

Android-Apps

Zeichnen auf der Canvas

Canvas-Demo



I. Zeichnen und Schreiben auf der Canvas ohne Verwendung weiterer Views

Zeichnen und Schreiben

Java - canvasDemo/src/eh/beispiele/zeichnen/zeichnen.java - Eclipse Platform

File Edit Refactor Run Source Navigate Search Project Window Help

Package Explorer: amando, BigBrother-Master, canvasDemo (src: eh.beispiele.zeichnen, zeichnen.java; gen [Generated Java Files]; Android 2.1-update1; assets; res: drawable-hdpi, drawable-ldpi, drawable-mdpi, layout, values; AndroidManifest.xml)

zeichnen.java

```

package eh.beispiele.zeichnen;

import android.app.Activity;

public class zeichnen extends Activity {
    DrawView drawView;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        //setContentView(R.layout.main); wird nicht benötigt!
        setContentView(new DrawView(this));
    }

    final class DrawView extends View {
        Paint paintSun = new Paint();
        Paint paintStar = new Paint();
        Paint paintText = new Paint();

        public DrawView(Context context) {
            super(context);

            paintSun.setColor(Color.YELLOW);
            // paintSun.setStrokeWidth(5); // Liniendicke
            // paintSun.setAntiAlias(true);
            // paintSun.setStyle(Style.FILL_AND_STROKE);
        }
    }
  
```

Neues Projekt anlegen

res/layout/main.xml kann gelöscht werden!

DrawView als einzige View , main.xml überflüssig

Zeichnen und Schreiben

```
final class DrawView extends View {  
    Paint paintSun = new Paint();           eine Art Zeichenstift  
    Paint paintStar = new Paint();  
    Paint paintText = new Paint();  
  
    public DrawView(Context context) {  
        super(context);  
        paintSun.setColor(Color.YELLOW);  
        // paintSun.setStrokeWidth(5); // Liniendicke  
        // paintSun.setAntiAlias(true);  
        // paintSun.setStyle(Style.FILL_AND_STROKE);  
        paintStar.setColor(Color.WHITE);  
        paintText.setColor(Color.MAGENTA);  
        //paintText.setTextScaleX(2.0f);  
        paintText.setTextSize(24);           Textattribute  
        paintText.setUnderlineText(true);  
    }  
}
```

Kann mit
verschiedensten
Stilmitteln
versehen
werden

Zeichnen und Schreiben

```
final class DrawView extends View {
    Paint paintSun = new Paint();
    Paint paintStar = new Paint();
    Paint paintText = new Paint();
```

```
public DrawView(Context context) {
```

```
    ~~~~
```

```
@Override
```

Überschriebene Methode der Klasse DrawView!

```
protected void onDraw(Canvas canvas) {
```

```
    super.onDraw(canvas);
```

```
    float x1,y1,x2,y2;
```

```
    x1 = 100;
```

```
    y1 = 100;
```

```
    canvas.drawCircle(x1, y1, 15, paintSun); Kreis
```

```
    for (int i = 0; i < 18; i++) {
```

```
        x2 = (float) (x1 + 50 * Math.sin( (float) i /18 * 2*Math.PI));
```

```
        y2 = (float) (y1 + 50 * Math.cos( (float) i /18 * 2*Math.PI));
```

```
        canvas.drawLine(x1,y1,x2,y2,paintSun); Linie
```

```
    }
```

```
    Path path = new Path(); geschlossenenes Polygon
```

```
    path.moveTo(0,10); path.lineTo(-2,2); path.lineTo(-10,0); path.lineTo(-2,-2);
```

```
        path.lineTo(0,-10); path.lineTo(2,-2); path.lineTo(10,0); path.lineTo(2,2);
```

```
    path.close();
```

```
    path.offset(40, 40); canvas.drawPath(path, paintStar);
```

```
    path.offset(120, 40); canvas.drawPath(path, paintStar);
```

```
    path.offset(90, 100); canvas.drawPath(path, paintStar);
```

kann mehrfach gezeichnet werden, offset jeweils relativ zum letzten offset!

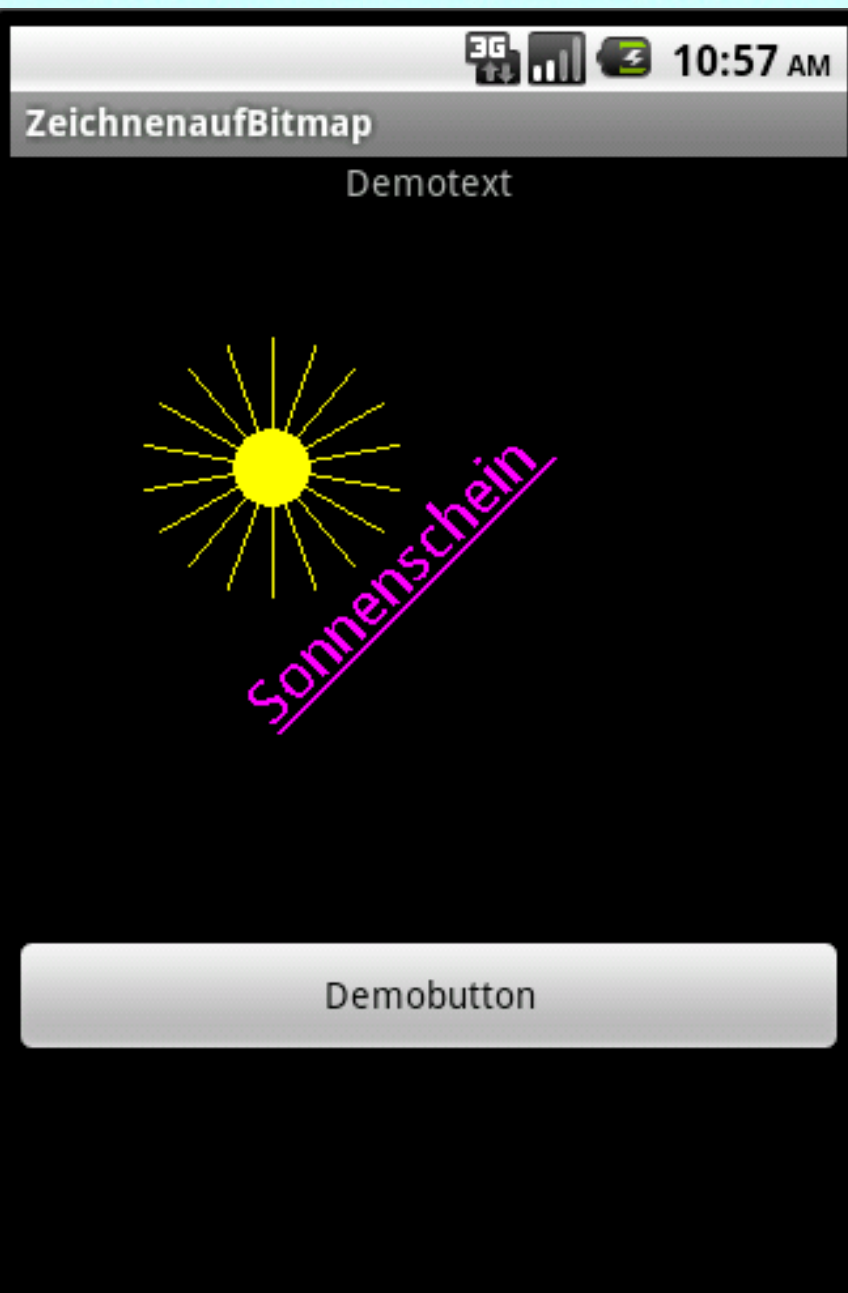
```
    canvas.drawText("Sonnenschein", 100, 200, paintText);
```

Textausgabe

```
    }
```

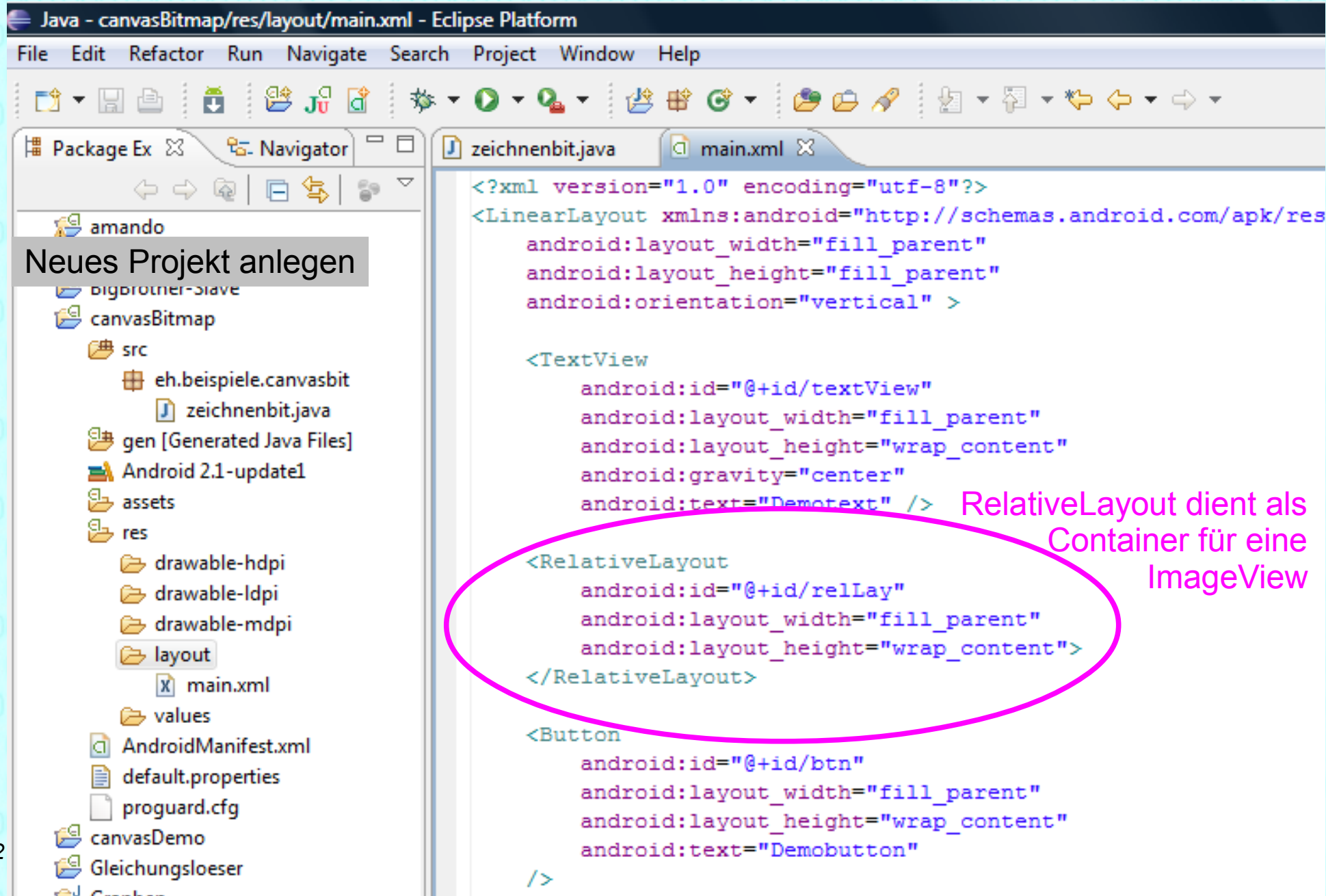
```
}
```

Canvas-Demo



II. Die Canvas
in Kombination
mit weiteren
Views

DrawView in Kombination



The screenshot shows the Eclipse IDE interface. On the left, the Package Explorer displays the project structure for 'amando'. The 'res' directory is expanded, showing 'layout/main.xml' selected. The main editor displays the XML code for 'main.xml'. The code defines a vertical LinearLayout containing a TextView, a RelativeLayout, and a Button. The RelativeLayout is circled in pink, and a pink annotation points to it, stating 'RelativeLayout dient als Container für eine ImageView'. The code is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res-
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/textView"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="Demotext" />
    <RelativeLayout
        android:id="@+id/relLay"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content">
    </RelativeLayout>
    <Button
        android:id="@+id/btn"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Demobutton"
    />
</LinearLayout>
```

RelativeLayout dient als
Container für eine
ImageView

DrawView in Kombination

```
zeichnenbit.java X main.xml
package eh.beispiele.canvasbit;

import android.app.Activity;

public class zeichnenbit extends Activity {

    private ImageView image;
    private RelativeLayout rl;
    private RelativeLayout.LayoutParams params;

    private Display display;
    private int screenHeight;
    private int screenWidth;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        // Bildschirmmaße
        display = getWindowManager().getDefaultDisplay();
        screenHeight = display.getHeight();
        screenWidth = display.getWidth();
    }
}
```

ImageView wird in RelativeLayout eingebunden

Breite und Höhe des
Bildschirms ermitteln

DrawView in Kombination

RelativeLayout finden und casten

```
// Bitmap ablegen
image = new ImageView(this);
rl = (RelativeLayout) findViewById(R.id.rellay);
params = new RelativeLayout.LayoutParams(screenWidth, screenHeight-200);
rl.addView(image, params);
```

Größe des ImageViews
definieren (so groß wie das
Display ohne die anderen Views)

Das ImageView dem RelativeLayout hinzufügen

```
// zeichnen auf dem Bitmap
Bitmap bitmap = Bitmap.createBitmap(screenWidth, screenHeight-200, Bitmap.Config.RGB 565);
Canvas canvas = new Canvas(bitmap);
```

Bitmap in der Größe des ImageViews erstellen

Zeichenflächen auf dem Bitmap

2 Byte pro Pixel
(rot 5Bit, grün 6Bit,
blau 5 Bit)

```
image.setImageBitmap(bitmap);
Paint paint = new Paint();
paint.setColor(Color.YELLOW);
float x1, y1, x2, y2;
x1 = 100;
y1 = 100;
canvas.drawCircle(x1, y1, 15, paint);
for (int i = 0; i < 18; i++) {
    x2 = (float) (x1 + 50 * Math.sin( (float) i /18 * 2*Math.PI));
    y2 = (float) (y1 + 50 * Math.cos( (float) i /18 * 2*Math.PI));
    canvas.drawLine(x1, y1, x2, y2, paint);
}
```

Bitmap auf dem ImageView platzieren

```
Paint paintText = new Paint();
paintText.setColor(Color.MAGENTA);
paintText.setTextSize(24);
paintText.setUnderlineText(true);
canvas.rotate(-45, 100, 200); // Drehwinkel in Grad, Koordinaten des Drehpunktes
```

DrawView in Kombination

```

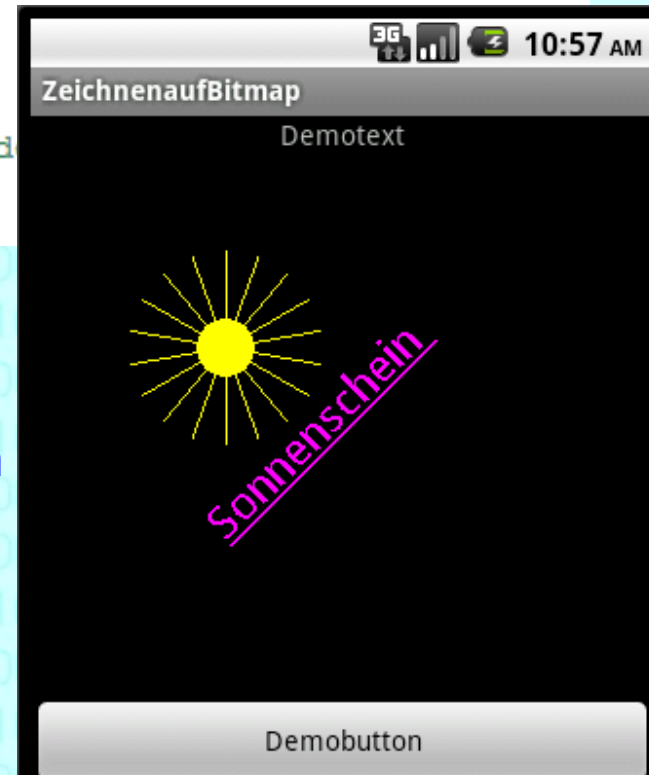
        canvas.drawLine(x1, y1, x2, y2, paint);
    }
    Paint paintText = new Paint();
    paintText.setColor(Color.MAGENTA);
    paintText.setTextSize(24);
    paintText.setUnderlineText(true);
    canvas.rotate(-45, 100, 200); // Drehwinkel in Grad, Koordinaten des Drehpunktes
    canvas.drawText("Sonnenschein", 100, 200, paintText);

    // oder Bitmap aus res/drawable laden
    // image.setImageResource(R.drawable.demo); // lädt d
}

```

Canvas.rotate(winkel)
dreht um linke obere Ecke

Alternativ kann auch eine vorhandene Grafik aus den Ressourcen geladen werden.
grafik.png oder grafik.jpg in res/drawable-hdpi kopieren (oder -mdpi oder ... egal, wird automatisch gefunden) und mit .setImageResource(R.drawable.grafik) einbinden



Übung

I

N

F

O

R

M

A

T

I

K

Erstellen Sie eine App, die eine Parabel zeichnet

- Sie benötigen eine Eingabe-Activity
- Die Parabel sollte in einer eigenen Activity gezeichnet werden
- Der Zoom könnte über ein Optionsmenü gesteuert werden (Version 2!)