



Mobile Technologies & Smart Devices

2020~2021

overtref jezelf



Huiswerk

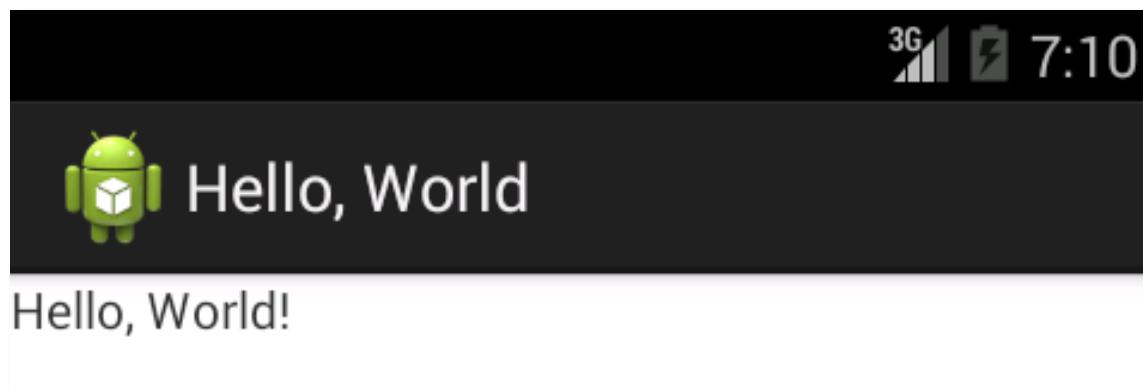
- ▶ Wat is een lifecycle

Mobile development

- ▶ Theorie mobile development a.d.h.v. Android
- ▶ Tentamen bevat:
 - ▶ Object Orientatie vragen
 - ▶ Algemene mobile development vragen
 - ▶ Platformspecifieke development vragen
- ▶ Platformspecifieke vragen op het tentamen gaan over Android
- ▶ Praktijkopdracht mag in iOS of ander framework dat native compileert, maar kennis Android wordt van iedereen verwacht!

Android Studio

- ▶ Ontwikkelomgeving
- ▶ SDK
- ▶ Emulator
- ▶ Management software



Mobile development

Mobile device \neq Desktop computer

- ▶ Kleiner
- ▶ Minder performance
- ▶ Minder energie
- ▶ Gebruikt in andere context
- ▶ Meer sensoren

Mobile development

Resources

- ▶ Één programma tegelijk / Full screen
- ▶ Zuinig met resources => batterij moet een dag meegaan

User / Context

- ▶ Één gebruiker / Altijd dezelfde gebruiker
- ▶ Mobiel (niet altijd zelfde plek / meer afleiding dan niet mobiel)
- ▶ Stopt onverwacht / gaat iets anders doen

Device

- ▶ Veel sensoren
- ▶ GPS, maar niet altijd
- ▶ Internet, maar niet altijd

Android

<https://source.android.com/security/>

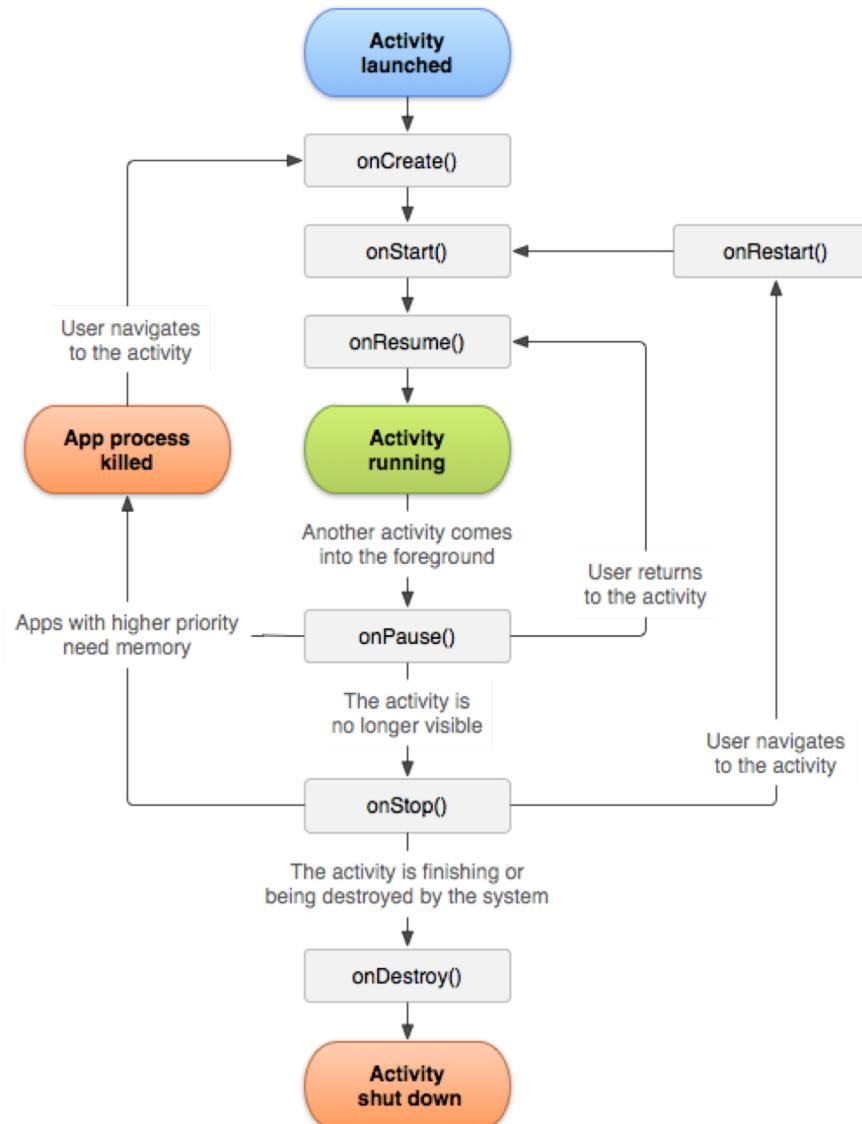


- ▶ Linux: het “hart van Android”
- ▶ Native Libraries: snel, in c/c++ geschreven
- ▶ Android Runtime: voor minimaal geheugengebruik
- ▶ Application Framework: Hier praten we mee
- ▶ Applications and Widgets: Op dat niveau ook onze app!

Android framework onderdelen

- ▶ Activity
 - ▶ user interaction
 - ▶ single screen

Android Activity life cycle



<https://developer.android.com/guide/components/activities/activity-lifecycle>

Resources

- ▶ In map /res staan de (niet java) resources
 - ▶ Meeste is xml
 - ▶ Gecompileerd tot R(.java)
 - ▶ In code via R benaderbaar
-
- ▶ Layouts
 - ▶ Strings

XML Layout

- ▶ ConstraintLayout (makkelijker met meerdere formaten)
 - ▶ FrameLayout (links boven)
 - ▶ LineairLayout (1 kolom)
 - ▶ RelativeLayout (ten opzichte van elkaar) – meest gebruikt
 - ▶ TableLayout
-
- ▶ Margins en andere vormgeving allemaal in xml
 - ▶ px = pixels
 - ▶ dp of dip = density independent pixels (160dpi)

XML Resources

- ▶ Strings.xml

Interactie: output

- ▶ Views
- ▶ Alles wat je kunt zien is een view in je resources (polymorfisme!)
- ▶ Ophalen uit resources en aanpassen via methods

Vb. TextView

- ▶ Geef een id, en
- ▶ Gebruik findViewById om op te zoeken in layout
- ▶ Cast view naar een textView
- ▶ setText()

```
hello = (TextView) findViewById(R.id.hello);  
hello.setText("Hello World");
```

Interactie: input

- ▶ Events
- ▶ Buttons (en alle andere user input) werkten via listeners
- ▶ Implementeer de juiste listener (interface!)
- ▶ Reactie in hook-method (callback)

Vb. Button

- ▶ In layout eenvoudig toe te voegen
 - ▶ Geef naam (id) voor referentie
 - ▶ Koppel XML met method
(Android studio genereert de code voor je)
- OF
- ▶ Gebruik findViewById om op te zoeken in layout en voeg OnClickListener toe (Button is ook een view: polymorfisme)
 - ▶ Implementeer de interface OnClickListener om de click af te handelen

```
Button testButton = (Button)findViewById(R.id.button_test);  
testButton.setOnClickListener(this);
```

```
@Override  
public void onClick(View v) {  
    // TODO Auto-generated method stub  
    toast.show();  
}
```

Debugging

▶ Logging

Log.e(): Errors

Log.w(): Warnings

Log.i(): Information

Log.d(): Debugging

Log.v(): Verbose

Log.wtf(): What a Terrible Failure

Debugging: Toast

- ▶ Toast messages zijn eenvoudige pop-ups
- ▶ Bedoeld voor snel bericht naar gebruiker
- ▶ Ook heel geschikt voor debugging omdat je ze niet in je layout hoeft op te nemen

```
Toast toast = Toast.makeText(this, "Settings saved",  
Toast.LENGTH_SHORT);  
toast.show();
```

Een tweede scherm

- ▶ Is gewoon een nieuwe activity
- ▶ = Nieuwe Activity (starten via Intent)
- ▶ Manifest, activity moet bekend zijn om er een intent naar te kunnen sturen
- ▶ Nieuwe activities komen bovenop de stack (lifecycle!)
- ▶ (Android studio kan dit allemaal voor je doen)

```
Intent i = new Intent(MainActivity.this, NextActivity.class);
// Launch the Activity using the intent
startActivity(i);
```

Activity sluiten

- ▶ Activity sluiten, finish()
- ▶ Activity wordt van de stack gehaald (lifecycle!)
- ▶ Na sluiten van main activity, sluit je app

Workshop

► Hello World / Hello Android

- ▶ IDE + Emulator
- ▶ Layout
- ▶ Resources (strings)
- ▶ Button
- ▶ 2e activity

Workshop: Emulator

- ▶ Emulatie van de hardware van een Android Device
- ▶ Nadeel: traag

Tips:

- ▶ Nooit afsluiten
- ▶ HAX installeren en Intel image gebruiken
- ▶ Gebruik echt device

NB. Xcode werkt met een simulator

Afbeeldingsbronnen

→ <https://mobilebcitcomputing.wordpress.com/2015/05/23/hello-world/>