

# Software Atelier II

“Wednesday Lectures”  
19/02/2014

**Davide Eynard**

Institute of Computational Sciences - Faculty of Informatics  
Università della Svizzera Italiana  
*davide.eynard@usi.ch*

# About me



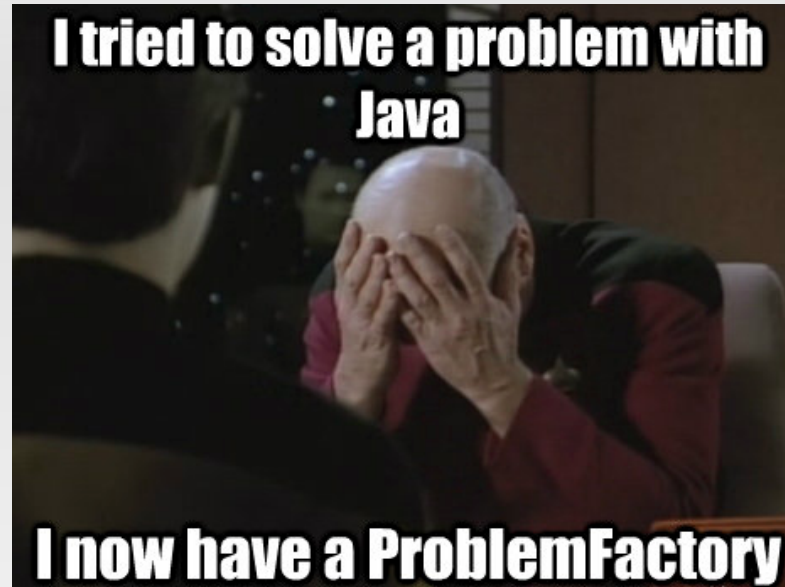
Davide Eynard

- [davide.eynard@usi.ch](mailto:davide.eynard@usi.ch)
- <http://davide.eynard.it>

# About Wednesday Lectures

- Learn how to build GUI
- Hands-on (computers are *mandatory*)
- Java + Android Developer Tools (ADT)
  
- SA II ↔ PF II
  - both in Java
  - communication flows bidirectionally
  - aiming for symbiosis

# Raise hand if...



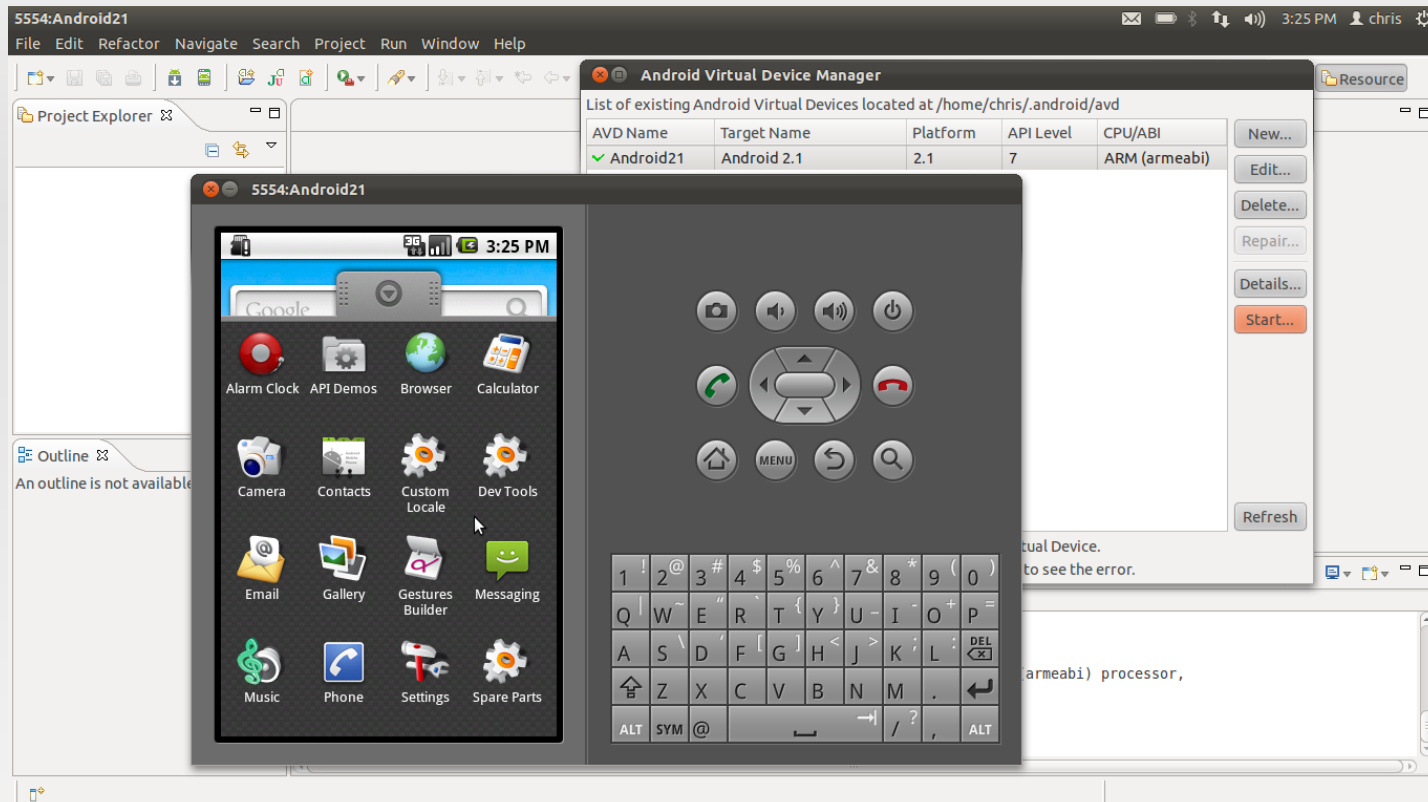
... you have programmed in Java before?

# Raise hand if...



... you have created a GUI before?

# Raise hand if...



... you have ever used Eclipse/ADT before?

# Why ADT?

- Last year: desktop GUI with NetBeans
  - more “classical” UI
  - less overhead from other components
  - advanced GUI builder
- This year: mobile GUI with ADT
  - more incentives (what is the most useful desktop app you can make with just one message box?)
  - directly run on your (or an emulated) mobile device
  - get introduced to a set of best practices for mobile UI design

# Why ADT?

... so why not a platform-independent framework?

- remember we want it to be java-based...
  - ... and we do not want to overcomplicate things!
- 
- Solution:
    - practice with ADT
    - extend to other platforms



# Raise hand if...



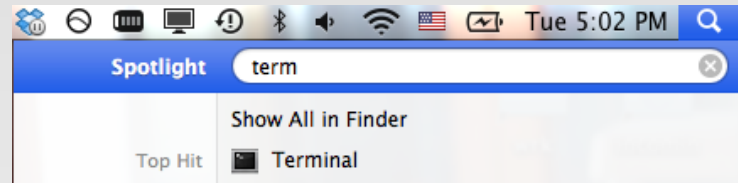
... you object to using ADT

# Plan for today

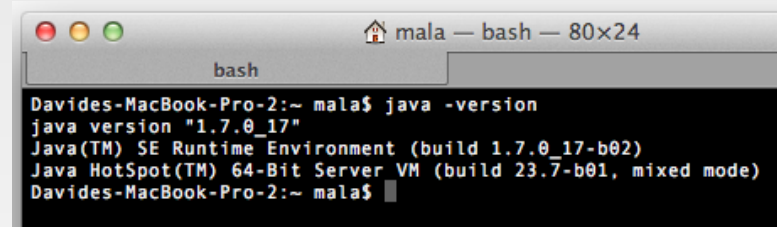
- Install and play with ADT
- Build our first Android app
- Run it!

# Step 1 – Check your java installation

- Open terminal



- Type  
`java -version`

A screenshot of a Mac Terminal window. The window title is 'mala -- bash -- 80x24'. The terminal shows the command 'java -version' being executed, with the following output:

```
Davides-MacBook-Pro-2:~ mala$ java -version
java version "1.7.0_17"
Java(TM) SE Runtime Environment (build 1.7.0_17-b02)
Java HotSpot(TM) 64-Bit Server VM (build 23.7-b01, mixed mode)
Davides-MacBook-Pro-2:~ mala$
```

# Step 2 – Install ADT

## ■ System requirements:

### Operating Systems

- Windows XP (32-bit), Vista (32- or 64-bit), or Windows 7 (32- or 64-bit)
- Mac OS X 10.5.8 or later (x86 only)
- Linux (tested on Ubuntu Linux, Lucid Lynx)
  - GNU C Library (glibc) 2.7 or later is required.
  - On Ubuntu Linux, version 8.04 or later is required.
  - 64-bit distributions must be capable of running 32-bit applications.

### Eclipse IDE

- Eclipse 3.6.2 (Helios) or greater
  - Note:** Eclipse 3.5 (Galileo) is no longer supported with the latest version of ADT.
- Eclipse JDT plugin (included in most Eclipse IDE packages)
- JDK 6 (JRE alone is not sufficient)
- Android Development Tools plugin (recommended)
- **Not** compatible with GNU Compiler for Java (gcj)

## ■ Download the SDK:

- go to <http://developer.android.com/sdk/>
- get the “ADT bundle”

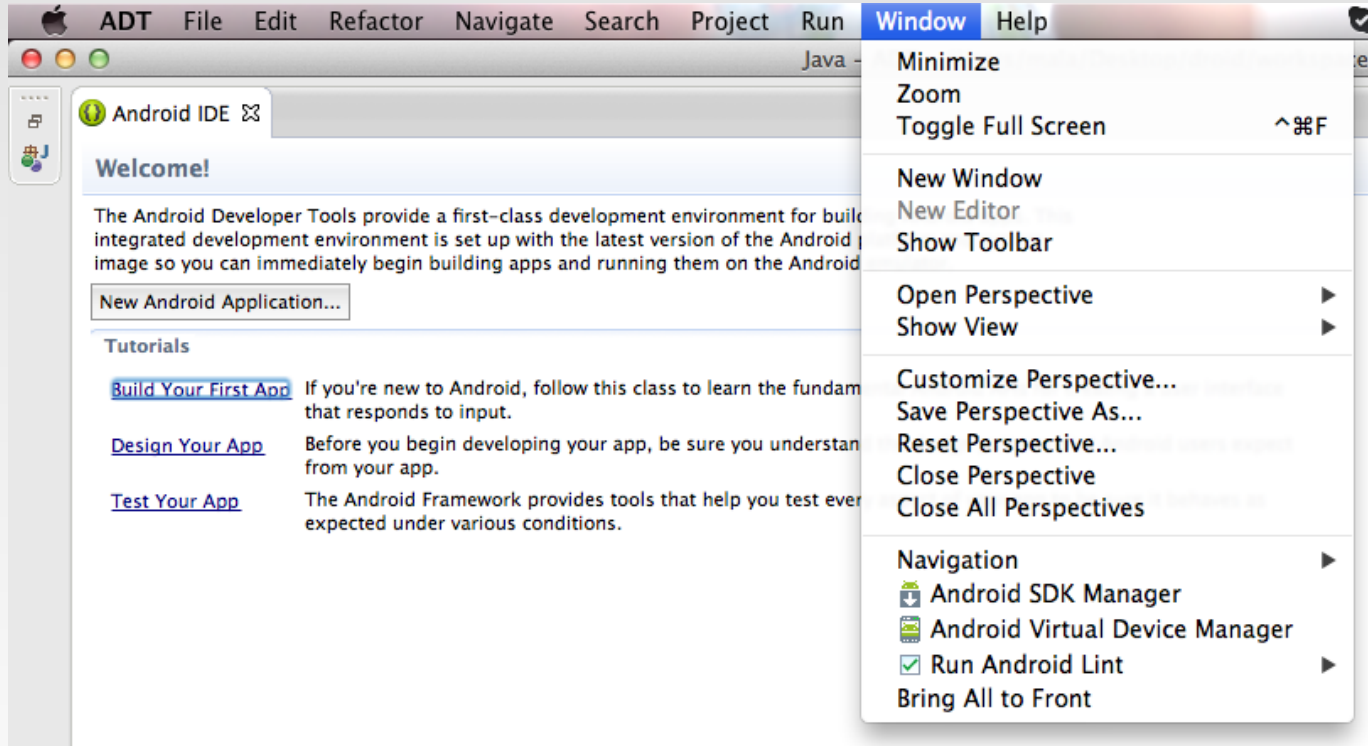
## ■ Unzip... et voilà! :-)

# Step 2 – ADT vs Android Studio

- Android Developer Tools
  - more mature and (hopefully) stable
  - based on Eclipse
  - ready-to-use bundle (including Android SDK Tools)
  - last version dated October 2013
- Android Studio
  - available as an *early access preview*
  - based on IntelliJ IDEA
  - not bundled with Android SDK (no, wait, now it is!)
  - last version dated January 2014

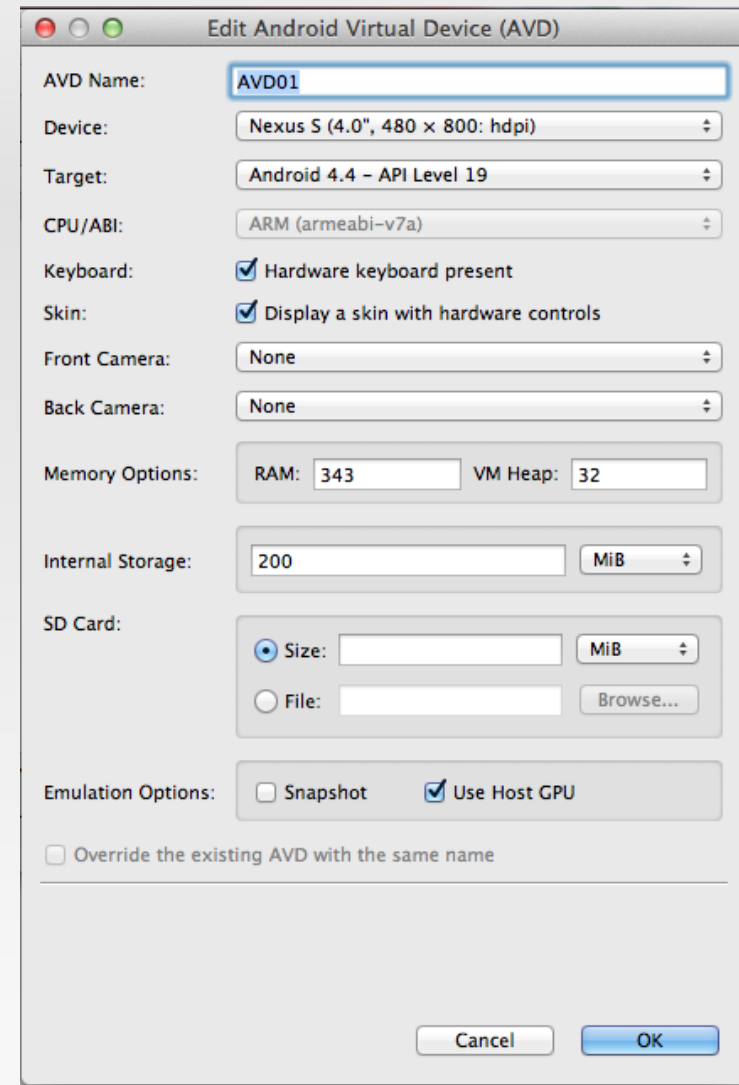
# Step 3 – Manage Virtual Devices

- Before creating your first project, select `Window > Android Virtual Device Manager`

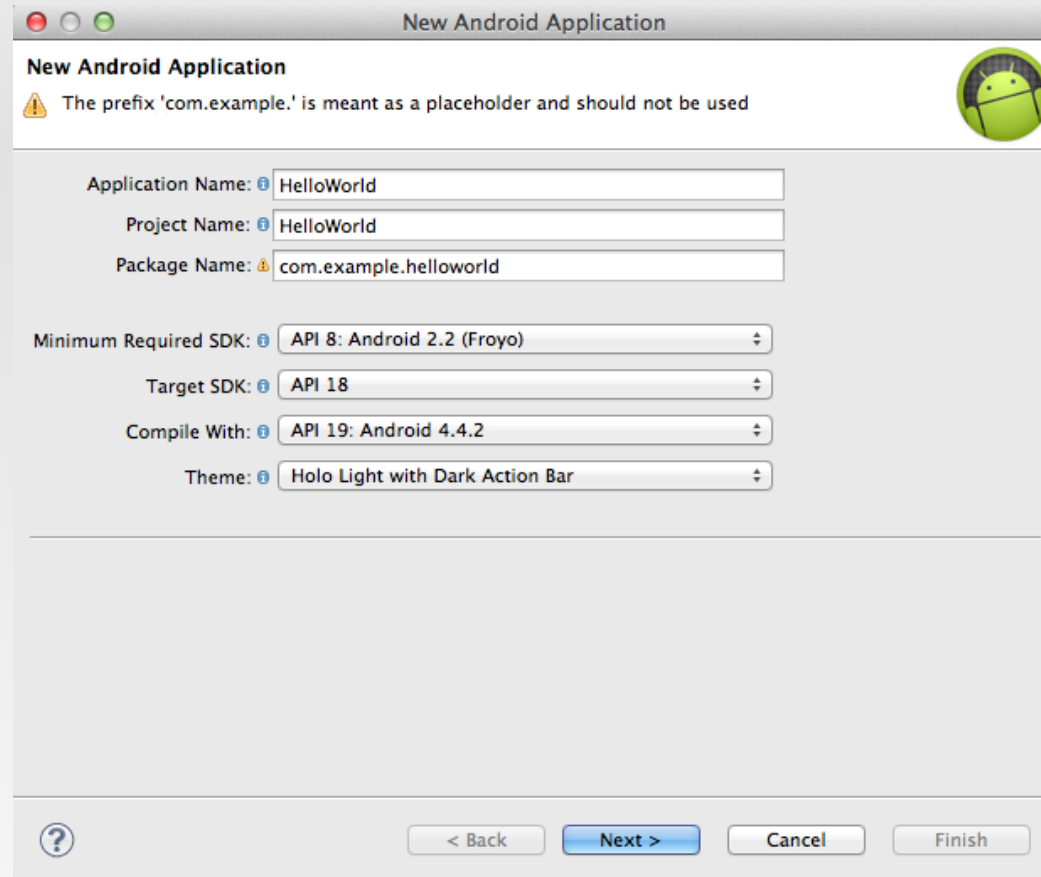


# Step 3 – Manage Virtual Devices



- Create a new device
- Check “device definitions” if you want to specify custom characteristics



# Step 4 – Create a new project




**New Android Application**

 The prefix 'com.example.' is meant as a placeholder and should not be used 

Application Name:

Project Name:


Package Name: 

Minimum Required SDK:

Target SDK:

Compile With:

Theme:





# Step 4 – Create a new project

The image shows the 'New Android Application' dialog box in an IDE, titled 'Configure Project'. It includes the following options:

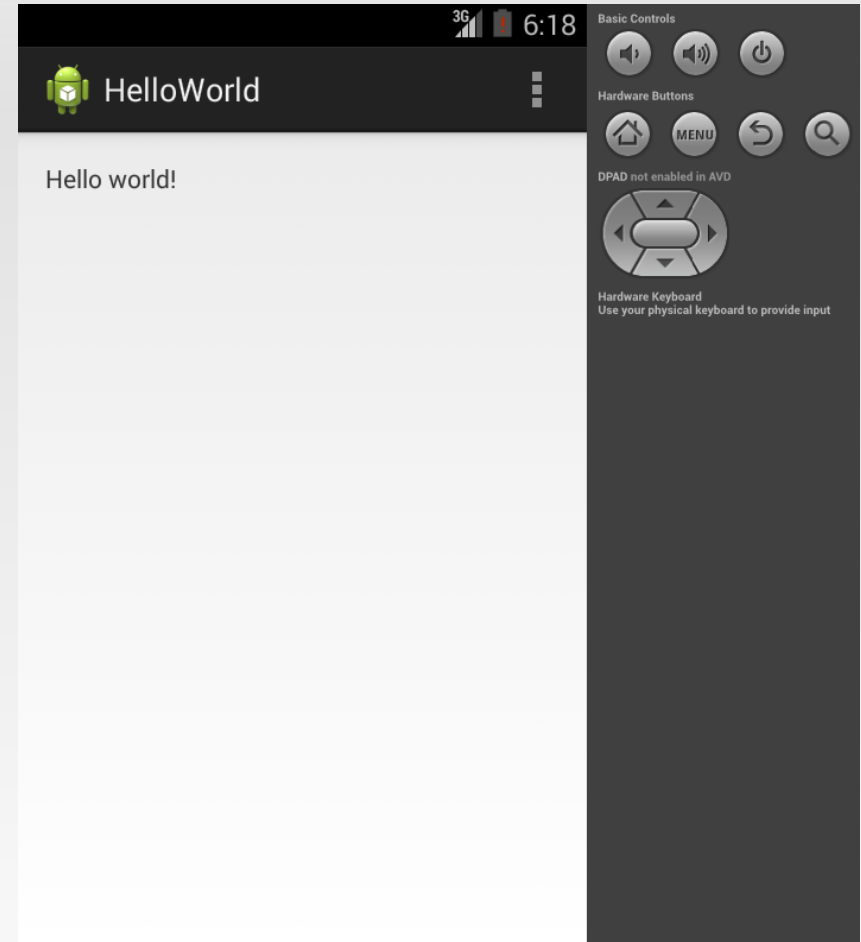
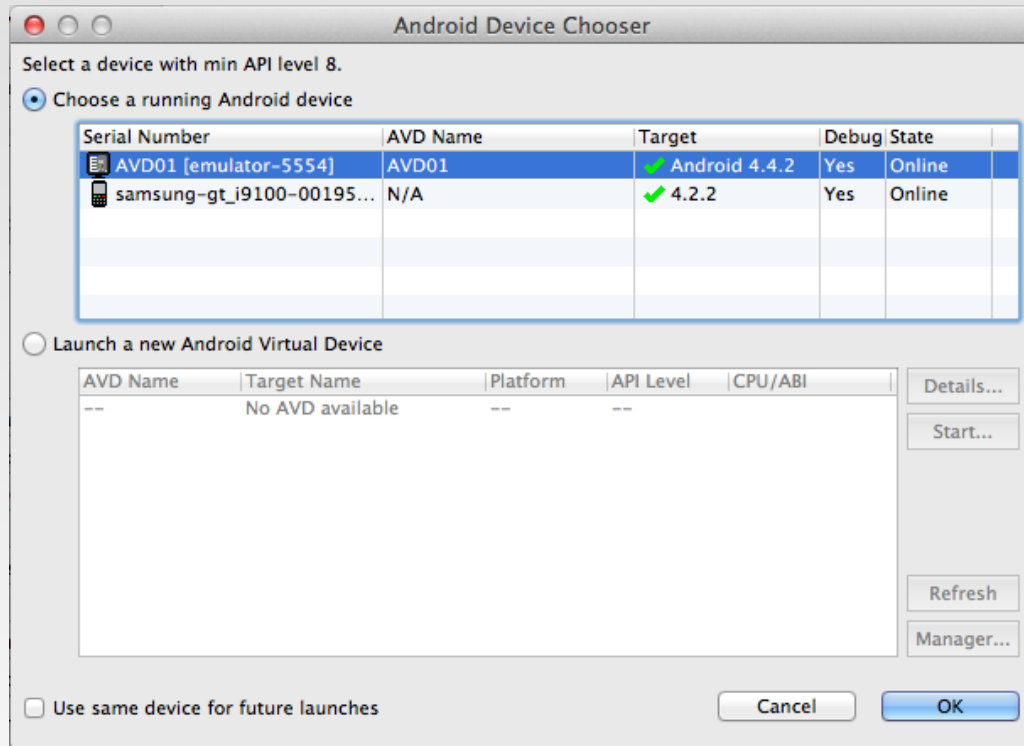
- Create custom launcher icon
- Create activity
- Mark this project as a library
- Create Project in Workspace

The Location field is set to `/Users/mala/Desktop/droid/workspace/HelloWorld` with a 'Browse...' button.

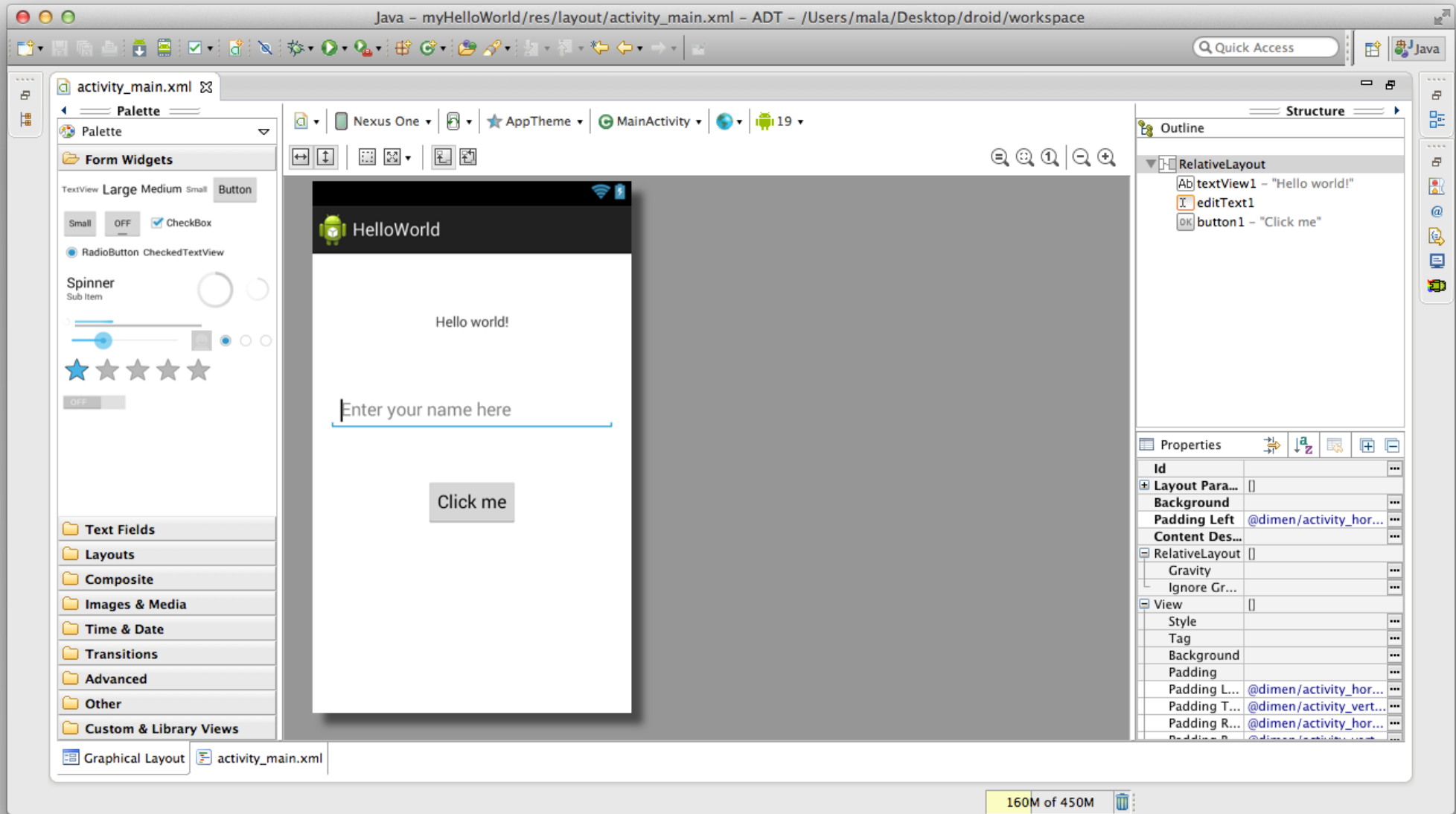
Two arrows point from the 'Create custom launcher icon' and 'Create activity' options to their respective preview windows:

- The 'Create custom launcher icon' preview shows a grid of Android icons for different screen densities: mdpi, hdpi, xhdpi, and xxhdpi. The 'Image File' field is set to 'launcher\_icon'.
- The 'Create Activity' preview shows a list of activity types: Blank Activity (selected), Fullscreen Activity, and Master/Detail Flow. Below the list, a description for 'Blank Activity' states: 'Creates a new blank activity, with an action bar and optional navigational elements such as tabs or horizontal swipe.'

# Step 5: run your app!



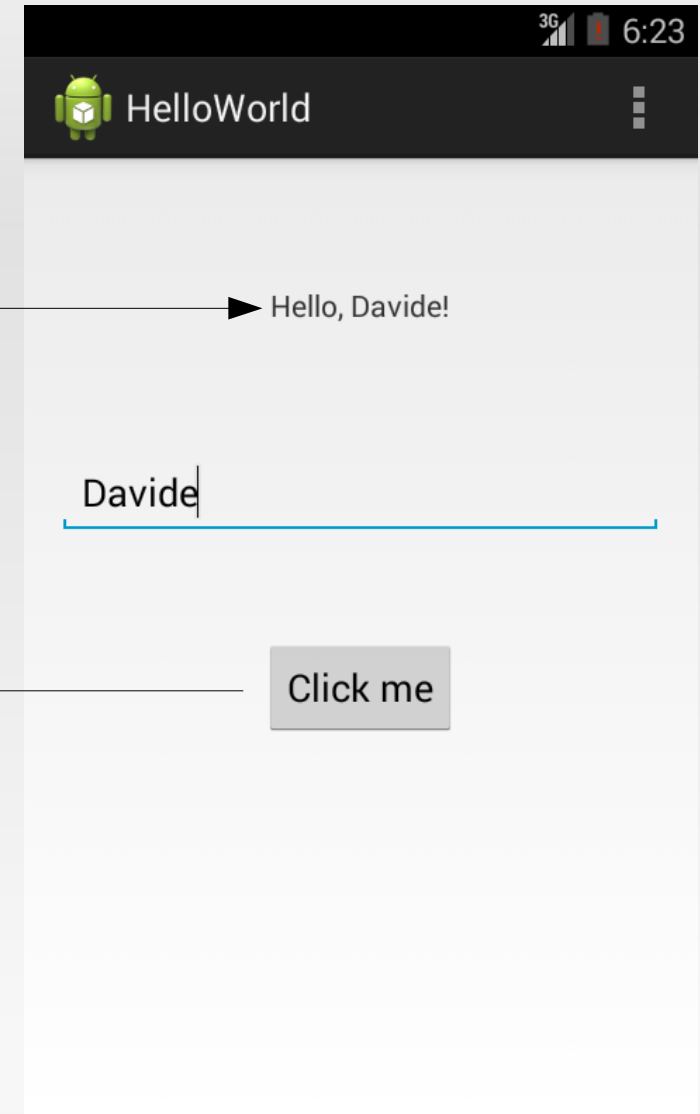
# Step 6: customize the UI



# Step 7: add interaction

```
/** Called when the user clicks the "Click me" button */  
public void sendMessage(View view) {  
    // Do something in response to button  
    TextView tv = (TextView) findViewById(R.id.textView1);  
    EditText et = (EditText) findViewById(R.id.editText1);  
  
    tv.setText("Hello, " + et.getText().toString() + "!");  
}
```

```
<Button  
    android:id="@+id/button1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_below="@+id/editText1"  
    android:layout_centerHorizontal="true"  
    android:layout_marginTop="48dp"  
    android:onClick="sendMessage"  
    android:text="@string/clickme" />
```



# Step 8: enjoy :-)

... and experiment!

## **A00 (first assignment)**

- invent an interesting way to process your input
- find a different way to show it (e.g. another widget or activity)
- **send me an email** (before Monday) if you have problems

# Useful links

- ADT installation:  
<http://developer.android.com/sdk/installing/bundle.html>
- Android app development workflow:  
<http://developer.android.com/tools/workflow/index.html>
- Step-by-step tutorials:  
<http://developer.android.com/training/index.html>

# Thank you!

Thanks for your attention!

Questions?