

# Cocos2d-x 安装、配置、使用指南

火炬工作室

Spring 2017

## Contents

<b>Table of Contents</b>	<b>1</b>
<b>1 为什么有这篇文档</b>	<b>1</b>
<b>2 开发平台 Windows / 目标运行平台 Windows</b>	<b>1</b>
2.1 下载和安装 Python . . . . .	1
2.2 下载和安装 Visual Studio . . . . .	2
2.3 下载和配置 cocos2d-x . . . . .	2
2.4 Cocos Creator? . . . . .	3
2.5 Cocos Studio? . . . . .	3
2.6 验证你的安装 . . . . .	3
<b>3 开发平台 Windows / 目标运行平台 Android</b>	<b>4</b>
3.1 下载和安装 JDK . . . . .	4
3.2 下载和配置 SDK . . . . .	4
3.3 新建 AVD . . . . .	5
3.4 下载和配置 NDK . . . . .	6
3.5 下载和安装 Apache Ant . . . . .	6
3.6 让 cocos2d-x 能够使用 SDK/AVD/NDK . . . . .	6
3.7 验证你的安装 . . . . .	8
<b>4 好了，安装完了，我该怎么使用</b>	<b>8</b>
4.1 新建项目 . . . . .	8
4.2 在 Windows 下运行项目 . . . . .	8
4.3 在 Android 模拟器下运行项目 . . . . .	8
4.4 在 Android 实机上运行项目 . . . . .	9
4.5 写成脚本，快速运行 . . . . .	9

4.6 多人协作? . . . . .	9
5 你说的这些我都懂，但我的环境就是配置不好 . . . . .	9
6 其他说明 . . . . .	9
6.1 为什么下载 Android SDK 6.0 . . . . .	9
6.2 为什么编译这么慢 . . . . .	9
6.3 为什么不使用 Eclipse / AS . . . . .	10
6.4 为什么有两个目标运行平台 . . . . .	10
6.5 是否升级到 cocos2d-x 3.15 . . . . .	10

## List of Tables

## List of Figures

## 1 为什么有这篇文档

是的，书籍上、网络上到处都有讲解如何安装和配置 cocos2d-x 开发环境。但它们都过时且杂乱。为了

1. 避免下载和使用存在 bug 和特性不全的旧版本软件
2. 使全组拥有相同的开发环境，便于相互协作和调试

本组成员请务必**根据本文档一点不差的配置开发环境**。可选的配置项都在文中有说明，请不要省去你认为不必要的配置，导致以后协作调试时不方便。本配置在某些方面会和书上的，乃至官方的有所不同。不同的地方，以本文档为准。

注：有些软件包可能会由于网络原因下载不下来，群文件里面可能会有。

## 2 开发平台 Windows / 目标运行平台 Windows

本节所配置的环境是后续所有章节的基础。

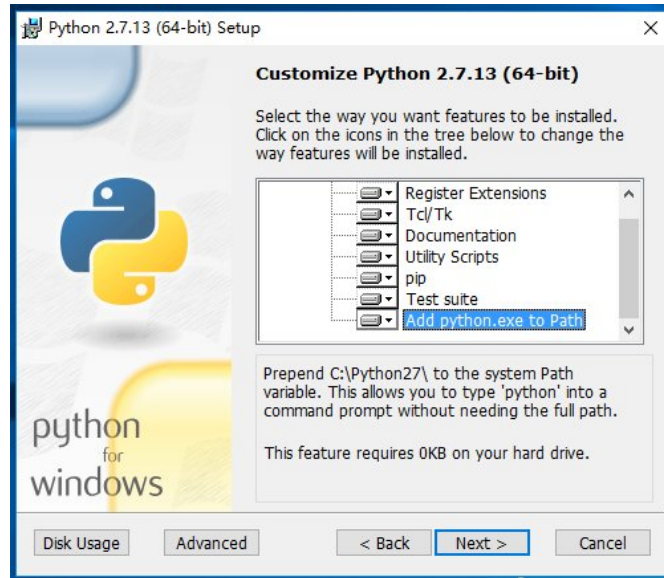
目标运行平台 Windows 使得我们可以很方便的在 Windows 平台下直接进行测试，而不用再去使用性能较弱的 Android 模拟器。

### 2.1 下载和安装 Python

cocos2d-x 大量使用 python 脚本来完成任务。

1. 下载 python 2.7.13: <https://www.python.org/ftp/python/2.7.13/python-2.7.13.amd64.msi>。

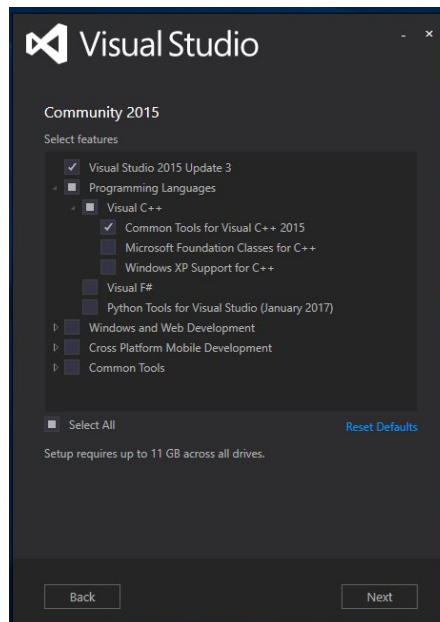
2. 安装时，在如图 ?? 界面选择 `add python.exe to Path`。这样就不用再手动添加 PATH 了。



## 2.2 下载和安装 Visual Studio

cocos2d-x 需要 2015 及以上的 VS 版本来更好的支持 C++11 特性。

如图 ??，安装时要自定义安装选项，注意勾选 “common tools for Visual C++ 2015”



## 2.3 下载和配置 cocos2d-x

cocos2d-x 下载后解压即可使用。

1. 下载 cocos2d-x 3.14.1: <http://www.cocos2d-x.org/filedown/cocos2d-x-3.14.1.zip>。  
下载完成后建议将其解压到某方便找到的位置，记其为 A；

2. 解压完成后将“A\tools\cocos2d-console\bin”加入系统变量 PATH 中，以保证能在 CMD 中使用 cocos 命令。

## 2.4 Cocos Creator?

No, No Cocos Creator。

Cocos 官网上虽然吹得天花乱坠，但它不支持 C++，只能用于 JS/Lua 项目，和我们的 C++ 项目一点关系都没有，帮不上我们任何忙。

## 2.5 Cocos Studio?

Cocos studio 项目在 2016 年 4 月的时候就 EOL 了。

官方声明如下：

Cocos Studio has been EOL'd as of April 2016. There will be no more releases of Cocos Studio. The last version of Cocos Studio is v3.10 and can be downloaded for Mac and Windows. This version of Cocos Studio requires version v3.10 of Cocos2d-x. This version of Cocos Studio works best in Windows 7 and OS X 10.10.

If you update Cocos2d-x to a version greater than v3.10, you may experience compatibility issues. If you run Cocos Studio on Windows 10 or OS X 10.11, you may experience compatibility issues.

在写这篇安装指南的时候，项目还没开始，游戏制作流程还没制定下来，无法知晓我们的项目是否需要使用 cocos studio。

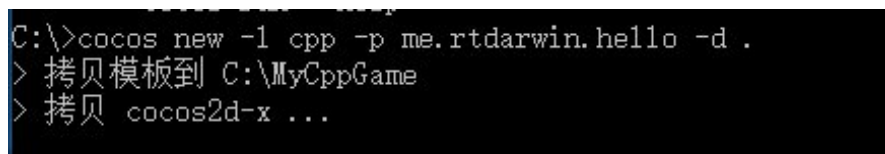
再加上现在市场上比较新的书籍上面关于 cocos studio 的介绍也较少，有介绍的也大多放到最后一章，这使得我们无法评估 cocos studio 的重要性。

所以本配置指南暂不建议安装 cocos studio。如果以后经过试验证明确实需要安装 cocos studio，届时会有另外一篇文档来说明。

## 2.6 验证你的安装

1. 在 CMD 中切换到某目录下 (记为 B)，输入如图 ?? 所示命令

`cocos new -l cpp -p me.rtdarwin.hello -d .` (将 me.rtdarwin.hello 替换为你喜欢的包名)。cocos 会在当前目录下建立一个名为 MyCppGame 的文件夹，里面即为新的项目。



```
C:\>cocos new -l cpp -p me.rtdarwin.hello -d .  
> 拷贝模板到 C:\MyCppGame  
> 拷贝 cocos2d-x ...
```

2. VS > 打开项目 B\MyCppGame\proj.win32\MyCppGame.sln
3. 在 VS 点击“本地调试器”，尝试编译运行 MyCppGame。首次编译可能需要较长时间，取决于电脑配置。编译完成后若弹出如 ?? 所示画面，则可认为环境搭建成功。



### 3 开发平台 Windows / 目标运行平台 Android

本节配置的环境建立在上一节所配置环境的基础上。如果你不需要将其安装到安卓上进行调试，那么上一节就足够了。

#### 3.1 下载和安装 JDK

// 假设你已经完成

#### 3.2 下载和配置 SDK

在 Android 平台上运行 cocos2d-x 游戏，并不需要在 Windows 安装 Eclipse 或者 Android Studio。这些 IDE 对 cocos2d-x 的支持也不是很好，有时会出现问题。

下面所做的配置，不影响你原先的安卓开发环境。当项目开发结束后，直接将本节配置所占用的目录（AndroidDev）删除即可。

1. 从郑州大学开源镜像站处 [10.66.0.111/android/SDKTools/](http://10.66.0.111/android/SDKTools/) 下载 `tools_r25.2.3-windows.zip`, 293M
2. 在某目录下（比如 D 盘根目录）下新建文件夹 `AndroidDev`，将上一步下载的 `tools.zip` 解压进去（记得消去嵌套的同名目录）  
注意：这个 `AndroidDev` 目录是我们这个配置所有安卓相关文件的存放目录，每个要安装的组件在其中占有一个子目录。
3. 在资源管理器下进入 `D:\Androiddev\tools\` 点击 `android.bat`，打开 SDK Manager
4. 在打开的 Android SDK Manager 窗口里面修改 SDK 设置：郑州大学 Android SDK 反代使用说明。<http://10.66.0.111/wiki/android.html>
5. 在打开的 Adnroid SDK Manager 窗口里面选择 Android 开发必需的 packages，进行安装：

- Tools::Adnroid SDK Platform-tools
- Tools::Adnroid SDK Build-tools
- Android 6.0(API 23)::SDK Platform
- Android 6.0(API 23)::Intel x86 Atom System Image
- Extras::Android Support Repository

上面提到即为我们环境搭建要用到的所有 packages。不过 SDK mgr 可能会自动勾选某些其他 packages，为了磁盘空间考虑，建议不安装。

注意不要更新到 Android SDK tools 25.3.0 以上（如果 SDK mgr 里面有的话）。这会导致我们所配置的 cocos2d-x 环境无法工作。如果你要了解具体原因，见 cocos2d-x v3.15 Release notes ([https://github.com/cocos2d/cocos2d-x/blob/v3/docs/RELEASE\\_NOTES.md](https://github.com/cocos2d/cocos2d-x/blob/v3/docs/RELEASE_NOTES.md)) 最后一段话。

### 3.3 新建 AVD

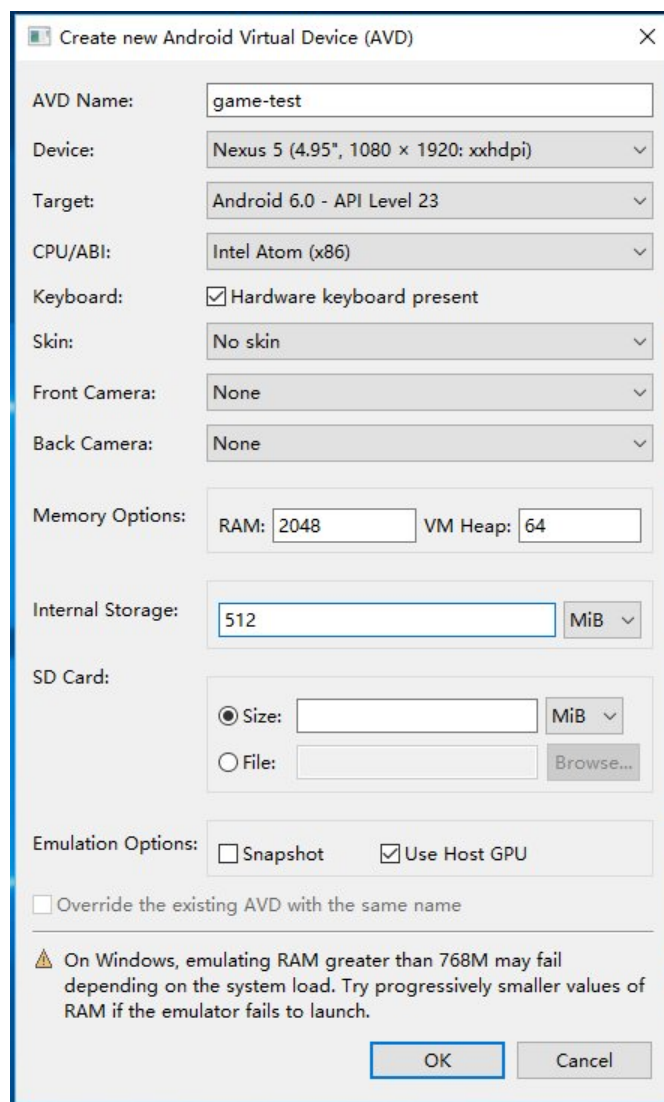
AVD 用于在电脑上模拟安卓设备。如果你确定仅使用自己的安卓手机调试的话，本节可以跳过。

1. 进入 CMD，使用命令

你的 AndroidDev 所在的位置（比如 C:）\AndroidDev\tools\android.bat avd  
打开 AVD 管理器

2. 新建如图 ?? 所示的 AVD。

注意：要勾选 “Use Host GPU”。勾选这个可以显著提高 AVD 的运行速度。



### 3.4 下载和配置 NDK

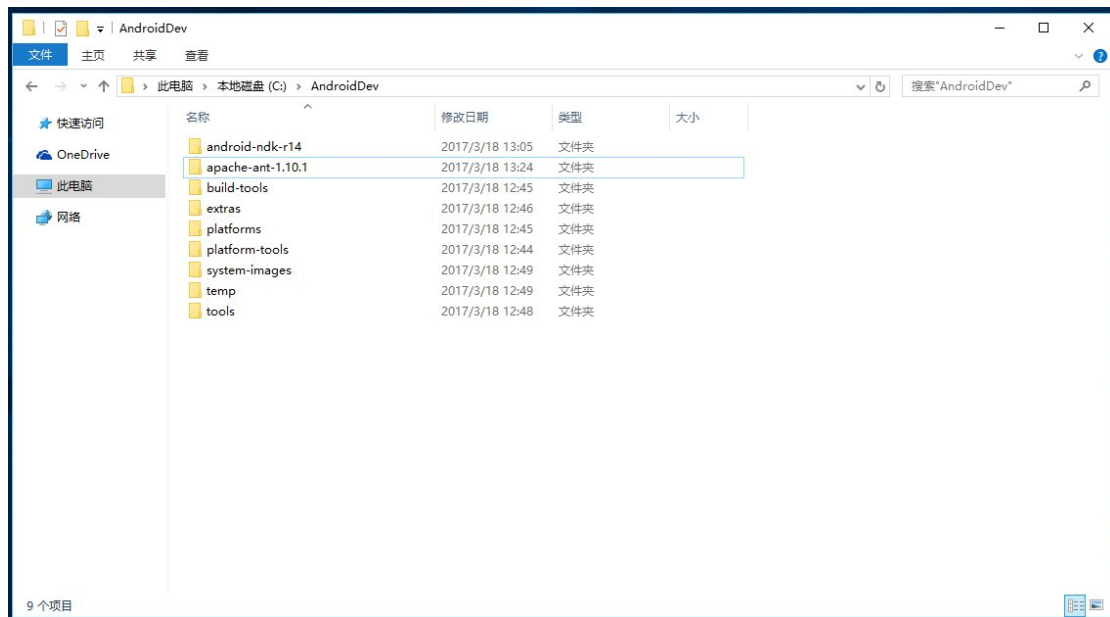
1. 从郑州大学开源镜像站处 <http://10.66.0.111/android/repository/> 下载 android-ndk-r14-windows-x86\_64.zip, 733M
2. 将其解压到我们在『下载和配置 SDK』一节所建的 AndroidDev 目录下。(记得消去嵌套的同名目录)

### 3.5 下载和安装 Apache Ant

1. 从 <https://www.apache.org/dist/ant/binaries/> 处下载 apache-ant-1.10.1-bin.zip。
2. 将其解压到 AndroidDev 目录下 (记得消去嵌套的同名目录)。

### 3.6 让 cocos2d-x 能够使用 SDK/AVD/NDK

以上步骤完成后, 应该在 AndroidDev 目录下形成如图 ?? 的目录。



但这样还不够，cocos2d-x 还不知道你的 SDK/NDK 装到哪里了。

1. 进入我们在 2.3 节将 cocos2d-x-3.14.1 解压至的目录 A。点击 `setup.py`
2. 弹出如图 ?? 命令行窗口提示要输入 NDK\_ROOT 的位置。我们输入前面将 NDK 所解压的位置后回车确认。（在我的电脑上是 `C:\AndroidDev\android-ndk-r14`，注意不要在最后加个反斜杠）

```

选择C:\Python27\python.exe
->Added COCOS_CONSOLE_ROOT=C:\cocos2d-x-3.14.1\tools\cocos2d-console\bin
->Check environment variable COCOS_X_ROOT
->Search for environment variable COCOS_X_ROOT...
->COCOS_X_ROOT not found

->Add directory "C:" into PATH succeed!

-> Add COCOS_X_ROOT environment variable...
->Added COCOS_X_ROOT=C:\

->Check environment variable COCOS_TEMPLATES_ROOT
->Search for environment variable COCOS_TEMPLATES_ROOT...
->COCOS_TEMPLATES_ROOT not found

->Add directory "C:\cocos2d-x-3.14.1\templates" into PATH succeed!

-> Add COCOS_TEMPLATES_ROOT environment variable...
->Added COCOS_TEMPLATES_ROOT=C:\cocos2d-x-3.14.1\templates

->Configuration for Android platform only, you can also skip and manually edit your environment variables

->Check environment variable NDK_ROOT
->Search for environment variable NDK_ROOT...
->NDK_ROOT not found

->Search for command ndk-build in system...
->Command ndk-build not found

->Please enter the path of NDK_ROOT (or press Enter to skip):
  
```

3. 提示要输入 ANDROID\_SDK\_ROOT, 我们输入 AndroidDev 目录的位置后回车确认。（在我的电脑上是 `C:\AndroidDev`，注意不要在最后加个反斜杠）
4. 提示要输入 ANT\_ROOT, 我们输入 AndroidDev 目录下 Ant 目录下 bin 目录的位置后回车确认。（在我的电脑上是 `C:\AndroidDev\apache-ant-1.10.1\bin`，注意不要在最后加个反斜杠）



### 3.7 验证你的安装

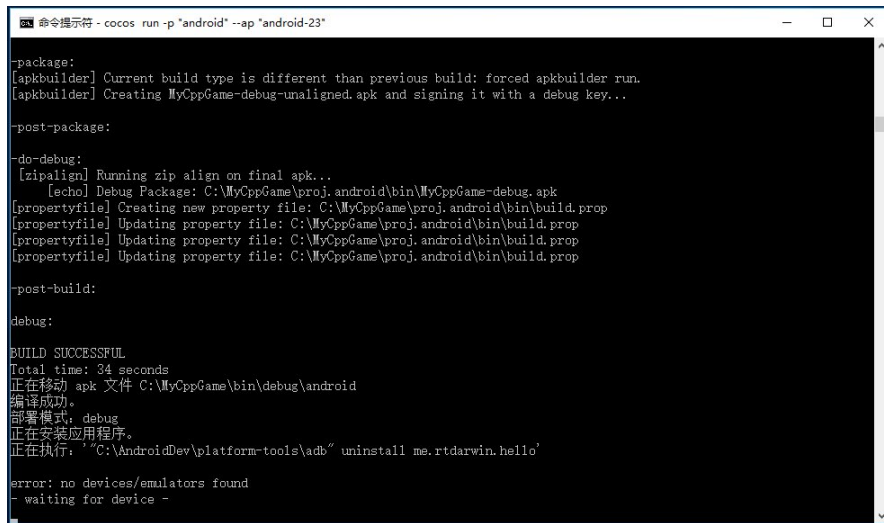
1. 使用 CMD cd 入我们在 2.6 节所建立项目的目录下。执行如图 ?? 所示的命令

```
cocos run -p android --ap android-23 --app-abi x86
```



```
C:\MyCppGame>cocos run -p "android" --ap 23
编译模式: debug
使用 Eclipse 工程: C:\MyCppGame\proj.android
在 Android SDK的platforms下未找到文件夹 '23', 请使用 --ap 指定需要的API版本。
C:\MyCppGame>cocos run -p "android" --ap "android-23"
编译模式: debug
使用 Eclipse 工程: C:\MyCppGame\proj.android
正在执行: "C:\AndroidDev\tools\android" update project -t android-23 -p C:\MyCppGame\proj.android'
```

2. 经过一个漫长的编译期后, 若出现如图 ?? 所示 — wait for device, 则可认为安装成功



```
命令提示符 - cocos run -p "android" --ap "android-23"

-package:
[apkbuilder] Current build type is different than previous build: forced apkbuilder run.
[apkbuilder] Creating MyCppGame-debug-unaligned.apk and signing it with a debug key...

-post-package:

-do-debug:
[zipalign] Running zip align on final apk...
[echo] Debug Package: C:\MyCppGame\proj.android\bin\MyCppGame-debug.apk
[propertyfile] Creating new property file: C:\MyCppGame\proj.android\bin\build.prop
[propertyfile] Updating property file: C:\MyCppGame\proj.android\bin\build.prop
[propertyfile] Updating property file: C:\MyCppGame\proj.android\bin\build.prop
[propertyfile] Updating property file: C:\MyCppGame\proj.android\bin\build.prop

-post-build:

debug:

BUILD SUCCESSFUL
Total time: 34 seconds
正在移动 apk 文件 C:\MyCppGame\bin\debug\android
编译成功。
部署模式: debug
正在安装应用程序。
正在执行: "C:\AndroidDev\platform-tools\adb" uninstall me.rtdarwin.hello'
error: no devices/emulators found
- waiting for device -
```

## 4 好了, 安装完了, 我该怎么使用

上面那么多安装步骤最终都是为了能正常使用 cocos 这个命令。也就是说配置好了, 你就可以忘掉上面那些东西了, 只需要使用 cocos 这个命令就可以了。

### 4.1 新建项目

```
cocos new -l cpp -p <what-you-like> -d <where-you-like>
```

### 4.2 在 Windows 下运行项目

使用 VS 打开项目目录/proj.win32/MyCppGame.sln, 然后点击调试。

### 4.3 在 Android 模拟器下运行项目

1. 在 CMD 下输入下面的命令, 打开 Android 模拟器管理器:

```
你的 AndroidDev 所在的位置 (比如 C:) \AndroidDev\tools\android.bat avd
```

2. 启动某个模拟器

3. CMD 在项目根目录下运行 `cocos run -p android --ap android-23 --app-abi x86`

## 4.4 在 Android 手机上运行项目

1. 连接你的手机
2. CMD 在项目根目录下运行 `cocos run -p android --ap android-23 --app-abi armeabi-v7a`

注意：本节所使用命令并不用来最终部署。关于部署，届时会有另外一篇文档来说明。

## 4.5 写成脚本，快速运行

每次使用 CMD 来输入命令是很不方便的。可以写几个脚本来自动运行，当我们需要的时候只要点一下就能执行任务了。

示例（这些示例脚本放在本文档所在目录的 scripts 目录下，如果你的环境配置目录和示例相同，直接将 scripts 目录拷贝到项目根目录下即可使用）：

1. sdk-mgr.bat: `C:\AndroidDev\tools\android.bat`
2. avd-mgr.bat: `C:\AndroidDev\tools\android.bat avd`
3. run-emulator.bat: `cd .. cocos run -p android --ap android-23 --app-abi x86`
4. run-phone.bat: `cd .. cocos run -m release -p android --ap android-23 --app-abi armeabi-v7a`

## 4.6 多人协作？

多人协作模型现在还没有定下来，届时会有另外一篇文档来说明。

# 5 你说的这些我都懂，但我的环境就是配置不好

第三章『开发平台 Windows / 目标运行平台 Android』配置起来确实很难，再加上文档是短时间写出来的，没有仔细检查。如果你无论如何都配置不好，很可能是文档本身的错误，请及时报于开发组长『张晨』。

## 6 其他说明

### 6.1 为什么下载 Android SDK 6.0

// TODO

### 6.2 为什么编译这么慢

C++ 编译的特点，就是慢。一般而言第一次编译慢，以后就快了（因为是基于 Makefile 的）。

## 6.3 为什么不使用 Eclipse / AS

简单的来说：没有必要。

Cocos 对特定平台的依赖很小，几乎在所有的平台上都是：平台特有的启动器将应用启动之后就全权交给 cocos 引擎，让 cocos 引擎自己来工作了。

谷歌已经明确表示不支持 Eclipse 了。Eclipse 版本的 ADT，在 2015 年 6 月谷歌就不再维护了。官方声明

( <https://developer.android.com/studio/tools/sdk/eclipse-adt.html>,  
<https://android-developers.googleblog.com/2015/06/an-update-on-eclipse-android-developer.html> )

如下：

The Eclipse ADT plugin is no longer supported per our announcement. Android Studio is now the official IDE for Android, so you should migrate your projects to Android Studio as soon as possible. For more information on transitioning to Android Studio, see [Migrate to Android Studio from Eclipse](#).

cocos2d-x 在 v3.15 的 release notes 中也提到以后会移除对 eclipse 项目的支持。

**cocos2d-x 对 Android Studio 支持不完善**。不过 v3.15（尚未发布）明确表示“full Android Studio supports”。使用现在的 v3.14.1 版本，配置复杂度和我们要做的几乎相同（是的，我亲身尝试过）。

**Eclipse/AS 作用很小**。我们不需要写任何 Android 平台相关的 Java 代码 — 不需要在 Eclipse 或者 Android Studio 里面进行任何编码和调试。Visual Studio 的调试功能比 Eclipse+CDT 或者 AS 要强的多。

开着 Eclipse/AS 也仅仅能辅助一下部署过程。而这个部署，我们可以用仅仅一条命令来完成。

**从头配置更简单**。每个人已经安装的 IDE 不同，已经配置的 Android 开发环境不同。配置新的环境远远比调整原有的环境简单和容易。

## 6.4 为什么有两个目标运行平台

是的，我们在开发的时候两个目标运行平台都要使用。

目标运行平台 Windows 性能强，启动快，是用于快速验证想法和代码的，平常大部分的工作都在 Windows 上进行。

目标运行平台 Android 启动较慢，性能较弱。但游戏的真实体验只有在手机上动手玩一玩才能知道。所以必要的时候（经常），我们还需要 Android 来了解真实的体验，以此来调整游戏参数。

## 6.5 是否升级到 cocos2d-x 3.15

这篇配置指南定稿的时候 v3.15 尚未发布。而且从 cocos2d-x 团队公布的开发进度来看，v3.15 发布可能还需要一段时间。所以建议使用 v3.14.1 版本进行开发。如果以后发现确实需要 v3.15，届时会有另外一篇文档来说明。