: org.cocos2dx.example:
27 set /P _PROJECTNAME=Please enter your project name:
28 if exist "%CD%\_%PROJECTNAME%" echo "%PROJECTNAME%" exists, please use another name & pause & exit 7
29 echo "Now cocos2d-x supports Android 2.1-updatel, 2.2, 2.3 & 3.0"
30 echo "Other versions have not tested."
31 call "%_ANDROIDTOOLS%\android.bat" list targets
32 set /P _TARGET=Please enter target id:
```

图 2-23 需要修改的 Cygwin、Android SDK 和 Android NDK 的目录

3) 双击 create-android-project.bat 文件, 在命令行模式下输入项目的包名项目名、SDK 版本等, 选择后完成新建, 如图 2-24 所示。

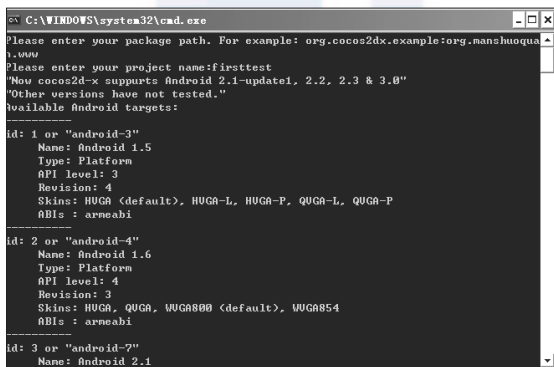


图 2-24 新建 Android 项目

4) 根据 2.2.1 节中的内容编译运行 firsttest 项目, 效果如图 2-25 所示。

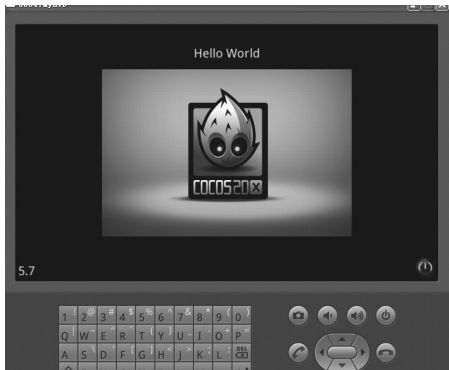


图 2-25 新建项目运行结果

5) 编辑 Android.mk 文件。

把游戏项目的 C++ 文件信息输入 Android.mk 文件中, 如图 2-26 所示。

```
LOCAL_PATH := $(call my-dir)
include $(CLEAR_VARS)
LOCAL_MODULE := game_logic

LOCAL_SRC_FILES := AppDelegate.cpp \
GameBullet.cpp \
GameEnemy.cpp \
GamePlayer.cpp \
GameSprite.cpp \
MapScene.cpp

LOCAL_C_INCLUDES := $(LOCAL_PATH)/../cocos2dx \
$(LOCAL_PATH)/../cocos2dx/platform \
$(LOCAL_PATH)/../cocos2dx/include \
$(LOCAL_PATH)/../CocosDenshion/include \
$(LOCAL_PATH)/../cocos2dx/luajava_support

LOCAL_LDLIBS := -L$(call host-path, $(LOCAL_PATH)/../android/libs/$(TARGET_ARCH_ABI)) \
-lcocos2d -lcocosdenshion \
-L$(call host-path, $(LOCAL_PATH)/../cocos2dx/platform/third_party)

include $(BUILD_SHARED_LIBRARY)
```

图 2-26 编辑 Android.mk 文件

注意 不同版本的 MK 文件的位置是不同的, Cocos2D-x 的 1.0.1-x-0.11.0 版本的是在 Classes 目录下, Cocos2D-x 的 2.0-rc2-x-2.0.1 版本的在 jni 目录下。修改位置不正确, 是起不到相应的效果的。

在 Win32 程序转换成 Android 项目的过程中, 由于二者在规则上的不同, 为了程序的可移植性更好, 需要注意以下内容:

- ❑ Win32 的程序对于资源的名称是大小写不敏感的, 但是在 Android 中, 却恰恰是大小写敏感的。比如图片的名称是 Example.png, 在 Win32 程序中如果写 example.png, 是可以找到相应的图片的, 但是在 Android 中却不可以。所以为了提高程序的可读性, 请保持大小写的敏感。
- ❑ 需要改变 Android 项目的横纵屏幕时, 在图 2-27 所示的位置上修改, 分别选择 landscape (横向) 和 portrait (纵向) 即可。

```
android:versionName="1.0">

<uses-sdk android:minSdkVersion="7" android:targetSdkVersion="$"/>

<application android:label="@string/app_name"
    android:debuggable="true"
    android:icon="@drawable/icon">

    <activity android:name=".ApplicationDemo"
        android:label="@string/app_name"
        android:screenOrientation="landscape"
        android:theme="@android:style/Theme.NoTitleBar.Fullscreen"
        android:configChanges="orientation">

        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>
<supports-screens android:largeScreens="true"
```

图 2-27 配置文件中配置屏幕方向

26 ❖ 第一部分 准备篇

□ VS 下的编码格式与 Android 下的编码格式不同,所以需要统一编码格式,或者使用转换工具转换。

如果需要添加新的库,这里以 Box2D 库为例,在 2.0 之前的版本需要经过如下步骤:

1) 修改工程目录下的 Classes 目录下的 Android.mk 文件,需要修改的位置如图 2-28 所示。

```
1 LOCAL_C_INCLUDES := $(LOCAL_PATH)/../cocos2dx \
2     $(LOCAL_PATH)/../cocos2dx/include \
3     $(LOCAL_PATH)/../cocos2dx/platform \
4     $(LOCAL_PATH)/../cocos2dx/platform/third_party/android/ \
5     $(LOCAL_PATH)/../chipmunk/include/chipmunk \
6     $(LOCAL_PATH)/../CocosDenshion/include \
7     $(LOCAL_PATH)/..
8
9
10 LOCAL_LDLIBS := -L$(call host-path, $(LOCAL_PATH)/test.android/libs/$(TARGET_ARCH_A
11     -lGLESv1_CM \
12     -lcocos2d -llog -lcocosdenshion \
13     -lbox2d -lchipmunk \
14     -L$(call host-path, $(LOCAL_PATH)/../cocos2dx/platform/third_party/
15
16 include $(BUILD_SHARED_LIBRARY)
```

图 2-28 修改 Classes 目录下的 Android.mk 文件

2) 修改 Android 工程目录下的 jni 目录下的 Android.mk 文件,需要修改的位置如图 2-29 所示。

```
subdirs := $(addprefix $(LOCAL_PATH)/../../.., $(addsuffix /Android.mk, \
Box2D \
chipmunk \
cocos2dx \
CocosDenshion/android \
))
```

图 2-29 修改 jni 目录下的 Android.mk 文件

3) 修改 jni 目录下的 Application.mk 文件,需要修改的位置如图 2-30 所示。

```
APP_STL := gnustdl_static
APP_CPPFLAGS += -frtti
APP_MODULES := cocos2d cocosdenshion chipmunk box2d game_logic tests
```

图 2-30 修改 jni 目录下的 Application.mk 文件

4) 修改入口 Activity 类,需要修改的位置如图 2-31 所示。

在 2.0 之后的版本中,由于设置了环境变量 NDK_MODULE_PATH,只需要修改 Android 工程目录下的 jni 目录下的 Android.mk 文件即可,如图 2-32 所示。

```
static {
    System.loadLibrary("cocosdenshion");
    System.loadLibrary("chipmunk");
    System.loadLibrary("box2d");
    System.loadLibrary("cocos2d");
    System.loadLibrary("game_logic");
    System.loadLibrary("tests");
}
```

图 2-31 修改入口 Activity 类

```
1 LOCAL_STATIC_LIBRARIES := curl_static_prebuilt
2
3 LOCAL_WHOLE_STATIC_LIBRARIES := cocos_tests_common
4 LOCAL_WHOLE_STATIC_LIBRARIES += cocos2dx_static
5 LOCAL_WHOLE_STATIC_LIBRARIES += cocosdeshion_static
6 LOCAL_WHOLE_STATIC_LIBRARIES += box2d_static
7 LOCAL_WHOLE_STATIC_LIBRARIES += chipmunk_static
8
9 include $(BUILD_SHARED_LIBRARY)
10
11 $(call import-module,tests)
12 $(call import-module,cocos2dx)
13 $(call import-module,cocos2dx/platform/third_party/android/prebuilt/libcurl)
14 $(call import-module,CocosDenshion/android)
15 $(call import-module,Box2D)
16 $(call import-module,chipmunk)
```

图 2-32 修改 Android 工程目录下的 jni 目录下的 Android.mk 文件

2.3 iOS 下的开发环境配置

iOS 操作系统是由苹果公司开发的手持设备操作系统，最早发布于 2007 年 1 月 9 日的 Macworld 大会上，最初是为 iPhone 设计的，后来陆续套用到 iPod Touch、iPad 以及 Apple TV 等苹果移动操作设备上。

iOS 与苹果以 Darwin 为基础，因此同样属于类 UNIX 的商业操作系统。原本这个系统名为 iPhone OS，直到 2010 年 6 月 7 日 WWDC 大会上被改名为 iOS。截至 2011 年 11 月，iOS 已经占据了全球智能手机系统市场份额的 30%。

Xcode 是苹果公司向开发人员提供的集成开发环境，用于开发 Mac OS 的应用程序。iOS SDK 是 iOS 系统的开发工具。Xcode 允许你开发基于 iOS 的 iPad、iPhone、iPod Touch 设备应用程序。只要有 Mac OS X Snow Leopard 10.6.2 以上版本 Mac OS 操作系统，便可安装 iOS SDK，可以使用 iPhone 模拟器进行调试或者使用真机进行调试。

在 iOS 上开发 Cocos2D-x 的应用也要使用 Xcode。Xcode 的安装文件下载地址为 <https://developer.apple.com/technologies/tools/>。

注意 下载之前需要注册苹果开发者，下载后双击 DMG 文件进行安装即可。

2.3.1 Cocos2D-x 软件安装与配置

首先将 Cocos2D-x 的压缩包解压到某一个目录下，这时便可运行 HelloWorld 项目了。

双击 proj.ios 目录下的 iOS 工程文件，用 Xcode 打开 HelloWorld 项目，然后单击运行，Cocos2D-x 的 HelloWorld 项目便可在 iPhone 模拟器和 iPad 模拟器上运行，效果如图 2-33 和图 2-34 所示。

下面在 Mac 系统上安装 Cocos2D-x 模板。首先启动终端（单击 Finder→前往→实用工具→终端），然后将目录切换到 Cocos2D-x 的解压目录，输入如下命令安装 Cocos2D-x 模板：

28 第一部分 准备篇



图 2-33 HelloWorld 在 iPhone 模拟器上的运行效果



图 2-34 HelloWorld 在 iPad 模拟器上的运行效果

```
sudo ./install-templates-xcode.sh
```

图 2-35 所示为终端下命令安装 Cocos2D-x 模板的过程。

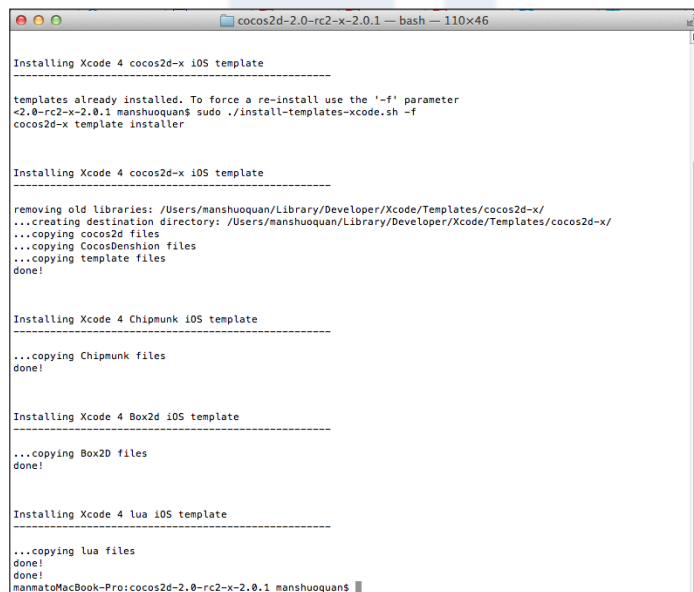


图 2-35 终端下命令安装 Cocos2D-x 模板的过程

注意 运行这段命令的过程中如果出现问题，请检查你的命令是否输入正确。

2.3.2 Cocos2D-x 工程目录介绍

Cocos2D-x 的 iOS 版本工程目录和 iOS 普通的工程目录类似，如图 2-36 所示。

各子目录说明如下：

- Classes: Cocos2D-x 的游戏逻辑类，也是我们要编写的类文件。
- cocos2dx: 引擎代码类。
- Frameworks: 引擎所使用的框架部分。
- ios: iOS 平台相关专用类。
- Products: 编译生成的包文件。
- Resources: 资源文件夹。

在 Xcode 4 环境下新建项目的目录略有不同，如图 2-37 所示。

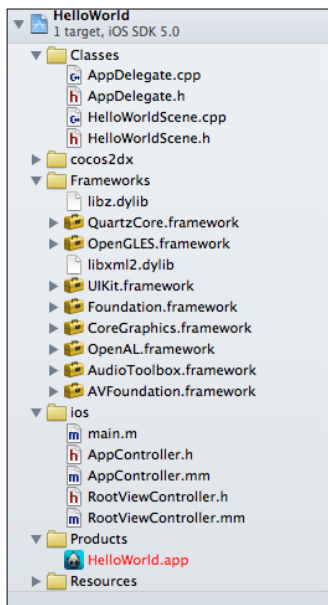


图 2-36 Cocos2D-x 的 HelloWorld 目录

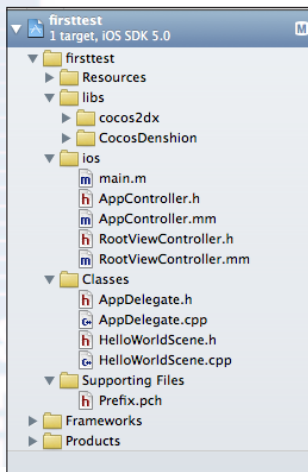


图 2-37 Cocos2D-x 新建工程的目录

各子目录说明如下：

- firsttest: 项目相关文件夹，其中包括资源文件夹 Resources、库文件夹 libs、iOS 平台相关代码文件夹、游戏逻辑类文件夹 Classes、支持的预编译头文件夹 Supporting Files。
- Frameworks: 引擎所使用的框架部分。
- Products: 编译生成的包文件。

2.3.3 新建项目及交叉编译

在 Xcode 下选择新建项目，在如图 2-38 所示的对话框中选择 cocos2d-x，然后选择需要的框架类型。

在如图 2-39 所示的对话框中输入项目名称等信息。

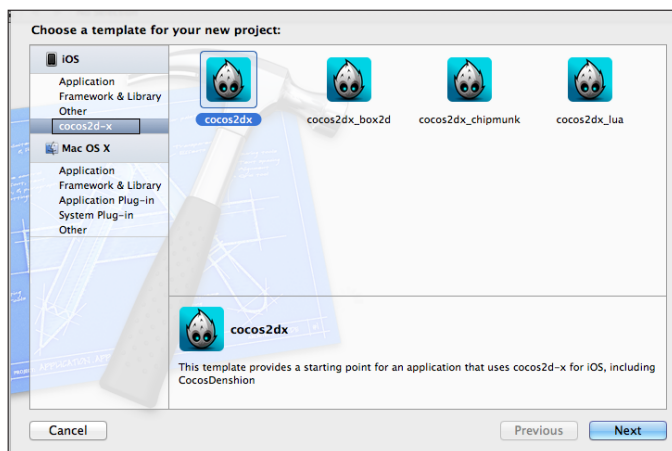


图 2-38 在 Mac 系统上新建 Cocos2D-x 项目

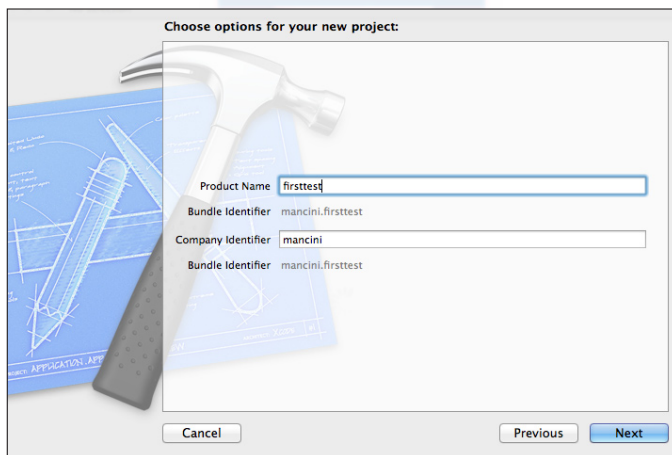


图 2-39 输入项目名称等信息

选择相应的模拟器，单击运行项目。图 2-40 和图 2-41 所示分别为新建项目在 iPhone 模拟器和 iPad 模拟器上的运行效果。



图 2-40 新建项目在 iPhone 模拟器上的运行效果



图 2-41 新建项目在 iPad 模拟器上的运行效果

2.4 使用 Doxygen 工具生成 Cocos2D-x 文档

Doxygen 是一种开源跨平台的工具，其功能是从程序源代码中抽取类、方法、成员的注释，形成一个和源代码配套的 API（Application Programming Interface，应用程序编程接口）帮助文档。Doxygen 工具完全支持 C、C++、Java、Objective-C 等语言，部分支持 PHP、C#。

Doxygen 可以根据代码中的注释，按照规则生成相应的文档。Cocos2D-x 的代码就依照了它的规则，并且提供了 doxygen.cocos2d-x 文件。这是 Cocos2D-x 代码的 Doxygen 配置文件，可以通过这个文件来生成 Cocos2D-x 文档。

首先下载 Doxygen，下载地址为 <http://www.doxygen.nl/download.html#latestsrc>。根据系统选择要下载的版本，建议下载安装版。下载完成后运行程序，出现如图 2-42 所示的界面。

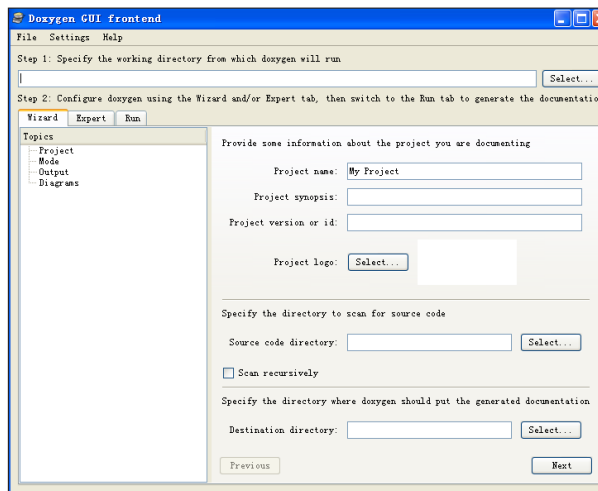


图 2-42 Doxygen 运行界面

32 ❖ 第一部分 准备篇

使用 File → Open 打开 Cocos2D-x 目录下的 doxygen 目录下的 doxygen.cocos2d-x 文件，界面如图 2-43 所示。

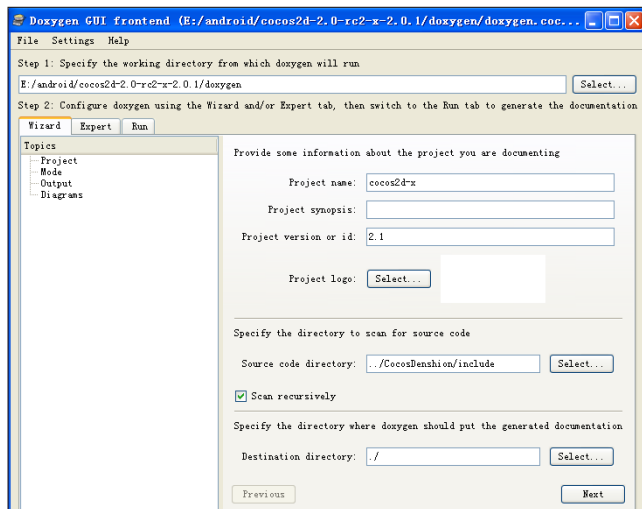


图 2-43 打开 doxygen.cocos2d-x 文件后的界面

打开配置文件后出现了 Cocos2D-x 版本配置信息，单击“Run”选项卡，单击“Run doxygen”按钮，运行生成文档，完成后如图 2-44 所示。

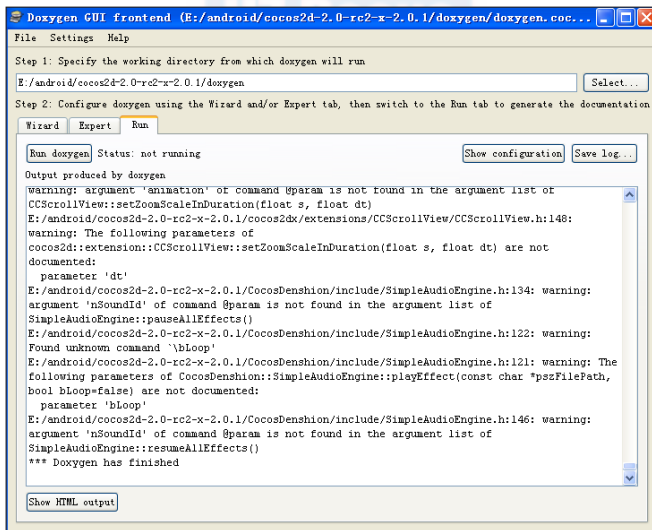


图 2-44 生成文档

打开 Cocos2D-x 目录下的 doxygen/html 目录下的 index.html 文件，如图 2-45 所示，可以使用 Cocos2D-x 文档了。



图 2-45 Cocos2D-x 文档

2.5 本章小结

本章介绍 Cocos2D-x 的项目在 Win32、Android、iOS 开发环境中的安装与配置，详细介绍了项目结构以及新建项目和交叉编译。其中，在 Win32 环境下，需要了解 Visual Studio 的使用以及模板的安装；在 Android 环境下，学习了 Android 环境的搭建，包括 Eclipse、Android SDK、ADT 的基本配置，以及 Eclipse C++ 环境的插件 CDT 和 Android 的 NDK 工具的配合使用，另外，在新建项目时还需要 Cygwin 的辅助；iOS 的开发在 Mac 上进行，需要安装 Xcode 工具。最后介绍了文档生成工具 Doxygen，使用该工具可以根据文档更好地学习和使用 Cocos2D-x。

学习本章内容，应该根据提示一步步完成跨平台系统的搭建，为今后的学习做准备。从第 3 章开始是本书的第二部分，将正式介绍 Cocos2D-x 的基础知识，首先介绍 Cocos2D-x 的核心类。