# **cocos2d-x 3.0rc开发指南：Windows下Android环境搭建**

# 安装工具

### 1. 配置JDK

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

本人的系统是Win7 64位版，但安装的是JDK7，Windows X86版。

假设安装目录在：*C:\Program Files (x86)\Java\jdk1.7.0\_21*；当然也可以是其他地方

* 设置环境变量：

*JAVA\_HOME=C:\Program Files (x86)\Java\jdk1.7.0\_21*

*CLASSPATH=.;%JAVA\_HOME%\lib;*

Path增加*%JAVA\_HOME%\bin;*

本文假设你知道怎么设置环境变量，如果不知道，请使用搜索引擎查找相关资料，设置完后打开cmd，输入java -version

如果出现下面提示，表明环境变量设置成功：

*C:\Users\arlin>java -version*

*java version "1.7.0\_21"*

*Java(TM) SE Runtime Environment (build 1.7.0\_21-b11)*

*Java HotSpot(TM) Client VM (build 23.21-b01, mixed mode, sharing)*

### 2. 下载Android SDK

Google为Android开发者提供了ADT(Android Developer Tools)，里面包含了SDK和Eclipse。

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

可以选择32位或64位版本，我选择的是32位版的，把它解压到一个目录，我这里是：*d:\adt-bundle-windows-x86\*，目录结构如下：

*--<adt-bundle-windows-x86>*

*--<eclipse>*

*--<sdk>*

*--SDK Manager.exe*

### 3. 下载Android NDK

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

同样有32位和64位版可选，我选择的是32位版的，把它解压到adt同目录下，现在adt的目录结构如下：

*--<adt-bundle-windows-x86>*

*--<eclipse>*

*--<sdk>*

*--<android-ndk-r8e>*

*--SDK Manager.exe*

### 4. 下载ANT

ANT是为了自动构建Android程序用。

下载地址是：*<http://ant.apache.org/bindownload.cgi>*

我选择的是：*apache-ant-1.9.3-bin.zip*，同样解压到adt目录下，现在adt的目录结构如下：

*--<adt-bundle-windows-x86>*

*--<eclipse>*

*--<sdk>*

*--<android-ndk-r8e>*

*--<apache-ant-1.9.3>*

*--SDK Manager.exe*

### 5. 下载Python

* 原来cocos2dx在不同平台下必须用不同的脚本来创建和构建工程，3.0以后都统一用python了，所以Windows下需要安装python环境：

下载地址：<https://www.python.org/ftp/python/2.7.3/python-2.7.3.msi>

原来下载了最新的版本，使用setup.py的时候发现有语法错误，所以最好使用2.7.3版本，亲测过没有问题

* 配置环境变量：最新版本可以自动加进path环境变量，但2.7.3不行，所以我们只能自己设置一下，假设我的Python安装在：d:\Python27\

把这个路径加到path环境变量中，当我们在cmd下输入python，出现下面字样，就说明环境变量设置正确了：

*C:\Users\arlin>python*

*Python 2.7.3 (default, Apr 10 2012, 23:31:26) [MSC v.1500 32 bit (Intel)] on win32*

*Type "help", "copyright", "credits" or "license" for more information.*

*>>>*

# cocos2d-x下载与配置

### 1. 下载cocos2d-x

目前最新版本是3.0RC，下载地址是：*<http://cocostudio.download.appget.cn/Cocos2D-X/3.0RC0/cocos2d-x-3.0rc0.zip>*

下载完解压到一个地方，我这里：*f:\cocos2dx\cocos2d-x-3.0rc0\*

### 2. 设置环境变量

Cocos2dx需要下面几个环境变量，用于后面的新建工程，构建工程等：

* COCOS\_CONSOLE\_ROOT: cocos控制台路径，控制台用于新建，构建，发行工程。
* NDK\_ROOT: NDK根目录
* ANDROID\_SDK\_ROOT: SDK根目录
* ANT\_ROOT: ANT根目录

这几个环境变量是通过cocos2d-x根目录下的setup.py来配置的，我们启动cmd，运行setup.py：

*F:\cocos2dx\cocos2d-x-3.0rc0>setup.py*

此时脚本需要我们提供NDK，SDK，ANT的路径，我们根据之前放的路径输入就是了，整个命令过程如下：

*Setting up cocos2d-x...*

*-> Adding COCOS2D\_CONSOLE\_ROOT environment variable... OK*

*-> Added: COCOS\_CONSOLE\_ROOT = F:\cocos2dx\cocos2d-x-3.0rc0\tools/cocos2d-cons*

*ole/bin*

*-> Looking for NDK\_ROOT envrironment variable... NOT FOUND*

*Please enter its path (or press Enter to skip): d:\adt-bundle-windows-x8*

*6\android-ndk-r8e*

*ADDED*

*-- Added: NDK\_ROOT = d:\adt-bundle-windows-x86\android-ndk-r8e*

*-> Looking for ANDROID\_SDK\_ROOT envrironment variable... NOT FOUND*

*Please enter its path (or press Enter to skip): D:\adt-bundle-windows-x8*

*6\sdk*

*ADDED*

*-> Added: ANDROID\_SDK\_ROOT = D:\adt-bundle-windows-x86\sdk*

*-> Looking for ANT\_ROOT envrironment variable... NOT FOUND*

*Please enter its path (or press Enter to skip): d:\adt-bundle-windows-x8*

*6\apache-ant-1.9.3\bin*

*ADDED*

*-> Added: ANT\_ROOT = d:\adt-bundle-windows-x86\apache-ant-1.9.3\bin*

*Set up successfull:*

*COCOS\_CONSOLE\_ROOT was added into registry*

*NDK\_ROOT was added into registry*

*ANDROID\_SDK\_ROOT was added into registry*

*ANT\_ROOT was added into registry*

*Please restart the terminal or restart computer to make added system variables t*

*ake effect*

命令最后提示我们重启终端或者重启系统，以让这些环境变量生效，一般我们关闭cmd和资源浏览器就行了，如果后面遇到环境变量找不到的错误，直接重启系统试试吧。

要测试环境变量是否生效，重新打开cmd，输入下面命令查看，正常应该是这样：

*F:\cocos2dx\cocos2d-x-3.0rc0>set NDK\_ROOT*

*NDK\_ROOT=d:\adt-bundle-windows-x86\android-ndk-r8e*

*F:\cocos2dx\cocos2d-x-3.0rc0>set ANT\_ROOT*

*ANT\_ROOT=d:\adt-bundle-windows-x86\apache-ant-1.9.3\bin*

*F:\cocos2dx\cocos2d-x-3.0rc0>set ANDROID\_SDK\_ROOT*

*ANDROID\_SDK\_ROOT=D:\adt-bundle-windows-x86\sdk*

*F:\cocos2dx\cocos2d-x-3.0rc0>set COCOS\_CONSOLE\_ROOT*

*COCOS\_CONSOLE\_ROOT=F:\cocos2dx\cocos2d-x-3.0rc0\tools/cocos2d-console/bin*

# 新建和构建工程

### 1. 新建工程

我们要用tools\cocos2d-console这个工具来新建工程，这个工具提供了下面几个功能：

* new        创建一个新的工程
* compile    编译当前工程，生成二进制文件，个人觉得写为build更贴切一些。
* deploy     发布程序到一个平台
* run        编译和发布，和运行程序

我们现在只用到new，让我们看看new提供了哪些参数，打到cmd，输入：

*cocos new --help*

得到下面帮助：

*usage: cocos new [-h] [-p PACKAGE\_NAME] -l {cpp,lua,js} [-d DIRECTORY]*

*[-t TEMPLATE\_NAME] [--no-native]*

*[PROJECT\_NAME]*

*创建一个新工程*

*必要参数：*

*PROJECT\_NAME          工程名*

*可选参数:*

*-h, --help            显示帮助信息*

*-p PACKAGE\_NAME, --package PACKAGE\_NAME*

*设置包名，如com.colin.mbgame*

*-l {cpp,lua,js}, --language {cpp,lua,js}*

*使用的主要语言，可以是：[cpp | lua | js]*

*-d DIRECTORY, --directory DIRECTORY*

*工程所在的目录*

*-t TEMPLATE\_NAME, --template TEMPLATE\_NAME*

*要创建的工程模板名*

*lua/js 工程参数:*

*--no-native           No native support.*

现在我们在cmd中输入：

*F:\cocos2dx>cocos new mygame1 -p com.colin.mbgame -l cpp -d f:\cocos2dx\mbgame*

得到下面输出：

F:\cocos2dx>python F:\cocos2dx\cocos2d-x-3.0rc0\tools\cocos2d-console\bin\/cocos.py

new mygame1 -p com.colin.mbgame -l cpp -d f:\cocos2dx\mbgame

Runing command: new

> Copy template into f:\cocos2dx\mbgame\mygame1

> Copying cocos2d-x files...

> Rename project name from 'HelloCpp' to 'mygame1'

> Replace the project name from 'HelloCpp' to 'mygame1'

> Replace the project package name from 'org.cocos2dx.hellocpp' to 'com.colin.mbgame'

这表明我们已经成功创建一个工程，到*f:\cocos2dx\mbgame\*看看，工程目录结构如下：

*--<mygame1>*

*--<Classes>*

*--<cocos2d>*

*--<proj.android>*

*--<proj.ios\_mac>*

*--<proj.linux>*

*--<proj.win32>*

*--<Resources>*

*--.cocos-project.json*

*--CMakeLists.txt*

可以看到各种平台下的工程都创建出来了，同时它把cocos2d整个框架都拷贝过来了，这种好坏就见仁见智了，

### 2. 构建Android工程

同时是用到cocos2d-console，这里要用的是compile这个命令，在cmd输入下面查看帮助：

*f:\cocos2dx\mbgame\mygame1>*cocos compile --help

*usage: cocos compile [-h] [-s SRC\_DIR] [-q] [-p PLATFORM] [-m MODE] [-j JOBS]*

*[--ap ANDROID\_PLATFORM] [--source-map]*

*把当前工程编译为二进制*

*可选参数:*

*-h, --help            显示帮助信息*

*-s SRC\_DIR, --src SRC\_DIR*

*工程根目录，比如上面应该是f:\cocos2dx\mbgame\mygame1*

*如果不设，就为当前目录（cmd定位到的那个目录）*

*-q, --quiet           less output*

*-p PLATFORM, --platform PLATFORM*

*选择一个平台 android|ios|mac|web|win32|linux*

*-m MODE, --mode MODE  编译模式 debug|release, 默认为debug.*

*-j JOBS, --jobs JOBS  半行编译，如果你有4核，可以设为4。*

*Android Options:*

*--ap ANDROID\_PLATFORM*

*指定SDK版本？没试过，不敢枉下定论，保留英文：*

*parameter for android-update.Without the parameter,the*

*script just build dynamic library for project. Valid*

*android-platform are:[10|11|12|13|14|15|16|17|18|19]*

*Web Options:*

*--source-map          Enable source-map*

现在我们在cmd输入：

*F:\cocos2dx\mbgame\mygame1>cocos compile -p android -j 4*

意思就是说我们要编译当前目录下的Android工程，同时可以有4个编译任务，所以我的4核机器就满负的跑了，如无意外，应该可以看到编译开始了，最后如果看到下面这几句，说明编译成功：

*BUILD SUCCESSFUL  
Total time: 7 seconds  
Move apk to F:\cocos2dx\mbgame\mygame1\bin\debug\android  
build succeeded.*

我们到*f:\cocos2dx\mbgame\mygame1\proj.android\bin\*，可以看到mygame1-debug.apk，说明构建完成。

我把它安装到手机上试跑，发现好像启动速度是快了很多，不知是不是心理作用。

# 在Eclipse下运行程序

还记得我们下的adt里面有eclipse吗，现在我们就来把工程导入到eclipse跑跑看

* 到*d:\adt-bundle-windows-x86\eclipse*打到*eclipse.exe*

第一次会提示设置工作区，你可以设置工作区放在哪里，我这里为了演示就默认了。

* 接下来我们要看看SDK和NDK的路径是否正确：
  + 选择*Windwos/Preferences*菜单项，打开选项对话框。
  + 选择*Android*结点，看看*SDK Location*是否正确，我这里为*F:\software\adt-bundle-windows-x86-20131030\sdk*，说明是正确的，如果没有值，就手动把SDK的根目录设进去。
  + 接着选择*Android/NDK*结点，在NDK Location输入*d:\adt-bundle-windows-x86\android-ndk-r8e*
* 接下来我们来导入工程，选择file/import，打开导入对话框：
  + 选择*Android/Existing Android Code Into Workspace*，然后*Next*。
  + 第一步要先把*libcocos2dx*导进来：
    - 在*Root Directory*输入*F:\cocos2dx\mbgame\mygame1\cocos2d\cocos\2d\platform\android\java*
    - 此时列出*libcocos2dx*工程，点击*finish*把工程导进来。
  + 第二步导入我们的工程：
    - 在*Root Directory*输入*F:\cocos2dx\mbgame\mygame1\proj.android*。
    - 此时会列出*mygame1*工程，点击*finish*把工程导进来。
  + 最后连接手机，选择运行或调试就可以跑程序了。
* 在导入工程到Eclipse时遇到几个坑，在这里记下来，免得后面的人继续踩这些坑：
  + 没有导*libcocos2dx*，只导*mygame1*时不会有编译错误，但跑起来后会*ClassNotFound*异常，说找不到*org/cocos2dx/lib/Cocos2dxHelper*类，这个问题搞了我好久，后来在论坛看到贴子才明白，希望后续官方有详细的文档。
  + 因为adt只包含了*android4.4(api-19)*的版本，所以默认使用的是这个版本，我自己拷了几个更低的版本，然后把*Build target*设为*android2.3(api-9)*，此时运行程序会提示下面错误：

*Unable to execute dex: java.nio.BufferOverflowException*

后来在网上查了一下，发现是*Android SDK Build tools*的BUG，把它升级到*19.0.3*后就好了：打开SDK Manager，找到19.0.3的Android SDK Build Tools，安装之。