Software Atelier II

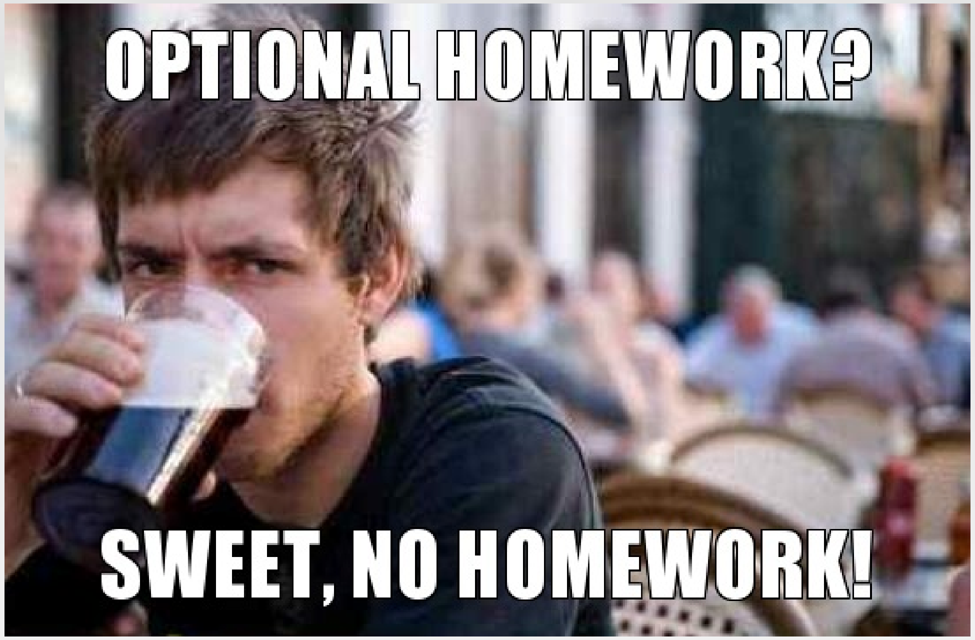
"Wednesday Lectures"
26/02/2014

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

Administrative

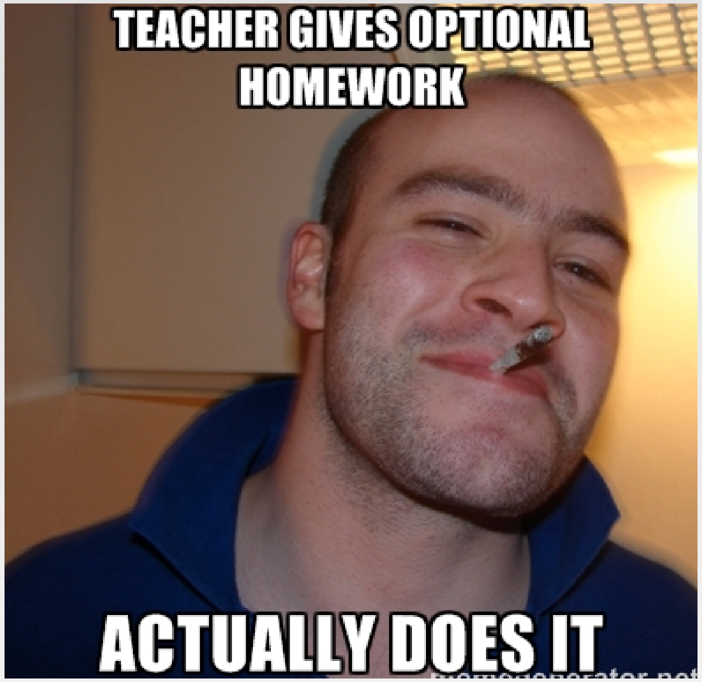
- We are using iCorsi (INFO.B198)
  - **go enroll!**
  - there is a forum for your questions
- There will be **homework**
  - **now**: standalone, individual
  - **later**: project lead-up, groups

Raise hand if...



… you did the optional homework?

For those who did it...



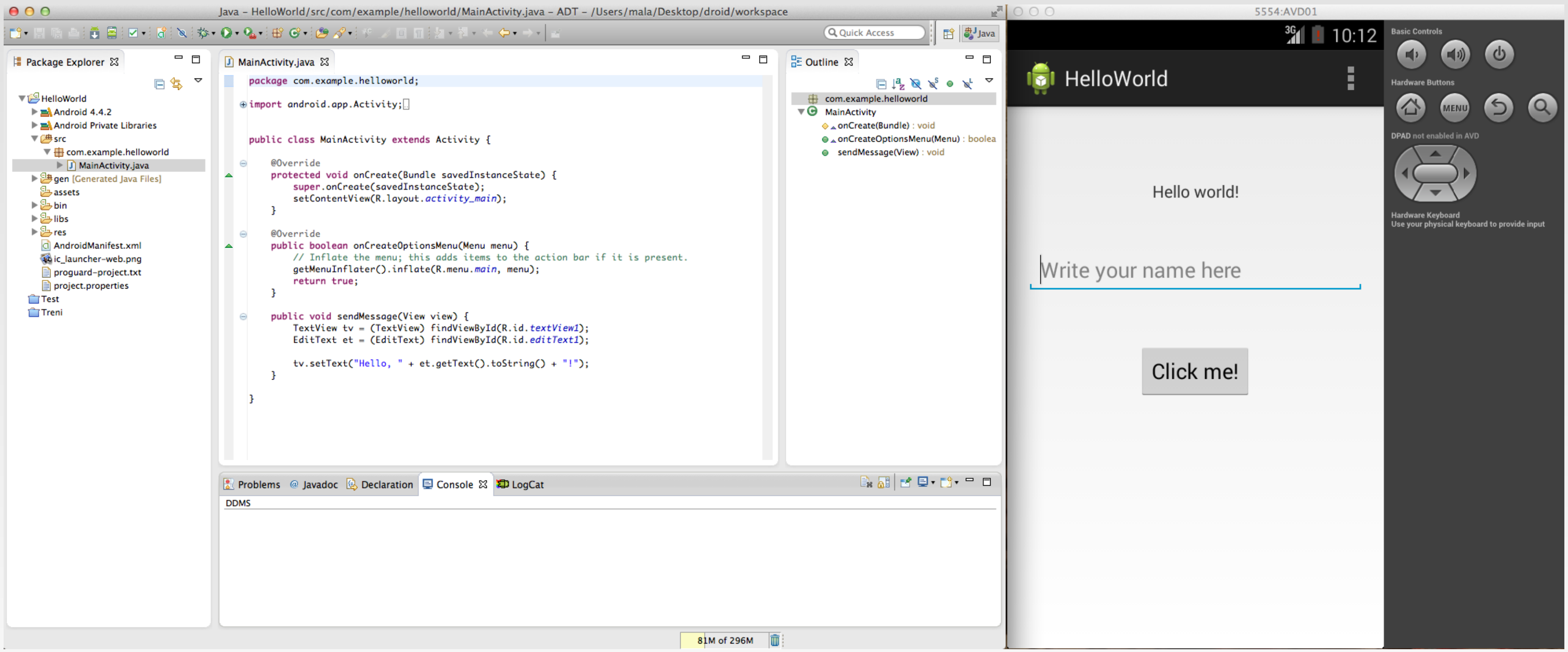
Well done! Let's talk about it :-)

Plan for today

- Recap
- Activities
- Layouts
- Views

Recap

- IDE installation + HelloWorld application



Java – HelloWorld/src/com/example/helloworld/MainActivity.java – ADT – /Users/mala/Desktop/droid/workspace

```java
package com.example.helloworld;

import android.app.Activity;


public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    public void sendMessage(View view) {
        TextView tv = (TextView) findViewById(R.id.textView1);
        EditText et = (EditText) findViewById(R.id.editText1);

        tv.setText("Hello, " + et.getText().toString() + "!");
    }

}
```

Recap

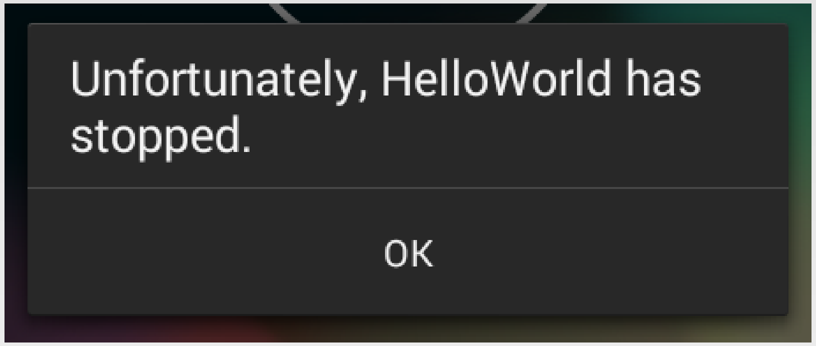
You created a new app, customized its *views*, added interaction to it, and ran it on the VM

- did you miss anything from this process? e.g.:
  - how can I solve that nasty bug I had?
  - why do I need such a bloated structure for a simple HelloWorld application?
  - where is the all the useful stuff saved?
  - how can I do <*add whatever idea you had here*>?

Recap

… How can I solve that nasty bug I had?

- I got some emails from you about the following error:


Unfortunately, HelloWorld has stopped.

  - related to the R.java resources file, it occurred when some changes were performed (e.g. SDK updated, layout file modified)
  - can be solved by executing Project > Clean before running the app
  - the LogCat utility was useful to diagnose the problem

- shall we use the forum / create a FAQ list from now on?

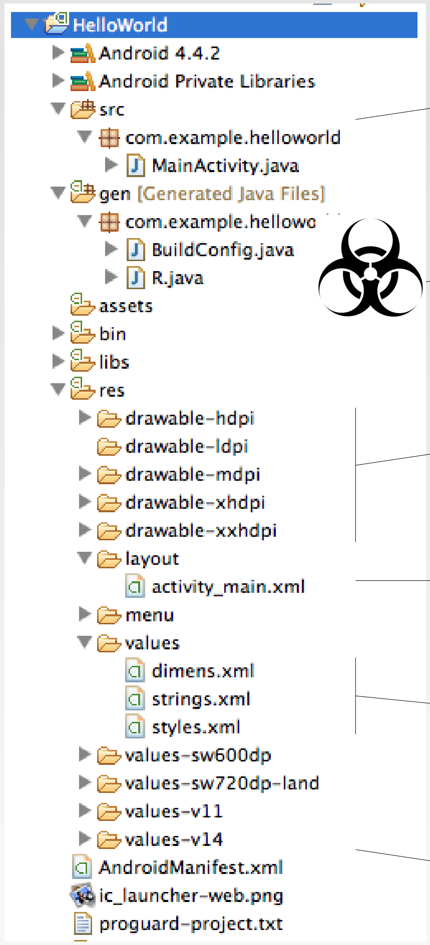
Recap

… Why do I need such a bloated structure for a simple HelloWorld application?

- first of all, the structure is the same for *every* application
    - what looks bloated now will just look *well organized* later
- then, the application itself is simple, but Android is not
    - different devices, different OS versions, different languages
- finally, you can ignore most of it
    - if you are not distributing your app, you can just use the defaults most of the times ;-)

Recap

... Where is all the useful stuff saved?



```
HelloWorld
  Android 4.4.2
  Android Private Libraries
  src
    com.example.helloworld
      MainActivity.java
  gen [Generated Java Files]
    com.example.hellowo
      BuildConfig.java
      R.java
  assets
  bin
  libs
  res
    drawable-hdpi
    drawable-ldpi
    drawable-mdpi
    drawable-xhdpi
    drawable-xxhdpi
    layout
      activity_main.xml
    menu
    values
      dimens.xml
      strings.xml
      styles.xml
    values-sw600dp
    values-sw720dp-land
    values-v11
    values-v14
  AndroidManifest.xml
  ic_launcher-web.png
  proguard-project.txt
```

- your source files
- the R class you often refer to (don't touch!)
- icons and images for different display densities
- your layouts
- values (constants) you are going to use in the app
- your application's Manifest file

Recap

… How can I do <*add whatever idea you had here*>?

- HAH! Good question :-)

  … limit of this class: mostly focused on UI

  … why don't you start from Android training and samples?

  … and if you still have problems, let's talk about it together

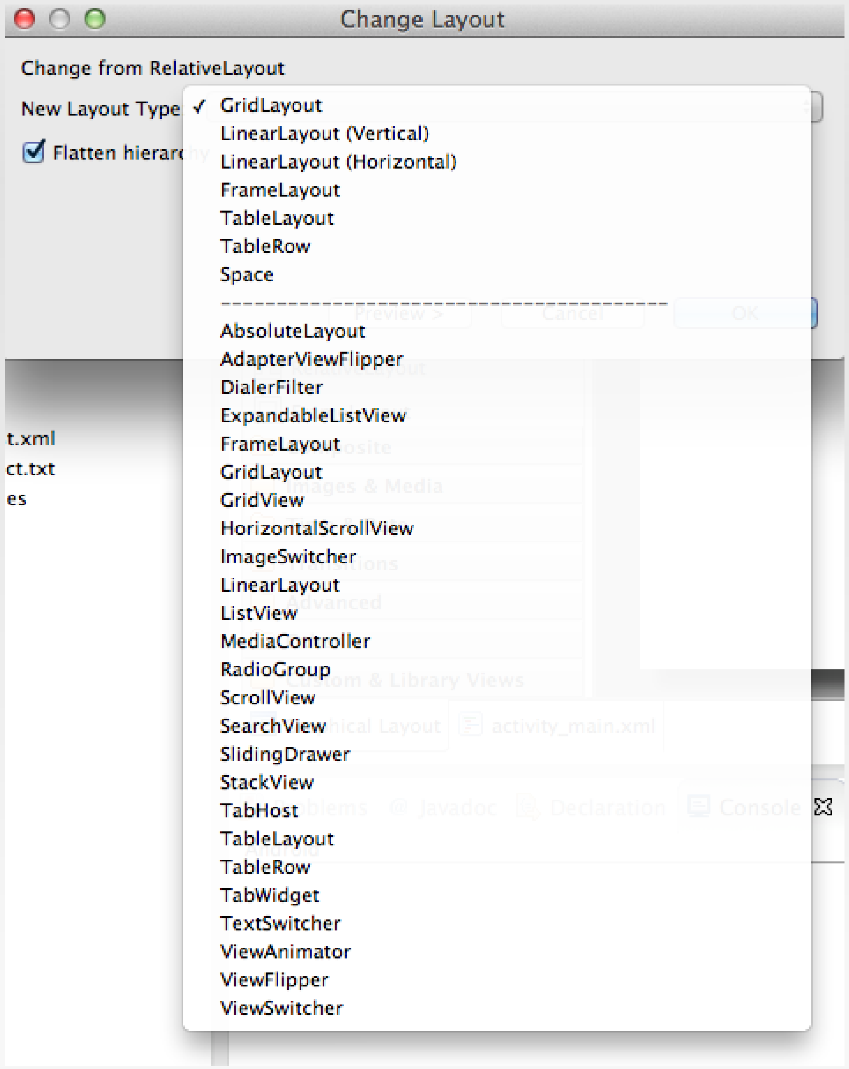
Activities

- similar to *controllers* in the MVC paradigm
- extended to *fragments* from Android 3
- typically match a visible screen
- composed by a *hierarchical collection of Views*
  - everything is a View, even Layouts!
  - views use strings, colors, styles, and graphic resources which are compiled and made available as resources
- need to *communicate* between activities => *Intents*

Layouts

- *XML* vs *programmatic*
  - in most of the cases you can choose between the two
  - some layouts need to be populated using an Adapter (see e.g. ListView and GridView)
- Layouts themselves are derived from the *View* class
- Many layout/view properties are common
  - width, height, padding, gravity, hint, text, ...
- Find more about the other layouts online

Layouts



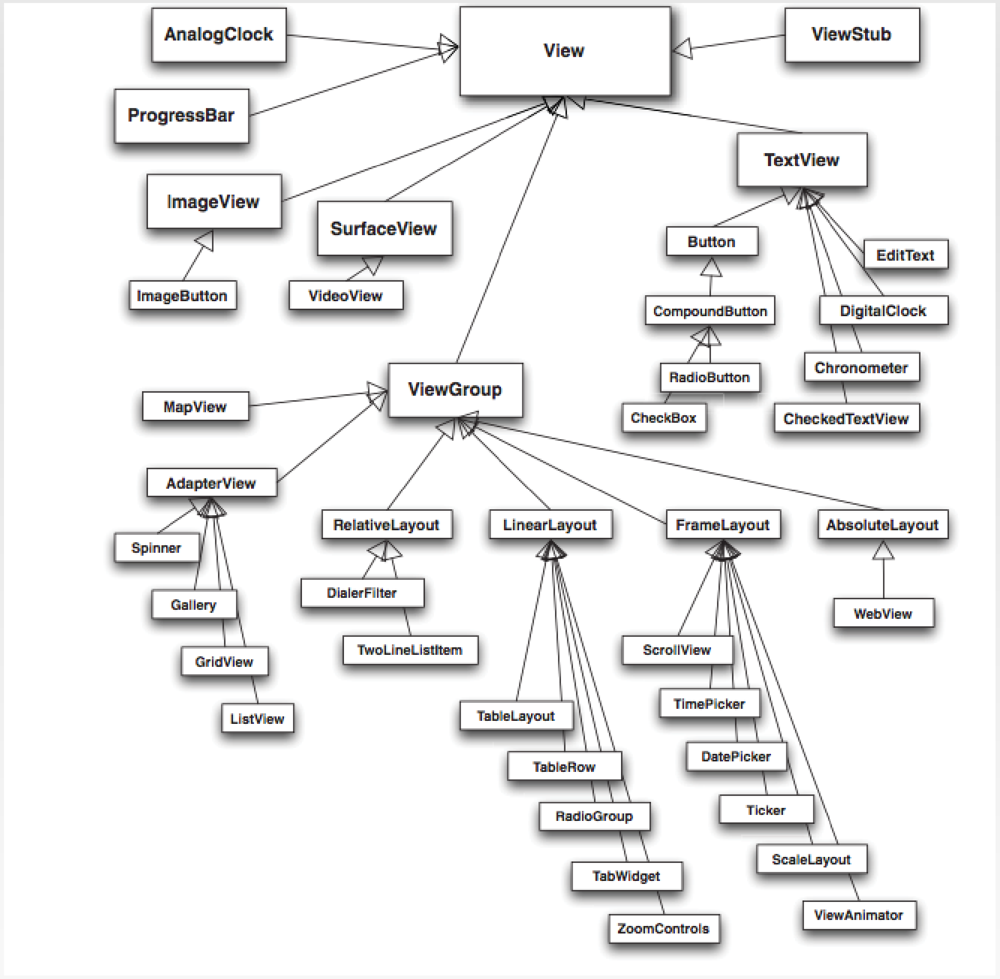
✓ GridLayout
LinearLayout (Vertical)
LinearLayout (Horizontal)
FrameLayout
TableLayout
TableRow
Space

AbsoluteLayout
AdapterViewFlipper
DialerFilter
ExpandableListView
FrameLayout
GridLayout
GridView
HorizontalScrollView
ImageSwitcher
LinearLayout
ListView
MediaController
RadioGroup
ScrollView
SearchView
SlidingDrawer
StackView
TabHost
TableLayout
TableRow
TabWidget
TextSwitcher
ViewAnimator
ViewFlipper
ViewSwitcher

Scary!

- See here for a detailed description of the main layouts
- Let us play with a demo (you can get it here)

Views



Scary!

- See here for a detailed description of each available View

Simple components – a comparison

Desktop

- Label
- Button
- Check box
- Radio button
- Text Field
- Text Area
- Password Field
- Spinner
- Slider
- Progress bar
- ...

Android (everything's a View)

- TextView
- Button / ToggleButton
- CheckBox / CheckedTextView
- RadioButton / RadioGroup
- EditText
  - textPersonName, textPassword, numberPassword, textEmailAddress
- Spinner
- SeekBar
- ProgressBar
- ...

Next steps

- Fragments
  - portion of UI in an activity
  - *Loaders* are used to monitor data sources and asynchronously load data
- Intents
  - a powerful mean for communication between views
  - “declaration of need”: they describe what you want to do
  - they are caught by intentFilters (declaration of capability)
- More advanced adapters

Useful tricks

- android.util.Log
  - see its output in the LogCat pane
- Window > Open Perspective > DDMS
  - a powerful interface with your Android VM
- Toasts
  - a quick'n'easy way to communicate directly on the device
- … name yours!

Assignment 01

- Follow this “Hangman Game Tutorial” (parts 1-2-3)
  - … do it step-by-step, you will learn much more!
- **Add your own interface**
  - play with layouts, change images, letters arrangement... let me see you understood how it works
- **Add your own data**
  - do something *completely new*: don't use an array, but get text from contacts, sms messages, a SQLite database...
- **Submit to iCorsi**
  - **deadline: Monday (03/03) night**

Useful links

- Android Developers - Design
  http://developer.android.com/design
- API Guides – App Components
  http://developer.android.com/guide/components/index.html
- Android Developers – Samples
  http://developer.android.com/samples/content.html
- Android in Action, 3rd Edition
  http://www.manning.com/ableson3/
-

Thank you!

Thanks for your attention!

Questions?