Chukong Technologies

Cocos3D Getting Started

PC-Android

Contens

1	Preparations		.0
	Create cocos3d-x project		
	Compile Android project		
	3.1	Configuration environment variable	. 1
	3.2	Compile lib file	. 1
	3.3	Generate apk file	. 2
	3.4	Run	.4

1 Preparation

- Software:
 - 1. Windows (In this case, we use Windows7 64 bit)
 - Jre (In this case, we use Jre -7u51-windows-x64)
 Download: http://www.java.com/en/download/manual.jsp
 - 3. Python (In this case, we use python2.7.5 and install at C:\)

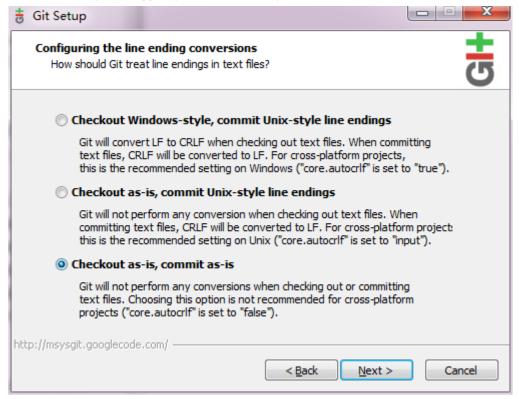
 Download: http://www.python.org/download/releases/2.7.5/
 - 4. Cygwin (In this case, we use Cygwin for 64-bit versions of Windows and install at D:\)

 Download: http://cygwin.com/install.html
 - 5. adt-bundle (In this case, we use adt-bundle-windows-x86_64-20131030 and unzip it at D:\)
 - Download: http://developer.android.com/sdk/index.html#download
 - 6. NDK (In this case, we use android-ndk-r9b-windows-x86_64 and unzip it at D:\)

 Download: https://developer.android.com/tools/sdk/ndk/index.html
- install Git (In this case we use git version 1.7.10-preview20120409)

Download: http://git-scm.com/download/win

Note: this step we suggest you choice third option shown below



• Get cocos3d-x source from GitHub: https://github.com/cocos2d/cocos3d-x (in this case, we put the source into E:\)

2 Create cocos3d-x project

Run Cygwin as Administrator and navigate to cocos3d-x\tools\project-creator directory

```
Your group is currently "mkpasswd". This indicates that your gid is not in /etc/group and your uid is not in /etc/passwd.

The /etc/passwd (and possibly /etc/group) files should be rebuilt. See the man pages for mkpasswd and mkgroup then, for example, run mkpasswd -l [-d] > /etc/passwd mkgroup -l [-d] > /etc/group

Note that the -d switch is necessary for domain users.

ck01-125@LvLong ~
$ cd E:/cocos3d-x/tools/project-creator

ck01-125@LvLong /cygdrive/e/cocos3d-x/tools/project-creator

$
```

Input "Python ./create_project.py -project fishjoy -package com.chukong.fishJoy"

```
$ python ./create_project.py -project fishjoy -package com.chukong.fishjoy
```

Finally, the newly created project will be located in cocos3d-x\projects

```
proj.android : Done!
proj.win32 : Done!
proj.ios : Done!
New project has been created in this path: E:\cocos3d-x\tools\project-creator/..
/../projects/fishjoy
Have Fun!
```

3 Compile Android project

3.1 Configuration environment variable

In this case, my environment variable as below:

ANDROID_SDK D:\adt-bundle-windows\sdk

NDK_ROOT D:\android-ndk-r9b

Add D:\cygwin\bin to Path tail.

Add C:\ Python27 to Path tail

3.2 Compile lib file

1. Run Cygwin as Administrator, navigate to cocos3d-x\projects\fishjoy\proj.android directory, and run build native.sh

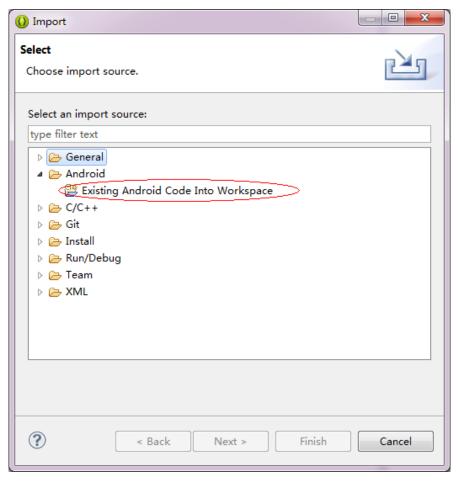
```
ck01-125@LvLong /cygdrive/e/cocos3d-x/projects/fishjoy/proj.android
$ ./build_native.sh |
```

The output as follows:

```
23
/cygdrive/e/cocos3d-x/projects/fishjoy/proj.android
                                                                    Compile++ thumb: cocos2dx_static <= CCTextFieldTTF.cpp
Compile++ thumb: cocos2dx_static <= CCTexture2D.cpp
Compile++ thumb: cocos2dx_static <= CCTextureAtlas.cpp
Compile++ thumb: cocos2dx_static <= CCTextureCache.cpp
Compile++ thumb: cocos2dx_static <= CCTextureETC.cpp
Compile++ thumb: cocos2dx_static <= CCTexturePVR.cpp
Compile++ thumb: cocos2dx_static <= CCTMXLayer.cpp
Compile++ thumb: cocos2dx_static <= CCTMXLayer.cpp
Compile++ thumb: cocos2dx_static <= CCTMXObjectGroup.cpp
Compile++ thumb: cocos2dx_static <= CCTMXTiledMap.cpp
Compile++ thumb: cocos2dx_static <= CCTMXXMLParser.cpp
Compile++ thumb: cocos2dx_static <= CCTMXMLParser.cpp
Compile++ thumb: cocos2dx_static <= CCTouchDispatcher.cpp
Compile++ thumb: cocos2dx_static <= CCTouchHandler.cpp
Compile++ thumb: cocos2dx_static <= CCTouchHandler.cpp
Compile++ thumb: cocos2dx_static <= CCTouchLandler.cpp
Compile++ thumb: cocos2
           armeabi
          armeabi]
          armeabi
          armeabij
           armeabi
           armeabi
           armeabi
           armeabi<sup>†</sup>
           armeabi
           armeabi
           armeabi
           armeabi
           armeabi
          armeabi<sup>:</sup>
          armeabi
   [armeabi] Compile thumb : cpufeatures <= cpu-features.c
[armeabi] StaticLibrary : libcpufeatures.a
[armeabi] SharedLibrary : libfishjoy.so
[armeabi] Install : libfishjoy.so => libs/armeabi/libfishjoy.so
make: 离开目录"/cygdrive/e/cocos3d-x/projects/fishjoy/proj.android"
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               Ξ
        k01-125@LvLong /cygdrive/e/cocos3d-x/projects/fishjoy/proj.android
```

3.3 Generate apk file

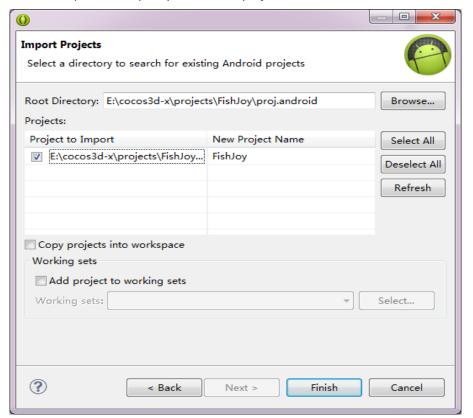
Run eclipse as Administrator, right-click on a blank space in the Package Explorer, select import



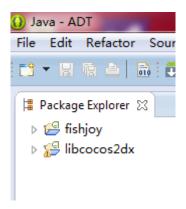
Select "Existing Android Code into Workspace" and then click the Browser button to import cocos2d-x lib for android projects



Repeat the above process, import your android project



Click the Finish button to complete the import process and now the Package Explorer window will appear both projects



3.4 Run

Right click fishjoy project -> Run as -> Android Application.

Note: at this step you may have permission problem as below:

```
(skipping file '.gitignore' due to ANDROID_AAPT_IGNORE pattern '.*')
Unable to add 'E:\cocos3d-x\projects\fishjoy\proj.android\assets\CloseNormal.png': Zip add failed

RROR: unable to process assets while packaging 'E:\cocos3d-x\projects\fishjoy\proj.android\bin\resources.ap_'

RROR: packaging of 'E:\cocos3d-x\projects\fishjoy\proj.android\bin\resources.ap_' failed
```

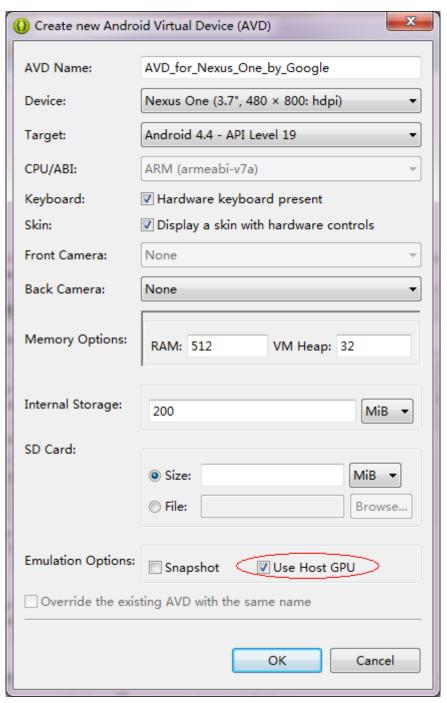
Double click the follow script



Now, right click proj.android folder select "管理员取得所有权限":



Select menu "Window->Andorid Virtual Device Manage" to create a avm(check "Use Host GPU")



Finally, start the avm

