Programming applications for the Android platform

Objectives

- Discover the Android system architecture.
- Understand the Android SDK and NDK
- Master Android application architecture
- Master the main programming tasks with Android
  - designing user interfaces
  - data storage and retrieval
  - network communications...
- Integrating your application in an Android system
  - calling system components
  - being callable by other components

Labs are conducted on i.MX6 or i.MX8 boards
We use the last open source version of Android, as available on the board.

Who should attend this course

- Engineers that must create programs for the Android platform

Prerequisite

- Advanced knowledge of Java programming (see our L4G - Java for Androidcourse)

Course Environment

- Theoretical course
  - PDF course material (in English) supplemented by a printed version for face-to-face courses.
  - Online courses are dispensed using the Teams video-conferencing system.
  - The trainer answers trainees' questions during the training and provide technical and pedagogical assistance.
- Practical activities
  - Practical activities represent from 40% to 50% of course duration.
  - Code examples, exercises and solutions
  - For remote trainings:
    - One Online Linux PC per trainee for the practical activities.
    - The trainer has access to trainees' Online PCs for technical and pedagogical assistance.
    - QEMU Emulated board or physical board connected to the online PC (depending on the course).
    - Some Labs may be completed between sessions and are checked by the trainer on the next session.
  - For face-to-face trainings:
    - One PC (Linux ou Windows) for the practical activities with, if appropriate, a target board.
    - One PC for two trainees when there are more than 6 trainees.
  - For onsite trainings:
    - An installation and test manual is provided to allow preinstallation of the needed software.
    - The trainer come with target boards if needed during the practical activities (and bring them back at the end of the course).
- Downloadable preconfigured virtual machine for post-course practical activities
- At the start of each session the trainer will interact with the trainees to ensure the course fits their expectations and correct if needed

Target Audience

- Any embedded systems engineer or technician with the above prerequisites.

Evaluation modalities

- The prerequisites indicated above are assessed before the training by the technical supervision of the traineein his company, or by the trainee himself in the exceptional case of an individual trainee.
- Trainee progress is assessed in two different ways, depending on the course:
  - For courses lending themselves to practical exercises, the results of the exercises are checked by the trainer while, if necessary, helping trainees to carry them out by providing additional details.
  - Quizzes are offered at the end of sections that do not include practical exercises to verifythat the trainees have assimilated the points presented
- At the end of the training, each trainee receives a certificate attesting that they have successfully completed the course.
  - In the event of a problem, discovered during the course, due to a lack of prerequisites by the trainee a different or additional training is offered to them, generally to reinforce their prerequisites,in agreement with their company manager if applicable.

Plan

First Day

Introduction

- History
- Overview
- The Android system architecture
  - Android services
  - Android frameworks

Android application architecture

- Structure of an Android Application
- Android application components
  - Activity
  - Service
  - Broadcast receiver
  - Content provider
- Manifest file
  - Application components declaration
  - Permissions

Exercise: *Hello world application*

Activities and user interface

- Activities life cycle
- Activity callbacks
  - onCreate
  - onStart&
- Intents and Intents filter
  - the Intent class
  - declaring Intent filters in manifest files
- Activity invocation with and without results
  - startActivity
  - startActivityForResult
- Tasks (activities stack) and navigation between activities
- Resources

    - Strings
    - images
    - layouts...
- Views
    - Buttons, labels and edition fields
    - View instantiation from a resource
- Layouts
    - Layout kinds
    - Components properties related to layouts
- Specialized views
    - ListView
    - Data binding (Adapter class and subclasses)

Exercise: *The simplified Notepad application*

Second Day

User Interactions

- User Input
    - Touch screen and keyboard
    - Software keyboard management
- User notifications
    - Dialog box
    - Status Bar
    - Toast
    - The Search dialog (SearchManager)
- Notifications and the Notification Manager
    - Notifications in the Status bar
    - Vibrating or flashing
- Push notifications

Exercise: *Enhancing the Notepad application*

- User interface adaptation
    - Depending on the language
    - Depending on screen characteristics (dimensions, orientation &)

Exercise: *Bilingual Hello world (English-French)*

Test and debug

- Using the debugger from Eclipse
- Logs
- Unit testing

Advanced User Interface

- User interface and multithreading
    - Accessing views from another thread

Exercise: *Multi-threaded user interface with buttons and progress bars*

- Custom control creation
    - By deriving directly the View class
    - By deriving an existing view
- 2D Drawing
    - Canvas and Shapes
    - Drawing from the main thread
    - Drawing from another thread
- Animations

Exercise: *Moving an image on the screen*

Third day

The Fragment API

- What is a fragment
- The fragment lifecycle
  - Interaction with the Activity lifecycle
  - The fragment-specific lifecycle callbacks
- Fragments and screen orientation
  - Using fragment to adapt to various screen sizes
  - Reacting to orientation changes
- The various specialized fragments
  - List fragments
  - Dialog fragments

Exercise: *Adapting the Notepad application to varying screen sizes using fragments*

The Android NDK

- The Android NDK
  - Defining Java methods in C++
  - JNI for Android
  - Using SWIG
- Integrating native code in a package
  - Using the NDK from Eclipse
  - Debugging native code

OpenGL/ES

- OpenGL and OpenGL/ES
  - OpenGL/ES versions
  - Java Access to OpenGL/ES
  - Native access using the NDK
- Base Concepts
  - Vertices and Triangles
  - Transformation
  - Drawing
- OpenGL/ES 1.1
  - The OpenGL/ES 1.1 Pipeline
  - Projection and Camera View
- OpenGL/ES 2.0
  - The OpenGL/ES 2.0 programmable pipeline
  - Vertex shaders
  - Fragment shaders
  - Applying transforms
- Android OpenGL Setup

Exercise: *Creating an animated 3D view in OpenGL/ES (from Java and through the NDK)*

Renderscript

- The Android Renderscript layer
  - The Android framework renderscript API
  - The Reflected layer mapping renderscript code to Java classes
- Renderscript code
  - The Renderscript C language
  - The renderscript compute engine
  - The deprecated renderscript graphics engine

Exercise: *Using Renderscript compute for transforming an image from color to black and white*

Fourth Day

Services

- Service declaration
- Starting and stopping a service
- RPC
  - Definition and implementation of an AIDL interface
  - Service binding and RPC invocation
- The Android binder

Exercise: *Creation of an application service*

- System services
  - What is a system service
  - Static and context-dependent services
  - Structure of a system service
  - The ServiceManager process
- Application Power Management

Exercise: *Use Android Power Management to prevent device getting asleep*

Multimedia

- Audio and video playback (MediaPlayer class)
- Audio and video capture (MediaRecorder class)

Exercise: *Implementation of an mp3 playback service*

Broadcast Receivers

- Installing a Broadcast Receiver
  - Static creation of broadcast receivers
  - Dynamic instantiation and registration
- Broadcasting intents
  - Normal broadcast
  - Ordered broadcast
- Using PendingIntent in broadcast receivers
- System broadcasted events

Exercise: *Implementation of a custom broadcast receiver*

Network

- Connections management
- Sockets
- HTTP requests
- WebView control
- Web Services

Exercise: *Socket communications*
Exercise: *Displaying a web page with a WebView*
Exercise: *Consuming a web service*

Fifth Day

Data management

- Storage
  - Shared preferences
  - Internal storage
  - External storage
  - SQLite

- Content provider
  - Communication with a content provider
  - Implementing a content provider

Exercise: *MP3 file storage alongside author and album names in a SQLite table*
Exercise: *Implementation of a content provider and a client to retrieve the MP3 files*

Interacting with the platform

- Contacts management
- Sending and receiving emails
- Placing a phone call
- Camera, video and still Pictures
- Main system events
- Broadcasted events

Exercise: *Application to take a picture and send it to a contact*

The Android Sensors

- Sensors in Android
  - The sensor types
  - The Sensor Manager
  - Accessing Sensors
- Framework Architecture
  - Sensor discovery
  - Sensor Calibration

Exercise: *Getting and displaying a sensor value (temperature...)*

Bluetooth

- Configuration
- Device search
- Services management (SDP)
- RFCOMM sockets

Exercise: *Bluetooth chat*

Location and Google Map API

- Location
  - Location framework with cells, WIFI or GPS
  - The LocationManager class
- Integrating Google Map API in an application
  - The MapView class

Exercise: *Using a MapView and displaying an image in it for points of interest*

Renseignements pratiques

Inquiry : 5 days