

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



4 – Activity

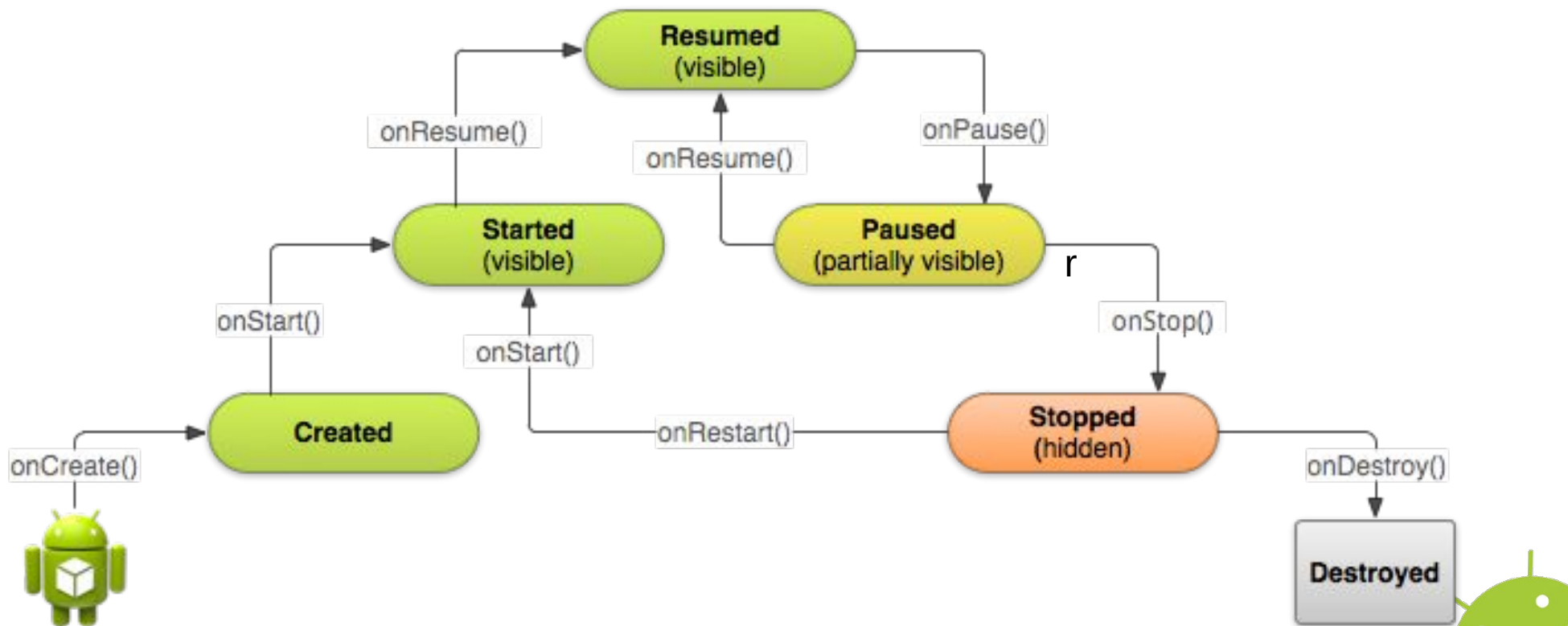
- Cicle de vida
- onCreate
- onStart
- onResume
- onPause
- onStop i onDestroy



4 – Activity

Cicle de vida

Les *Callbacks* permeten executar codi en cada punt del cicle de vida de l'activity → fent override (**sempre cridar al super**)



4 – Activity

onCreate

Callback que executa el sistema quan s'inicia (*lanuch*) l'activity

- Cridarà al `super.onCreate()`
 - **Inflem** el layout per a l'activity:
 - `setContentView(R.layout.my_activity)`
 - Si cal, inicialitzem les Views que farà servir per interactuar
 - `findViewById(R.id.button1)`
 - Podem començar a posar Logs amb `Log.i("Tag","Log a escriure");`
-
- Inicialitzar tot allò que necessita l'aplicació i no consumeix recursos escassos



4 – Activity

onStart

Callback que executa el sistema quan es mostra l'activity

- Cridarà al `super.onStart()`
- Iniciem aquells recursos que la nostra app necessita sempre que sigui visible (càmera, localització...)



4 – Activity

onResume

Callback que executa el sistema quan es mostra l'activity al *foreground*

- Cridarà al `super.onResume()`
- Inicialitzem aquells recursos que siguin escassos i necessita la nostra app per funcionar
- Iniciem tasques que consumeixen bateria, CPU o memòria



4 – Activity

onPause

Callback que executa el sistema quan l'activity deixa d'estar al *foreground*

- Cridarà al `super.onPause()`
- Alliberem aquells recursos que siguin escassos i necessita la nostra app per funcionar
- Aturem tasques que consumeixen bateria, CPU o memòria

El sistema android GARANTEIX que es crida al onPause



4 – Activity

onStop

Callback que executa el sistema quan l'activity deixa d'estar visible

- Cridarà al `super.onStop()`
- Alliberem aquells recursos que haguem inicialitzat al `onStart`

El sistema android NO GARANTEIX que es crida al onStop → Cal alliberar recursos escassos i guardar dades al `onPause`



4 – Activity

onDestroy

Callback que executa el sistema quan s'alliberen tots els recursos de l'activity

- Cridarà al `super.onDestroy()`
- Rarament s'utilitza

El sistema android NO GARANTEIX que es crida al onDestroy →
Cal alliberar recursos escassos i guardar dades al onPause

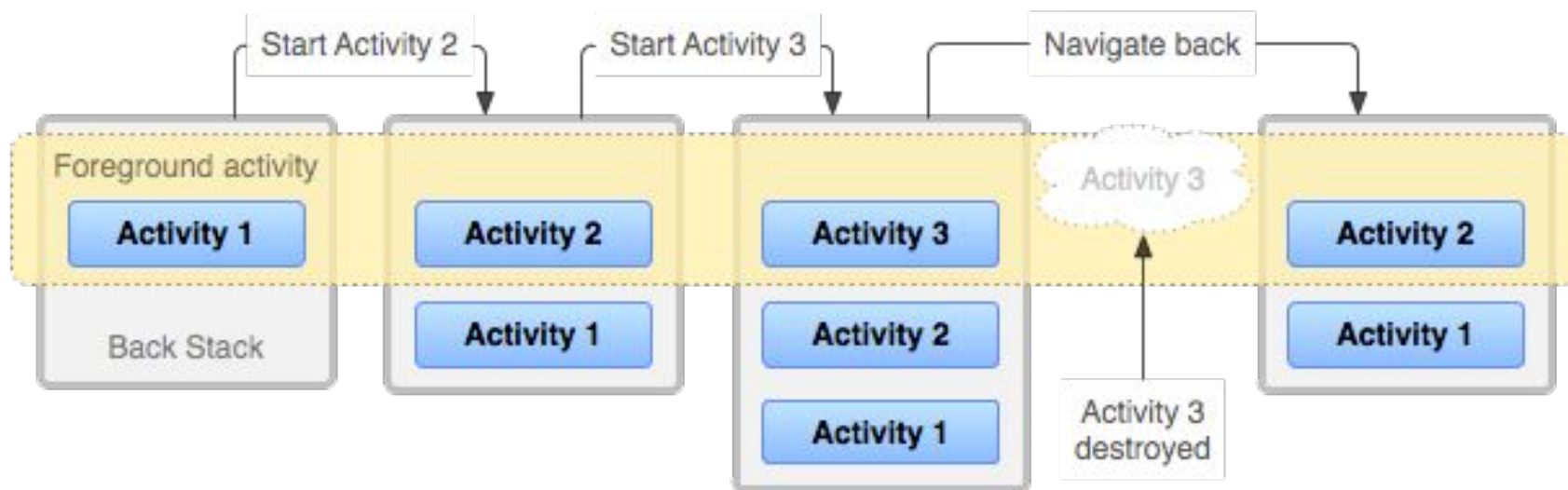


4 – Activity

Backstack

Quan una activity s'inicia, s'afegeix a una **pila** d'activities (**activity stack**)

En apretar el botó *back* l'activity es tanca i s'obre la que hi ha a la pila, sobre de tot



4 – Activity

Provem-ho

Fem una app de lot amb el flash

Afegir Log.x()!!!

- Nou Project, amb un layout amb un ToggleButton (o botó amb imatge?)
- A l'activity, afegir:
 - Variable d'instància per a saber si està encès el flash
 - Variable d'instància per a guardar la Camera
 - Variable d'instància amb els parametres de la càmera
 - un mètode per a encendre el flash
 - Un mètode per apagar el flash
 - Un mètode per obtenir la càmera
- Fer servir el PackageManager per saber si el dispositiu té flash
- Mostrar una alerta en cas que no en tingui, i acabar l'activity **finish()**
- **Sobreescriure les *callbacks* del cicle de vida:**
 - *OnCreate* → *comprovar si té flash, configurar el botó*
 - *OnStart* → *obtenir la càmera*
 - *OnResume* → *si estava encès, tornar a encendre el flash*
 - *OnPause* → *si estava encès, apagar el flash*
 - *OnStop* → *alliberar la càmera*
- Permisos i features al **AndroidManifest**

