

- Teil I – Augmented Learning
 - Vortragender: Krischan Udelhoven

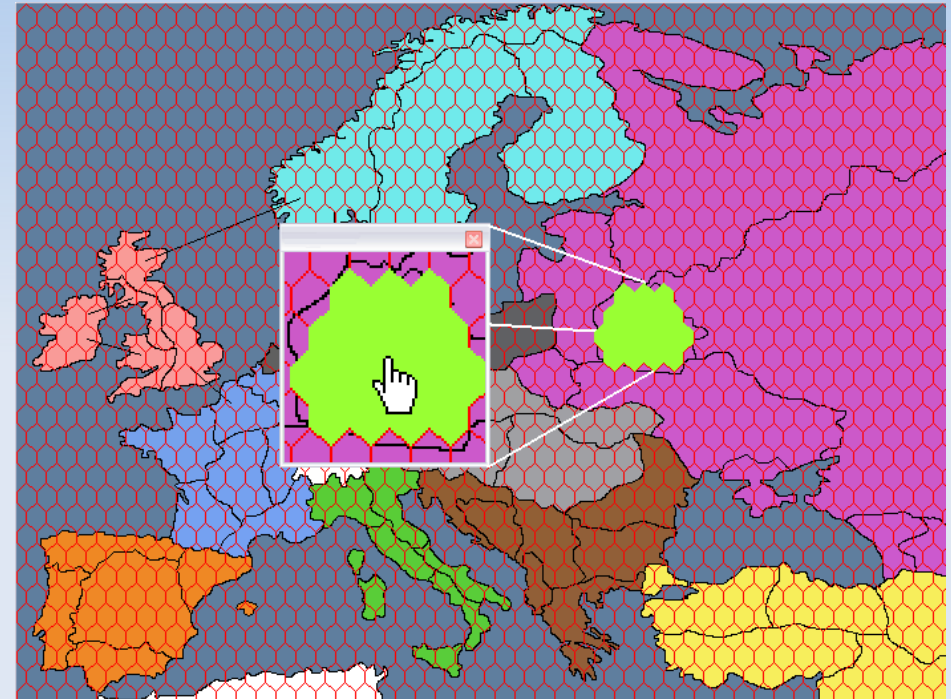
- Teil II – Android
 - Vortragender: Folker Hoffmann

Inhalt

1. Warum Android?
2. Was ist Android?
3. Entwicklung mit Android

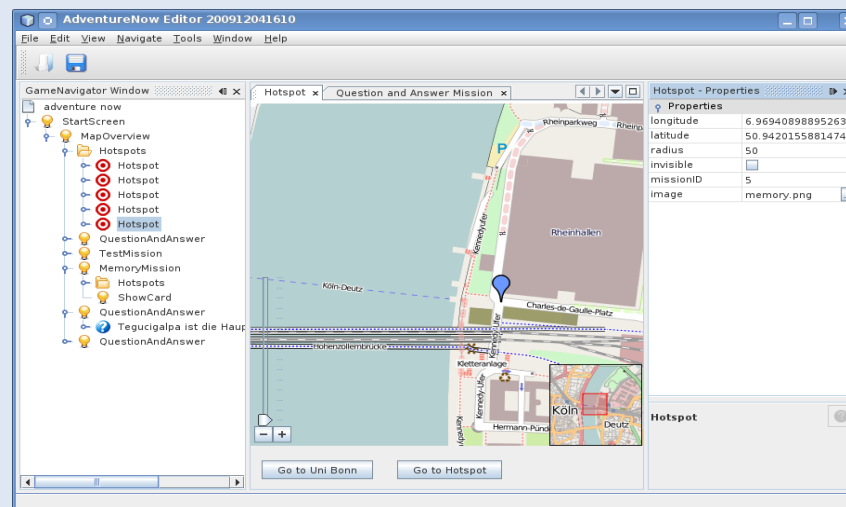
Warum Android?

- Ursprünglich: Adventure Anywhere erweitern
- Kein Source Code
- Angebot: Interfaces aber: Keine Zeit wegen EU Projekt



Warum Android?

- Deswegen Adventure Now!
- Lernspiel mit Lokalisierung
- Android Basis
- Eigener Editor

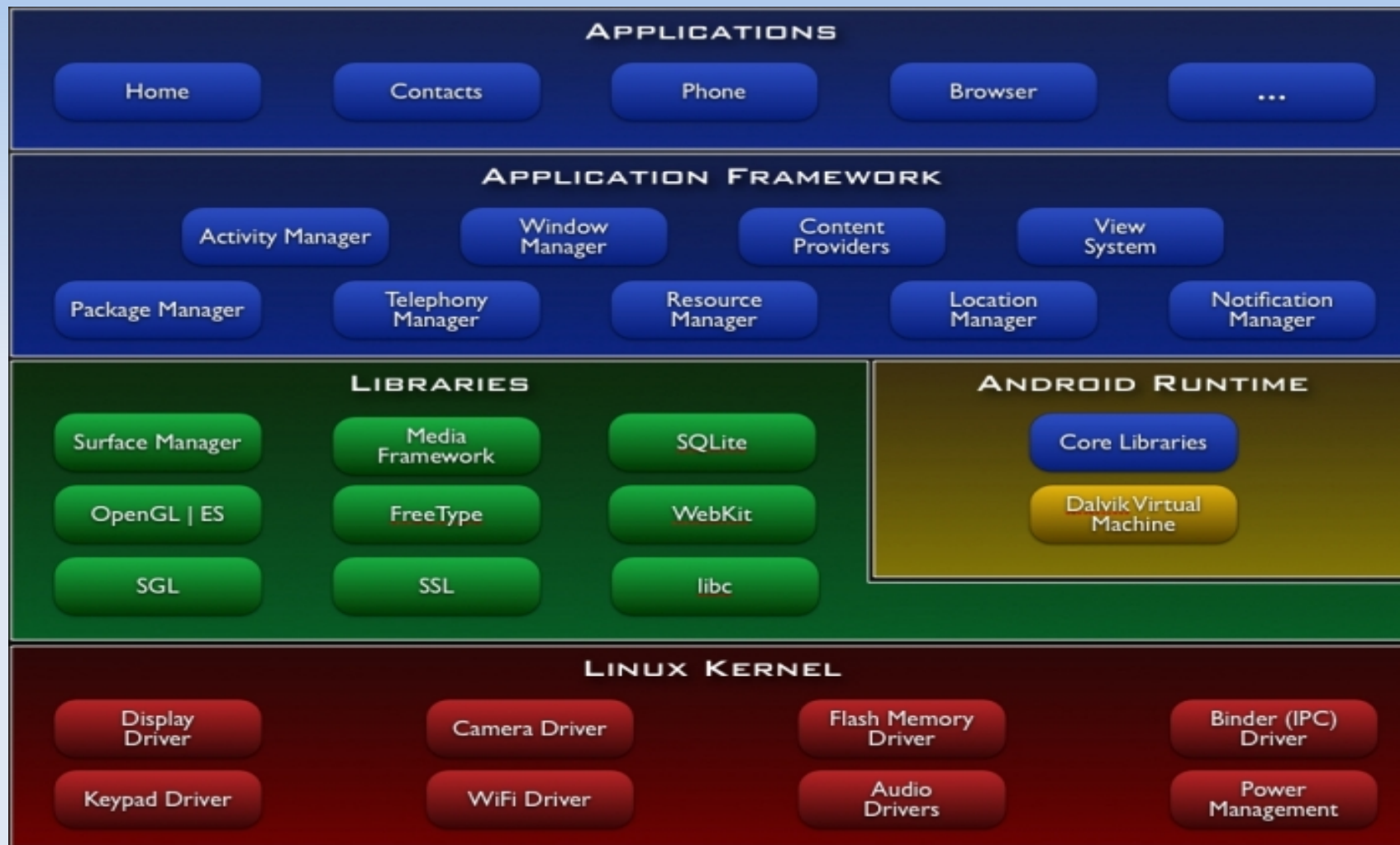


Android

- Betriebssystem für Smartphones
- Entwickelt von der Open Handset Alliance (Unter der Führung von Google)
- Open Source
- Veröffentlichung: Oktober 2008
- Anwendungsentwicklung mit Java



Aufbau von Android



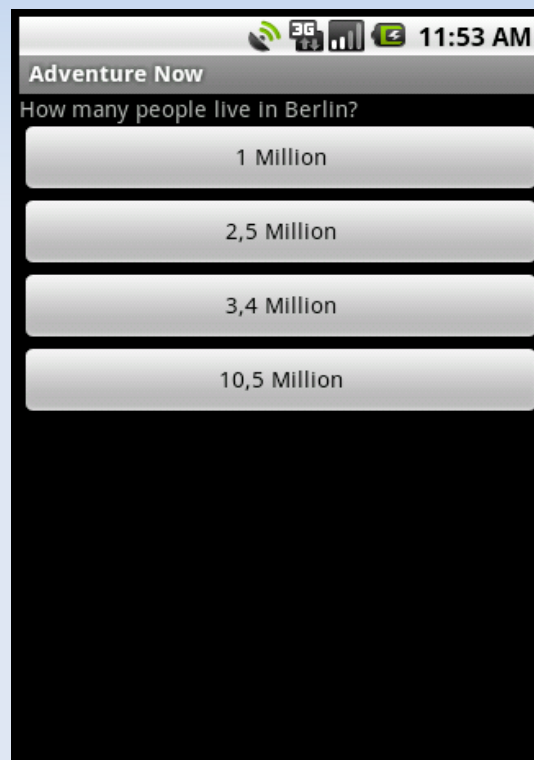
Quelle: <http://developer.android.com/images/system-architecture.jpg>

Anwendungsentwicklung

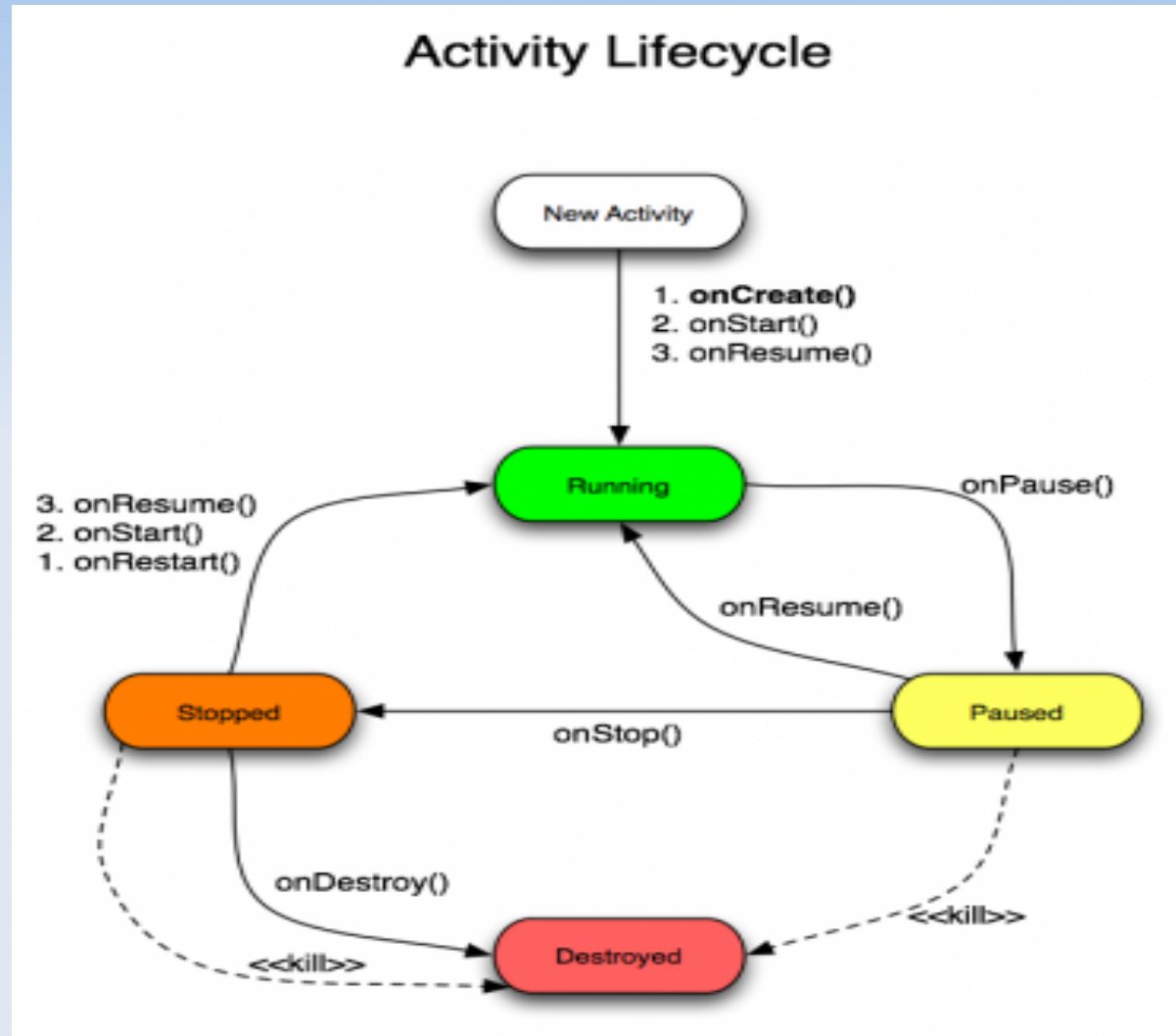
- Activity
- Ressourcen
- GUI
- Intents
- Grafik
- Sound
- Lokalisierung
- Sensoren

Activity

- Fenster (meistens Fullscreen)



Activity



Quelle: <http://mobisynth.wordpress.com/2009/07/24/android-activity-lifecycle/>

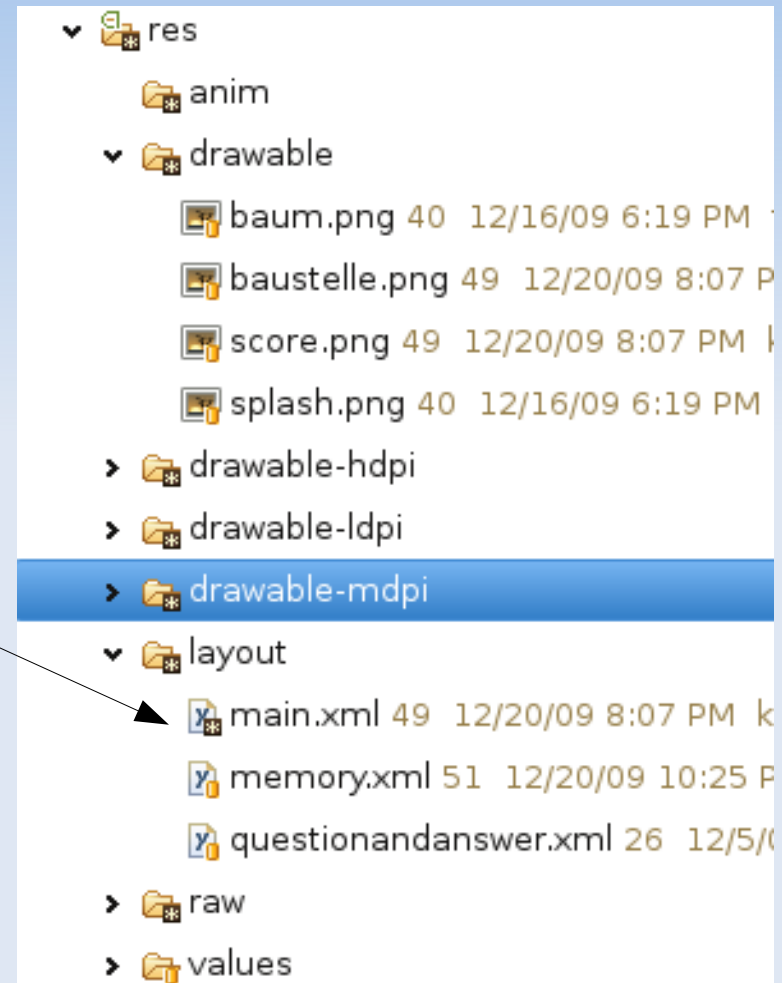
Activity

- Ableiten von `android.app.Activity`
- Oder: `com.google.android.maps.MapActivity`

```
1: public class MyActivity extends Activity {  
2:     public void onCreate(Bundle savedInstanceState) {  
3:         super.onCreate(savedInstanceState);  
4:         setContentView(R.layout.main)  
5:     }  
6: }
```

Ressourcen

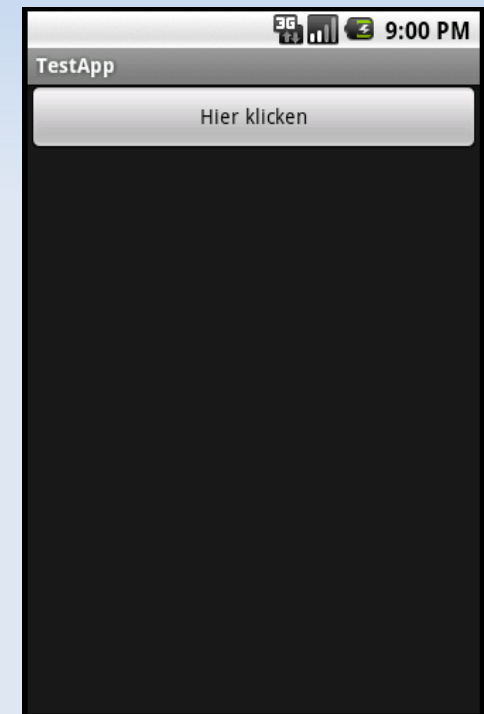
- Unterverzeichnis res/
- Klasse R
 - Enthält ID's
- Bsp.: R.layout.main
- Werden beim kompilieren in das Application Package kopiert



GUI

- Prozedurales Design (Java)
- Deklaratives Design (XML)
- Beispiel: (Prozedural)

```
1: public class MyActivity extends Activity {  
2:     private Button button;  
3:     private LinearLayout ll;  
4:     public void onCreate(Bundle savedInstanceState) {  
5:         super.onCreate(savedInstanceState);  
6:         ll = new LinearLayout(this);  
7:         ll.setOrientation(LinearLayout.VERTICAL);  
8:         button = new Button(this);  
9:         button.setText(„Hier klicken“);  
10:        ll.addView(button);  
11:        setContentView(ll);  
12:    }
```



- Beispiel: (Deklarativ)

res/layout/main.xml:

```
1: <?xml version="1.0" encoding="utf-8"?>
2: <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3:     android:orientation="vertical">
4:     <Button
5:         android:id="@+id/mybutton"
6:         android:text="Hier klicken!" />
7: </LinearLayout>
```

- Beispiel: (Deklarativ)

```
1: public class MyActivity extends Activity {  
2:     private Button button;  
3:     public void onCreate(Bundle savedInstanceState) {  
4:         super.onCreate(savedInstanceState);  
5:         setContentView(R.layout.main);  
6:         button = (Button) findViewById(R.id.mybutton);  
7:     }  
8: }
```

Intents

- Activities werden vom Framework erstellt
- Indirektes Aufrufen über Intents
- Zwei Typen
 - Explizite Intents
 - (konkrete Activity)
 - Implizite Intents
 - (ruft Activity auf, die gewisse Anforderungen erfüllt)

Explizite Intents

- android.content.Intent

```
1: public class MyActivity extends Activity {
2:     private Button button;
3:     public void onCreate(Bundle savedInstanceState) {
4:         //... Initialisation
5:         button.setOnClickListener(new OnClickListener() {
6:             public void onClick(View v) {
7:                 Intent i = new Intent(this, OtherActivity.class);
8:                 startActivity(i);
9:             });
10: }
11: }
```


Grafik

- Wichtig für Spiele
- Erstellen einer Klasse, die von View erbt

```
1: public class MyActivity extends Activity {  
2:     public void onCreate(Bundle savedInstanceState) {  
3:         super.onCreate(savedInstanceState);  
4:         setContentView(new GraphicView(this));  
5:     }  
6: }
```

View

```
1: public class GraphicView extends View
2:     public GraphicView(Context context) {
3:         super(context);
4:     }
5:     @Override
6:     protected void onDraw(Canvas canvas) {
7:         Paint myPaint = new Paint();
8:         myPaint.setColor(Color.RED);
9:         //drawRect(left, top, right, bottom)
9:         canvas.drawRect(0,0, 100, 50
                        myPaint);
10:    }
11: }
```

- Basisklasse für GUI Komponenten
- Von View leiten z.B. ab: Button, LinearLayout
- onDraw wird aufgerufen, wenn der View neu gezeichnet werden muss

Paint

```
1: public class GraphicView extends View
2:     public GraphicView(Context context) {
3:         super(context);
4:     }
5:     @Override
6:     protected void onDraw(Canvas canvas) {
7:         Paint myPaint = new Paint();
8:         myPaint.setColor(Color.RED);
9:         //drawRect(left, top, right, bottom)
9:         canvas.drawRect(0,0, 100, 50
                        myPaint);
10:    }
11: }
```

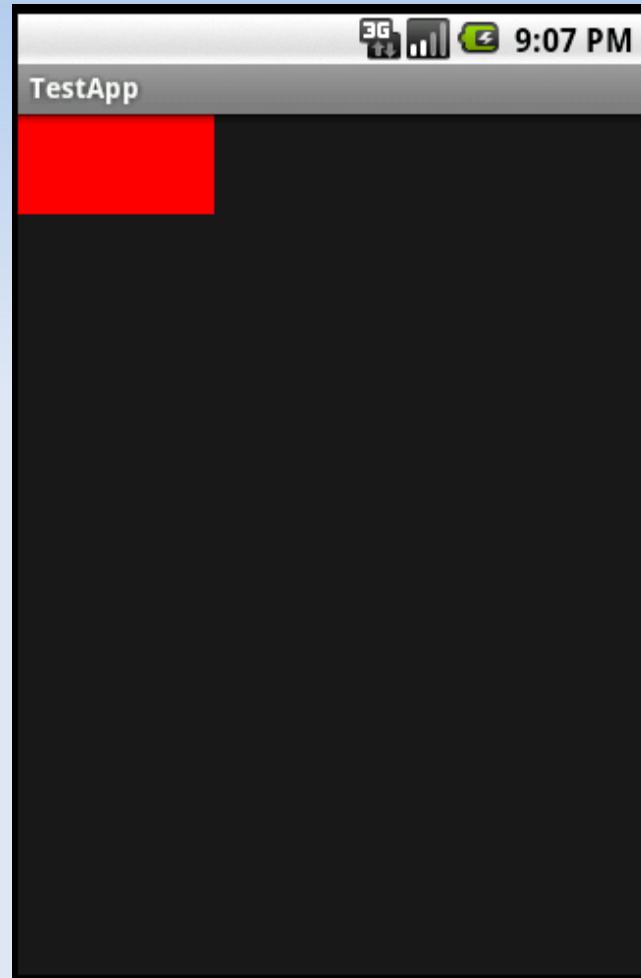
- Enthält Parameter für den Zeichenvorgang
- z.B.: Farbe, Antialiasing an/aus

Canvas

```
1: public class GraphicView extends View
2:     public GraphicView(Context context) {
3:         super(context);
4:     }
5:     @Override
6:     protected void onDraw(Canvas canvas) {
7:         Paint myPaint = new Paint();
8:         myPaint.setColor(Color.RED);
9:         //drawRect(left, top, right, bottom)
9:         canvas.drawRect(0,0, 100, 50,
                        myPaint);
10:    }
11: }
```

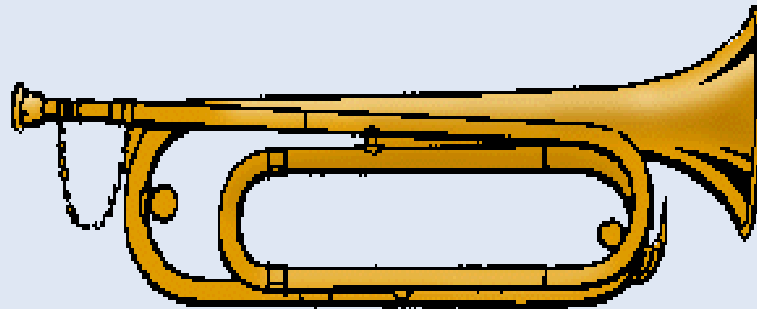
- Zeichenfläche
- Enthält Funktionen zum Zeichnen
- Braucht paint Objekt
- Bsp.:
 - drawRect
 - drawBitmap
 - drawText

Grafik



Sound

- MediaPlayer Klasse
- Unterstützt u.a. WAV, MP3, OGG



(Bild: www.musicgraphicsgalore.net)

Sound

```
1: public class MyActivity extends Activity {  
2:     MediaPlayer mp;  
3:     public void onCreate(Bundle savedInstanceState) {  
4:         super.onCreate(savedInstanceState);  
5:         mp = MediaPlayer.create(this, R.raw.mysound);  
6:         mp.setLooping(true);  
6:     }  
7:     protected void onPause() {  
8:         mp.pause();  
9:     }  
10:    protected void onResume() {  
11:        mp.start();  
12:    }
```

Lokalisierung

- Eingebaute Unterstützung für GPS / Location
- LocationManager
- Interface LocationListener
 - Callback Funktionen, z.B. onLocationChanged



(Bild: Wikipedia)

Lokalisierung

```
1: public class MyActivity extends Activity implements LocationListener{
2:     private LocationManager Imgr;
3:     private final long MIN_TIME = 1000; //in milliseconds
4:     private final float MIN_DISTANCE = 1; //in meters
5:     public void onCreate(Bundle savedInstanceState) {
6:         super.onCreate(savedInstanceState);
7:         Imgr = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
8:         Imgr.requestLocationUpdates(LocationManager.GPS_PROVIDER,
           MIN_TIME, MIN_DISTANCE, this);
9:     }
10:    public void onLocationChanged(Location location) {...}
11:    public void onProviderDisabled(String provider) {...}
12:    public void onProviderEnabled(String provider) {...}
13:    public void onStatusChanged(String provider, int status, Bundle extras) {...}
14: }
```

Sensoren

- Unterstützung für Sensoren
 - Orientierung, Licht, Beschleunigung, Nähe, Druck, Temperatur
- SensorManager
- SensorEventListener

Bsp.: Orientation

```
1: class MyActivity extends Activity implements SensorEventListener {  
2:     private SensorManager mgr;  
  
3:     public void onCreate(Bundle savedInstanceState) {  
4:         super.onCreate(savedInstanceState);  
5:         mgr = (SensorManager) getSystemService(SENSOR_SERVICE);  
6:         mgr.registerListener(this, Sensor.TYPE_ORIENTATION,  
           SensorManager.SENSOR_DELAY_FASTEST);  
7:     }  
  
8:     public void onAccuracyChanged(Sensor sensor, int accuracy) {}  
  
9:     public void onSensorChanged(SensorEvent event) {  
10:         ...  
11:     }  
12: }
```

Literatur

- Ed Burnette: Hello, Android, 2008
- <http://developer.android.com/>