

Installing OSG for Android on Linux OS

Written and editing by James and Sahar

Build the environment for OSG on Android

1. Download and install "android sdk"

2. Download "android ndk" and do following steps:

- Unzip the ndk and put the ndk folder into the directory you want (we use the path "/home/username/" as example for the following steps)
- Open the directory "/home/username/" and press Ctrl+H to show hidden files
- Open the hidden file ".cshrc" if the shell is "tcsh"
- Add the following 2 lines in ".cshrc"
#android
setenv PATH /home/username/android-ndk-rX #
(X is the version of ndk, depends on what version of ndk you download)
- Open a terminal, cd into "/home/username/" and enter the command "**source .cshrc**"

3. Install OpenSceneGraph(OSG):

- Create a folder for OSG (e.g. create a folder "OSGAndroid" under /home/username/, full path: /home/username/OSGAndroid)
- Under the folder "OSGAndroid", open a terminal and enter the command "**svn co <http://www.openscenegraph.org/svn/osg/OpenSceneGraph/tags/OpenSceneGraph-3.0.1>**" (here is OSG version 3.0.1)
- The following steps are based on the website:
<http://forum.openscenegraph.org/viewtopic.phpt=10076>
 - Create 2 folders "build", "osginstall" under the main OSG folder (e.g. /home/username/OSGAndroid/OpenSceneGraph-3.0.1)
 - Download the 3rdparty package from this website:
"<http://www2.ai2.upv.es/difusion/osgAndroid/3rdpartyAndroid.zip>"
 - Unzip the 3rdparty package and put it into the main OSG folder
 - cd into the "build" folder and open a terminal
 - In the terminal, enter this:

```
cmake .. -DOSG_BUILD_PLATFORM_ANDROID=ON
-DDYNAMIC_OPENTHREADS=OFF -DDYNAMIC_OPENSCENEGRAPH=OFF
-DOSG_GL_DISPLAYLISTS_AVAILABLE=OFF
-DOSG_GL_MATRICES_AVAILABLE=ON
-DOSG_GL_VERTEX_FUNCS_AVAILABLE=ON
-DOSG_GL_VERTEX_ARRAY_FUNCS_AVAILABLE=ON
-DOSG_GL_FIXED_FUNCTION_AVAILABLE=ON
-DOSG_CPP_EXCEPTIONS_AVAILABLE=OFF -DOSG_GL1_AVAILABLE=OFF
-DOSG_GL2_AVAILABLE=OFF -DOSG_GL3_AVAILABLE=OFF
-DOSG_GLES1_AVAILABLE=ON -DOSG_GLES2_AVAILABLE=OFF -DJ=4
-DCMAKE_INSTALL_PREFIX=/home/username/OSGAndroid/OpenSceneGraph-
3.0.1/osginstall
```

be aware of the last 2 parameters, -DJ=number of CPU cores in your computer, and the install path

- after it configure, enter the command "**make**" in terminal, after it finished, enter the

command **"make install"** (make and make install will take some time)

4. Assume you downloaded and installed "Eclipse IDE for Java Developers" and installed android sdk with it. Then we install CDT for Eclipse (allow you to edit .cpp, .h files in Eclipse)

- In Eclipse, go to "Help" (on the menu bar)--> "Install New Software..." --> "Add...", enter "CDT" in the field "Name", enter "<http://download.eclipse.org/tools/cdt/releases/galileo>" in the field "Archive...". Then just check all items and next → next → finish to install CDT

To compile native (c++,c) code for Android:

1. Open a terminal and cd into the Eclipse workspace then cd into the classes folder (e.g. `/home/username/workspace/project_name/bin/classes`)
2. Enter the command `"javah -jni package.name.ClassThatHasNativeFunctions"`
3. Open a terminal and cd into `"/home/username/workspace/project_name"`, you can see the header file for java native functions is generated (named as `"package_name_ClassThatHasNativeFunctions.h"`)
4. In Eclipse, create a folder named `"jni"` under your Android project, copy the header file (`package_name_ClassThatHasNativeFunctions.h`) to the jni folder
5. Create another .cpp file (say `"native.cpp"`) to include the header (`package_name_ClassThatHasNativeFunctions.h`) and implement the functions
6. Create a file named `"Android.mk"` which is the make file for Android native code, edit `Android.mk`

An example of `Android.mk`:

```
-----  
LOCAL_PATH := $(call my-dir)  
  
include $(CLEAR_VARS)  
include /some/libraries/you/need  
  
LOCAL_MODULE := NativeLib  
LOCAL_SRC_FILES := native.cpp  
LOCAL_LDLIBS += -llog -ldl  
  
include $(BUILD_SHARED_LIBRARY)  
-----
```

Optional:

You can add a file named `"Application.mk"`

An example of `Application.mk`

```
-----  
APP_STL := gnustl_static  
APP_CPPFLAGS := -frtti -fexceptions  
APP_ABI := armeabi-v7a #armeabi-v7a  
                #specify which architecture(s) your application supports  
-----
```

7. To compile the native code, open a terminal and cd into
"/home/username/workspace/project_name" and enter the command "ndk-build"
8. Press the "Run" button in Eclipse to run the application on Android

Note: Be sure to add the following lines in the java code where you declare the java native functions

```
static{  
    System.loadLibrary("NativeLib");  
    (must be the same as the module name you specified in  
     "LOCAL_MODULE" in Android.mk)  
}
```