

Temari

- 1) Introducció
- 2) Framework Android
- 3) Projectes Android i Android SDK
- 4) Activity
- 5) Fragments, Views i ListViews
- 6) Intents
- 7) Layouts i Custom Views
- 8) Resources i Themes
- 9) Dialogs, Menus i WebView
- 10) Persistència de dades
- 11) Tasques en Background i internet
- 12) SQLite i content providers
- 13) Notificacions



5 – Fragments, Views i Listviews

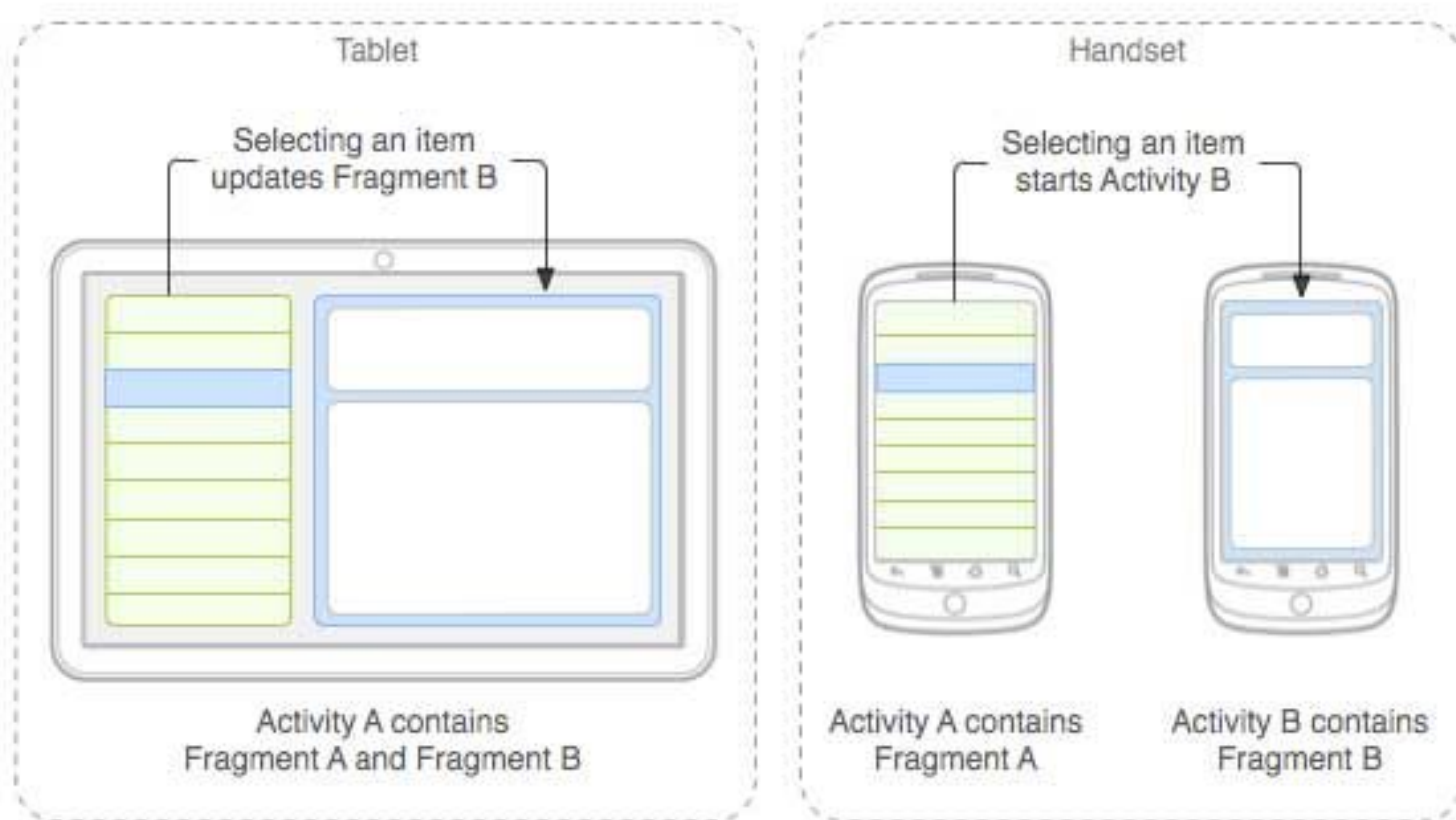
- Fragments
- Views
- ListView i ListActivity



5 – Fragments, Views i ListViews

Fragments, què són?

Apareixen al Android 3.0, representen una **porció de la UI d'una activity**, i es poden reusar en més d'una activity.



5 – Fragments, Views i ListViews

Fragments

- Una activity pot tenir més d'un fragment
- Els fragments es poden afegir i treure de l'activity en el layout (XML) o en el codi dinàmicament (java)
- Tenen un cicle de vida propi, amb callbacks pròpies, però vinculades a les de l'activity
- Un mateix fragment es pot fer servir en diferents activities
- Si volem tenir Uis diferents segons característiques de la pantalla (mida, landscape-portrait...) → necessitem fragments
- Similar que amb l'Activity, cal sobreescriure la classe **Fragment**



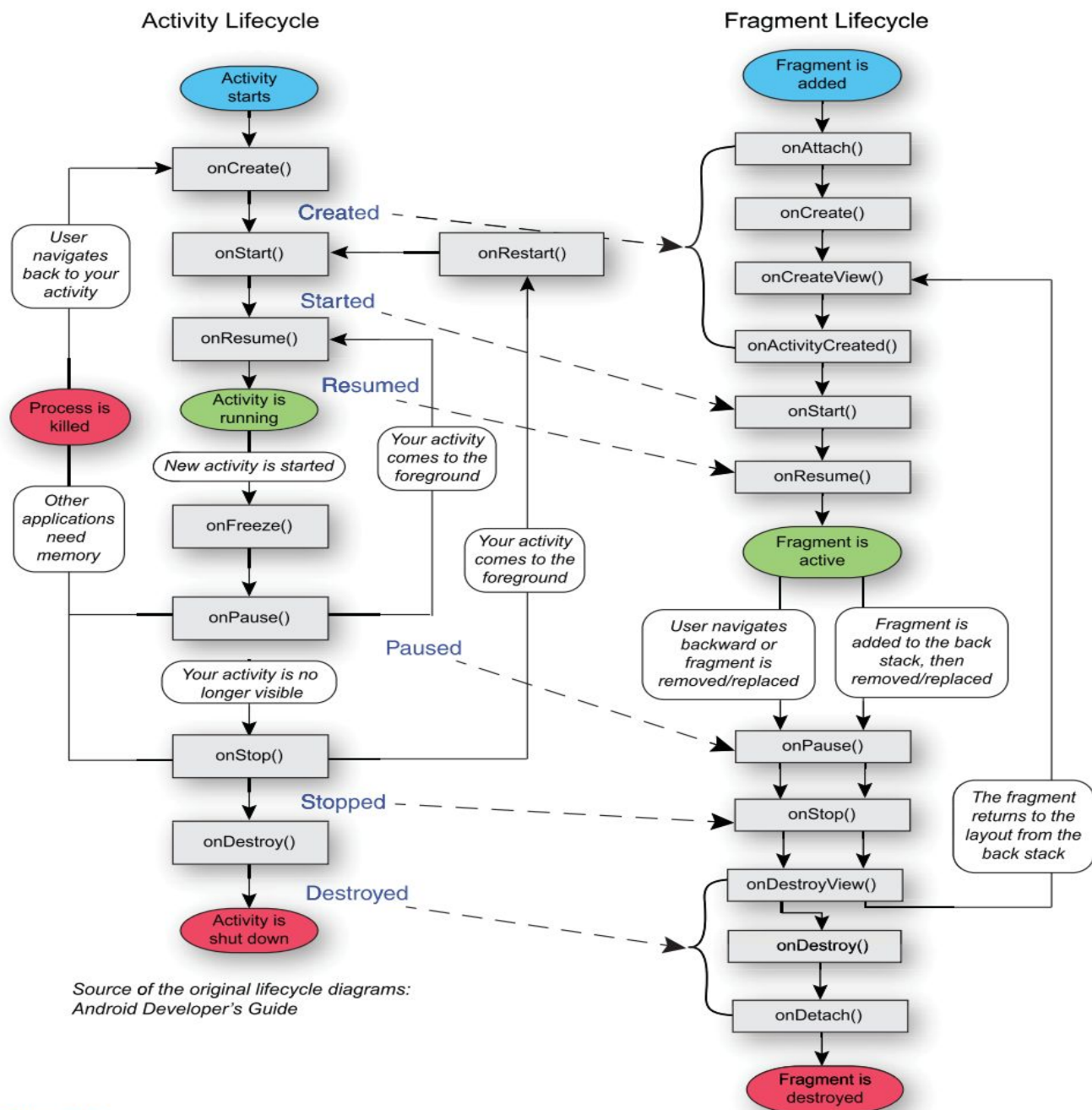


Figure 20.1 Activity and fragment lifecycles



5 – Fragments, Views i ListViews

Fragments, Cicle de vida

- Al mètode **onCreateView()** inicialitzem la UI → inflar-la amb un layout
- **OnCreate()** inicialitzem components i dades que volem conservar quan es pausi/pari el fragment (pause/stop)
- **OnAttach()** executat quan s'afegeix el fragment a l'activity. Utilitzar per a compartir events amb l'activity → amb interface
- **OnPause()** executat quan l'usuari abandona el fragment. Aquí cal guardar les dades que han de persistir.



5 – Fragments, Views i ListViews

Fragments, afegir al layout

- Fent servir el tag <fragment> al layout de l'activity
- Especificant la *class* del Fragment
- Posant-hi un id, per a referenciar-lo despres

```
<fragment  
    android:id="@+id/listFragment"  
    android:layout_width="0dp"  
    android:layout_weight="1"  
    android:layout_height="match_parent"  
    class="com.example.android.rssreader.MyListFragment"  
></fragment>
```



5 – Fragments, Views i ListViews

Fragments, afegir programàticament

- Cal obtenir un FragmentManager

```
FragmentManager fm = getFragmentManager();
```

- Iniciar una transaccio

```
FragmentTransaction ft = fm.beginTransaction();
```

- Fer les modificacions que calguin

```
ft.add(R.id.your_placehodler, new YourFragment());  
ft.replace(R.id.your_placehodler, new YourFragment());  
Fragment fragment =  
fm.findFragmentById(R.id.your_placehodler);  
ft.remove(fragment);
```

- Fer commit

```
ft.commit();
```

Id d'un layout
(Framelayout)



5 – Fragments, Views i ListViews

Fragments, provem-ho

- Fem un exemple
- Fer una app que:
 - tingui 3 fragments, cadascun d'ells amb un botó.
 - En cas que la disposició sigui *portrait*, es mostra només el primer fragment, en cas que sigui *landscape*, els dos primers
 - En portrait, en fer click al botó, cal obrir el següent fragment, en fer enrere, tornar a l'anterior. En landscape, el 3^r fragment s'obre en el lloc del segon



5 – Fragments, Views i ListViews

Views

- És la *parent class* de tots els elements interactius de la UI
- És responsable de **mostrar-se** i **recollir els events**
- **ViewGroup** n'és una subclasse, que és un contenedor per a altres Views
- Es poden afegir a través d'un XML o directament al codi
- Un cop afegides, depenent del que volguem, configurarem:
 - **Propietats**: depenent de la view, en tindrà unes o altres
 - **Focus**: si volem que inici l'*user input* hem de cridar al **setFocus()**
 - **Listeners**: objectes que contenen el mètode o mètodes a executar quan passa un event
 - **Visibilitat**: es pot controlar si es veu o no amb *setVisibility(int)*



5 – Fragments, Views i ListViews

Views

- Remenem una mica per els widgets que venen per defecte
 - Button
 - ToggleButton
 - Checkbox
 - ImageView
 - ...
 - LinearLayout
 - RelativeLayout
 - ...



5 – Fragments, Views i ListViews

ListView i ListActivity

- ListView és una View pensada per a mostrar llistats verticals d'elements
- Dues opcions per a mostrar una ListView:

Si l'Activity conté **només**
una ListView



ListActivity
(recomanat)

Si l'Activity conté **més**
Elements a part de la
ListView



ListView



5 – Fragments, Views i ListViews

ListActivity

- Subclass d'Activity, que la seva UI és una ListView
- Per a fer una ListActivity cal:
 - Després de Crear el projecte, crear una class que extengui ListActivity

```
public class MyListAct extends ListActivity{ ...}
```

- Crear un array de Strings amb el que volem mostrar

```
final String[] ANDROID_OS={"Cupcake", "Donut",...};
```

- Crear un ArrayAdapter al onCreate()

```
setListAdapter(new ArrayAdapter<String>(this,  
android.R.layout.simple_list_item1, ANDROID_OS));
```

Provem-ho!



5 – Fragments, Views i ListViews

Adapters

- Classes del SDK que connecten les dades amb la presentació
- Existeixen vàries per defecte:
 - ArrayAdapter: adapta un array o List
 - CursorAdapter: adapta un cursor (accés a BDD o a ContentProvider)
 - Una que creem nosaltres extenent BaseAdapter
- Especifica quants elements hi ha, quin element hi ha a cada posició, quin layout es fa servir per pintar cadascun dels elements...
- Si les dades canvien (afegim o treiem coses del array, o de la BDD...) cal cridar a **notifyDataSetChanged()**



5 – Fragments, Views i ListViews

ListView

- Es pot fer servir com a widget dins d'un Layout complex
- Cal crear un Layout que serà el que aplicarà als items de la llista
- Cal afegir-li un **adapter** (per exemple un ArrayAdapter), que lliga:
 - Els elements a mostrar
 - El layout a aplicar a cada element
 - La activity

Provem-ho!



5 – Fragments, Views i ListViews

Events en una ListView

- Afegir un event en fer click sobre un element

```
listView.setOnItemClickListener(...)
```

- Afegir un event en fer un click llarg sobre un element:

```
listView.setOnItemLongClickListener(...)
```



5 – Fragments, Views i ListViews

Lab → Simple TO-DO app

- Fer una app que tingui un camp de text i un botó, per afegir elements a la llista
- En fer un click llarg sobre un item, l'esborra

Intentar-ho!

