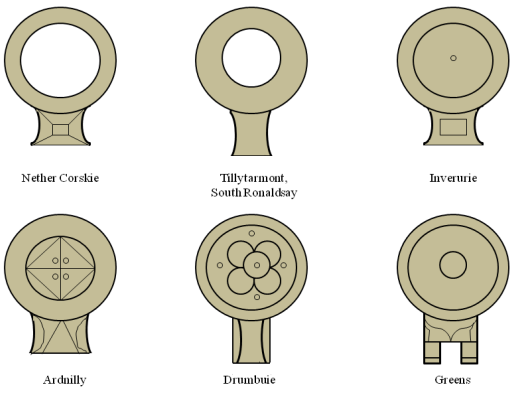
T8 - Application Development

T-DEV-800

PictsManager

Bootstrap



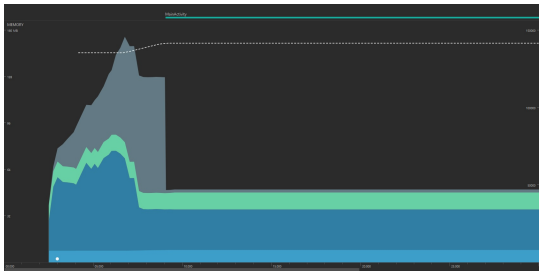
1.4

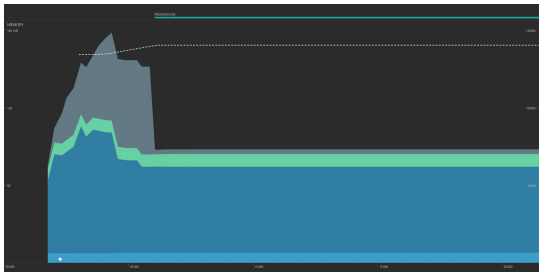
ANDROID PROJECT

Start off on the grounds of the T-DEV-700 project, to create an activity in a new Android project. Your goal is to display a picture from the Internet in your activity.

In the XML of your activity, add an ImageView object.From your code, load a picture from a URL and/or from your telephone storage.

Keep an eye on the memory consumption of your application. What do you notice?





Later on, you'll have to display several pictures at once on one of your views. Moreover, an app which is available on all types of devices (including those with small memory) is more likel to reach a large audience.

In order for your application to be less consuming, you'll have to transform the orginal picture: redimensioning, compression, other…

Compression reduce weight AND quality!

With a simple compression algorithm, lightly compress your picture then load it again in your activity. Analyze and compare your memory consumptions.

Via a dedicated tool, compress your picture, then load it again in your activity. Analyze and compare your memory consumptions.

Find and/or create more pictures, load them, transform them, load them again… etc… Analyze and compare your memery consumptions.

the documentation for Android as well as some libraries should be of great help.

Working with a database

+First Step

Correct handling of a database requires a sequence of choices, which should be made carefully according to the nature, the structure and the size of your data, like:

- How should your data be organized ?
- What database Management System is suitable ?
- How does your Java application connect and request your database ?
- How do you map your Java Objects onto your database ?
- What is the permission system and how is it enforced ?

You will make these decisions on time, preferably at the start of the project. For the purpose of this Boot-Strap only, we will enforce some decisions, but do not let them become the default solution for your project.

First of all, build on some server a simple relational database with two tables. First table is called Owner and has two fields (id and name); second table is called Picture and has three fields (id, url and owner_id).
In your project add a class called DataManager, which contains two methods:

- the first one connects to your database
- the second one takes a user name as an argument, and sends a request to your database querying the urls of all pics owned by that user.

Do not forget the good habits. Handle the dependencies in your project definition, cover your new methods with unit tests. Be careful that 1- there are many different sources of errors when dealing with database connections 2- for your test procedure to remain unitary, you will have to mock the database behaviour

+Go further

Now that you have a minimal interaction with your database, we can start considering the next steps:

- Instead of urls you want to fetch the pictures themselves. How do you store and request them?
- Try fetching all pictures of width and length between 60 and 90 px.
- Try replacing the Relational model by a XML based one.
- Go get huge data and see how much time requests take for various models.