פרוייקט במסגרת הטכניון – מעבדת GIP

נושא הפרוייקט : זיהוי כדור באמצעות פלאפון ושימוש במידע כ"עכבר" בעולם תלת מימדי בUNITY.

תוכנה : android studio , Unity

התהליך:

ירושה והרחבה של UnityPlayerActivity -

* לוקחים את קובץ classes.jar מC:\Program Files\Unity\Editor\
* Data\PlaybackEngines\AndroidPlayer\Variations\mono\Release\Classes
* פותחים בפרוייקט תיקיה חדשה libs ומעתיקים אותו לשם
* נכנסים ל File -> Project Structure
* App ( או הפרוייקט שאתה עובד בו..) <- dependencies
* לוחצים על ה + ובוחרים file dependency
* ובוחרים את הקובץ שהעתקנו לשם....
* כעת ניתן להרחיב את UnityPlayerActivity

הכנסת OpenCV לפרוייקט שלך בandroid studio

* שימוש במדריכים של [http://blog.codeonion.com/tutorials/opencv-for-android-tutorials/](http://l.facebook.com/l.php?u=http%3A%2F%2Fblog.codeonion.com%2Ftutorials%2Fopencv-for-android-tutorials%2F&h=SAQFv1w5Y)

תקשורת בין אנדרואיד לUnity

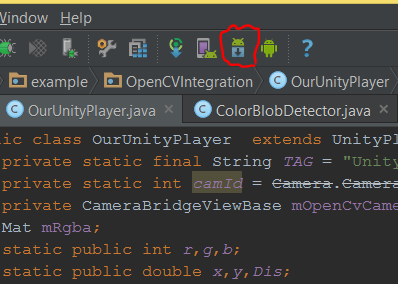
כדי לפתח בVR אלייך להקבל קובץ SSIG שנותן אישור לפלאפון לפתוח את האפליקציה שלך.

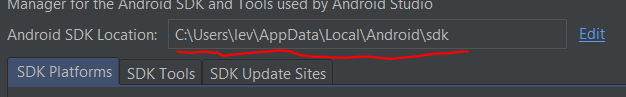
כדי לקבל קובץ זה נכנס ל:

<https://dashboard.oculus.com/tools/osig-generator/>

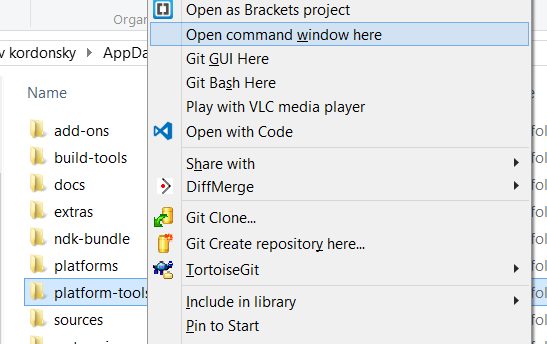
האתר מבקש Device ID כדי להשיג אותו נצטרך לבצע מספר דברים..

נכנס לandroid studio -> SDK manager



נכנס לתיקיה הבאה : 

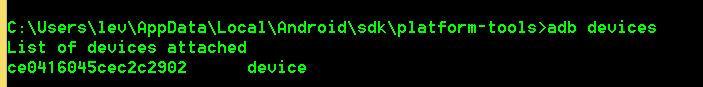
בתיקיה נלחץ shift+rightMouse על platform-tools ונפתח CMD בתיקיה הזאת



כעת יש גישה בCMD ל adb שזה מנהל הdevices של אנדרואיד.

נכתוב adb devices

ונקבל



זה הdevice ID שאנחנו צריכים בשביל האתר .

בעזרתו נקבל קובץ SSIG שאותו נכניס לתיקית unity ( הסבר אחכ)

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>

Open new Unity Project .

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

Add script to Sphere



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

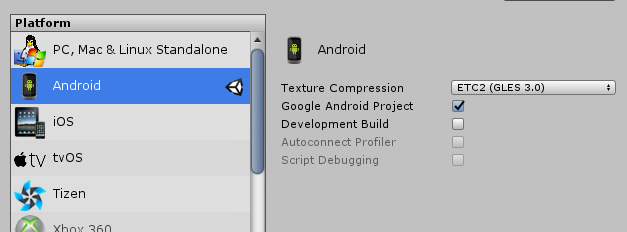
\*ajc will be the java class that we will build later , this class has static methods getR,getG,getB that return int.

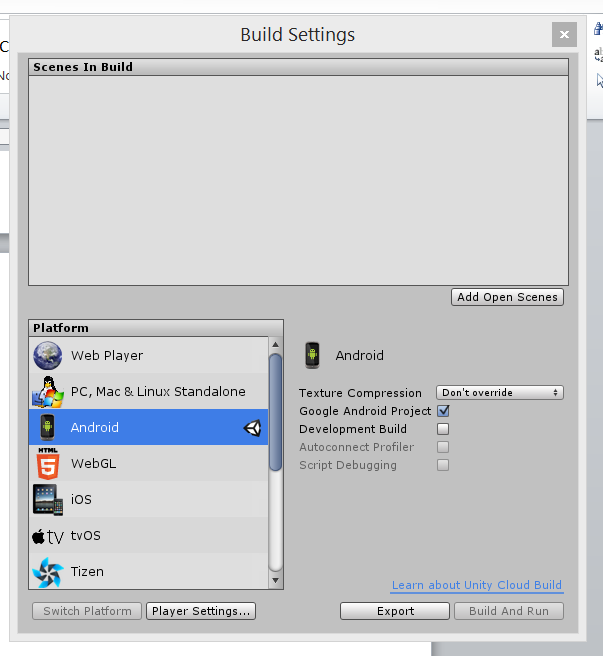
Go to File -> Build Settings

Choose Android and check Google Android Project

( u 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 Studio :

File -> New -> Import Project -> choose the folder you exported to

(as for this unity version, you will have two folders – OVRPlugin and #yourname – choose #youname folder.

Notice that android studio will ask you to choose a folder when you use import project – choose a new folder name so it will ask you if you want to create one...

From now on we will only use this new folder.

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

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

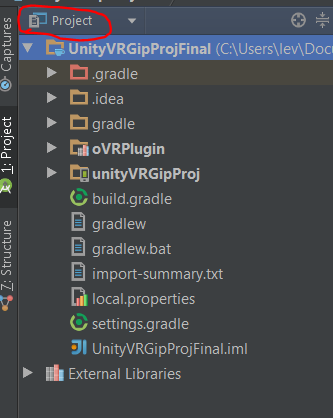
To #yourNewFolder \oVRPlugin\src\main\res\

Android Studio should now be open to your project. (in case its not open yet.. )

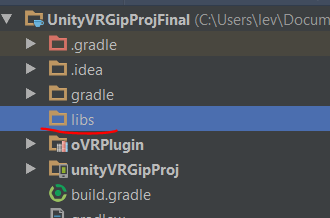
Build the project just to make sure everything is ok

**OpenCV static initialization:**

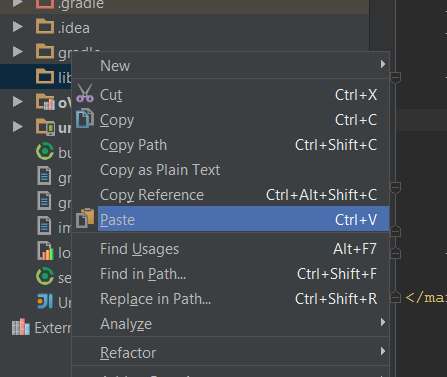
Go to Project view



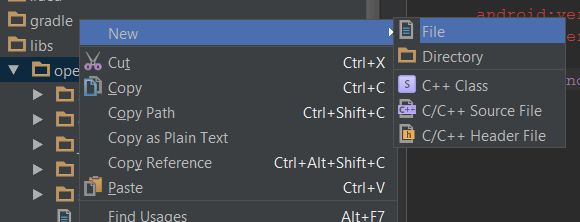
Add new folder "libs" (right mouse on main folder -> new -> directory)



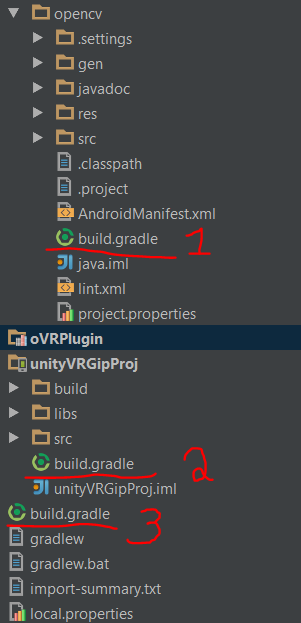
Copy the folder OpenCVFolder\sdk\java into your new "libs"

 call the new folder "opencv"

Add new file in "opencv" new folder



Call it "build.gradle"

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

1 is now empty

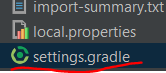
Copy this to it:

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

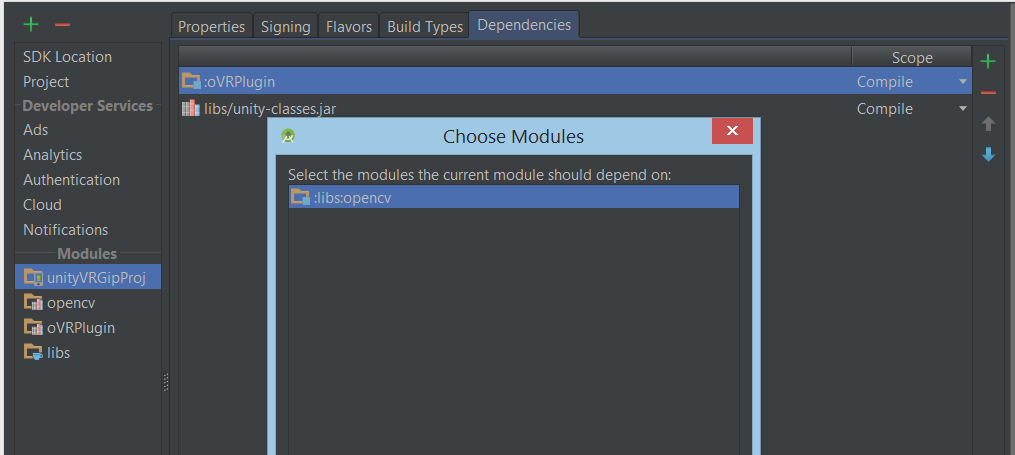
Select #yourAppName

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

