Android: Activities and Views

29 Mar 2010
CMPT166
Dr. Sean Ho
Trinity Western University

TRINITY WESTERN UNIVERSITY

Outline for today

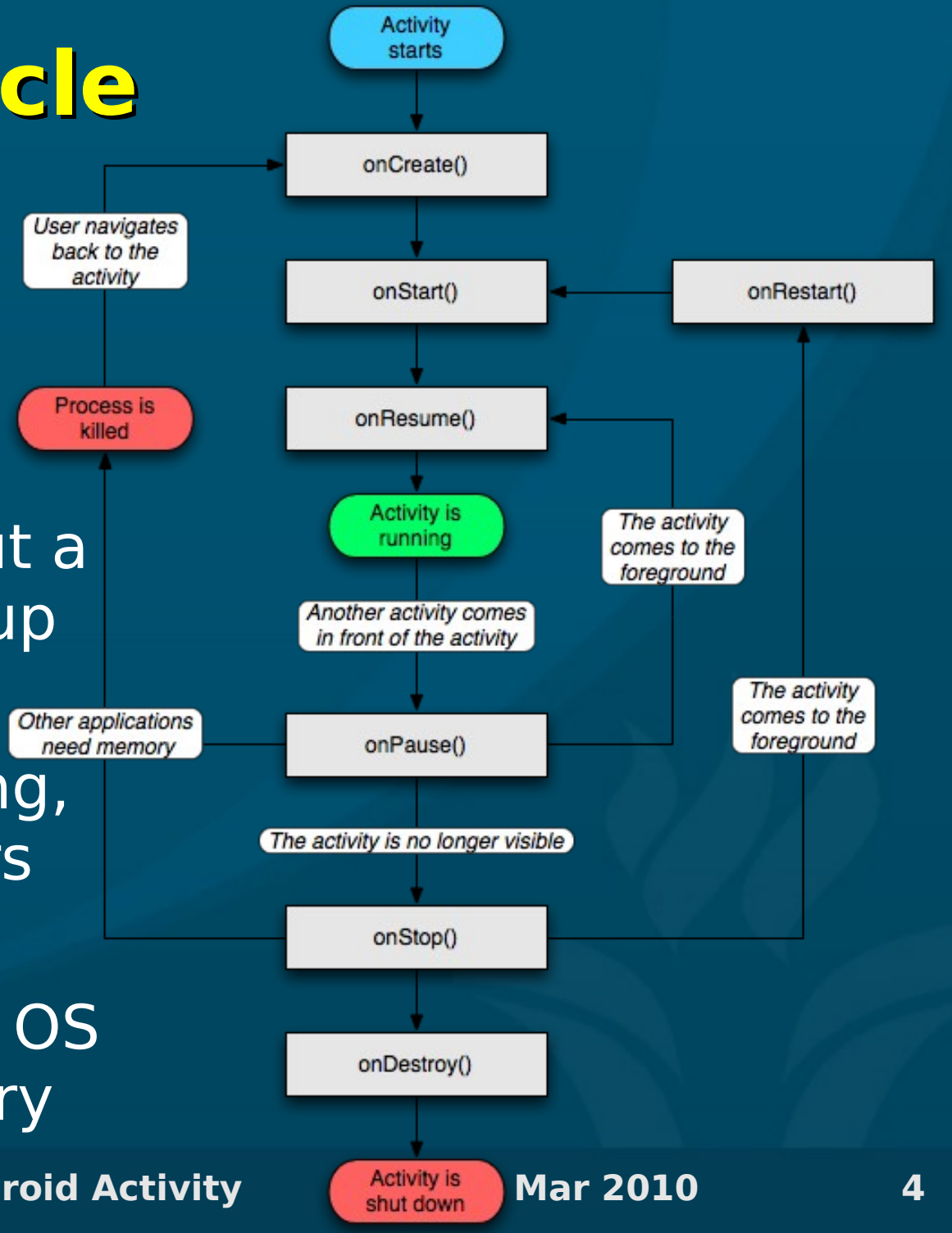
- Android Activity: life cycle, states
  - Life cycle methods
  - Saving persistent / transient state
- Android Views (widgets):
  - Widgets and layouts
  - XML configuration, layout editor

Kinds of Android applications

- Activity: present UI for one interactive task
  - e.g.: get username+password, display map
- Service: background task, often w/o UI
  - e.g.: play music, fetch file over network
- Broadcast Receiver: respond to announcements
  - e.g.: if timezone changes, battery low, etc.
- Content Provider: access/query a datastore
  - e.g.,: music library, student database, etc.
- We will focus on Activities, as the simplest kind

Activity life cycle

- Four states:
- Active: running and in foreground
- Paused: running, but a dialog has popped-up on top of it
- Stopped: still running, but hidden by others
- Dead: terminated, perhaps by Android OS when low on memory



Life cycle methods

- Activity exists between onCreate/onDestroy():
  - Initial setup and final tear down of resources
- Activity is visible between onStart/onStop():
  - onRestart() also called when return to fore
- In foreground between onResume/onPause():
  - In foreground means accepting user input
  - onPause: commit unsaved changes, etc.
- A paused activity might be destroyed before it ever resumes!

Saving state

- Persistent state should be saved in onPause()
  - e.g. draft of a message being composed
  - Write to storage: preferences, SQL database, app-specific file, or SD card
- Transient state: use onSaveInstanceState()
  - e.g. how user filled out form before “submit”
  - Save in a Bundle, which is passed to both onCreate() and onRestoreInstanceState()
  - Use this, e.g., to fill out the form again when user goes “Back” to this activity

Views (widgets)

- View is Android's widget class (c.f. JComponent)
- Subclasses include: Button, TextView (label), EditText (text area), Spinner (pull-down list), ...
  - Or make your own subclass to customize!
- ViewGroups are layout managers: LinearLayout, GridView, TableLayout, TabHost,..
- In the activity's onCreate() method: call setContentView() to declare the activity's main View (panel):
  - **TextView tv = new TextView(this);**
  - **setContentView(tv);**

“Hello, Android!” tutorial

- Only one activity: HelloAndroid
  - Package: domain name, application name
- onCreate() method: called when activity is run
- The parameter is the saved state Bundle:
  - Use this to restore transient state
  - Also pass up to superclass onCreate()
- Create a view (widget): TextView
  - Set the text to “Hello, Android!”
  - Set as the main view for the activity
- Set a layout as main view if want > 1 widget!

XML layout

- Laying out widgets can be complex in code
- You may use an XML config file for your layouts:
  - Create a file under res/layout/*.xml
  - XML is like HTML: <tag> ... </tag>
- Specify layouts, widgets, font/colour/text/etc.
  - Eclipse ADT has a WYSIWYG layout editor!
- The XML layout gets compiled into an object in the R class (auto-generated; don't edit directly!)
  - Refer to R.layout.myLayout (follows name of the XML file)