Lecture 5 — XML, Android, IDEs

Patrick Lam & Jeff Zarnett
p.lam@ece.uwaterloo.ca & jzarnett@uwaterloo.ca

Department of Electrical and Computer Engineering
University of Waterloo

November 14, 2015

Part I

XML

eXtensible Markup Language

You need to know a bit about XML (e**X**tensible **M**arkup **L**anguage) to build Android applications

By the way, this is examinable material.

For further reading, you can consult many resources on the Web. Here's one:

`http://www.w3schools.com/xml/default.asp`

XML in one Line

XML is a *structured document format*.

Structured Document Format

All XML documents therefore have the same format.

XML has no intrinsic meaning.

It just separates content from structure.

It is meant to be human-readable/writable.

If you are familiar with HTML, there are a number of similarities, but XML is more general.

XML Document Terminology

Every XML document begins with
`<?xml version="1.0" encoding="utf-8"?>`

Documents have tags, which appear between angle brackets:
`<example>`

Tags are opened and closed, so that tag must be followed by `</example>`, which closes the `example` tag.

We can also have "self-closing" tabs, (notational convenience) such as `<approval/>`.

A tag may also have attributes, like colour in this example:
`<square colour="red"/>`.

XML Parsing

XML is intended to be human-readable and human writeable.

Computers can read and comprehend XML by “parsing” the document.

There are a number of well-known XML parsers.

Use one of them if you need to programmatically examine some XML and do not write your own.

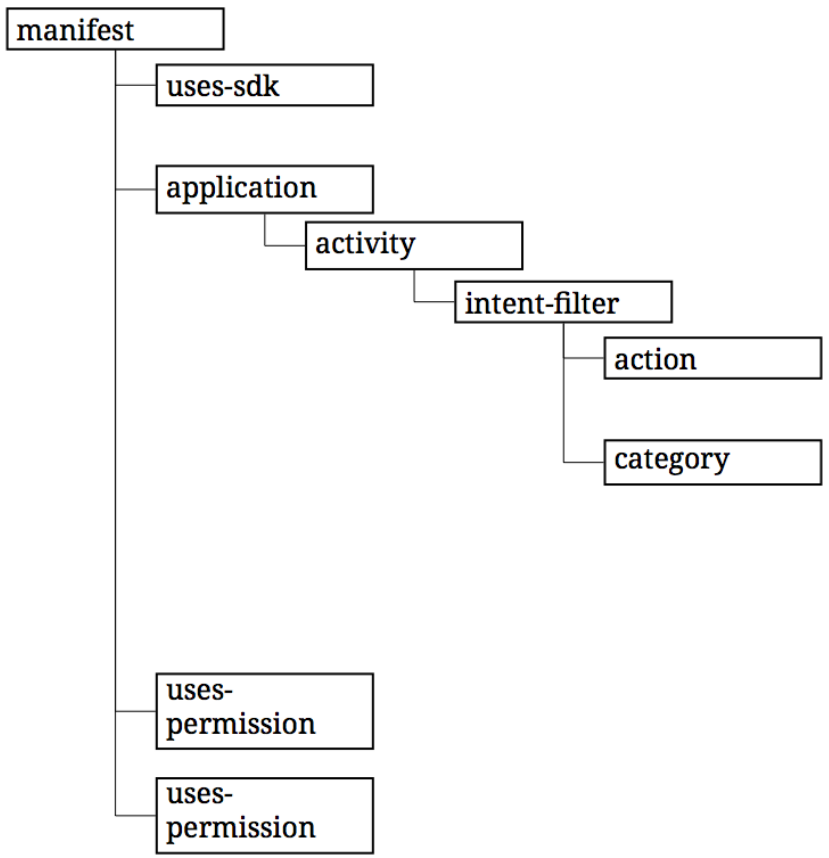
Android Manifest

Example XML Document: Android Manifest.

(Shown from the notes because it's too large for slides).

XML Tree Structure

XML is tree-structured: at the top level, there is a *root* element. We can convert the textual form into a tree.



Part II

Android Programming

Android: Introduction

By now you have no doubt taken a look at the labs.

You need to implement them using Android, either on your own phone or one of the devices we provide in the labs.

We will take some time to introduce you to Android programming.

Android: Activity

"An activity is a single, focused thing that the user can do."

Usually, an Activity corresponds to a full-screen window which the user may interact with. For instance, the user may:

- set up a timer;
- read off sensor values; or
- make a phone call.

Android: Activity

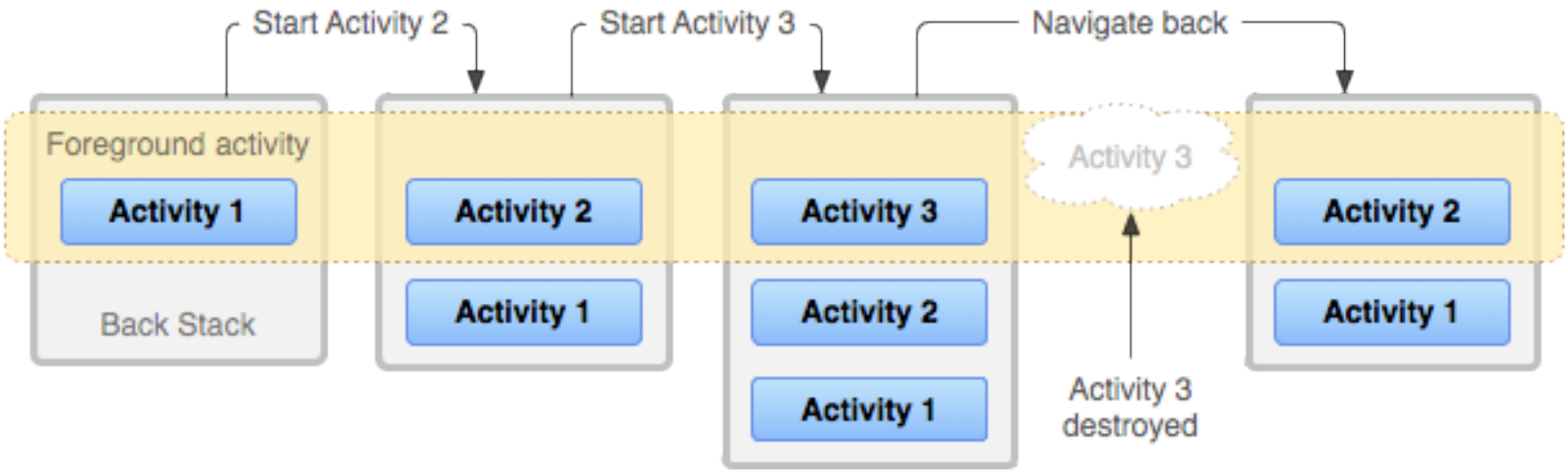
Applications may contain multiple activities, each of which corresponds to a thing that the user wants to do.

Android organizes activities into tasks.

A task consists of a last-in, first-out stack of activities, possibly from different applications.

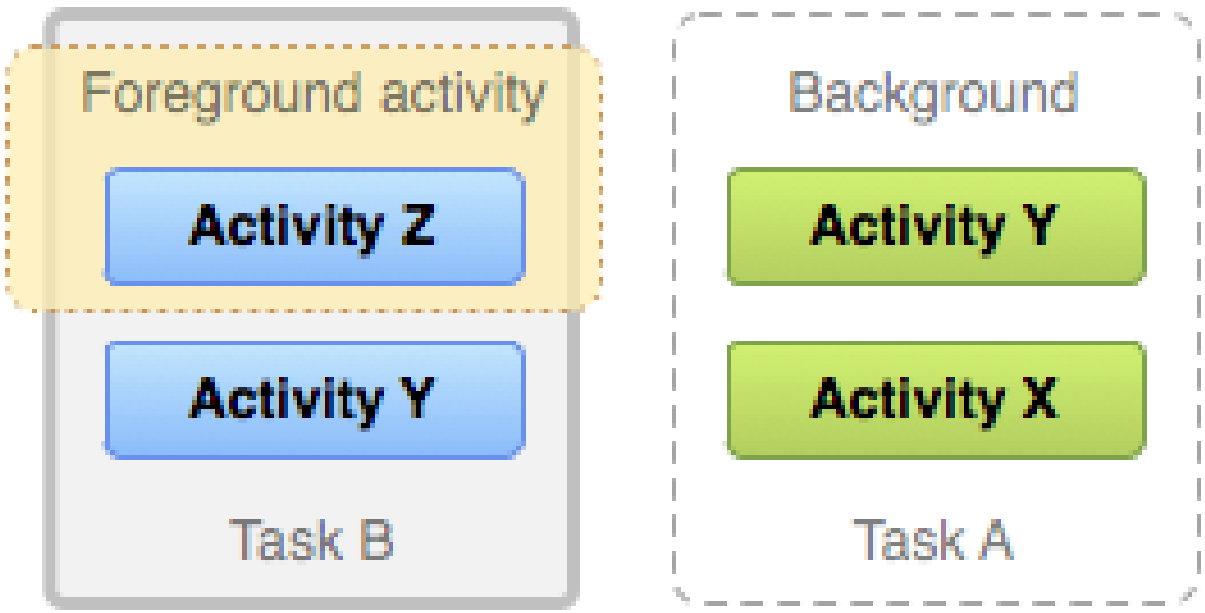
The Back Button

The Back button pops the topmost activity off the stack and gets rid of it.



It is also possible to switch between tasks.

Switching tasks puts a different activity and its stack in the foreground, and puts the old activity in the background.



Doing Something in the Activity

You'll be writing a lot of code in `onCreate()`.

It gets executed when the activity starts.

Typically, it will set up the user interface, namely:

- creating widgets;
- setting up event listeners;

Note that you must call **`super.onCreate()`**.
This is done for you in the autogenerated boilerplate code.

Retrieving Widgets

If you've declared widgets in the XML file, you can use the **`findViewById()`** method to get a hold of them.

Note:

- you need to cast the return value, e.g.
  `tv = (TextView) findViewById(R.id.t);`
- you must save the XML file to get the right ids on the R object.

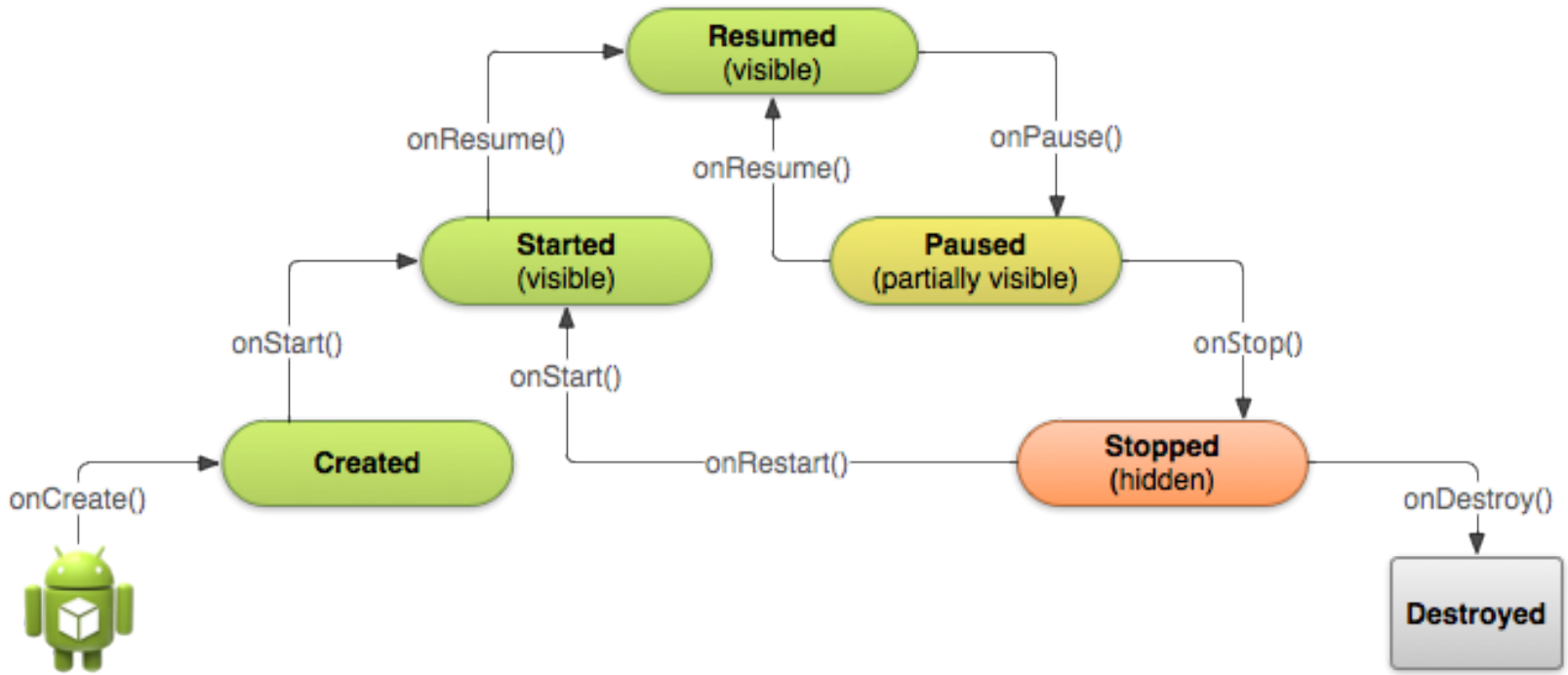
Programmatically Adding Widgets

We ask you to add widgets programmatically in Lab 1. There are two steps:

1. Create the widget: `tv = new TextView(getApplicationContext());`
2. Add it to the Activity: `addView(tv);`

Activity Lifecycle

Although we've only talked about `onCreate()`, there are numerous other methods on `Activity` which Android calls at various times.



Part III

Integrated Development Environments

In the Dark Ages

Write code in editor (`vi`, `emacs`, `notepad.exe`)

Invoke command-line compiler

Can we make better tools? Yes: IDE!

What is an IDE?

Contains:

- editor, plus
- compiler, plus
- debugger

Integrated into a single environment.

The Eclipse IDE

Eclipse is a fully-featured modern IDE.

Notes:

- runs on Linux, Mac, Windows;
- it is free software: you can extend and modify it;
- initially developed by IBM Ottawa.

Beyond Core IDE Components

Eclipse (and other IDEs) also support:

- revision control systems (you've used this);
- documentation and modelling (e.g. UML);
- autocomplete and refactoring.

Syntax Highlighting

```
// This file must exist first!
private static File outputFile = new File("files/mkdir.txt");

/**
 * @param args
 * @throws IOException
 */
public static void main(String[] args) throws IOException {

    BufferedWriter bw = new BufferedWriter(new FileWriter(outputFile));
```

Syntax Highlighting

```
// This file must exist first!
private static File outputFile = new File("files/mkdir.txt);
```

Templates

IDEs often allow you to start your project from a template; easier than starting from scratch

- Android Test Project (you'll use these);
- Java Project
- Java Class
- Java Interface

Continuous Compiling

```
// This file must exist first!
private static File outputFile = new File("files/mkdir.txt);
```

Notice the red squiggle under the error.

Project Development Workflow

0. Figure out what you'll need to do.
1. Start a new project from a template or, more realistically, check it out from a version control repository.
2. Make the edits that you need.
3. Test your edits by running the application.
4. Debug your edits.
5. Commit your files to the version control repository.

Eclipse demo

We'll show a little demo of Eclipse (time permitting):

1. Create a new project from a template.
2. Go over syntax highlighting.
3. Show you content assist.
4. Demonstrate the "Quick Fix" feature.