|  |
| --- |
| Beijing CHUKONG Technology Co |
| Getting Started with Cocos3D |
|  |
|  |
| **Cocos Team** |
| **2014/2/26** |

|  |
| --- |
| This document explains how to get up and running quickly with Cocos3D. |

目录

[Welcome 0](#_Toc381264735)

[ABOUT COCOS3D 0](#_Toc381264736)

[COCOS3D Features 0](#_Toc381264737)

[Setting up Development Environments 0](#_Toc381264738)

[Create a New cocos3d project on multi-platforms 0](#_Toc381264739)

[Windows Environment Setup 1](#_Toc381264740)

[Mac OS X Environment Setup 6](#_Toc381264741)

[Support 0](#_Toc381264742)

[Website 0](#_Toc381264743)

[Ticketing System 0](#_Toc381264744)

[Feedback 1](#_Toc381264745)

# Welcome

Welcome to CHUKONG® COCOS3D® 1.0. This guide describes what COCOS3D is, how to use and

install the SDK, and where to look for additional information.

## ABOUT COCOS3D

cocos3d is a significant extension to cocos2d, a popular, well-designed framework for building cross plantform games and applications that play out in 3D. Although on the one hand it is possible to start with the cocos3d Application template and develop a 3D application without knowing too much about the workings of cocos3d, to get the most out of this document, you should familiarize yourself with cocos2d. You can learn more about cocos2d at the cocos2d Wiki.

# Setting up Development Environments

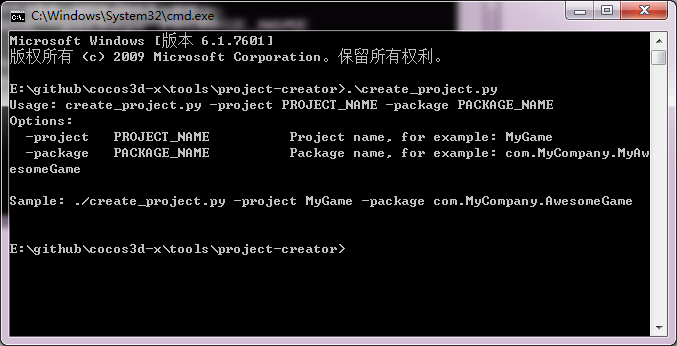
## Create a New cocos3d project on multi-platforms

 Preparation  
Before create your game with cocos3d-x, setup your environment first.

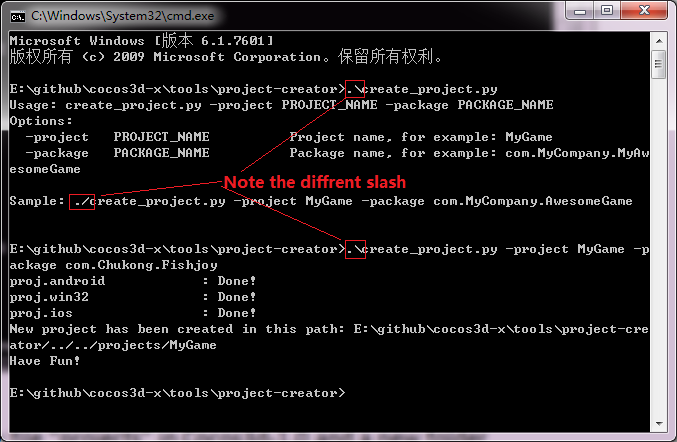
1. download the “Git” and setup (<http://git-scm.com/download>) , maybe you also need “TortoiseGit” the Interface to Git Version Control(<http://code.google.com/p/tortoisegit/wiki/Download?tm=2>)

2. clone Cocos3d from (<https://github.com/cocos2d/cocos3d-x>).

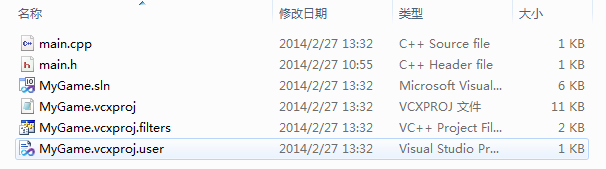
3. download python version 2.72(<http://www.python.org/download/releases/2.7.2/>).

 Create multi-platforms cocos3d project  
Installed the python you just downloaded. Locate the directory which you Cocos3d-1.0 in, then run the create\_project.py at cocos3d/tools/project-creator/ in command line, then you will see below  


Input again follow the instructions



Check out your directory, you will find a new file “projects” in Cocos3d-1.0 and a new folder “MyGame” you just created in “Cocos3d-1.0/projects”. Now import the “proj.win32” in the VS2010 and enjoy the programming.



## Windows Environment Setup

#### Setup android development environment

This document describes how to run a simple ‘HelloCpp’ sample under NDK with Windows (using Cygwin), or Linux.  
Your NDK version must be r5 or above.

##### NOTE:

This document doesn’t describe how to set up NDK. Please refer to the following website if you wish to learn more about the Android NDK, and how to download and install it:  
<http://developer.android.com/sdk/ndk/overview.html>

##### Your Cygwin’s version should be 1.7 or above

This is described in the NDK documentation.

##### NOTE

We recommend you to use Eclipse as your IDE when programming for Android, not only for cocos2d-x projects, but all other apps as well, because it is the IDE that Google supports. The Android tutorials on this site will also be using Eclipse for its examples, making it easier to follow. Other IDEs are fine, but we do not discuss them here.

##### 2. Compiling

You should define environment variable NDK\_ROOT before compiling.

“NDK\_ROOT” means the path of NDK you installed in.

##### Linux

To compile the project, simply run build\_native.sh in your shell. In Linux, this simply means opening up a shell prompt such as *bash*.

Here is an example:

cd /usr/workspace/cocos3d-x/HelloCpp/proj.android/

./build\_native.sh

This assumes that you put cocos3d-x in /usr/workspace, and left HelloCpp in its default location.

##### Windows using Cygwin

The same applies for Windows, but you will need to use Cygwin’s *bash*, rather than the command prompt. This is because build\_native.sh is a shell script meant for Linux, which calls a Linux-based tool included in Android-NDK. If you were to create an NDK app without using cocos2d-x, you would still have to call the tool, *ndk\_build*, from Cygwin. build\_native simply automates the process of building the project.

Here is an example:

bash

cd /cygdrive/c/cocos3d/samples/HelloCpp/proj.android

./build\_native.sh

This assumes that you put cocos3d in C:/, and left HelloCpp in its default location.

##### 3. Running

Running HelloCpp is the same as other ndk samples.

##### 3.1 Build Project

This step will generate R.java automatically, it will also compile the java code and generate the .apk.

##### 3.2 Clean Project

If you recompile the native code and want to pack the .so to .apk, you should do this step. Step 1 will not pack the .so to .apk except the first time.

##### 3.3 Run Project

Right click the project, select “run as”, then select “Android Application”. Don’t forget to start your emulator before running.

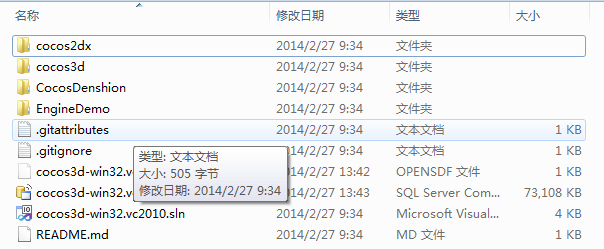
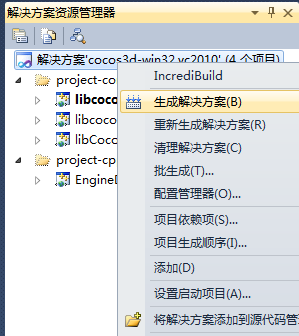
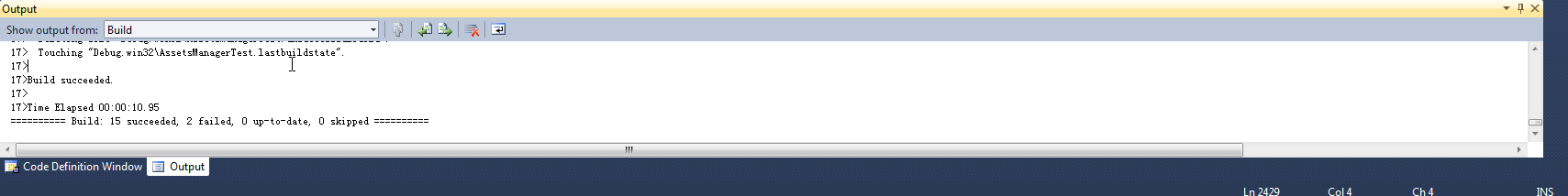
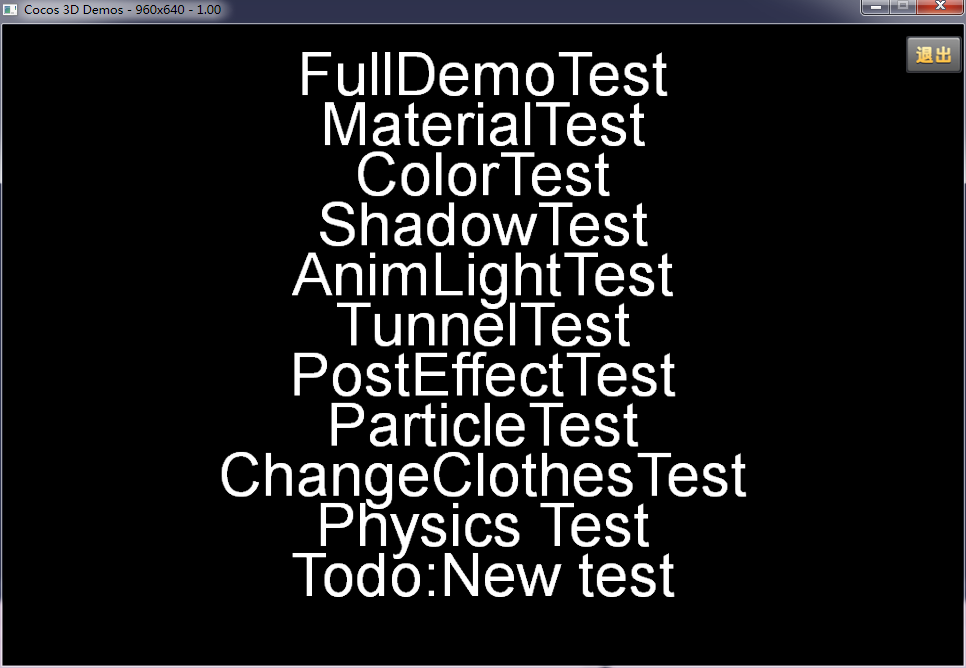
##### 4. Warning

You should do “clean project” after recompiling native code. If not, the latest .so will not be packed into .apk.



#### Setup win32 development environment

**Using Cocos3d version 1.0 in VS2010**

1. import cocos3d-win32.vc2010.sln in VS2010 at workingdirectory/cocos3d-1.0. As below.  
     
   Then Select “Build Solution” as below  
     
   About 5 minutes later, if you see the below in your Output Window, congratulation!  
   
2. Test  
   Now let’s do some test. Select HelloCpp and choose “Set as StartUp Project” as below  
     
   Start debugging  
     
   

## Mac OS X Environment Setup