Android App Programming

Lecture 1: Introduction, Activities
October 10, 2017

Thomas Nowak
Université Paris-Sud
Course organization

- Classes Tuesday 1-4 pm in room I013-015
- Labs every week, right after lecture
- https://www.lri.fr/~nowak/teaching/android/
Grading

1. Reading quizzes: 10%
2. Practical exam 1: 20%
3. Practical exam 2: 20%
4. Final project: 50%
Reading quizzes

- Advance reading before each class
- Short quiz before lecture
- ~5 minutes on Moodle
- A few multiple choice questions each time
Exams

- Two practical exams
- ~2 hours
- Create a small app
- Checks your understanding of the content of the preceding unit
Course overview

- Unit 1:
  - Activities
  - Intents
  - Web services
  - Asynchronous tasks
  - Peripherals
  - Databases
Course overview

• Unit 2:
  • Android Runtime
  • User Interface Design
  • RESTful services
  • Dealing with disconnectivity
  • Testing and profiling
  • Kotlin
Course overview

- Unit 3:
  - Design and develop your own app
  - Groups of 4
  - Version control via git
  - Final presentations and demos
Resources

- Android and Java APIs and documentation
- Lecture slides on course website
- Lots of online articles and tutorials
Introduction

• Getting started with Android Studio
• Create first app
• Single screen app: single Activity
App Inventor 2

- Online app creation tool
- Generates APKs
- Good for fast prototyping
- Graphical programming language
- Can become cumbersome for larger projects
initialize global xVel to 0
initialize global yVel to 0
initialize global DPIConst to 40

when Clock1.Timer
do set global xVel to get global xVel + \(-0.00581 \times \) Clock1.TimerInterval \(\times \) sin Orientation
  set Ball1.X to Ball1.X + get global DPIConst \times get global xVel
  if Ball1.X \(\leq\) 0 and get global xVel \(\leq\) 0 then set global xVel to 0
  if Ball1.X \(\geq\) Canvas1.Width \(\times\) 2 \times Ball1.Radius then set global xVel to 0
  set global yVel to get global yVel + \(-0.00581 \times \) Clock1.TimerInterval \(\times \) sin Orientation
  show warnings Ball1.Y to Ball1.Y + get global DPIConst \times get global yVel

Privacy Policy and Terms of Use
Java crash course

• Interfaces/classes:
  MainActivity extends AppCompatActivity

• Callbacks:
  boolean onOptionsItemSelected(MenuItem item)

• Lambda expressions:
  view -> System.out.println(view)
Android Studio

- IDE for Android applications
- Includes management of virtual devices (emulator)
- Tools for testing, profiling, packaging, shipping
- Lots of tutorials on the web
Layout files

- All UI elements can be created in Java code
- Most of the time, it isn’t
- Description in XML format
- Elements created at startup
• Dynamically generated class during compilation

• Links XML layout files to Java

• Contains all assets of the application (layout files, images, strings, etc.)

• Field names defined in layout editor

• Ex: `R.id.toolbar`
Activities

- Single block of user interaction
- In the simplest case, one screen
- Can be a window or part of the screen
- To display something, set its content:
  - `setContentView(View)`
Views

- Basic UI component
- E.g., Button, TextView, ImageView, menus, etc.
- Has a set of mutable properties (e.g., text size)
- Can register listeners:

  ```java
  button.setOnClickListener(this);
  ```
Lab 1

1. Setup Android Studio on your machine with your device
2. Create a simple Activity and explore the layout editor
3. Add simple interactivity (a button)
Advance reading

• Next week’s topic:

• Intents (a.k.a. switching Activities)

• Ch. 11.4–11.8 and 42.1–42.2 of Android Studio 3.0 Development Essentials
Questions