



Android Entwicklung

MobileDevCamp Munich

31.10.2009

Markus Junginger



greenrobot_de

greenrobot

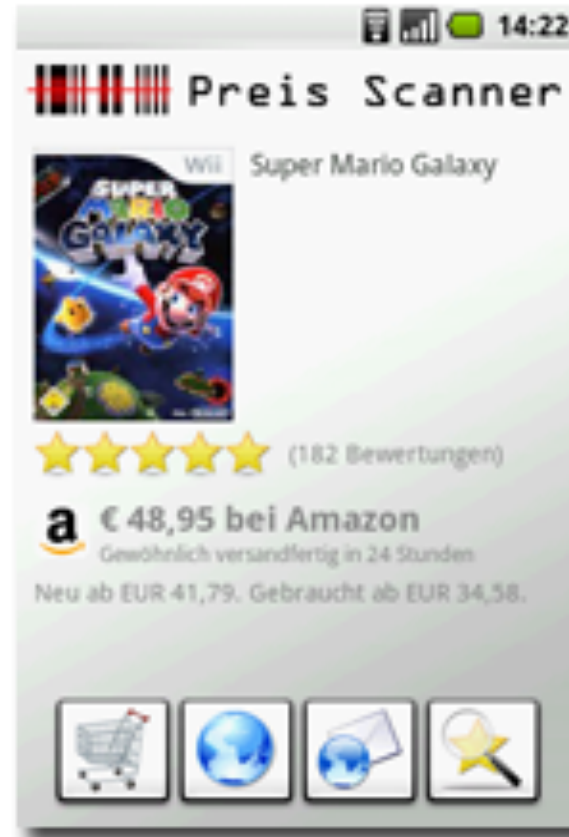
greenrobot

Preis Scanner



Barcode einscannen

Mit der Handy-Kamera.
Oder Code eintippen.



Produktübersicht

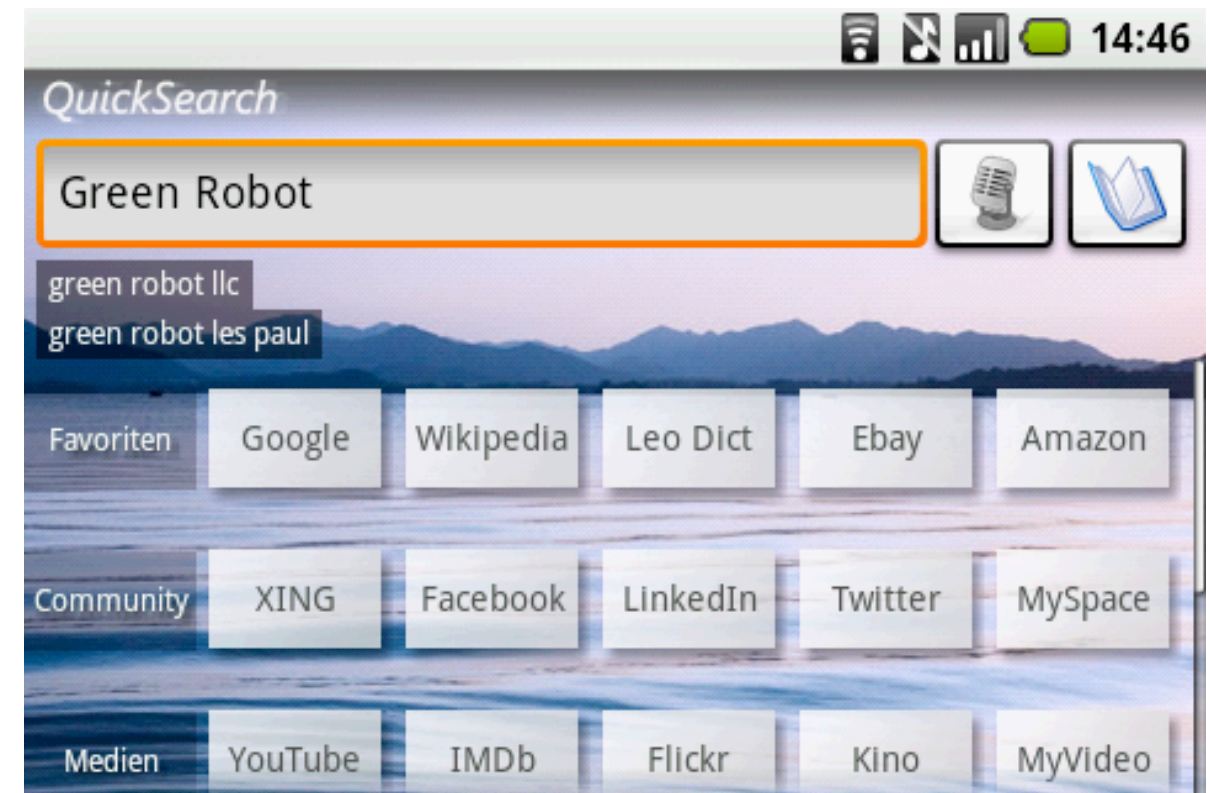
Übersicht über Preise und
Bewertungen.



Produktdetails

Bewertungen lesen.
Per Email verschicken.
Weblinks (Infos, Preise, Tests).
Bestellen.

Apps



greenrobot

Über mich

- Erste Android App: 2007
- Erstes Mobile Projekt: 2001
- 10 Jahre Java & Internet Erfahrung
- 20 Jahre Entwicklungserfahrung
- Android Beratung & Entwicklung Telco
- Android Technologie Startup

Outline

- Einführung
- Activities & Intents
- Hello World
- User Interface (UI), Fallbeispiel Listen
- Typische Tasks (Entwicklung)
- Erfahrungswerte

Warum ist Android interessant?

- 20 Android-Geräte bis Ende 2009 geplant
- Prognose: Wachstumsrate von 900%¹
- Nicht nur Handys (Tablets, Haus, Auto, ...)
- Gute Adoption durch Industrie (Preis, ...)
Beispiel HTC 2010: Android > WiMo
- Telekom, Vodafone, O2, E-Plus
- Neu: Google Maps Navigation

1) [Strategy Analytics](http://greenrobot.de/mobile-news/900-wachstum-fuer-android-in-2009-studie), <http://greenrobot.de/mobile-news/900-wachstum-fuer-android-in-2009-studie>

APPLICATIONS

Home

Contacts

Phone

Browser

...

APPLICATION FRAMEWORK

Activity Manager

Window
Manager

Content
Providers

View
System

Package Manager

Telephony
Manager

Resource
Manager

Location
Manager

Notification
Manager

LIBRARIES

Surface Manager

Media
Framework

SQLite

OpenGL | ES

FreeType

WebKit

SGL

SSL

libc

ANDROID RUNTIME

Core Libraries

Dalvik Virtual
Machine

LINUX KERNEL

Display
Driver

Camera Driver

Flash Memory
Driver

Binder (IPC)
Driver

Keypad Driver

WiFi Driver

Audio
Drivers

Power
Management

Vier Entwicklungsarten

- Android SDK: Entwicklung mit Java
- Webseiten (Chrome/WebKit)
- Cross Platform Apps
- Native Entwicklung
 - NDK in Verbindung mit Android SDK
 - Eigene Android Firmware

Vier Entwicklungsarten

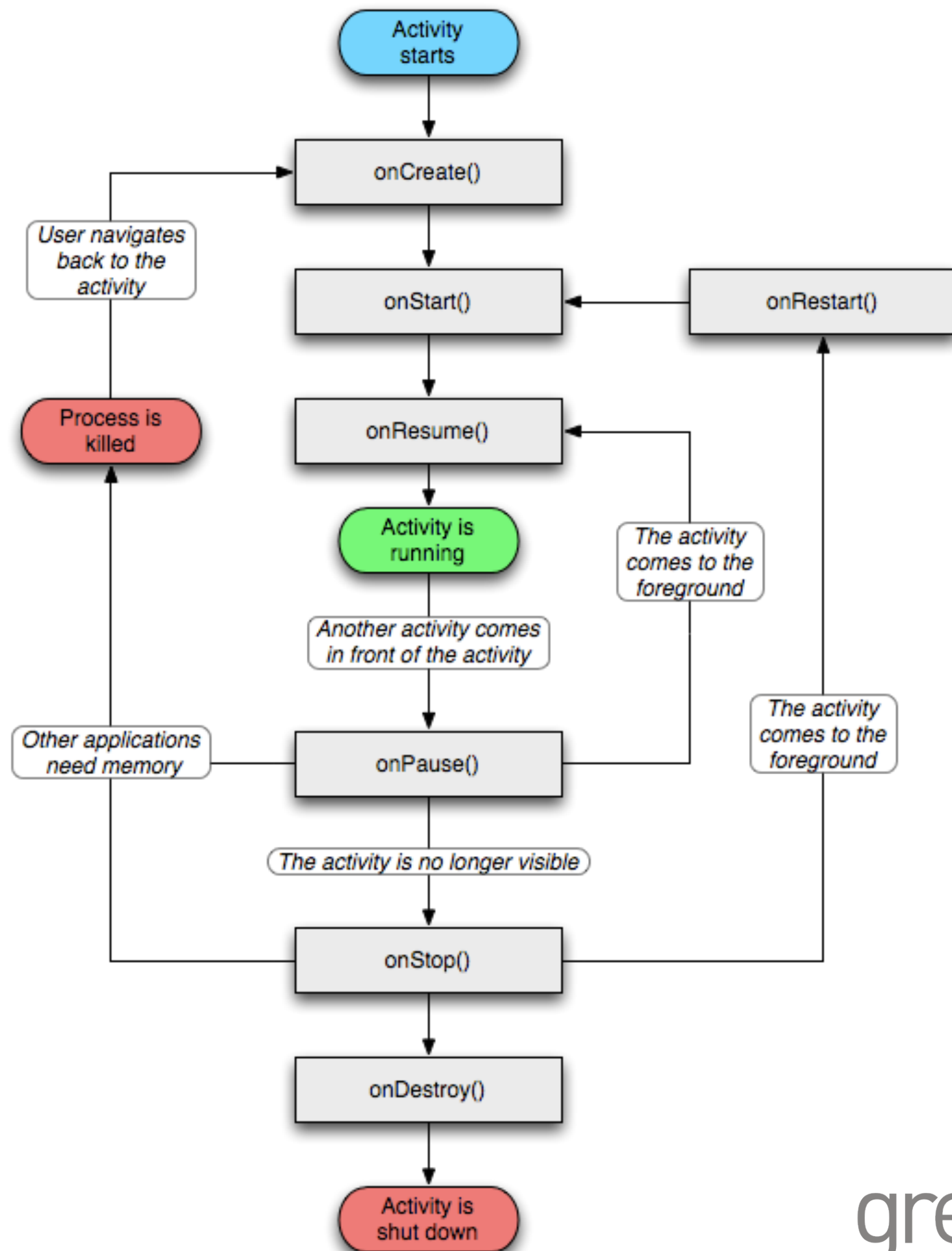
- **Android SDK: Entwicklung mit Java**
- Webseiten (Chrome/WebKit)
- Cross Platform Apps
- Native Entwicklung
 - NDK in Verbindung mit Android SDK
 - Eigene Android Firmware

Entwicklung mit Android SDK

- Java 5 (Scala, JRuby)
- Java 5 SE APIs teilweise vorhanden (io, nio, lang, util, math, etc.)
- Proprietäre Android APIs (android.*)
- Java Bytecode wird in DEX umgewandelt
- Dalvik Virtual Machine
- IDE: Eclipse (oder von Hand)
- Device Emulator

Activities

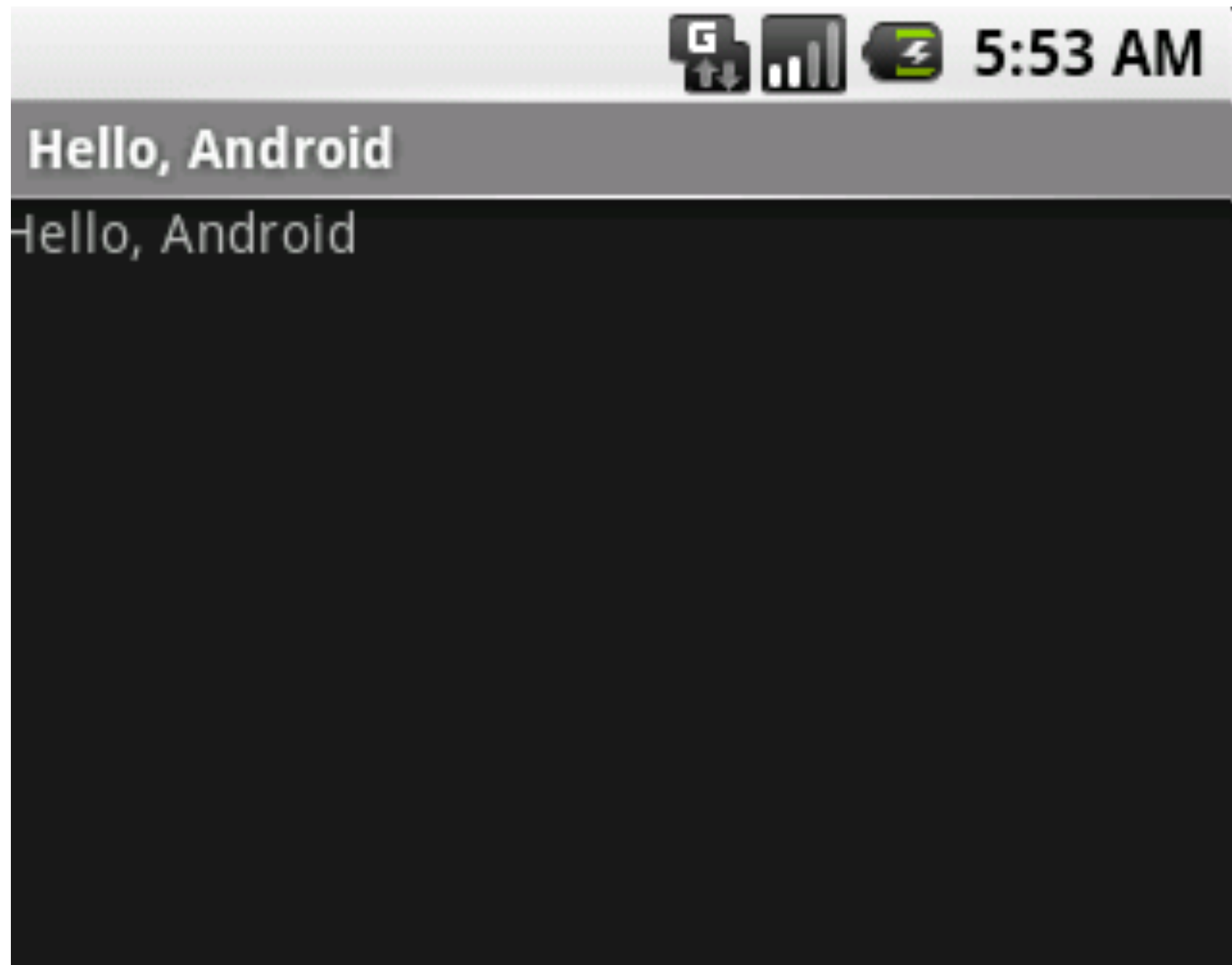
- Einstiegspunkt in Android Programm
- Aktiver Programmteil
- Eine Activity setzt (mind.) ein View (GUI)
- Activities unterliegen Life Cycle
- onResume und onPause (Persistenz!)
- Android OS kann Activities killen
- Alternative: Background Services



Intents

- Eine Absicht etwas zu tun ohne zu definieren wie es getan werden soll
- Vernetzung von Apps über lose Kopplung
- Angelehnt an URLs (Beispiel: „tel:123“)
- Auch für das Aktivieren eines neuen Zustand einer App: Starten einer Activity

Hello World

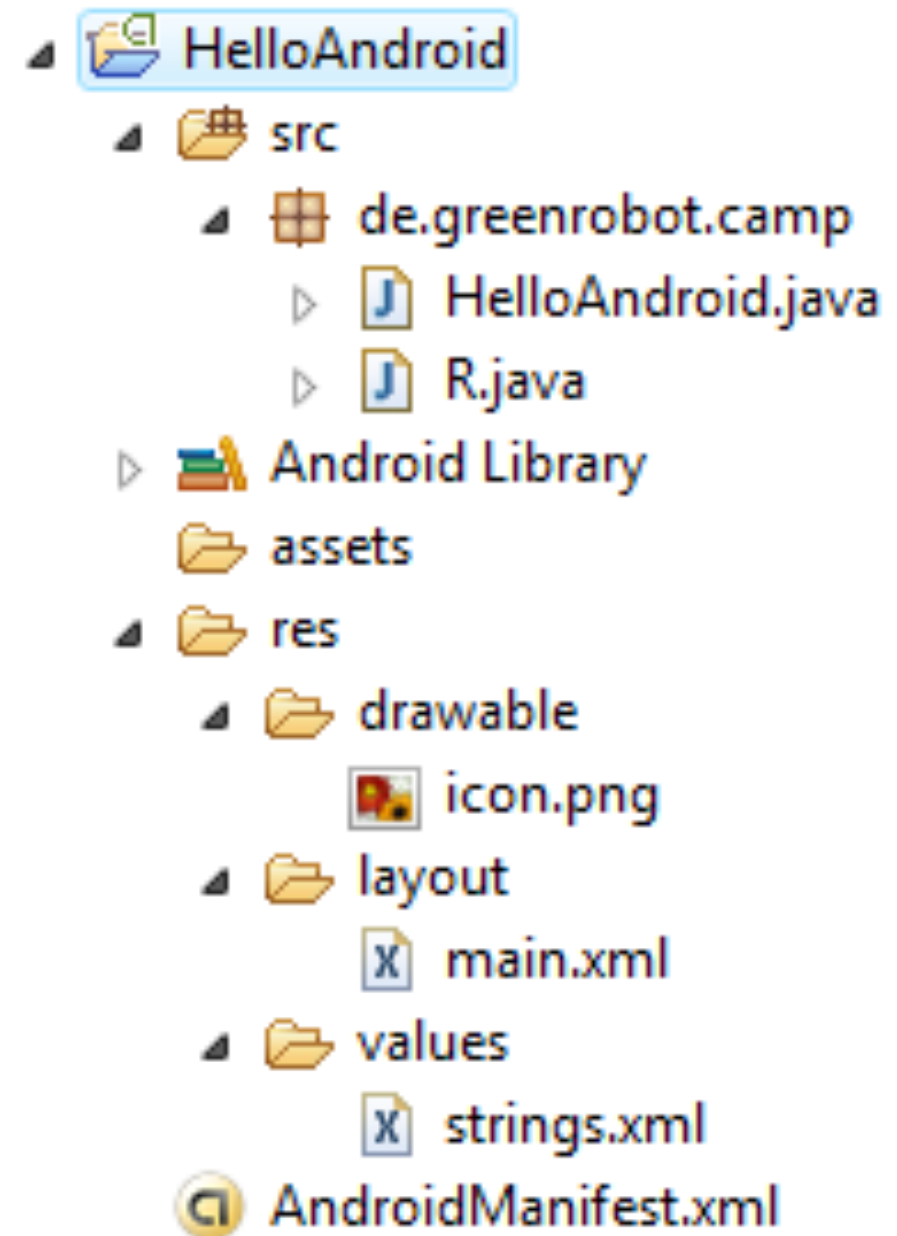


Hello World

```
public class HelloAndroid extends
    Activity {
    @Override
    public void onCreate(Bundle state) {
        super.onCreate(state);
        TextView tv = new TextView(this);
        tv.setText("Hello, Android");
        setContentView(tv);
    }
}
```

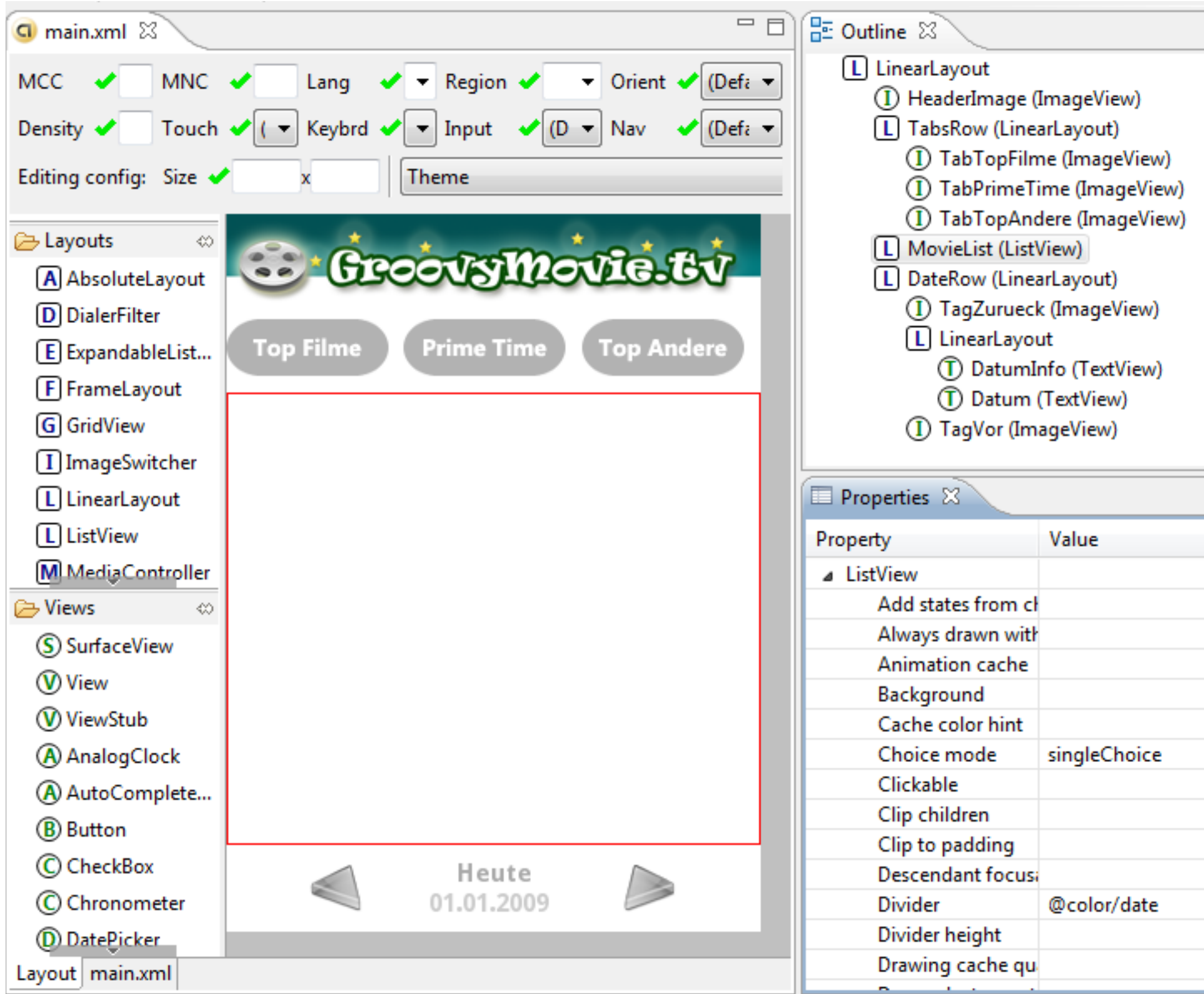
Android Projekt anlegen

- IDE erstellt Gerüst
- R.java ist generiert
- res Verzeichnis
- AndroidManifest.xml



UI Prinzipien: Klassisch...

- Als Ressourcen oder programmatisch
- UI Komponenten: Basisklasse View
- Layouts (Linear, Grid, Table, Relative, ...)
- UI Thread: Änderungen an der GUI
- Events (OnClick, ...)
- Visueller GUI Editor in Eclipse
- Low-Level Canvas API



Beispiel ListView

- Adapter (MVC) hält die Daten der Liste vor
- Adapter erzeugt einzelne Zeilen (MVC)



ListView: Adapter

```
public class MovieAdapter extends ArrayAdapter<Movie> {  
    ...  
    public View getView(int position, View convertView, ViewGroup parent) {  
        LinearLayout layout;  
        if (convertView instanceof LinearLayout) {  
            layout = (LinearLayout) convertView;  
        } else {  
            layout = (LinearLayout) inflater.inflate(R.layout.movie_row, null);  
        }  
        Movie movie = movies.get(position);  
        TextView title = (TextView) layout.findViewById(R.id.MovieTitle);  
        title.setText(movie.getTitleWithYear());  
        return layout;  
    }  
}
```

Adapter Performance

- Inflate jede Zeile: schlechte Performance
- View Recycling (convertView): Faktor 3
- Zusätzlich ViewHolder: bis zu Faktor 4

UI Performance

- Hintergrund
 - Bitmaps vorskalierten (bis zu Faktor 2!)
 - Theme ohne Hintergrund (@null) (Donut)
- Invalidate: nur Teilbereiche wenn möglich
- Layouthierarchie (8k Stack vor Donut)
- ViewStub für ausgeblendete Views
- Custom Views anstatt Layout Hierarchy

Asynchrone Verarbeitung & UI

- UI Thread darf niemals blockiert werden
- Nach 5s Android System Dialog
- Netzwerkzugriffe, etc. in eigenem Thread
- ProgressDialog, Progress Indicator
- UI Update aber wieder im UI Thread
 - Verschiedene Möglichkeiten, z.B. AsyncTask

Client-/Server Kommunikation

- Apache HttpClient 4
- Kein SOAP out of the box → kSOAP
- REST mit XML
 - XML: SAX, DOM, Pull aber kein XPath
- REST mit JSON
 - JSON integriert
 - Einfach zu parsen

Persistenz

- Files für XML, JSON, Java Objekte, ...
- Preferences mit Key/Value Paaren
→ Vorteil: einfache UI Erstellung
- SQLite
- Über Applikations-/Prozessgrenzen:
Content Provider

Performance

- Hohe Performance für normale Apps
- Optimierungen nötig für:
 - Background Services (be nice)
 - Spiele, Multimedia (flüssiger Ablauf)
- GC legt VM für 100-400ms lahm
- Regel #1: GC vermeiden. Keine neuen Objekte in zentralen Schleifen anlegen
- Versteckte Anlage von Objekten

Android Market

- Google
- Alternative App Stores existieren
- Registrierung notwendig, einmalig \$25
- Signieren mit Zertifikat (self-signed)
- Kein Review Prozess, sofort im Market
- Bezahlung über Google Checkout
- 70% der Einnahmen an Entwickler

Pluspunkte

- Schneller Einstieg für Java Entwickler
- All-in: Eclipse-basierte IDE
- Open Source
- Sicherheit (Permissions, Prozesse, User)
- Loose Coupling zwischen Apps (Intents)
- UI Customization mit Themes
- GPS, Kamera, Beschleunigungssensor

Minuspunkte

- Nach schnellem Einstieg komplexe Details
- Dalvik VM erinnert an Java 1.0/1.1 VM
- Homescreen Widgets sehr eingeschränkt
- Fehlende Event Unterstützung UI Editor
- Ressourcen (R) vs. mehrere Projekte
- Kein Desktop Unit Testing out-of-the-box
- Android Market Bugs

Weitere Ressourcen

- developer.android.com
- SDK: Dev Guide mit guten How-tos, Referenz und API Demos mit Source
- Android Developer Blog
- android-hilfe.de Forum
- XING Gruppe



<http://kumpa-game.com>

Release: droidcon (3./4.11.)

greenrobot

Vielen Dank! Q&A

[groups.google.de/group/
android-muenchen](https://groups.google.de/group/android-muenchen)

Markus Junginger

markus@greenrobot.de

<http://greenrobot.de>

<http://jars.de>

Twitter: [greenrobot_de](https://twitter.com/greenrobot_de)



greenrobot

