

# Diving into Android

---

By Jeroen Tietema

# Requirements

- Android SDK<sup>1</sup>
- Android Studio (or your IDE / editor of choice)
- Emulator (Genymotion) or a real device.

---

<sup>1</sup> See <https://developer.android.com>

# Structure of an Android project

```
<rootFolder>
  + app
    | + src
    |   \ main
    |     + assets
    |     + java
    |     + res
    |     \ AndroidManifest.xml
    |   \ build.gradle
  + gradle
  + build.gradle
  + gradlew
  + gradlew.bat
  \ settings.gradle
```

# Overview of Android components

- **Intents** : used to communicate actions or events
- **Activities** : a screen
- **Services** : a background process
- **BroadcastReceivers** : subscribing to broadcasts from android or other apps
- **ContentProvider** : share data across apps

# AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.myapp">

    <uses-permission android:name="android.permission.INTERNET" />

    <uses-sdk android:minSdkVersion="11" android:targetSdkVersion="21" />

    <application
        android:name=".MyApplication"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme">

        // application components go here

    </application>
</manifest>
```

# AndroidManifest.xml (cont.)

```
<activity
    android:name=".MyActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

# Activities

```
class MyActivity extends Activity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.my_layout);  
    }  
  
    @Override  
    protected void onSaveInstanceState(Bundle outState) {  
        super.onSaveInstanceState(outState);  
        // save your state  
    }  
}
```

# Life-cycle methods

- \* `onCreate(Bundle)`
  - \* `onStart()`
    - \* `onResume()`
    - \* `onPause()`
    - \* `onStop()`
  - \* `onDestroy()`

See javadoc of Activity

# Launching

```
Intent myIntent = new Intent(context, MyActivity.class);  
context.startActivity(myIntent);
```

# Stopping

```
activity.finish();
```

# XML Layouts

## src/main/res/layout/my\_layout.xml

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

    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />

    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>
```

# Working with your Views in Java code

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.my_layout);  
  
    TextView myText = (TextView) findViewById(R.id.text);  
    Button myButton = (Button) findViewById(R.id.button);  
  
    myText.setText("Hello world!");  
    myButton.setOnClickListener(this);  
}
```

# Widgets

- Button
- TextView
- ProgressBar
- ImageView
- Checkbox
- RadioButton

# Layouts

- LinearLayout
- RelativeLayout
- FrameLayout
- ViewPager
- AdapterView
- RecyclerView

# Resources

## **res**

- + **drawable**
- + **drawable-xhpdi**
- + **drawable-xxhdpi-land**
- + **drawable-nodpi**
- + **layout**
- + **layout-land**
- + **layout-sw600dp**
- + **values**
- + **values-nl**
- \ **values-v21**

# Density independent pixels

1dp = 1 pixel on a 160 dpi screen. This is called a mdpi screen.

mdpi = 1 pixel

hdpi = 1.5 pixels

xhdpi = 2 pixels

xxhdpi = 3 pixels

xxxhdpi = 4 pixels

# Using resources (drawables)

Given that you have a drawable `res/drawable/icon.png`

```
myImageView.setImageResource(R.drawable.icon);
```

Android will automatically pick the right resources from the most appropriate folder.

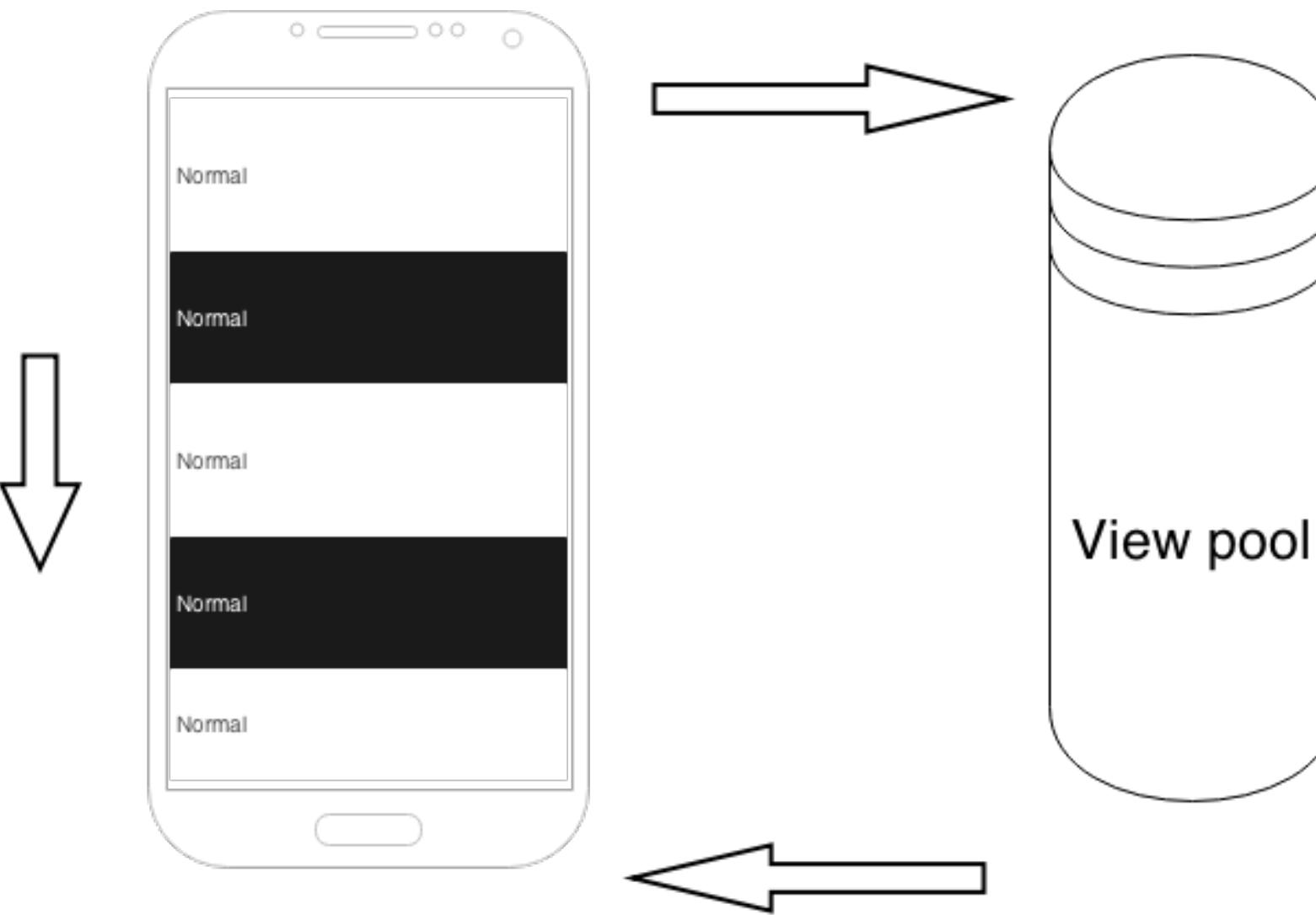
# Using resources (strings)

```
<resources>
    <string name="hello_world">Hello world! </string>
<string>

myText.setText(R.string.hello_world);
```

# ListViews and Adapters

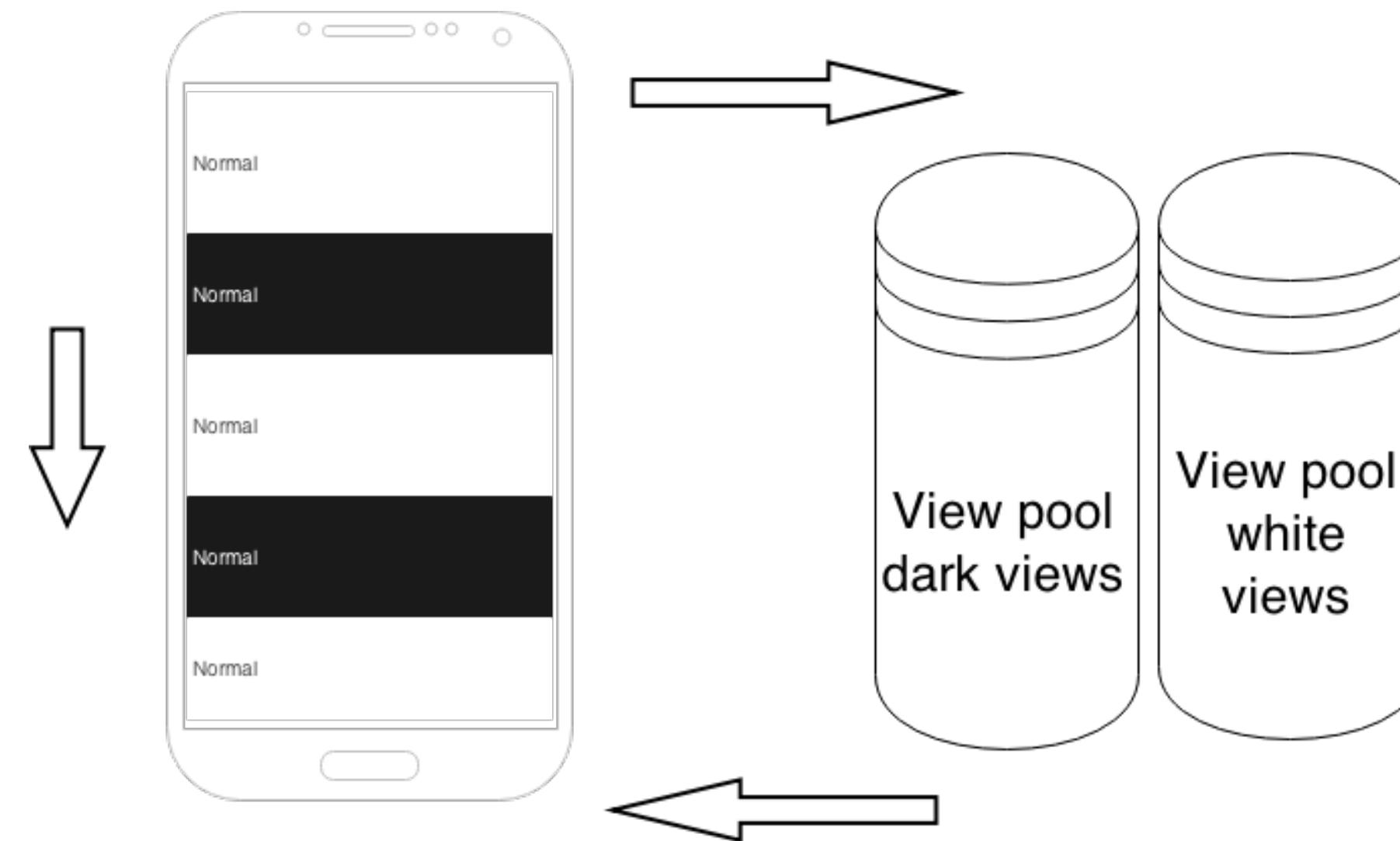
# View reuse



# BaseAdapter

```
class MyAdapter extends BaseAdapter {  
  
    public int getCount() {}  
  
    public Object getItem(int position) {}  
  
    public long getItemId(int position) {}  
  
    public View getView(int position, View convertView, ViewGroup parent) {}  
}
```

# Multiple view types



# Multiple view types

```
public int getViewTypeCount() {}
```

```
public int getViewType(int position) {}
```

Android will make sure you get back the correct View in  
your convertView.

# Concurrency

# The UI / Main Thread

- 60 fps == 16 ms per frame
- Blocking the UI thread for more than a couple of seconds pops the “This app is not responding” dialog
- No disk or network IO on the main thread
- Loading Bitmaps straight into right resolution minimizes GC

# Threading constructs

- Everything from Java (1.6)
- AsyncTask
- IntentService
- Handler

# AsyncTask

```
class MyTask extends AsyncTask<String, Void, Bitmap> {
    public void onPreExecute() {
        // called on UI thread
    }

    public Bitmap doInBackground(String... args) {
        // called in Background
    }

    public void onPostExecute(Bitmap result) {
        // called on UI thread
    }
}
```

# IntentService

```
class MyIntentService extends IntentService {  
  
    protected void onHandleIntent(Intent i) {  
        // this is on the background thread!  
    }  
  
}
```

# Handlers

Handlers are bound to the Thread you create them on (if you don't explicitly specify one).

Works only if the Thread has a Looper.

You can use them to:

- post message / runnables to a different thread
- schedule runnables in the future.

# Questions?

---

Just ask [jeroen@tietema.net](mailto:jeroen@tietema.net)