

# Android

*Looking beyond the obvious*

**10/20/11**

**Cocoaheads Siegen**

**Dipl. Inform. Simon Meurer**



# Agenda

- General
- Android „Desktop“
- Developer's View
- Publishing

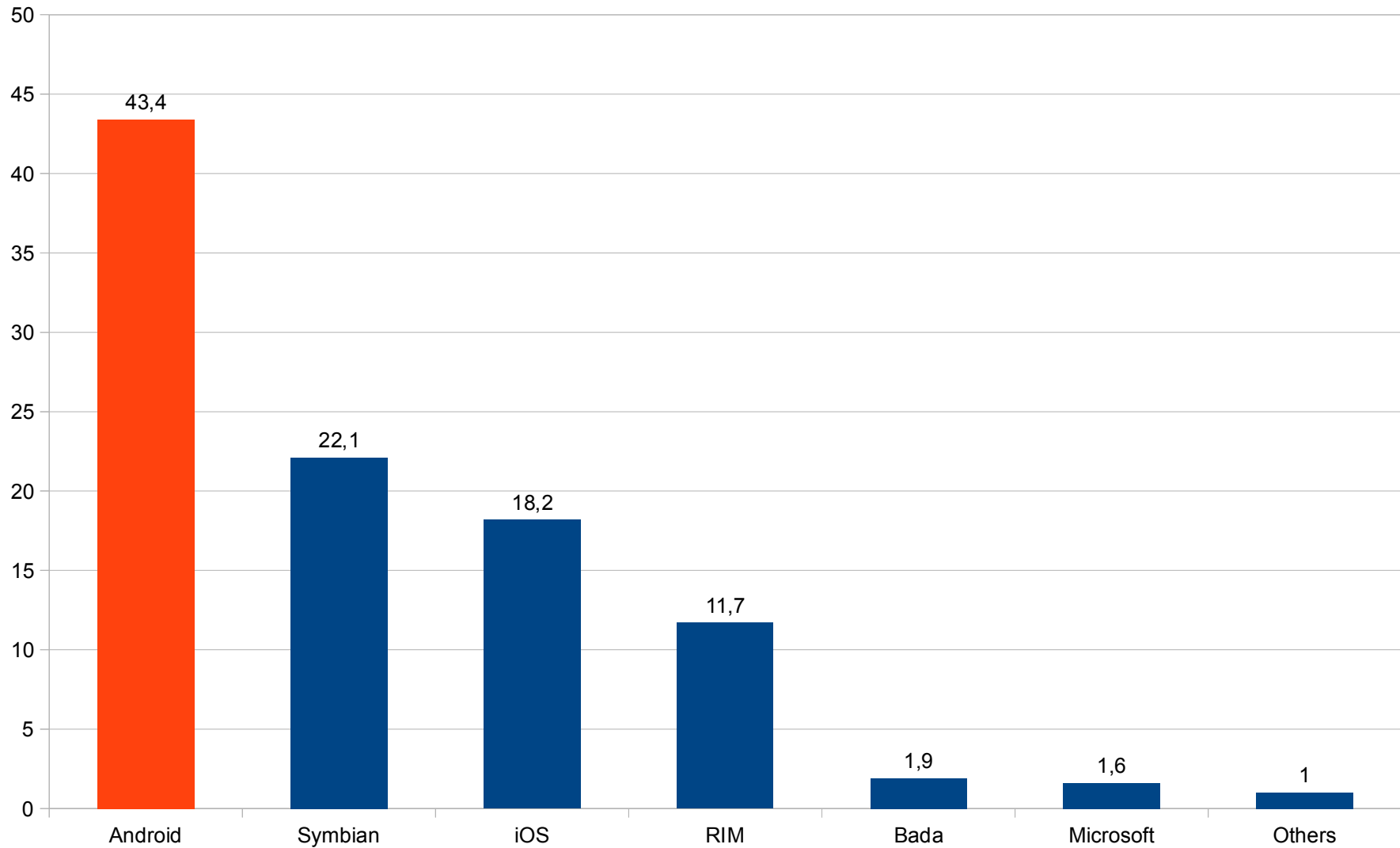


# What is Android?

- OS and Software-platform for *different* Mobile Devices
- Developed by Open Handset Alliance
- Based on Linux-Kernel 2.6
- Biggest rival of iOS on Smartphone OS-Market



# Market share



10/20/11

2nd quartal 2011 Market share (%) - source: Gartner



# Sample Devices



10/20/11



# Android Versions

- 2.x – for mobile phones (act. 2.3.6)
- 3.x – for tablets (act. 3.2)
- 4.x – gets 2.x and 3.x together

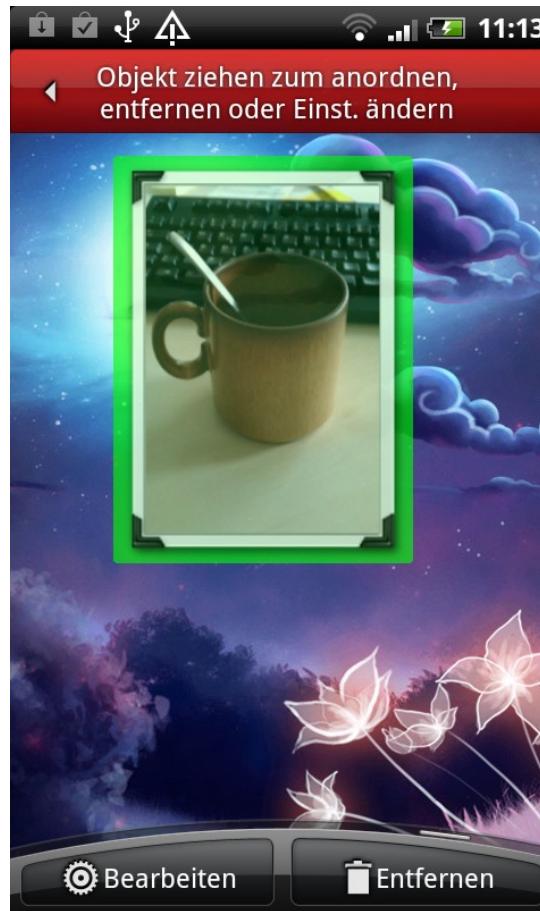
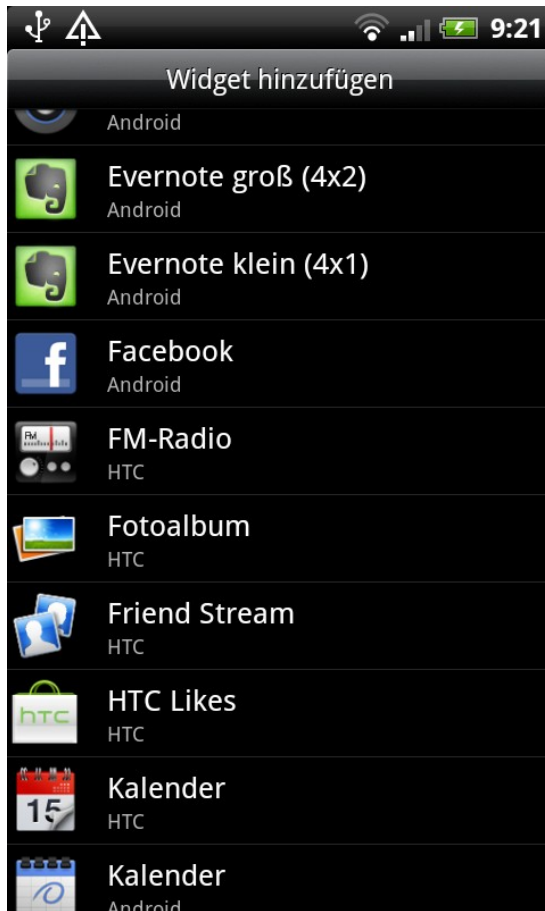


# Android „Desktop“

- Apps in Menu
  - Can be moved to „Desktop“
- Themes and Live-Background
- Shortcuts
- No. of Desktops set by Theme

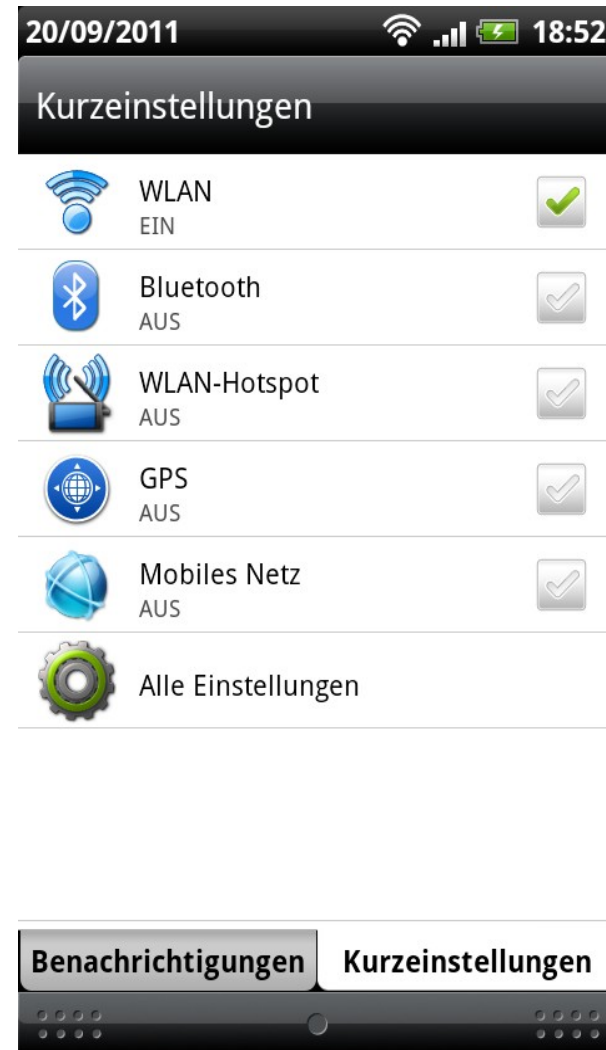
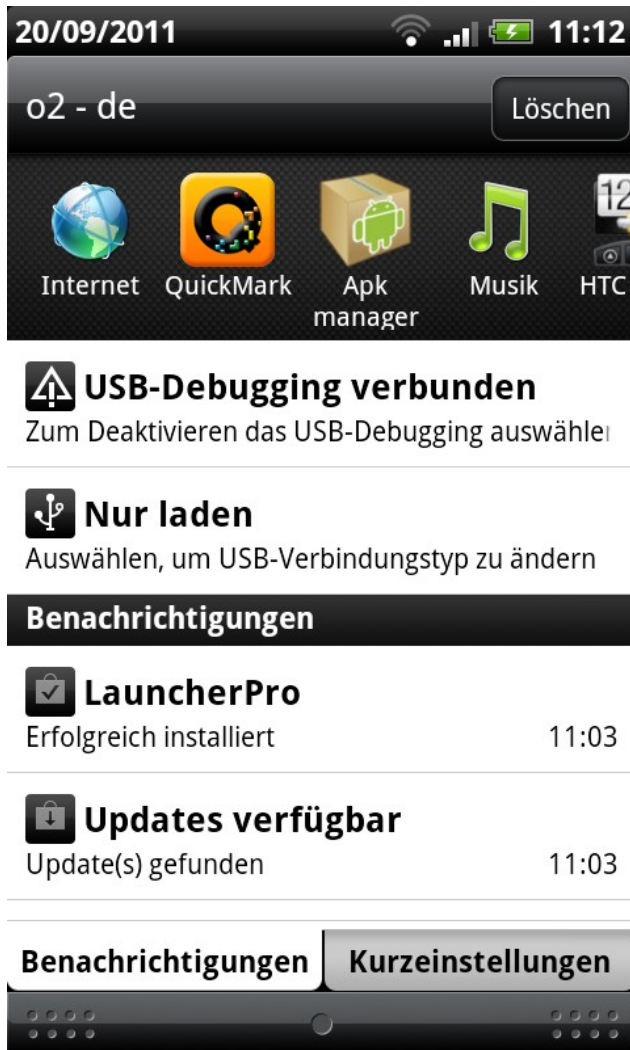


# Widgets





# Notifications and Preferences

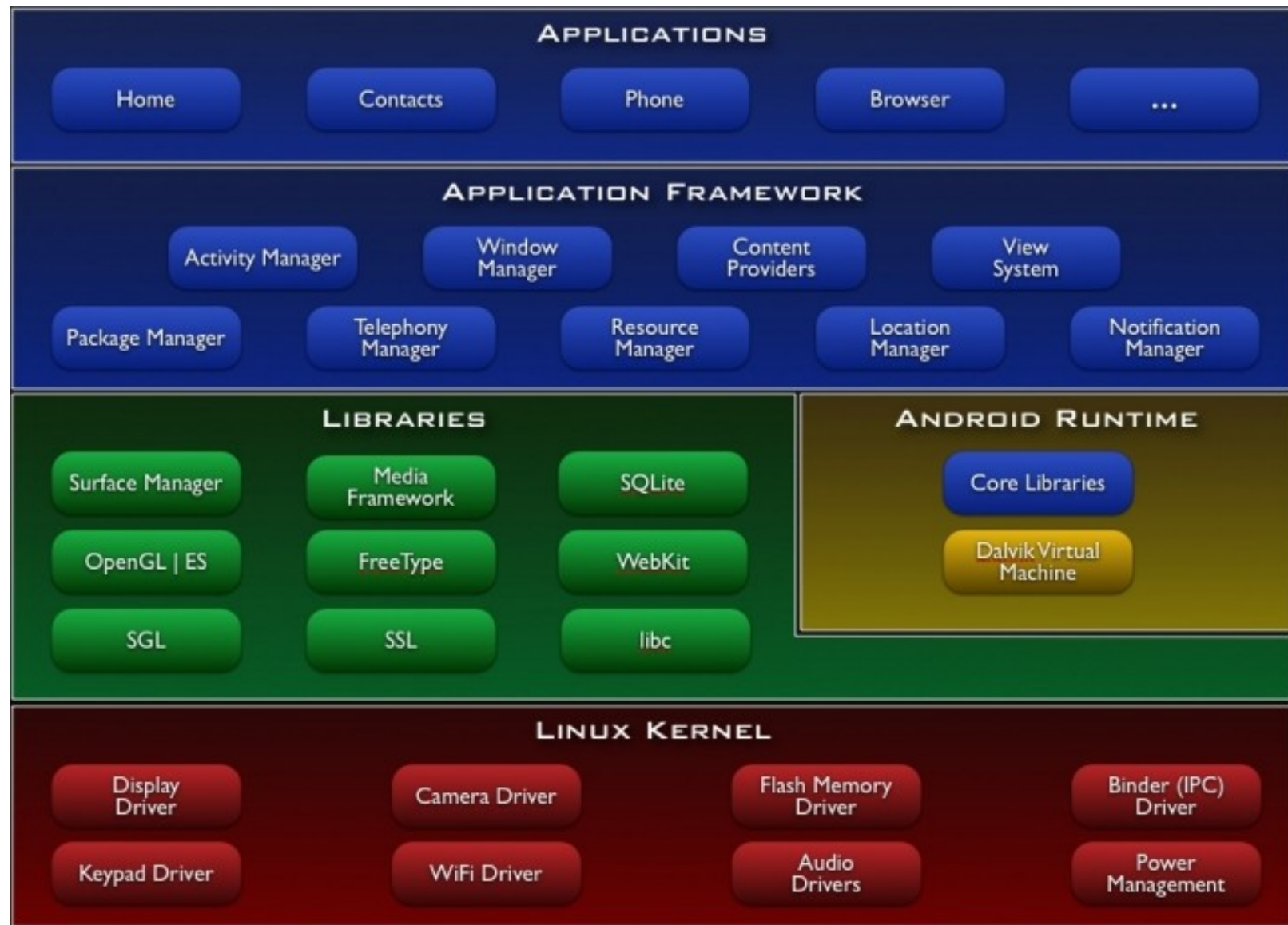


# Developer's View

- Architecture
- Language
- Key Concepts
- Menues
- Environment



# Architecture



# Language

- Android is Java, right?

Yes, but:

- No Constructors (for GUI-Classes)
- No Swing or SWT
- No `System.out.println(...)`
- Limited memory
- ...



# Key Concepts

- Activities
- Layouting with XML
- Intents
- Services
- Content Providers
- Resources



# Activities

- UI-Screen (better: logic)
- An app has usually more than one
- Activity Stack
  - New Activities pushed on stack
  - Back pops them of

# Example (HelloWorldActivity.java)

```
package com.test.helloworld;

import android.app.Activity;

import android.os.Bundle;

public class HelloWorldActivity extends Activity {

    /** Called when the activity is first created. */

    @Override

    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.main);

    }

}
```



# Layouting with XML

- GUI is build in XML
- Tags = Elements, Attributes = Properties
- Different Resolutions possible
  - No Absolute Layout, instead:
    - LinearLayout, RelativeLayout, TableLayout,  
...
- Place in XML = Place in Layout



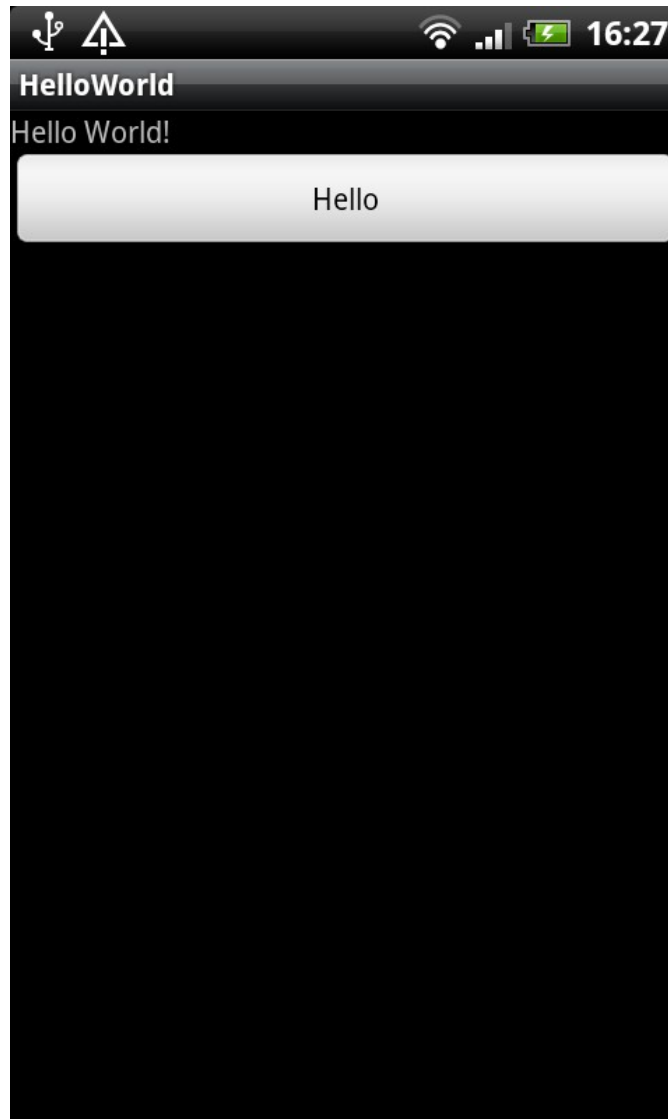
# Example (main.xml)

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:orientation="vertical" android:layout_width="fill_parent"  
    android:layout_height="fill_parent">  
    <TextView android:id="@+id/textview" android:layout_width="fill_parent"  
        android:layout_height="wrap_content" android:text="Hello World!" />  
    <Button android:id="@+id/button" android:layout_height="wrap_content"  
        android:layout_width="match_parent" android:text="Hello" />  
</LinearLayout>
```



# HelloWorld



# Clicking the button

```
public class HelloWorldActivity extends Activity {  
    public void onCreate(Bundle savedInstanceState) {  
        ...  
        Button b = (Button) findViewById(R.id.button);  
        b.setOnClickListener(new OnClickListener() {  
            @Override  
            public void onClick(View arg0) {  
                Toast.makeText(HelloWorldActivity.this, "Hello World!", Toast.LENGTH_SHORT).show();  
                Log.v("HelloWorldActivity", "Hello clicked");  
            }  
        });  
    }  
}
```



# Toasting and Logging

- Toast
  - displaying short Text for a short while
  - **NOT** for debugging
- Log
  - is for logging on console
  - Different levels:
    - d, e, i, v, w



# So, what about other Activities?

- Problem: No Constructors for GUI-Classes

→ Intent:

- Describes a spec. Action
- e.g. „pick a photo“, „take a picture“, „create person“
- Standards: „send an email“, ...
- Can be registered as standard
- Intents extras = Values of Constructors



# Example (Intents)

onClick of Hello-Button:

```
Intent i = new Intent(HelloWorldActivity.this, HelloWorldActivity2.class);
```

```
i.putExtra("Hello", "World");
```

```
startActivity(i);
```

HelloWorldActivity2:

```
public class HelloWorldActivity2 extends Activity{  
    protected void onCreate(Bundle savedInstanceState) {  
        ...  
        TextView t = (TextView) findViewById(R.id.textview2);  
        t.setText(getIntent().getStringExtra("Hello"));  
    }  
}
```



# How do I get results?

1. `startActivityForResult(Intent, requestCode)`
2. Generate result and put it in the intent
3. set resultCode and `finish()` the activity
4. in `onActivityResult(requestCode, resultCode, Intent data)` is your result



# Services

- Task that runs in background
- e.g. Music Player
  - Can be started by Activity1
  - Changed by Activity2
    - e.g. Activity2 says „stop“ or „nextTrack“
- „Live Wallpaper“

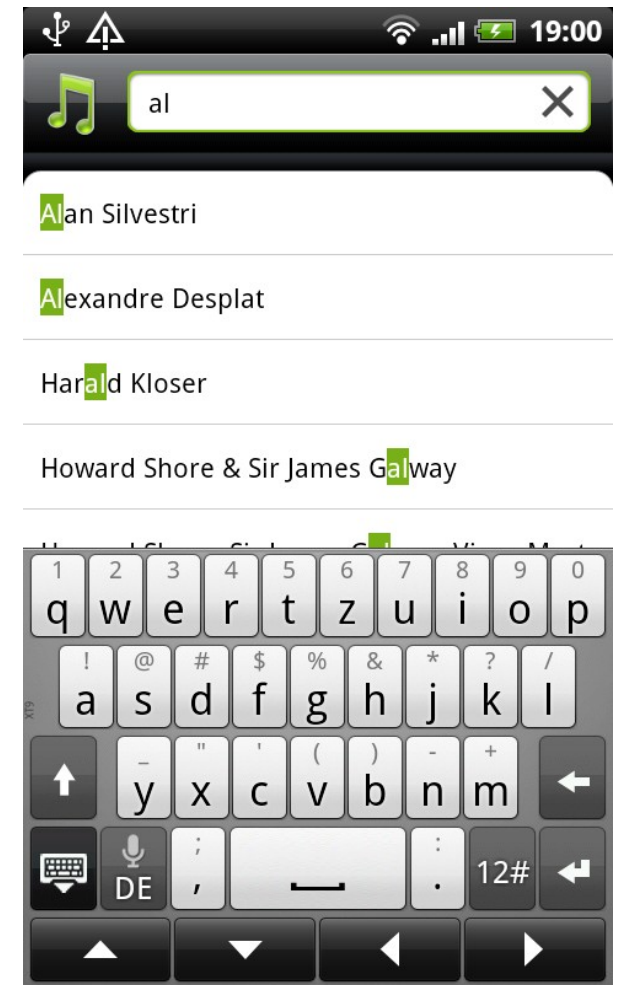
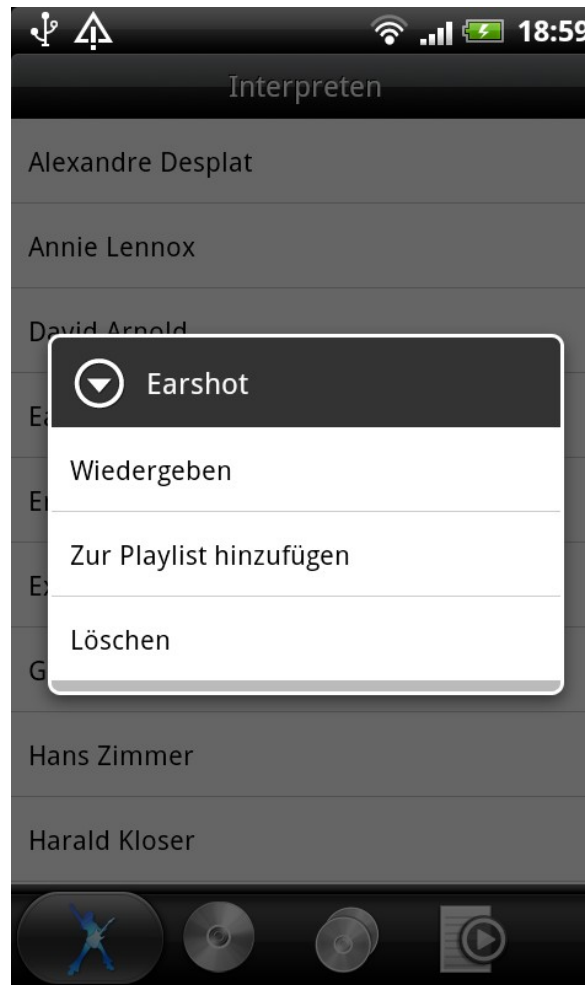
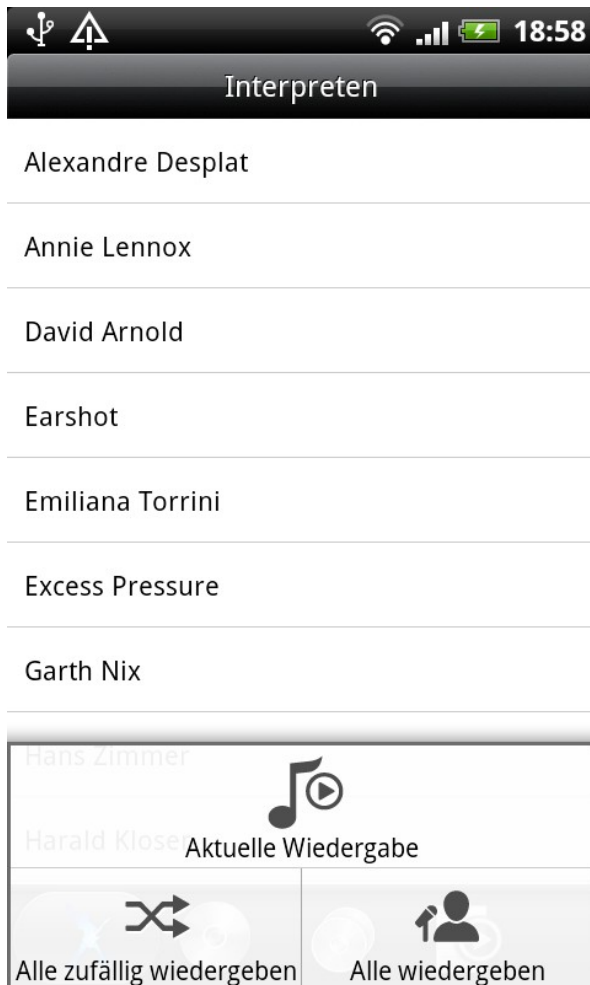


# Resources

- Manage data with xml
- Resource-Class is generated automatically
  - holds ids for the elements
- Contains:
  - Layouts, Strings, Images, Values, Menus, Settings



# Menu, Contextmenu and Search

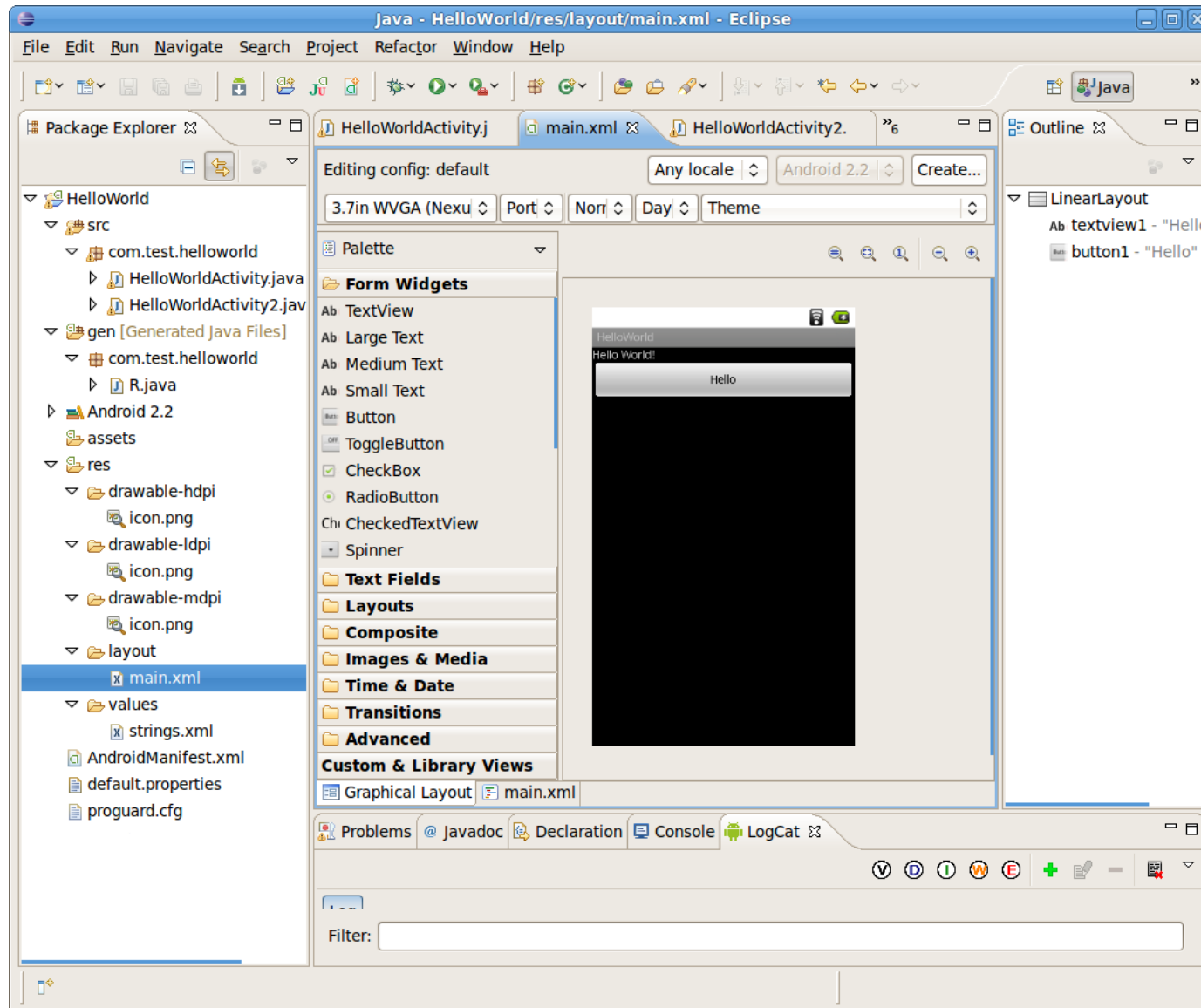


# Development environment

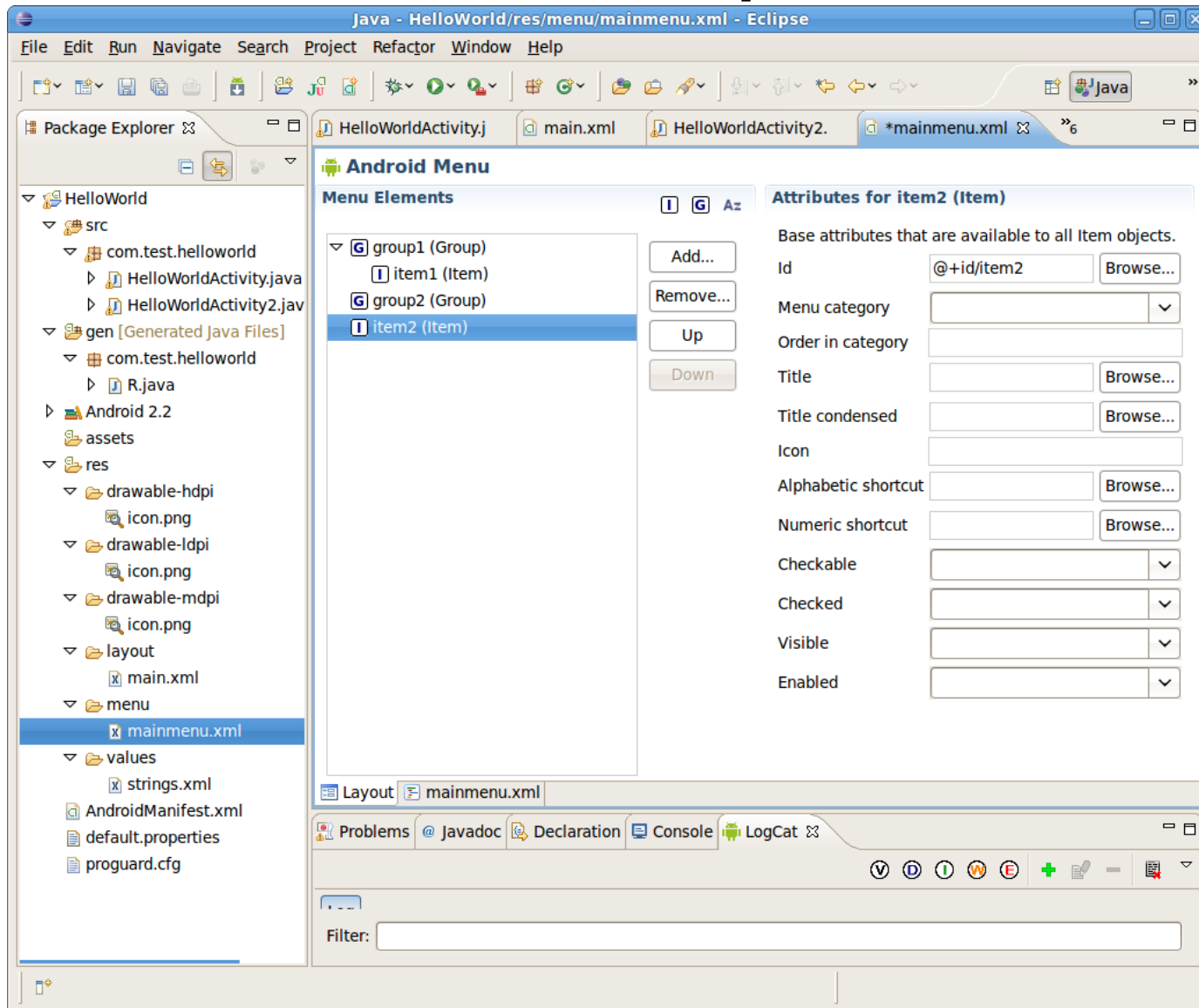
- Recommended: Eclipse
- Fully integrated:
  - GUI-Builder
  - XML-“Helper“
  - Console
  - Debugging-Tools



# GUI-Builder



# XML-“Helper”



# Console and Debugging-Tools

The screenshot displays the DDMS interface within the Eclipse IDE. The main window is titled "DDMS - HelloWorld/res/menu/mainmenu.xml - Eclipse". The interface includes a menu bar (File, Edit, Run, Navigate, Search, Project, Refactor, Window, Help) and a toolbar. The left sidebar contains the "Devices" panel, showing a list of virtual devices. The "Emulator Control" panel is also visible, featuring controls for telephony status and actions. The main area is divided into two panes: the "File Explorer" and the "LogCat" panel.

**File Explorer**

Name	Size	Date	Time	Permissions	Info
data		2011-09-20	11:03	drwxrwx--x	
mnt		2011-09-21	07:21	drwxrwxr-x	
system		2011-08-26	13:56	drwxr-xr-x	
app		2011-08-26	13:56	drwxr-xr-x	
bin		2011-08-26	13:55	drwxr-xr-x	
build.prop	5770	2011-08-26	13:56	-rw-r--r--	
customize		2011-08-26	13:56	drwxr-xr-x	
etc		2011-08-26	13:55	drwxr-xr-x	
fonts		2011-08-12	09:26	drwxr-xr-x	
framework		2011-08-26	13:55	drwxr-xr-x	
lib		2011-08-26	13:56	drwxr-xr-x	
media		2011-02-07	13:24	drwxr-xr-x	
tts		2011-02-07	13:24	drwxr-xr-x	
usr		2011-08-12	09:27	drwxr-xr-x	

**LogCat**

Time	pid	tag	Message
09-21 09:12:34	D 1327	NetUtils	ifc_get_info addr=0 !
09-21 09:12:35	D 1327	WifiStat	Disabling interface
09-21 09:12:35	W 2753	wpa_supp	l2_packet_receive - rcvfrom
09-21 09:12:36	D 1327	WifiStat	Not get IP address yet, shall not do WiFi offload
09-21 09:12:36	D 1327	WifiStat	Not get IP address yet, shall not do WiFi offload
09-21 09:12:36	V 1873	SSHDroid	EventsReceiver: NETWORK_STATE_CHANGED_ACTION (exists: true, connected: false)
09-21 09:12:36	D 1327	Connecti	getMobileDataEnabled returning false
09-21 09:12:36	D 1327	Connecti	getMobileDataEnabled returning false
09-21 09:12:36	D 1327	WifiStat	Not get IP address yet, shall not do WiFi offload
09-21 09:12:36	D 1327	Connecti	getMobileDataEnabled returning false



# Where do I publish my App?

- Android Market:
  - 510,000 Apps in Sept. 2011
  - uses Google Checkout for billing
  - takes 30% of the app-price
- Until now:
  - No controlles!



# Publishing on Android Market

## Requirements:

- Google Account
- register as developer on [developer.android.com](http://developer.android.com)
  - pay 25\$
  - only Creditcards
- For charged apps:
  - register at Google Checkout





# Publishing on Android Market

## Requirements for the App:

- App must be signed
  - expires after 22.10.2033
- AndroidManifest.xml:
  - *android:versionCode* and *android:versionName* must be defined
  - *android:icon* and *android:label* must be defined



# Publishing without Android Market

- Complete free:
  - Generate .apk
  - Put .apk on your website
  - .apk must be copied to device and installed (e.g. with APK-Manager)
- Problems:
  - No (normal) User will do that!
  - Billing, integration, updates



# Publishing without Android Market

- Alternative Markets:
  - Amazon App Store (about 18,000 Apps)
  - SlideME (about 10,000 Apps)
  - AndAppStore (about 2,000 Apps)
  - OnlyAndroid
- But:
  - Google can delete your app at any time!
  - Not so many customers



# Lessons learned

- Android is **NOT** Java
- XML-Layouting is not always fun
- Complicated solutions for simple problems
- Intents are sometimes mysterious
- Looks half-baked sometimes
- Docu is not as good as iOS-Docu



# But...

- Costs:
  - Development for free
  - Publishing 25\$ once
- Market share
- Mainly Java
- Freedom of choice
- ...

