

# Dingen die je zou kunnen willen (met Android)

Challenges in Computer Science Seminar  
Marijn Schraagen  
12-03-2013

# Sensoren: algemeen

- class implements `SensorEventListener`
- In class: `private SensorManager m`  
en `private Sensor s1, s2, ...`
- In `onCreate`: sensormanager initialiseren
- `m = (SensorManager) getSystemService(Context.SENSOR_SERVICE);`
- In `onResume`: sensors toewijzen en Listeners instellen
- `s1 = m.getDefaultSensor(Sensor.TYPE_X);`
- `m.registerListener((SensorEventListener)this,s1,SensorManager.SENSOR_DELAY_X);`
- In `onPause`: Listener uitzetten (batterij)
- `m.unregisterListener(this);`
- dat is alles

# Sensoren: uitlezen

- `public final void onSensorChanged(SensorEvent e)`
- Voor alle sensoren die geregistreerd staan tegelijkertijd
- Schrijf per sensor een if-statement (of case), bv.  
`if(e.sensor.equals(m.getDefaultSensor(Sensor.TYPE_X)))`
- waarden uitlezen met `e.values[n]`
- verplicht: `public void onAccuracyChanged`

# Sensoren: afhandeling

- Niet verstandig om de UI te updaten vanuit `onSensorChanged`
- Teveel updates
- Beter: update variabelen en gebruik die in gewone UI-updates met knoppen of `Timer/postDelayed` (zie verder)

# Sensoren: licht

- `SENSOR_TYPE_LIGHT`
- Een losse waarde: `e.values[0]`
- Waardes van 0 .. >2500 (daglicht)
- Binnen: 0 .. ~600

# Sensoren: accelerometer

- `SENSOR_TYPE_ACCELEROMETER`
- Drie waarden: x, y, z
- Bereik: -10..10
- Gebruik voor 2d-beweging alleen x en y

# Niet echt een sensor: touch

- `setOnTouchListener` op een View
- Definieer in-place
- Methods van eigen class mogen worden gebruikt
- Typecasten voor methods van andere classes

# onTouch

- Net als onClick
- `MijnView.setOnTouchListener(new OnTouchListener(){  
 @Override  
 public boolean onTouch(View v, MotionEvent e){  
 float tx = e.getX(); //ook: e.getY();  
 ((MijnViewClass) v).methode(tx);  
 return(true);  
 }  
});`
- Vergeet de ; niet



# Afbeeldingen aanpassen

- Kleuren, belichting, alpha (transparency):
- `LightingColorFilter`
- Multiplication en addition van alpha, rgb
- `int mul = Color.rgb(100,255,0);`
- `int add = 0x000000;`
- `LightingColorFilter lcf = new LightingColorFilter(mul,add);`
- `ImageView.setColorFilter(lcf);`

# Afbeeldingen aanpassen

- Kopieer afbeeldingsbestand naar res/drawable
- Een of meerdere versies
- `ImageView a1 = (ImageView) findViewById(R.id.afbeeldingsid);`
- `A1.setImageResource(R.drawable.bestands naam);`
- Geen hoofdletters, geen extensie

# Canvas

- Maak eigen klasse, extends View
- Voeg deze klasse toe in je xml
- `<nl.liacs.project.EigenView  
android:layout_width="800dp"`
- In klasse: `onDraw(Canvas c)`
- `c.drawVanAlles`
- In main-klasse:
- `Private EigenView v1 = (EigenView) findViewById`
- `v1.eigenMethodes(...)`
- `v1.invalidate(); //verzoek onDraw`

# Herhaald uitvoeren

- extends TimerTask
- Of: Handler.postDelayed

# Voorbeeld-app

- <https://svn.liacs.nl/ccss/2013/team0/trunk/SensorTest/>

