# Android development with Unity3D and OpenCV

Sascha Wernegger

August 5, 2016

## 1 Setup

To ease things a little up for the OpenCV and Android Setup install NVidia Codeworks. Be sure to Check all options for during the installation to be sure. It should install a folder called NVPACK where you can find the Android SDK and NDK and the OpenCV Library for Android with working samples. Codeworks comes with an Eclipse version where you simply can import the sample projects and modify them.

### 1.1 ADB Logcat

The Logcat tool is able to output the Log of the Android device. Its located in the Android-SDK folder under Platform tools but should be available system wide after SDK installation. Open console here. To output the log enter

```
$ adb logcat
```

and run the application. You Can use Filters to output only certain types of logs.The priority is one of the following character values, ordered from lowest to highest priority:

V - Verbose (lowest priority)

D - Debug

I - Info (default priority)

W - Warning

E - Error

F - Fatal

S - Silent (highest priority, on which nothing is ever printed)

Like this you can remove all with lesser priority than E.

```
$ adb logcat *:E
```

You can also output the log to a file.

```
$ adb logcat *:E > myfile.txt
```

To debug for logs from your application run the following.

```
$ adb -d logcat <your package name>:<log level>
```

-d denotes an actual device and -e denotes an emulator. If there's more than 1 emulator running you can use

```
-s emulator-<emulator number>
```

## 1.2 Unity first Android Test

Download Unity3D installer and run it. Follow instructions and be aware to add Android features. After installation create a new project and open it. Now you come up with a default scene. Save the scene and give it a name. Under *Edit* → *Preferences*, select *External Tools* and select the root folders of the JDK, Android SDK and NDK (install them if necessary, and to be sure select all options for SDK).Click on *File* → *Build Settings* and select Android as target and press switch to platform. Select Player Settings and open the *Other Settings* in the Inspector. Enter a *BundleIdentifier*. Now You can hit *Build and Run* under *File*

## 1.3 Unity Call Java from C-Sharp

The first working sample i produced followingthis Guide.

TODO:

- Also call android library from C-Sharp
- Call android library with OpenCV from C-Sharp

## 1.4 OpenCV

The use of OpenCV is slightly different from the other languages. The most important packages are:

- Core
- Imgproc
- Feature2d
- Video
- Calib3d

The packages are also differeing between the 2.4 version then from newer where more things from the core where added to the Imgproc package.

## 1.5 Android Studio Shortcuts

- **CTRL + ALT +L** Format Code
- **CTRL + I** Add unimplemented methods
- **CTRL + O** Override methods.

- **CTRL + ALT + L** Format code.
- **ALT + 1** Show project.
- **ALT + 6** logcat.
- **SHIFT + ESC** Hide project - logcat.
- **CTRL + F9** Build.
- **CTRL + F10** Build and run.
- **CTRL + F9** Build.
- **CTRL + R** Find and replace.
- **CTRL + F6** Refactor class or method
- **CTRL + SHIFT + NumPad + -** Collapse all.
- **CTRL + SHIFT + NumPad + +** Expand all.

You can use Eclipse Short-cut key in Android Studio too.
File → Settings → Keymap → Choose Eclipse from Keymaps dropdown¿