Guide:

\*Note this guide is ok for version Unity 5.4.0f3 and OpenCV 3.1 \*

Preperations:

\* download Unity from <https://unity3d.com/get-unity/download>

\* download Android Studio from <https://developer.android.com/studio/index.html>

\* download OpenCV for android from <http://opencv.org/downloads.html>

First you need to get SSIG key to develop on your phone.

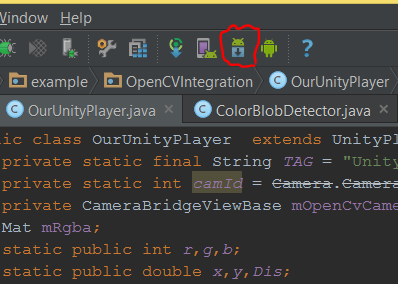
Plug your phone to the computer

( notice it should be in "developer mode" .. if you don’t know how to do it check on google ☺ )

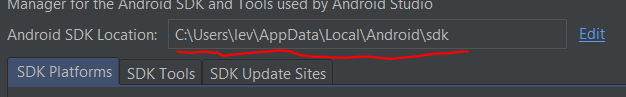
To get This key you should enter: <https://dashboard.oculus.com/tools/osig-generator/>

Follow those instructions:

Go to Android Studio -> SDK manager

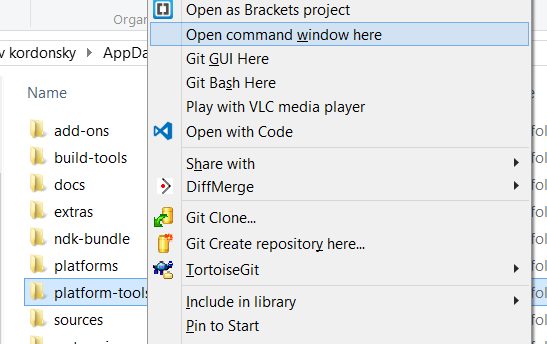


Go to the SDK folder:

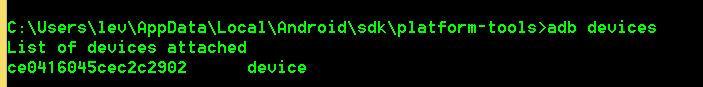


In that Folder , Press shift+right Mouse on platform-tools to enter cmd in that folder

(you can do at any other cmd way you like..)



Now write "adb devices" and you should get :



This is your device ID – you need this to the website to get your SSIG key.

Put it in a folder you can find later.. you will need it.

**Unity Project Start**

Open new Unity Project .

Add Sphere (GameObject -> 3D object -> sphere)

Add script to Sphere



\*Notice the sphere's name is "Sphere"

\*ajc will be the java class that we will build later , this class has static methods getR,getG,getB that return int.

Open a new folder in your Assets, name it "Plugins".

In Plugins open a folder named "Android".

In Android open a folder named "assets". (final path should be Assets \Plugins\Android\assets)

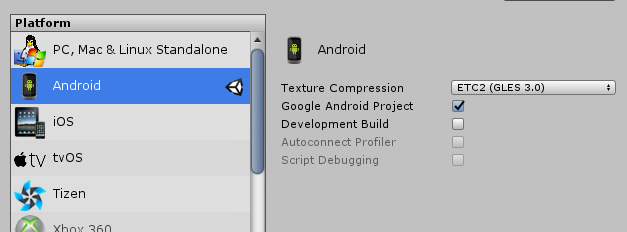
Copy Your SSIG Key to that folder!

Go to File -> Build Settings

Choose Android and check Google Android Project

( You might need to install SDK,NDK for that.. do that with android studio's SDK manager)

Change "Texture Compression" to ETC2 (GLES 3.0)



Export to a folder

**Android Project Start**

Open Android Studio :

File -> New -> Import Project -> choose the folder you exported to (from previous page)

(as for this unity version, you will have two folders – OVRPlugin and #yourname )

choose #yourname folder.

Notice that android studio will ask you to choose a folder when you use import project – choose a new folder name

And accept the "do you want to create new folder"

From now on we will only use this new folder.

For some reason I had some files missing in my OVRPlugin folder so I copied

#yourNewFolder\#yourAppName\src\main\res -> drawable and values (copy both folders)

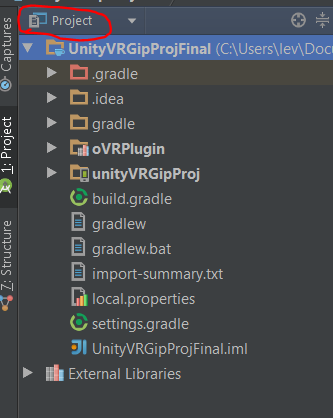
To #yourNewFolder \oVRPlugin\src\main\res\

Do those steps in case you see problems in Android Studio .

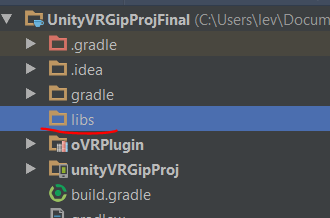
Build the project just to make sure everything is ok

**OpenCV static initialization:**

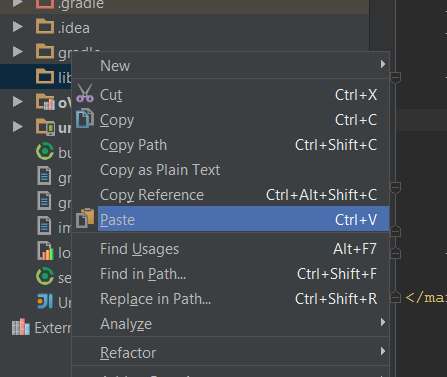
Go to Project view



Add new folder "libs" (right mouse on main folder -> new -> directory)

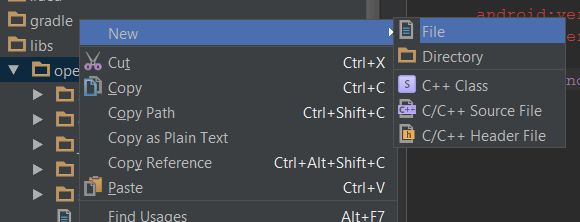


Copy the folder OpenCVFolder\sdk\java into your new "libs"



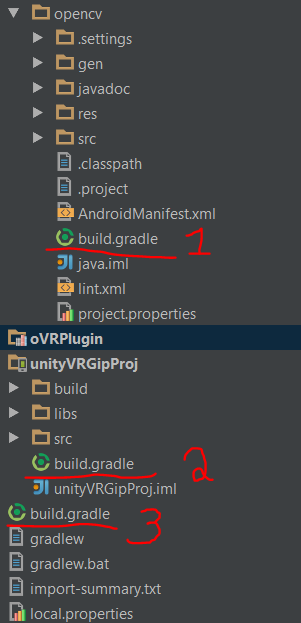
call the new folder "opencv"

Add new file in "opencv" new folder and Call it "build.gradle"



Notice now you have 3 "build.gradle" (actually 4 but OVRPlugin doesn’t matter..)

1 is now empty



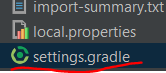
Copy this to the new file:

apply plugin: 'com.android.library'  
  
buildscript {  
 repositories {  
 jcenter()  
 }  
 dependencies {  
 classpath 'com.android.tools.build:gradle:1.5.0'  
 }  
}  
  
android {  
 compileSdkVersion 23  
 buildToolsVersion "23.0.2"  
  
 defaultConfig {  
 minSdkVersion 9  
 targetSdkVersion 23  
 }  
  
 sourceSets{  
 main{  
 manifest.srcFile 'AndroidManifest.xml'  
 java.srcDirs = ['src']  
 resources.srcDirs = ['src']  
 res.srcDirs = ['res']  
 aidl.srcDirs = ['src']  
 }  
 }  
}

\*Notice that "buildscript" part should be similar to same code in file3(main folder)

And "android" part – the versions – should be similar to file2 (yourAppName folder)

Go to setting.gradle (main folder)



and add:

include ':libs:opencv'

you can click Sync now and it should be ok… ☺

Go to OpenCVFolder\sdk \native\libs

Now in this folder you have many folders – but you need only one of them , it depends on your phones architecture.

For Samsung S7 I only needed " armeabi-v7a"

Copy the folder to

#yourNewFolder\#yourAppName\src\main\jniLibs

Go to File -> Project Structure

Select #yourAppName

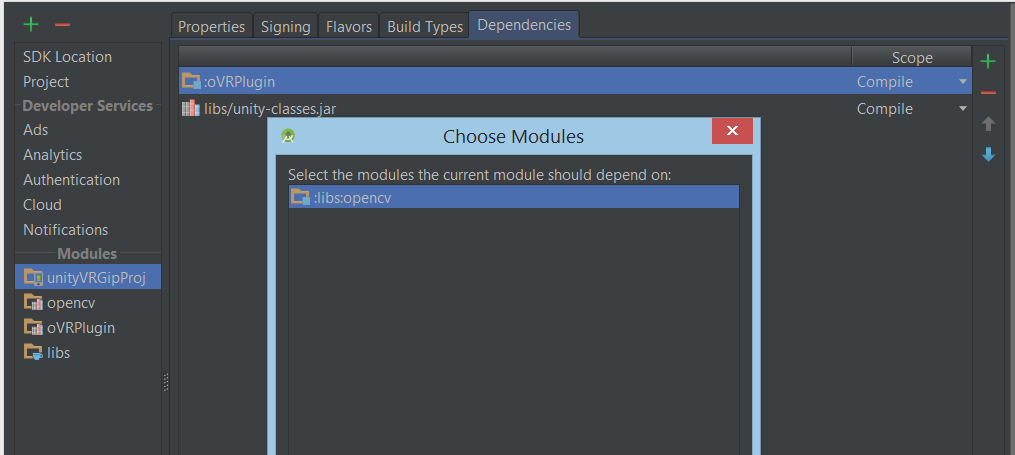
Click Dependencies

Click +

Select Module Dependencies

Select :libs:opncv and press OK

OpenCv is now integrated in your project.



To initialize it in your app use:

static {  
 if (!OpenCVLoader.*initDebug*()) {  
 Log.*d*(*TAG*, "Internal OpenCV library not found. Using OpenCV Manager for initialization");  
 } else {  
 Log.*d*(*TAG*, "OpenCV library found inside package. Using it!");  
 }  
}

Add

<uses-permission android:name="android.permission.CAMERA"/>

To your manifest

Create a new Class named UnityTalk to talk to Unity scripts.



Create a new Class to be your Main Class (We called it OurUnityPlayer)

Example in next page!

\* make sure you override from UnityPlayerActivity

\* in your manifest make sure starting activity is yours:



\* OnCameraFrame is where you make your ImageProcessing



Hope You Made It!!

For any question please send us mail to:

[Levk3112@gmail.com](mailto:Levk3112@gmail.com)

Or

[Noyhess@gmail.com](mailto:Noyhess@gmail.com)

With the headline "UnityOpenCV Question"

And we will be glad to help.