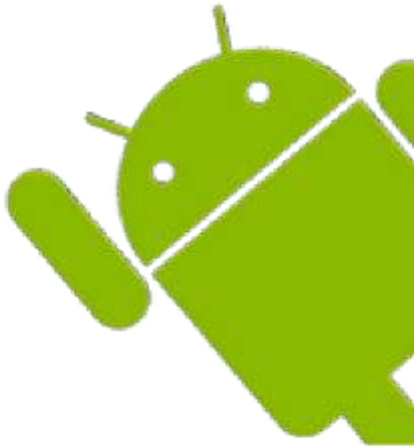
Mobile app development – week 9



Camera

Android components
- Services

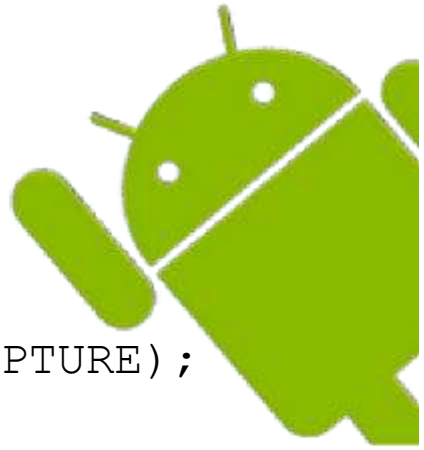
Concurrency

Second mandatory assignment

Course evaluation

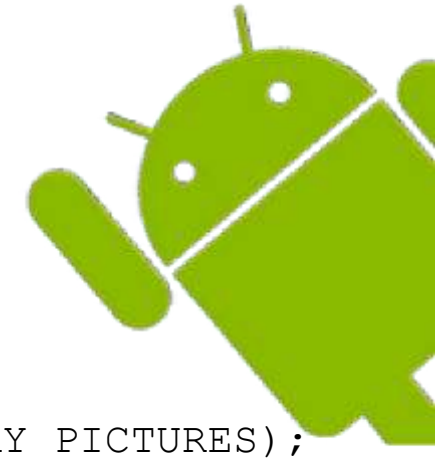
Exercises

Using the camera – textbook chapter 16



```
 private ImageView mPhotoView;
….. onCreate    {
  Intent captureImage=
              new Intent(MediaStore.ACTION_IMAGE_CAPTURE);

  Uri uri = Uri.fromFile(mPhotoFile);
  captureImage.putExtra(MediaStore.EXTRA_OUTPUT, uri);

  startActivityForResult(captureImage, REQUEST_PHOTO);
..
}

public void onActivityResult(int requestCode,
                             int resultCode, Intent data){
 if (resultCode == Activity.RESULT_OK) {
   Bitmap bitmap = PictureUtils
       .getScaledBitmap(mPhotoFile.getPath(), … );
   mPhotoView.setImageBitmap(bitmap);
 }
}
```

Handling images



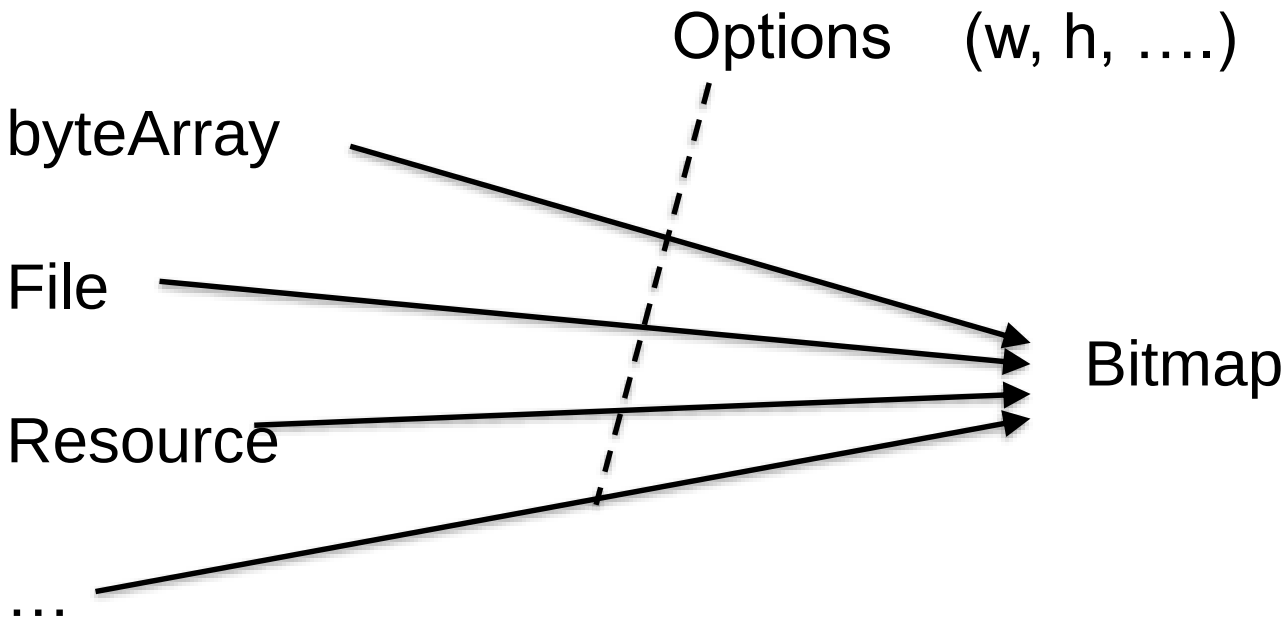
Files:

```
public File getPhotoFile(Thing thing) {
  File externalFilesDir = mContext
              .getExternalFilesDir(Environment.DIRECTORY_PICTURES);

  return new File(externalFilesDir, thing.getPhotoFilename());
}
```

Scaling

```
public class PictureUtils {   ….
  public static Bitmap getScaledBitmap(String path,
              int destWidth, int destHeight) {

  }
}
```

BitmapFactory

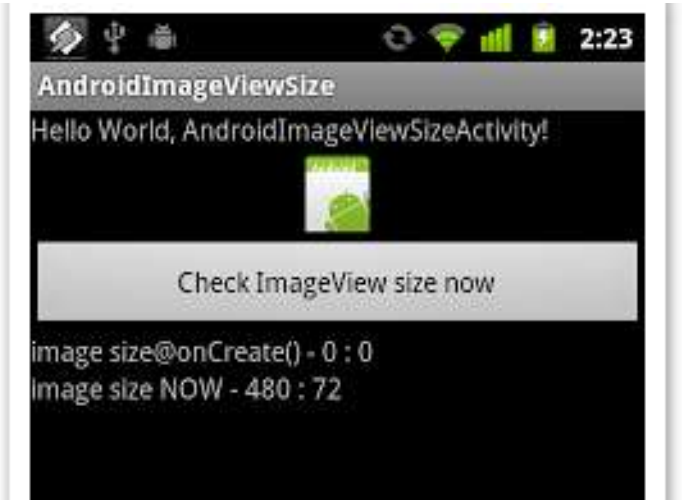


http://developer.android.com/training/displaying-bitmaps/load-bitmap.html

Beware of getWidth/getHeigth

Finding the dimensions of a View:

```
…. onCreate( …

image = (ImageView)findViewById(R.id.image);  // 72 px * 72 px

image.getWidth() ==
image.getHeight() ==
```

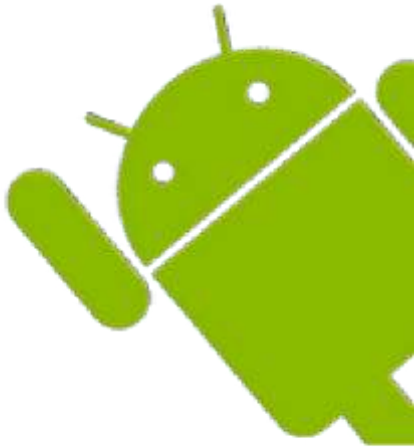


http://android-er.blogspot.dk/2011/07/get-width-and-height-of-imageview.html

http://www.sherif.mobi/2013/01/how-to-get-widthheight-of-view.html

Can you find a better link?

Android components



- **Activities**
- **Services**
- **Broadcast Receivers**
- **Content Providers**

2. Missing Network connection

```
ConnectivityManager connMgr = (ConnectivityManager)
                        getSystemService(Context.CONNECTIVITY_SERVICE);
```

http://developer.android.com/guide/components/fundamentals.html

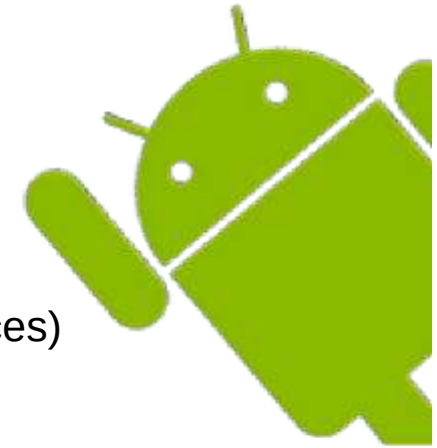
Synchronization service

SQLite

Server

```
public class DBSyncService extends IntentService {

      @Override
      public void onHandleIntent(Intent i){
        …
      }
  }
```

Concurrency - terminology



thread — Independent stream of execution (shared resources)

process — Linux process (own resources – not shared)

task — e.g. in AsyncTask which is a thread

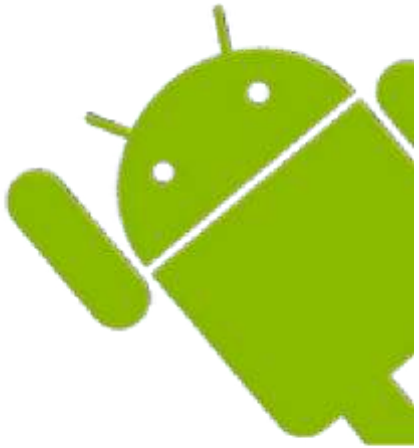
service — An activity without an UI.
All componets of an app run in the same process

processor (core) — Physical unit capable of executing code

…

http://developer.android.com/guide/components/processes-and-threads.html

Android threads - terminology



main thread: the thread started from the (Linux) process of the app

UI thread: other name for main thread

worker thread: thread started from another thread e.g. main

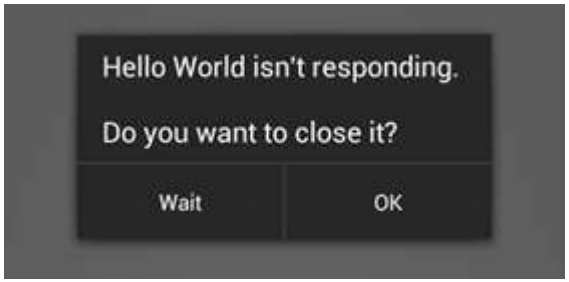
background thread: other name for worker thread

Android threading and the UI

Initially an application gets a separate process with running the **main thread**

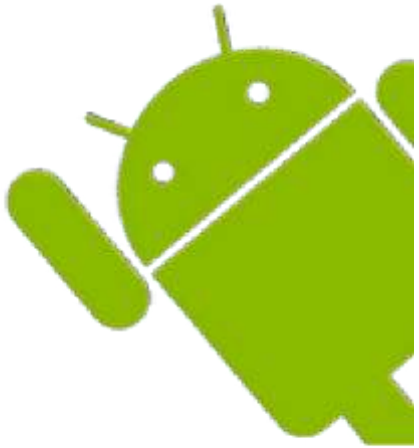
Only the main thread may access user interface e.g.:

- Updating a list view
- Adding views at runtime
- Displaying a toast message
- Response to onClick…



If an Android app is unresponsive for more than 5 seconds, an Application Not Responsive (ANR) dialog pops up,

runOnUIThread



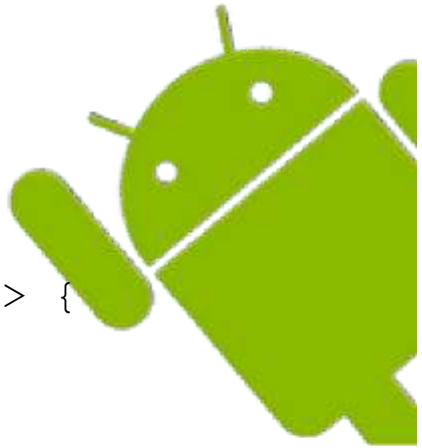
```
private void publishProgress( … ) {

  runOnUiThread(new Runnable() {
    @Override
    public void run() {
      updateResults(… ); // may use UI e.g. Views
    }
  });
}
```

publishProgress may be called from a worker thread

(in the same activity)

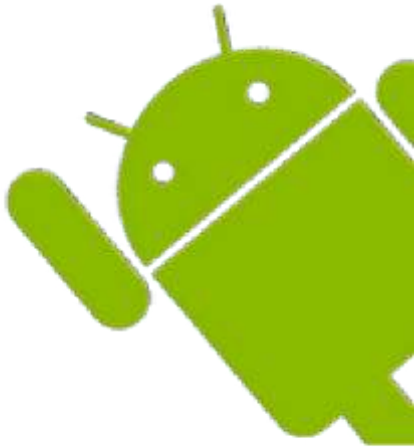
http://www.intertech.com/Blog/android-non-ui-to-ui-thread-communications-part-1-of-5/

Using AsyncTask to search for Things



```
private class searchClass extends
                              AsyncTask<String, Void, String> {
  String response = "????";
  Boolean found= false;         String mWhat;

  protected String doInBackground(String... param) {
    int i = 0;
    Boolean found = false;
    mWhat= param[0];
    int s = thingsDB.size();
    while ((!found) && (i < s)) {
      found = thingsDB.get(i).getWhat().equals(mWhat);
        i = i + 1;
    }
    return (found) ? thingsDB.get(i - 1).getWhere() : "????";
  }
  @Override
  protected void onPostExecute(String result) {newWhere.setText(result);}
}
```

Process states for an Android app

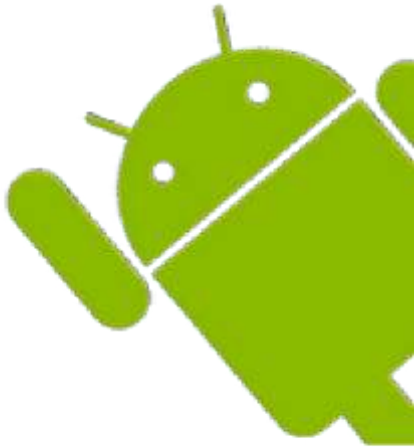


1. Foreground process
2. Visible process
3. Service process
4. Background process
5. Empty process

Android will nurture these processes like its own children

Android will slaughter and bury these processes the second it can get away with it

IntentServices



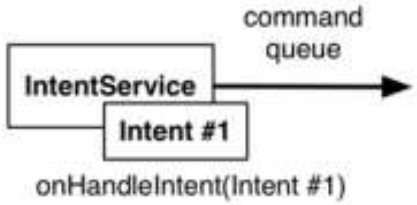
```
public class DBSyncService extends IntentService {

    public static Intent newIntent(Context context) {
      return new Intent(context, DBSyncService.class);
    }

    @Override
    public void onHandleIntent(Intent i){
      …
    }
  }
```

Manifest file:

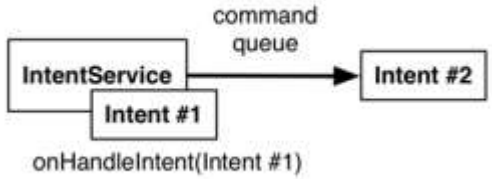
```
<application          ... >
<activity
  android:name=".TingleActivity"
  android:label="@string/app_name" >
...
</activity>

<service android:name=".DBSyncService" />
</application>
```

Servicing intents

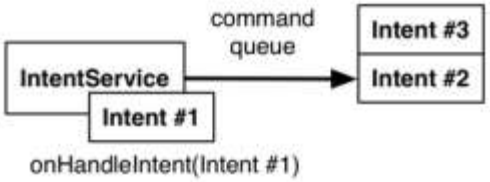
1. Command Intent #1 Received
Service Created



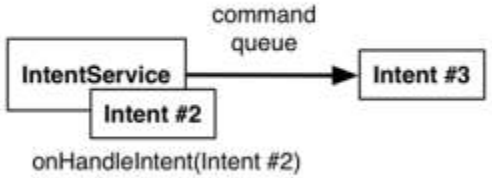
2. Command Intent #2 Received



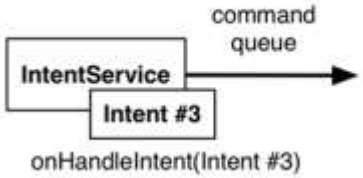
3. Command Intent #3 Received



4. Command Intent #1 Finished

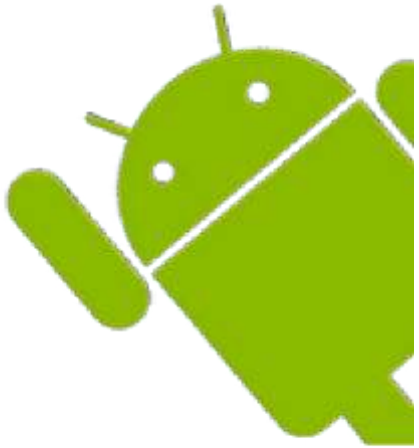


5. Command Intent #2 Finished



6. Command Intent #3 Finished
Service Destroyed

Background networking



```
@Override
protected void onHandleIntent(Intent intent) {
  if (!isNetworkAvailableAndConnected()) {   return;  }
  …
}
```

```
private boolean isNetworkAvailableAndConnected() {
 ConnectivityManager cm=
          (ConnectivityManager) getSystemService(CONNECTIVITY_SERVICE);
 boolean isNetworkAvailable= cm.getActiveNetworkInfo() != null;

 boolean isNetworkConnected = isNetworkAvailable
                                  &&cm.getActiveNetworkInfo().isConnected();
 return isNetworkConnected;
}
```

Second Mandatory Assignment

You decide on the user interface and functionality !!

Second mandatory: minimal functionality

The minimal functionality of your final version of the Tingle app must include:

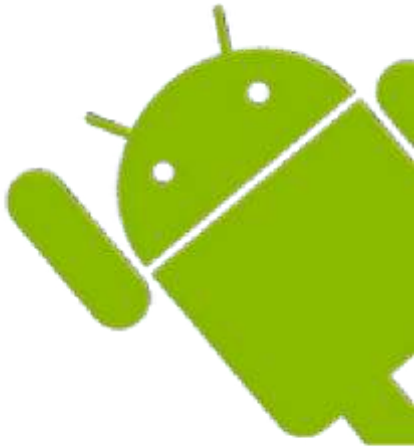
- Using a **database** to store things (Could be SQLite or a database on a server)
- Registration of things with **text entry** (as in the first Tingle versions)
- Registration of things with a **barcode (including lookup** of product info over the network)
- **Searching** for things including a principle/algorithm for ordering of multiple hits

Second mandatory: documentation

Documentation: You must submit a documentation (in pdf format) explaining your solution. The documentation must contain these sections:

- most important design choices, including functionality and user interface
- short user guide
- testing of the app – see below
- problems with your app (if there are any) e.g. if something does not work completely as you want

Second mandatory: design choices



Reflection of key decissions like:

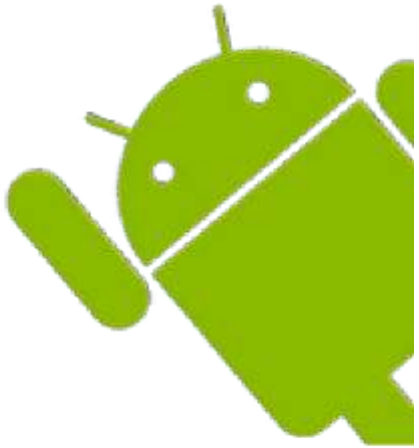
- additions to minimal functionality
- usability
- efficiency (e.g. how pictures are stored)

Testing

I have tested the app manually by adding Things, flipping the phone, deleting things from the list, switching activity using the "List of all things" button, and any combination imaginable of the above.

		Window 2 opened, worked
2. List of Things	• Scrolling through all items • Tapping an item to get information • Long tapping and item to delete • Long tapping and item to deleting and deleting/not deleting	Worked -||- -||- Deleted/didn't delete respectively, worked
3. Landscape	• Trying to add an item with text in both fields • Trying to add an item with one or more blank fields • Trying to add an item with whitespace characters • Trying to add emoticons as Things • Trying to add Thing with a lot of text, so the top of the app was not visible. • Clicking the "List all items" button • Scrolling through all items • Tapping an item to get information • Long tapping and item to delete • Long tapping and item to deleting and	Worked -||- -||- -||- -||- Window 2 opened, worked Worked -||- -||- Deleted/didn't delete

Possible Tingle extensions



Crowdsourcing of registrations
    separating private and public things?

Efficient handling of images
    searching images?

Ordering search results

Synchronisation with server database(s)

Using location (indoor tracking)

…

Second mandatory: final exam

You must have an approved solution to take the final exam.

Question from last years exam:

Explain what activities are, and describe their life cycles.

Course evaluation: NEXT WEEK

Jan Leschly, CEO /
Tennis pro (ATP 10)
If you are not counting points, you are just warming up

follow-up:

https://www.itu.dk/Om-ITU/Organisation-tal-og-fakta/Tal-og-fakta/Kvalitet-og-studiemiljoe/Kvalitet

This course: lecture on April 27 (or May 6)