

Android APIs 2.0 & 2.1

What's new?



m2d2, 23.02.2010

Markus Junginger

greenrobot

Outline

- Historie Android 1.0 - 2.0
- Android 2.0 APIs
- Bluetooth
- Quick Contacts
- Multitouch
- Live Wallpaper

Android 1.1

- API-Level 2
- API Änderungen sehr überschaubar
- Bugfix Release
- `android:minSdkVersion` ist Pflicht-Attribut
- Kleine App Verbesserungen
 - Suche nach POI (Google-Maps-App)
 - MMS Anlagen lokal speichern

Android 1.5 (cupcake)

- API-Level 3
- UI-Elemente überarbeitet
- Virtual Keyboard
- Home-Screen Widgets
- Copy & Paste
- Performance Verbesserungen



Android 1.6 (donut)

- API-Level 4
- Android-Market Update
- Quick-Search-Box
- Native VPN-Unterstützung
- Text-to-speech Engine
- Auflösungen und Pixeldichten
- Energieverbrauch AppWidget
- Native Entwicklung mit dem NDK



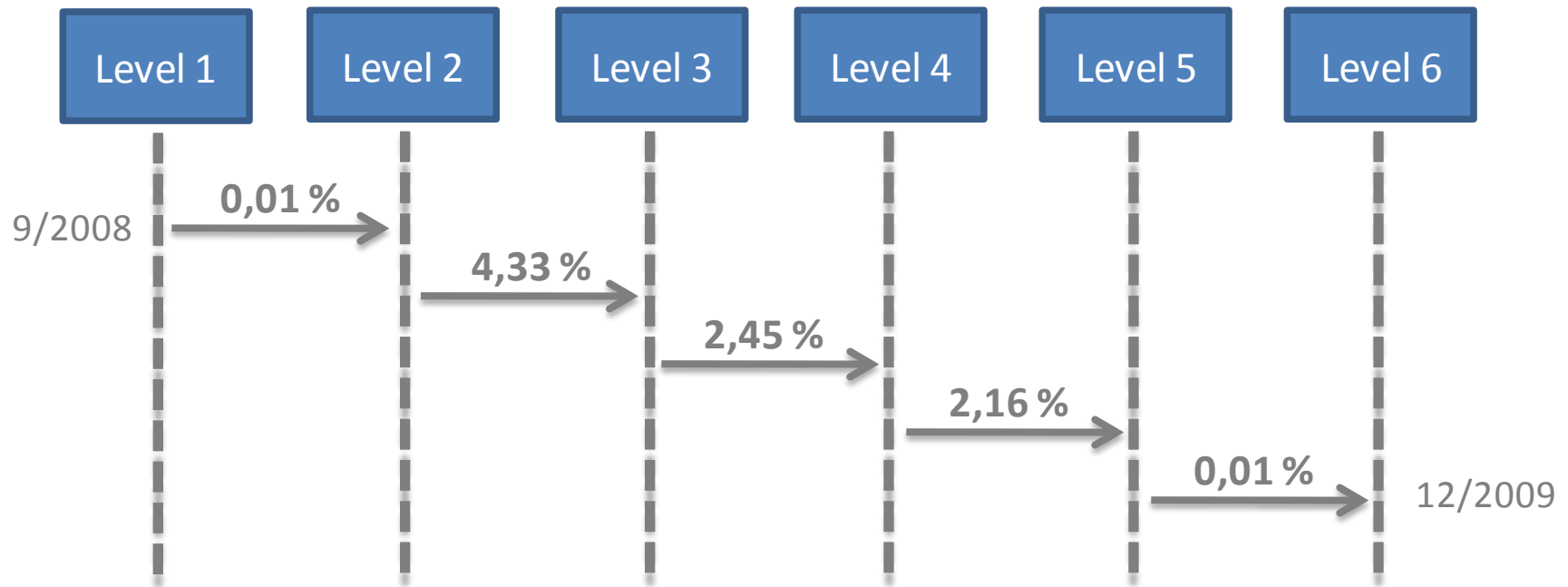
Android 2.0 (eclair)

- API-Level 5 und 6 (2.0.1)
- Multitouch API
- Bluetooth API
- Verbesserte virtuelle Tastatur
- Quick-Search (SMS, MMS)
- Quick-Contact-Widget
- Aktualisierter Browser
- Mehrere Email-Konten synchronisieren
- Kamera-Funktionen erweitert



Android API Differences Report

$$\text{Percentage difference} = \frac{100 * (\text{added} + \text{removed} + 2 * \text{changed})}{\text{sum of public elements in BOTH APIs}}$$



Bluetooth

- Android BT Stack basiert auf Bluez 4.47
 - Linux Standard (von SIG anerkannt)
- Bluetooth 2.1 EDR (2,1 Mb/s)
- Emulator unterstützt kein Bluetooth
- `permission.BLUETOOTH`
 - Mit Paired-Devices verbinden
- `permission.BLUETOOTH_ADMIN`
 - Finden und Verbinden von BT-Geräten

Bluetooth Workflow

- BluetoothServerSocket binden
- BT Discovery starten (→ BluetoothDevice)
- Verbindung über BluetoothSocket zu einem BluetoothDevice herstellen
- Mit Streams lesen und/oder schreiben

Bluetooth API: Klassen (1)

- BluetoothAdapter: BT-Zentrale
 - Status
 - Discovery starten
 - Server BT-Socket
- BluetoothDevice: Remote BT-Gerät
 - Properties (Name, Adresse, ...)
 - Client-BT-Socket

Bluetooth API: Klassen (2)

- BluetoothSocket
 - Analog TCP Socket (`java.net.Socket`)
 - Client-Verbindung aufbauen und verwalten
- BluetoothServerSocket
 - Analog TCP ServerSocket (`java.net.ServerSocket`)

Code: Bluetooth Check

```
BluetoothAdapter bluetoothAdapter =  
    BluetoothAdapter.getDefaultAdapter() ;  
if (bluetoothAdapter != null) {  
    if(!bluetoothAdapter.isEnabled()) {  
        Intent intent = new Intent  
            (BluetoothAdapter.ACTION_REQUEST_ENABLE) ;  
        startActivityForResult(intent, 1);  
        // check for RESULT_OK in onActivityResult  
    }  
} else {  
    // No BT available  
}
```

Code: Geräte suchen

```
registerReceiver(new MyReceiver(), new  
    IntentFilter(BluetoothDevice.ACTION_FOUND));  
bluetoothAdapter.startDiscovery();
```

```
class MyReceiver extends BroadcastReceiver()  
{  
    @Override  
    public void onReceive(Context context,  
        Intent intent) {  
        if(intent.getAction().equals(  
            BluetoothDevice.ACTION_FOUND) {  
            // See next slide  
        }  
    }  
}
```

Code: Geräte suchen (2)

```
// Continued from last slide inside onReceive()  
  
BluetoothDevice device = (BluetoothDevice)  
intent.getExtras().get(BluetoothDevice.EXTRA_DEVICE);  
  
Log.d(TAG, "name:" + device.getName() + " address: "  
    + device.getAddress());  
  
BluetoothClass clazz=device.getBluetoothClass();  
boolean hasAudio =  
    clazz.hasService(BluetoothClass.Service.AUDIO);
```

Code: Bluetooth Server

```
UUID uuid = uuid.fromString("27648B4D-D854-5674-  
FA60E4F535E44AF7");
```

```
BluetoothServerSocket serverSocket =  
    adapter.listenUsingRfcommWithServiceRecord("MyBlue  
toothApp", uuid);
```

```
BluetoothSocket socket = serverSocket.accept();  
InputStream in = socket.getInputStream();  
OutputStream out = socket.getOutputStream();  
// Read, write, and close
```

Code: Bluetooth Client

```
UUID uuid = uuid.fromString("27648B4D-D854-5674-  
FA60E4F535E44AF7"); // identical UUID
```

```
BluetoothDevice device = adapter.  
    getRemoteDevice("00:11:22:33:44:55");
```

```
BluetoothSocket socket = device.  
    createRfcommSocketToServiceRecord(uuid);
```

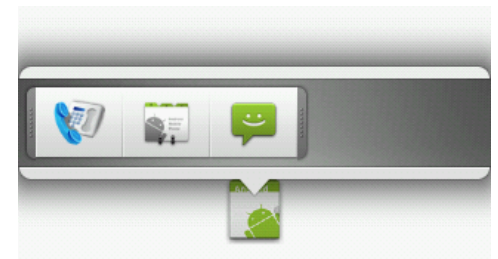
```
socket.connect();  
InputStream in = socket.getInputStream();  
OutputStream out = socket.getOutputStream();  
// Read, write, and close
```

Bluetooth Profile (keine API)

- Protokoll zum Datenaustausch
- Werden beim Verbindungsaufbau mitgeteilt
- Android 1.0 Profile
 - SDP, L2CAP, GAP, RFCOMM, SPP
- Android 1.5
 - AVCTP, AVRCP, GAVDP, AVDTP, A2DP
- Android 2.0 neue Profile
 - OOP (Object Push Profile)
 - PBAP (Phonebook Access Profile, read only)

Quick Contact

- Neues Widget (abgeleitet von ImageView)
`android.widget.QuickContactBadge`
- Schneller Zugriff auf Kontaktdaten
- Leicht in eigene App integrierbar
- Referenzierung des Kontakts via URI,
Email-Adresse oder Telefonnummer



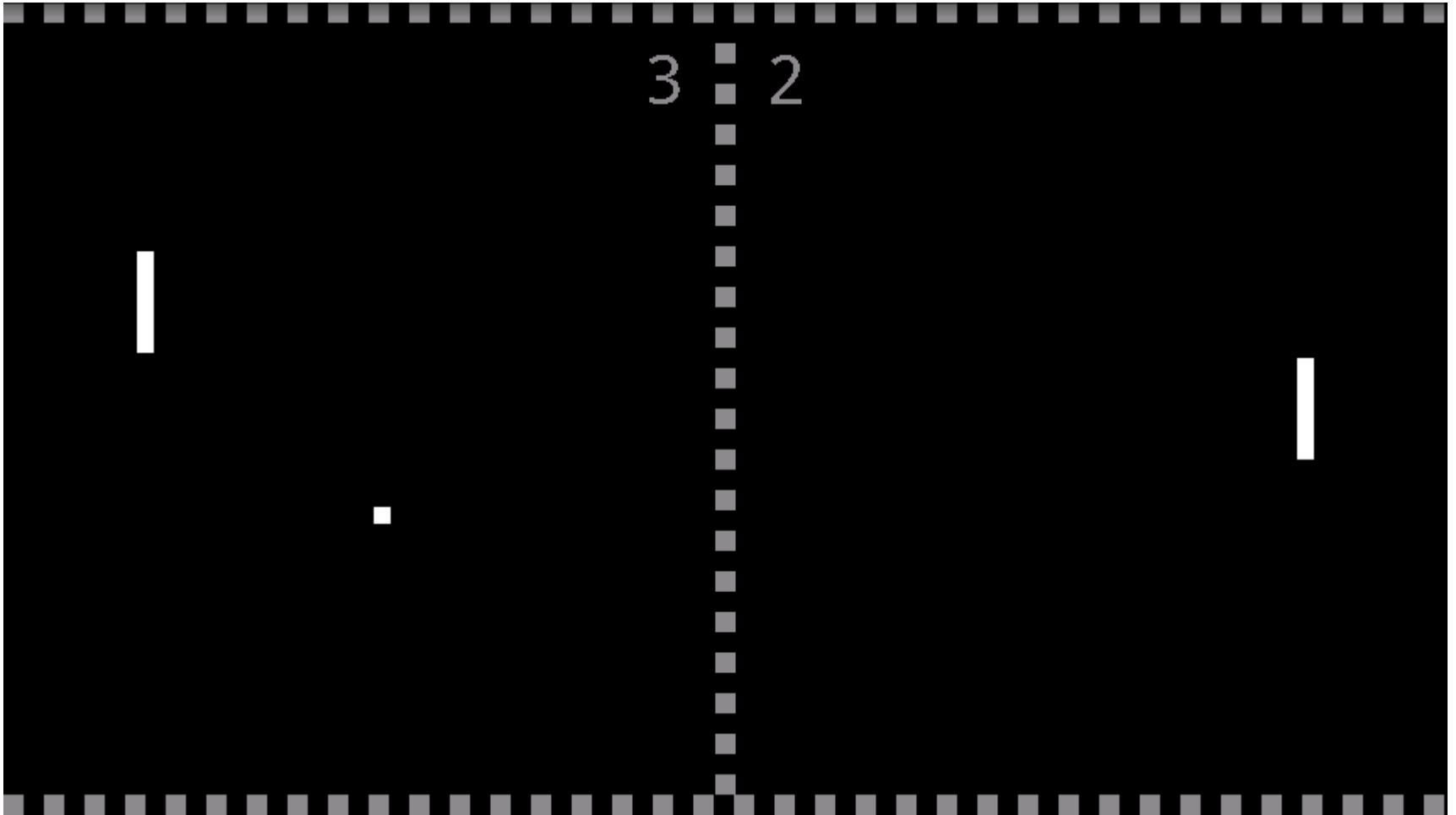
Quick Contact - Beispiel

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.contacts);  
  
    String phoneNumber = "0891234";  
    QuickContactBadge badge = (QuickContactBadge)  
        findViewById(R.id.QuickContactBadge);  
    badge.assignContactFromPhone(phoneNumber, true);  
}
```

Touch und Multitouch UIs

- Touch: wesentlich für moderne mobile UIs
- iPhone hat es vorgemacht
- Multitouch erlaubt Gesten (Pinch & Zoom)
- G1 Hardware Multitouch-Gesten fähig
- API Multitouch Support ab Android 2.0
- Basiert auf Touch API

Anwendungsbeispiel 😊



(Single) Touch Events

- Seit Android 1.0
- Auf MotionEvent registrieren
 - Activity.onTouchEvent
 - onTouchListener
- MotionEvent Daten
 - Action: Down, Move, Up
 - X und Y Koordinaten

(Single) Touch Events: Code

```
@Override
public boolean onTouch(View v,
    MotionEvent event) {
    if (event.getAction() ==
        MotionEvent.ACTION_DOWN) {
        int x = event.getX();
        int y = event.getY();
        doSomething(x, y);
    }
}
```

Multitouch API Dokumentation

- Keine API Demos, kein Blogpost
- API Docs teilweise verwirrend
(z.B. ACTION_POINTER_1_DOWN)
- Android Developer Mailing Liste
http://groups.google.com/group/android-developers/browse_thread/thread/962b1a27a31e450d
- Try & Error (Loggen von Events)

Android 2.0 Multitouch API

- MotionEvent: zusätzliche Daten
- Pointer
- `event.getPointerCount()`
- `event.getX(pointerIndex)`
- Analog: `getY`, `getPressure`
- `event.getPointerId(pointerIdx)`

Multitouch: Action Kodierung

- Action bei Multitouch

```
int action = event.getAction() &  
MotionEvent.ACTION_MASK;
```

- Pointer ID aus Action bestimmen

```
int id = (event.getAction() &  
MotionEvent.ACTION_POINTER_ID_MASK) >>  
MotionEvent.ACTION_POINTER_ID_SHIFT;
```

Multitouch: Neue Actions

- ACTION_POINTER_DOWN
- ACTION_POINTER_UP
- Pointer ID aus Action Kodierung
- Beispiel Ablauf (ohne Move):
 - ACTION_DOWN: 1. Finger
 - ACTION_POINTER_DOWN: 2. Finger
 - ACTION_POINTER_UP: 1. oder 2. Finger
 - ACTION_UP: Verbliebener Finger

Beispiel Ablauf (ohne Move)

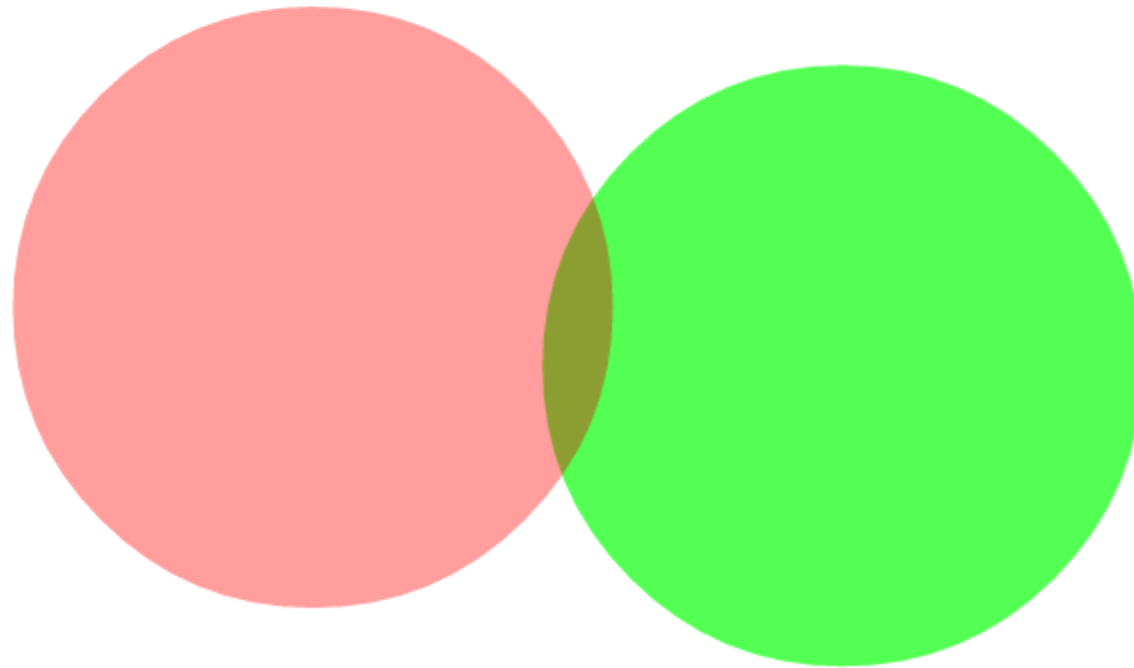
- ACTION_DOWN: 1. Finger
- ACTION_POINTER_DOWN: 2. Finger
- ACTION_POINTER_UP: 1./2. Finger
- ACTION_UP: Verbliebener Finger

Multitouch Code-Beispiel



17:55

Android 2.0 Samples



Multitouch Code-Beispiel 1/2

- Farben setzen bei UP/DOWN

```
int action =
    event.getAction() & MotionEvent.ACTION_MASK;

int id = (event.getAction() &
    MotionEvent.ACTION_POINTER_ID_MASK) >>
    MotionEvent.ACTION_POINTER_ID_SHIFT;

if (action == MotionEvent.ACTION_DOWN || action ==
    MotionEvent.ACTION_POINTER_DOWN) {
    colors[id] = colorsDown[id];
} else if (action == MotionEvent.ACTION_UP || action
    == MotionEvent.ACTION_POINTER_UP) {
    colors[id] = colorsUp[id];
}
```

Multitouch Code-Beispiel 2/2

- Positionen setzen

```
int count = event.getPointerCount();  
for (int i = 0; i < count; i++) {  
    int pointerId = event.getPointerId(i);  
    float x = event.getX(i);  
    float y = event.getY(i);  
    points[pointerId] = new PointF(x, y);  
}
```

Multitouch Anmerkungen

- Keine Beschränkung: Anzahl der Punkte
- Motorola Milestone Milestone
 - Erstes Android 2.0 Gerät
 - Unterstützt nur 2 gleichzeitige Punkte (Finger)
- HTC Geräte noch beschränkter:
Bounding-Box zwischen 2 Fingern

Live Wallpaper

- Aktiver Home Screen Hintergrund ab API 2.1
- Animationen, Interaktion
- Als Service realisiert (WallpaperService)
- Implementierung über WallpaperService.Engine
- XML Metadaten



WallpaperService im Manifest

```
<service android:label="@string/wallpaper_label"  
    android:name=".MyWallpaper1"  
    android:permission =  
        "android.permission.BIND_WALLPAPER">  
    <intent-filter>  
        <action android:name =  
"android.service.wallpaper.WallpaperService" />  
    </intent-filter>  
    <meta-data  
        android:name="android.service.wallpaper"  
        android:resource="@xml/myxmlfile" />  
</service>
```

WallpaperService Class

```
public class MyWallpaper extends WallpaperService {  
    @Override  
    public Engine onCreateEngine() {  
        return new MyWallpaperEngine();  
    }  
  
    // onCreate, onDestroy, ...  
}
```

WallpaperService.Engine

- Life-Cycle Methoden
`onCreate, onDestroy`
- Offset-Änderung
`onOffsetsChanged`
- Surface Callbacks
`onSurface...`
- Interaktion (Touch)
`onTouchEvent`

Zeichnen des Wallpapers

- SurfaceHolder

```
final SurfaceHolder holder = getSurfaceHolder() ;  
Canvas c = holder.lockCanvas() ;  
if (c != null) {  
    drawSomething(c) ;  
    holder.unlockCanvasAndPost(c) ;  
}
```

- Asynchron: mit Thread oder Handler

Wallpaper Einstellungen

- Im Manifest Verweis auf Datei in res/xml/

```
<?xml version="1.0" encoding="UTF-8"?>
<wallpaper xmlns:android =
    "http://schemas.android.com/apk/res/android"
    android:thumbnail="@drawable/thumbnail"
    android:description="@string/description"
    android:settingsActivity="PreferenceActivity"/>
```

Wunschliste Android 3.0

- JIT Compiler
- Hardware-beschleunigtes Rendering
- OpenGL ES 2.0
- Market Verbesserungen
 - Bezahlungsmöglichkeiten
 - In-App Käufe
- iTunes Pendant für Desktop (Synchronisieren & Verwalten)
- Einheitliches Multitouch

Vielen Dank! Q&A

kontakt@greenrobot.de

<http://greenrobot.de>

[@greenrobot_de](#)



greenrobot