NTNU
Kunnskap for en bedre verden

Aktiviteter og intensjoner

Mildrid Ljosland, Institutt for datateknologi og informatikk (IDI), NTNU
Lærestoffet er utviklet for faget IFUD1042 Applikasjonsutvikling for Android

Resymé: Tema for denne leksjonen er lærebokas kapittel 2og 3. Vi ser på to grunnleggende begreper, aktivitet (activity) og intensjon (intent) samt noen grunnleggende UI-komponenter.

2.1 Aktivitet

En Android-applikasjon kan bestå av null eller flere aktiviteter, vanligvis minst en. En aktivitet er et vindu som inneholder et brukergrensesnitt til applikasjonen. Hvis vi ønsker flere forskjellige brukergrensesnitt, for eksempel et startbilde, og deretter et eller flere grensesnitt som bruker kan bevege seg mellom, lager vi flere aktiviteter (eller fragmenter – det skal vi komme tilbake til i en senere leksjon) og sørger for at de ulike aktivitetene skifter på å vises fram på skjermen.

En aktivitet lages ved å lage en Java-klasse som er en avledet klasse fra klassen Activity, det vil si at vi skriver

```
class minAktivitet extends Activity {

….
```

Android Studio (i hvert fall den versjonen jeg bruker) lager følgende start på java-filen:

```
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;


public class MainActivity extends ActionBarActivity {
```

Jeg ønsker ikke å bruke ActionBar, og endrer til extends Activity, og i stedet for

```
import android.support.v7.app.ActionBarActivity;
```

skriver jeg

```
import android.app.Activity;
```

Klassen Activity har diverse metoder som vi kan omdefinere i vår klasse. Den som iallfall må omdefineres, er onCreate(). Den som lages automatisk for oss ser slik ut:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}
```

(Husk å sette på @Override hver gang du skal omdefinere en metode. Da vil kompilatoren sjekke at metodehodet stemmer, og du risikerer ikke at du ved et uhell definerer en helt ny metode istedenfor å omdefinere en allerede eksisterende. Det er nemlig de metodene som allerede er definert, og som eventuelt er omdefinert, som blir brukt i diverse call-back-situasjoner. Har du laget en annen metode, selv om den kanskje har samme navn, vil den ikke bli brukt, og programmet fungerer ikke slik du har tenkt.)

Andre metoder som vi kan omdefinere er onStart(), onResume(), onPause(), onStop(), onDestroy(), onRestart(). Dette er metoder som kalles underveis i livsløpet til aktiviteten. Studer figur 2-1 i læreboka der dette livsløpet er beskrevet. I ActivityDemo vises hvordan man kan legge inn utskrifter i de ulike metodene, slik at man kan se når de blir kalt. Eksperimenter med dette! Prøv også ut LogCat som forklart i tilknytning til figur 2.2
Merk at alle aktiviteter må registreres i AndroidManifest.xml. Der skal hver aktivitet ha sitt <activity>-element.

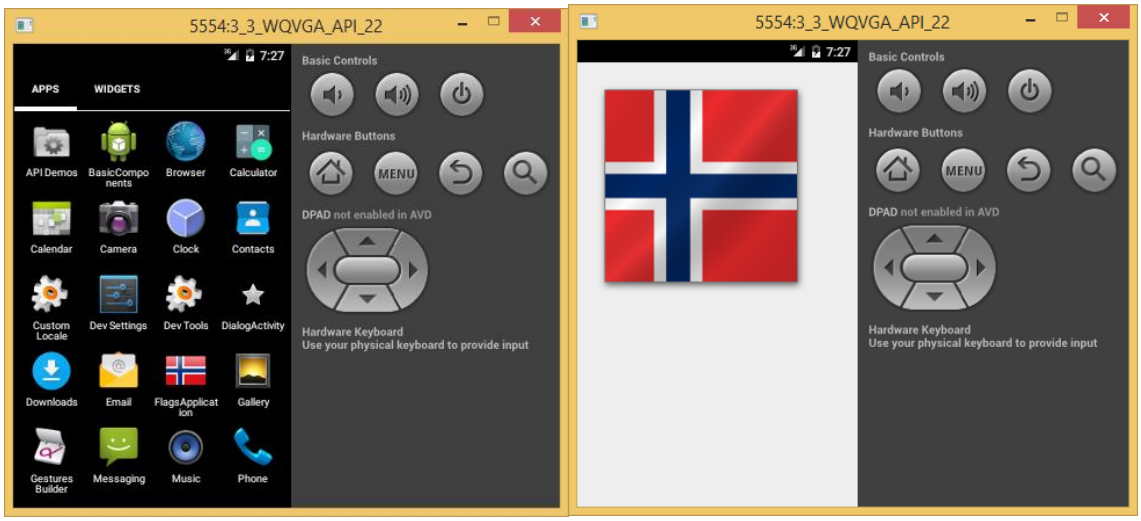
Eksempel

Vi skal lage en applikasjon som viser fram et flagg.

Først definerer jeg et nytt prosjekt FlagsApplication med aktiviteten NorwayActivity og plasserer tre png-bilder i en av drawable-katalogene (likegyldig hvilken). Det ene bildet, som jeg har kalt icon.png, er et norsk flagg i liten størrelse, egnet som ikon for applikasjonen. Deretter har jeg to større bilder, et av norsk flagg (norway.png) og et av det britiske flagget (uk.png). Alle flaggbildene har jeg funnet på Internett, og det er lett å finne mange flere. (Husk at filnavn ikke kan inneholde store bokstaver, bindestrek og diverse andre tegn, så filnavnene må vanligvis endres.)

Deretter fjerner jeg det TextView-elementet som lages automatisk i activity_norway og legger i stedet inn et ImageView (ved i design-modus å dra inn i bildet av skjermen en ImageView som finnes under Widgets). Når bildet er valgt, kan jeg lete fram src i Properties-vienduet, og ved å klikke på det tomme feltet, får jeg opp en boks der jeg kan bla meg gjennom Drawable-katalogen og finne norway.png. Til slutt endrer jeg ikonet for i manifestfila til icon og label til title_activity. Title_activity definerer jeg i strings.xml (kan gjøres enten direkte i xml-filen eller ved å åpne editoren).

Da gjenstår bare å kjøre applikasjonen. Vi ser at FlagsApplication har fått det norske flagget som icon, og vi får vist det fram



Neste trinn er å lage en tilsvarende applikasjon med det britiske flagget. Jeg kopierer activity_norway.xml og kaller kopien activity_uk.xml. Bildet endrer jeg til uk.png.

Tilsvarende kopierer jeg NorwayActivity.java og endrer navnet på kopien til UKActivity.java. Klassenavnet endrer seg automatisk. setContentView settes til R.layout.activity_uk. UKActivity må registreres i manifest-filen ved å legge den til i Application-fanen.. Manifest-fila ser nå slik ut:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="no.hist.itfag.flagsapplication" >

    <application
        android:allowBackup="true"
        android:icon="@drawable/icon"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
```

```
            android:name=".NorwayActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER"
/>
            </intent-filter>
        </activity>
        <activity android:name=".UKActivity"
            android:label="@string/title_activity"
            android:icon="@drawable/icon">
        </activity>
    </application>

</manifest>
```

For å få vist fram det britiske flagget må vi flytte intent-filteret fra NorwayActivity til UKActivity i manifest-filen.

2.2 Intensjon

Nest etter Activity, er Intent den viktigste komponenten i en Android-app. Intent er en budbringer mellom ulike aktiviteter, intert i en applikasjon eller mellom ulike applikasjoner. Den kan gi beskjed om at en annen aktivitet skal startes, sende med nødvendige data og returnere data fra denne aktiviteten. Vi har allerede møtt den i manifest-fila, i form av et

intent-filter med en action og en category. Action forteller hvilken aktivitet som skal startes, mens kategoriene grupperer de eksisterende intensjoner i logiske grupper. I det foregående eksesmplet sørger manifest-filen for at intensjon android.intent.action.MAIN starter vår NorwayActivity.

Når vi skal bruke intensjoner, lager vi et objekt av klassen Intent. Der spesifiserer vi navnet på den aktiviteten vi ønsker å starte, og kaller meoden startActivity(), for eksempel slik:

```
startActivity(new Intent("no.hist.itfag.UKActivity"));
```

I tillegg må vi legge den inn i Manifest-fila, f.eks. slik:

```
<activity
    android:name=".UKActivity"
    android:label="@string/title_activity" >
    <intent-filter>
        <action android:name="no.hist.itfag.UKActivity" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>

</activity>
```

Når den har kategori DEFAULT, kan den tas i bruk av andre aktiviteter, både intert i den samme applikasjonen, og fra andre applikasjoner.

StartActivity

I vårt eksempel er det nokså uhensiktsmessig å måtte gå inn i manifest-filen for å få vist fram det britiske flagget. Det ville vært mye kjekkere om bruker kunne styrt det fra applikasjonen. Det kan vi få til ved å legge inn en knapp. Inni klassen NorwayActivity legger vi metoden:

```
public void onClick(View v) {
  startActivity(new Intent("no.hist.itfag.UKActivity"));
  finish();
}
```

Denne metoden definerer hva som skal hende når knappen trykkes. Her sier vi at vi skal starte en ny aktivitet, og hvilken aktivitet som skal startes, defineres i en Intent. Vi passer også på å avslutte den kjørende aktiviteten med finish(), slik at det ikke hoper seg opp med aktiviteter som ligger i bakgrunnen og tar opp ressurser.

I activity_norway.xml legger jeg inn en button. For den setter jeg properties onClick lik onClick og text lik @string/change_activity. Denne fila ser dermed slik ut:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
    android:layout_height="match_parent"
android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
tools:context=".NorwayActivity">

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/imageView"
        android:src="@drawable/norway" />
```

```
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/change_activity"
        android:id="@+id/button"
        android:layout_below="@+id/imageView"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="58dp"
        android:onClick="onClick" />

</RelativeLayout>
```

Etter at jeg har lagt inn intentfilter

```
            <intent-filter>
                <action android:name="no.hist.itfag.UKActivity" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
```

i UKActivity i manifest-fila, kan jeg kjøre applikasjonen.

Vi ønsker også å kunne skifte fra britisk til norsk flagg, og legger inn de samme tingene i UKActivity.java og activity_uk.xml som vi har i NorwayActivity og activity_norway, samt intentfilter DEFAULT for NorwayActivity.

Men her skal vi i tillegg legge inn muligheten av å skifte flagg uten å skifte aktivitet. ”Endre aktivitet” fører oss tilbake til NorwayActivity, mens ”Endre flagg” skifter flagg i UKActivity. Vi må altså ha to knapper i activity_uk.xml, og to metoder i UKActivity.java

Her er programkoden:

```
package no.hist.itfag.flagsapplication;

import android.content.Intent;
import android.os.Bundle;
import android.app.Activity;
import android.view.View;
import android.widget.ImageView;

public class UKActivity extends Activity {
    private int flagValue = R.drawable.uk;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_uk);
    }

     public void onClick1(View v) {
        startActivity(new Intent("no.hist.itfag.NorwayActivity"));
        finish();
    }
    public void onClick2(View v) {
        if (flagValue == R.drawable.norway){
            flagValue = R.drawable.uk;
        }
        else {
            flagValue = R.drawable.norway;
        }
        ImageView flag=(ImageView)findViewById(R.id.imageView);
        flag.setImageResource(flagValue);
    }
}
```

For å holde rede på hvilket flagg som er satt, har jeg innført objektvariabelen flagValue. Den viser i utgangspunktet til det britiske flagget, men endres hver gang vi velger ”Endre flagg”. Merk at dette er et heltall. Ta fram R.java, så ser du hele lista over mulige ressurser. Men det er bare de som er drawable som kan brukes i ImageView. Til slutt finner vi den grafiske komponenten som viser fram flagget, og endrer hvilket bilde som skal vises fram.

StartActivityForResult

Hvis vi vil at en aktivitet skal returnere en verdi til den aktiviteten som startet den, bruker vi metoden startActivityForResult(). Da må vi også definere metoden onActivityResult(). Den blir kalt når den aktiviteten vi startet, er ferdig. I denne metoden får vi tilbake resultatet i form av en ny Intent, samt at vi får med en resultCode som forteller om alt er OK eller ikke. Her er et eksempel (se fullstendig kode lenger ned):

```
…
startActivityForResult(new Intent("no.hist.itfag.UKActivity"), request_Code);

…
public void onActivityResult(int requestCode, int resultCode, Intent data) {
      if (requestCode == request_Code && resultCode==RESULT_OK) {
            flaggverdi = data.getIntExtra("flag", R.drawable.uk);

      …

      }

}
```

request_Code er et tall som sikrer at det er den riktige aktiviteten som behandles – vi kan ha startet flere ulike. Den returnerte verdien finnes i objetktet data, og kan ha ulike typer. Her har vi valgt å la det være en int. Uansett type har den et navn (her flag, og en verdi. For int-verdier (og andre tallverdier) kreves også en defaultverdi som brukes hvis ingenting er satt, her er defaultverdien R.drawable.uk.

I den aktiviteten som startes (her UKActivity), må dataene settes. Det kan gjøres slik:

```
Intent intent = new Intent();
intent.putExtra("flag", flaggverdi);
setResult(RESULT_OK, intent);
finish();
```

Vi navngir altså verdien og putter den inn i en intent. Så setter vi resultatet til OK og avslutter denne aktiviteten. Da gjenopptas den forrige og dens onActivityResult() kalles.

Tilbake til vårt eksempel:

Nå skal vi endre applikasjonen slik at NorwayActivity ber om et resultat fra UKActivity. Det betyr at vi istedenfor startActivity() bruker startActivityForResult() i NorwayActivity, og setter resultatet med putExtra() i UKActivity. Resultatet vi skal utveksle, er hvilket flagg UKActivity viser fram.

I NorwayActivity endrer vi onClick til følgende:

```
    public void onClick(View v) {
//    startActivity(new Intent("no.hist.itfag.UKActivity"));
//    finish();
      startActivityForResult(
                  new Intent("no.hist.itfag.UKActivity"), request_Code);
      }
```

Nå må vi droppe finish for å unngå at NorwayActivity avsluttes øyeblikkelig når UKActivity er ferdig, vil skal jo tilbake til den når UKActivity er ferdig. request_Code lar vi være en objektvariabel med verdien 1.

Så lager vi metoden onActivityResult(). Her lager vi det slik at NorwayActivity også viser det samme flagget som UKActivity viste. Her er metoden:

```
@Override
 public void onActivityResult(int requestCode, int resultCode, Intent data) {
      if (requestCode == request_Code && resultCode==RESULT_OK) {
            flagValue = data.getIntExtra("flag", R.drawable.uk);
            ImageView flag=(ImageView)findViewById(R.id.imageView1);
            flag.setImageResource(flagValue);
      }
 }
```

I UKActivity må vi sette resultatet. Da endrer vi onClick1() slik:

```
    public void onClick1(View v) {
//     startActivity(new Intent("no.hist.itfag.NorwayActivity"));
       Intent intent = new Intent();
       intent.putExtra("flag", flagValue);
       setResult(RESULT_OK, intent);
       finish();
}
```

Vi starter altså ikke NorwayActivity på nytt, vi går tilbake til den aktiviteten vi hadde fra før. Dette skjer automatisk når UKActivity avsluttes ved finish(). Og i NorwayActivity kalles onActivityResult, der intent sendes med.

Sende med opplysninger i en Intent

Til slutt skal vi endre applikasjonen slik at NorwayActivity sender med opplysning om hvilket flagg UKActivity skal vise fram. Det gjør vi ved å bruke putExtra() for intent før vi sender den av gårde.

```
    public void onClick(View v) {
 //    startActivity(new Intent("no.hist.itfag.UKActivity"));
 //    finish();
 //          startActivityForResult(new Intent("no.hist.itfag.UKActivity"),
request_Code);
       Intent intent = new Intent("no.hist.itfag.UKActivity");
       intent.putExtra("flag", flagValue);
       startActivityForResult(intent, request_Code);
       }
```

Så må UKActivity hente den fram i onCreate():

```
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.uk);
        flagValue = getIntent().getIntExtra("flag", flagValue);
       ImageView flag=(ImageView)findViewById(R.id.imageView1);
       flag.setImageResource(flagValue);
    }
```

Denne versjonen av applikasjonen finner du i pakket versjon sammen med denne leksjonen.

2.3 Noen grunnleggende UI-komponenter

TextView og andre enkle komponenter

Den aller enkleste UI-kmponenten, TextView, har du allerede støtt på flere ganger. Den brukes til å vise fram tekst til brukeren. En liknende, EditText, brukes når brukeren skal skrive noe. For å få tak i det brukeren har skrevet, må vi første hente fram EditText-objektet, deretter bruke metoden getText().

Klassen Button har du også møtt allerede, og så at vi kan forbinde knappen med en onClick-metode som blir utført nå knappen klikkes.

ImageButton fungerer akkurat som Button, men den vises fram som et bilde. Da må vi naturlig nok fortelle hvilket bilde som skal vises fram, det kan gjøres ved å skrive scr=…. inni xml-definisjonen av knappen.

ImageView ligner på TextView iogmed at den viser fram noe, men altså et bilde istedenfor tekst.

CheckBox og ToggleButton har begge to tilstander, valgt eller ikke valgt. Den eneste forskjellen på dem er hvordan de tegnes opp.

RadioButton har også to tilstander, valgt eller ikke valgt. Men her samler vi vanligvis flere slike knapper i en RadioGroup. Dette gjør at bare en i gruppa kan være valgt på ethvert tidpunkt.

For alle komponentene gjelder at vi bør gi dem en id i den tilhørende xml-filen. Da kan vi finne dem igjen i java-programmet ved å bruke findViewById(R.id.et_eller_annet). Da får vi ut koponenten som et View-objekt (som de er subklasser av). Vanligvis foretar vi en casting til akkurat riktig klasse, slik at vi kan bruke de metodene som er spesiell for denne klassen.

ProgressBar kan brukes til å gi bruker tilbakemelding om hvor langt en pågående oppgave er kommet. Dette er nyttig hvis oppgaven tar såpass lang tid at bruker kan begynne å lure på om noe har gått galt. Slike oppgaver bør gjøres i en egen tråd, noe vi behandler mer inngående i senere leksjoner.

For å velge dato og klokkeslett har vi klassene DatePicker og TimePicker. For TimePicker kan man velge mellom 12-timersklokke og 24-timersklokke. Disse kan settes ved å kalle metodene setIs24HourView(true) eller …(false).

Toast, AlertDialog og Notification

Toast er en enkel måte å gi tilbakemelding til brukeren på. Den statiske metoden makeText() lager et Toast-objekt, mens metoden show() viser den fram. Så for å få vist fram teksten «Dette er en test», kan du skrive `Toast.makeText(this, "Dette er en test", Toast.LENGTH_LONG).show();` eller `Toast.makeText(this, R.string.tekst, Toast.LENGTH_LONG).show();` hvis teksten er lagt inn i strings.xml. `this` brukes hvis du er i en metode i Activity-klassen, ellers må du bruke `getActivity()` eller `getApplicationContext()`, avhengig av hvilken klasse du er i.

AlertDialog er et vindu med en eller flere knapper som kommer opp på toppen av det vanlige vinduet. Der kommer det en meldig og for eksempel en OK-knapp som bruker trykker på for å få dialog-vinduet til å forsvinne igjen.

Først må vi framskaffe et Builder-objekt:

```
AlertDialog.Builder builder = new AlertDialog.Builder(this);
```

Så kan vi sette den teksten som skal vises: `builder.setTitle(tittel);`

Hvis vi vil ha en OK-knapp, kan den lages slik:

```
bilder.setPositiveButton("OK", new MinLytter());
```

`MinLytter` er en klasse som for eksempel kan lages slik:

```
class MinLytter implements DialogInterface.OnClickListener {
      public void onClick(DialogIntrerface dialog, int which) {
                  dialog.dismiss();
      }
}
```

Alle disse setningene kan slås sammen til en setning på følgende måte:

```
new AlertDialog.Builder(this).setTitle(tittel).
      setPositiveButton("OK", new DialogInterface.OnClickListener() {
            public void onClick(DialogIntrerface dialog, int which) {
                  dialog.dismiss();
            }
       }
      );
```

new AlertDialog.Builder() lager et nytt Builder-objekt. På dette objektet anvendes metoden setTitile(). Returverdien fra denne metoden er det samme Builder-objektet, så da kan vi bruke metoden setPositiveButton() på det samme objektet. Også denne metoden har Builder-objektet som returverdi. Vi kan fortsette å anvende andre metoder, for eksespel setNegativeButton() og setNeutralButton(), så lenge vi ønsker siden alle metodene har Builder-objektet som returverdi.

I neste leksjon får du høre mer om hvordan man bruker lyttere.

En annen type Builder er Notification.Builder. Med den kan vi lage en «notification», som er en melding som kommer i statuslinjen øverst på skjermen. Hvis man ønsker, kan man lage meldingen slik at et eller annet utføres når brukeren berører meldingen. Da brukes en Intent, nærmere bestemt en PendingIntent, altså en Intent som ligger klar til å bli utført en gang i framtiden. Se læreboka for nærmere detaljer om dette.