Activity et cycle de vie d'une Activité

frederic.guinand@univ-lehavre.fr

IUT informatique / Université du Havre

Plan

- le fichier AndroidManifest.xml
- exécution d'une Activity
- Activity : cycle de vie
- résumé
- Questions ?

Présentation

- Un composant Activity peut être considéré comme un composant qui fournit un écran pour interagir avec l'utilisateur au travers d'une interface graphique (GUI) décrite dans des fichiers xml séparés du code java
- une application peut être composée de plusieurs Activity correspondant à plusieurs actions et à plusieurs écrans
- chaque Activity *A* est indépendante et peut être démarrée par une Activity *B* n'appartenant pas nécessairement à l'application (si l'Activity *A* autorise une telle action)
- pour implémenter une Activity, le développeur doit étendre la classe Activity
  e.g. public class MyFirstApplication extends Activity

Activity

- lorsque une app est lancée par l'utilisateur, celle-ci démarre avec une première Activity qui peut être considérée comme le *main*
- examen du fichier AndroidManifest.xml de notre app.

AndroidManifest.xml (ancienne version)

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="mobapp.myfirstapplication"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="14"
        android:targetSdkVersion="19" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MyFirstActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Contraintes liées au SDK (anciennes versions)

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="mobapp.myfirstapplication"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="14"
        android:targetSdkVersion="19" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MyFirstActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Contraintes liées au SDK (nouvelle version)

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 26
    defaultConfig {
        applicationId "drivelog.guinand.lecteurcodebarres"
        minSdkVersion 23
        targetSdkVersion 26
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:26.1.0'
    implementation 'com.android.support.constraint:constraint-layout:1.0.2'
    implementation 'com.google.android.gms:play-services-vision:11.8.0'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.1'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.1'
}
```

AndroidManifest.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="drivelog.guinand.lecteurcodebarres">

    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.READ_SMS" />
    <uses-permission android:name="android.permission.SEND_SMS" />
    <uses-permission android:name="android.permission.RECEIVE_SMS" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Drivelog : Lecteur de codes à barres"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".LecteurCodesBarres"
            android:screenOrientation="landscape">
        </activity>
        <activity
            android:name=".Accueil"
            android:screenOrientation="portrait">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".ConfigurationLecteurDrivelog"
            android:screenOrientation="portrait" />
        <activity android:name=".SuperviseurDrivelog"></activity>
    </application>

</manifest>
```

Permissions

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="drivelog.guinand.lecteurcodebarres">

    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.READ_SMS" />
    <uses-permission android:name="android.permission.SEND_SMS" />
    <uses-permission android:name="android.permission.RECEIVE_SMS" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Drivelog : Lecteur de codes à barres"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".LecteurCodesBarres"
            android:screenOrientation="landscape">
        </activity>
        <activity
            android:name=".Accueil"
            android:screenOrientation="portrait">
            <intent-filter>
```

Composition de l'application

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="Drivelog : Lecteur de codes à barres"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity
        android:name=".LecteurCodesBarres"
        android:screenOrientation="landscape">
    </activity>
    <activity
        android:name=".Accueil"
        android:screenOrientation="portrait">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity
        android:name=".ConfigurationLecteurDrivelog"
        android:screenOrientation="portrait" />
    <activity android:name=".SuperviseurDrivelog"></activity>
</application>
```

Attributs de l'application

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="Drivelog : Lecteur de codes à barres"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity
        android:name=".LecteurCodesBarres"
        android:screenOrientation="landscape">
    </activity>
    <activity
        android:name=".Accueil"
        android:screenOrientation="portrait">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity
        android:name=".ConfigurationLecteurDrivelog"
        android:screenOrientation="portrait" />
    <activity android:name=".SuperviseurDrivelog"></activity>
</application>
```

sauvegarde sur carte SD autorisée

icône de l'application

nom de l'application tel qu'il apparaîtra sur l'écran du téléphone

Description de l'Activity

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="Drivelog : Lecteur de codes à barres"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity
        android:name=".LecteurCodesBarres"
        android:screenOrientation="landscape">
    </activity>
    <activity
        android:name=".Accueil"
        android:screenOrientation="portrait">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity
        android:name=".ConfigurationLecteurDrivelog"
        android:screenOrientation="portrait" />
    <activity android:name=".SuperviseurDrivelog"></activity>
</application>
```

Nom de la classe

Attribut de l'Activity (ici l'écran se mettra en mode portrait)

Signaux auxquels l'app va répondre

Avec cette action et cette catégorie cette activity pourra être lancée depuis un clic sur l'écran du téléphone

L'application est composée de 4 Activity

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="Drivelog : Lecteur de codes à barres"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity
        android:name=".LecteurCodesBarres"
        android:screenOrientation="landscape">
    </activity>
    <activity
        android:name=".Accueil"
        android:screenOrientation="portrait">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity
        android:name=".ConfigurationLecteurDrivelog"
        android:screenOrientation="portrait" />
    <activity android:name=".SuperviseurDrivelog"></activity>
</application>
```

L'application est composée de 4 Activités

AndroidManifest.xml

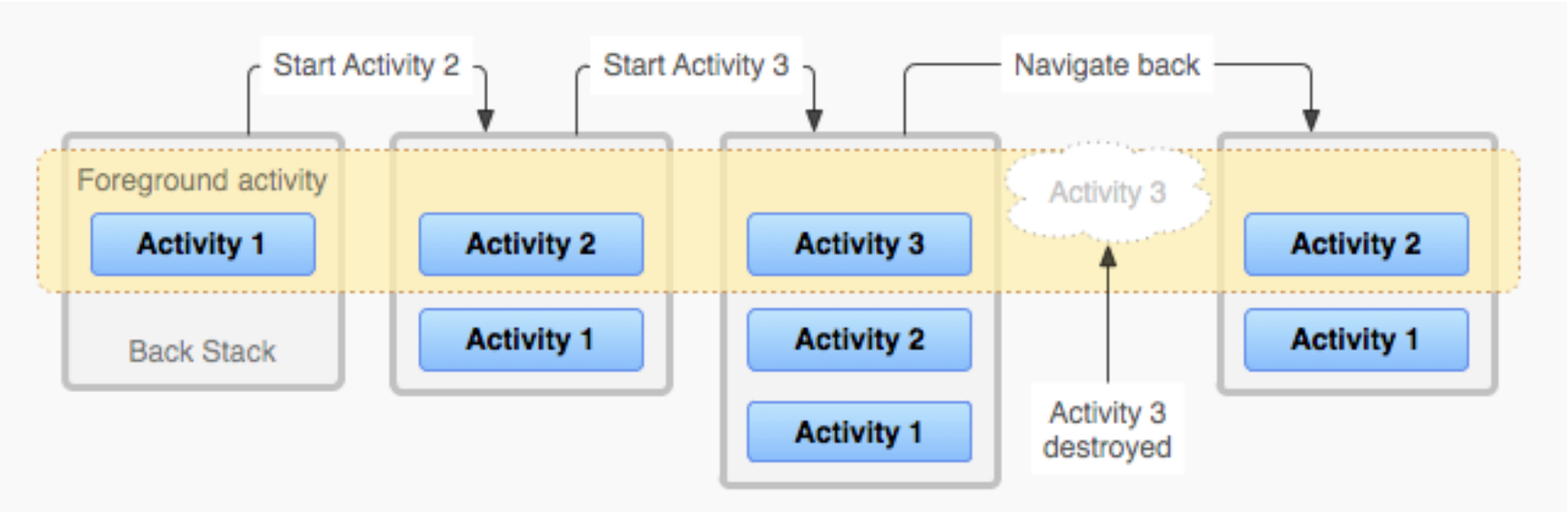
clef de voute d'une app

- point d'entrée
- composants (Activity, Receiver, etc.)
- valeurs minimum et courante du SDK
- les permissions
- les messages auxquels l'Activity est sensible : intent-filters

Exécution d'Activity

- une Activity peut démarrer une autre Activity pour, par exemple, exécuter de nouvelles actions
- les Activity sont organisées selon une pile (appelée Back Stack).
- l'Activity en cours d'exécution se situe au sommet de la pile
- lorsqu'une nouvelle Activity est démarrée, l'Activity courante est stoppée et la nouvelle est placée au sommet de la pile et devient la seule Activity en mesure d'interagir avec l'utilisateur
- lorsque l'Activity courante se termine, elle est dépilée et l'Activity suivante dans la pile reprend la main

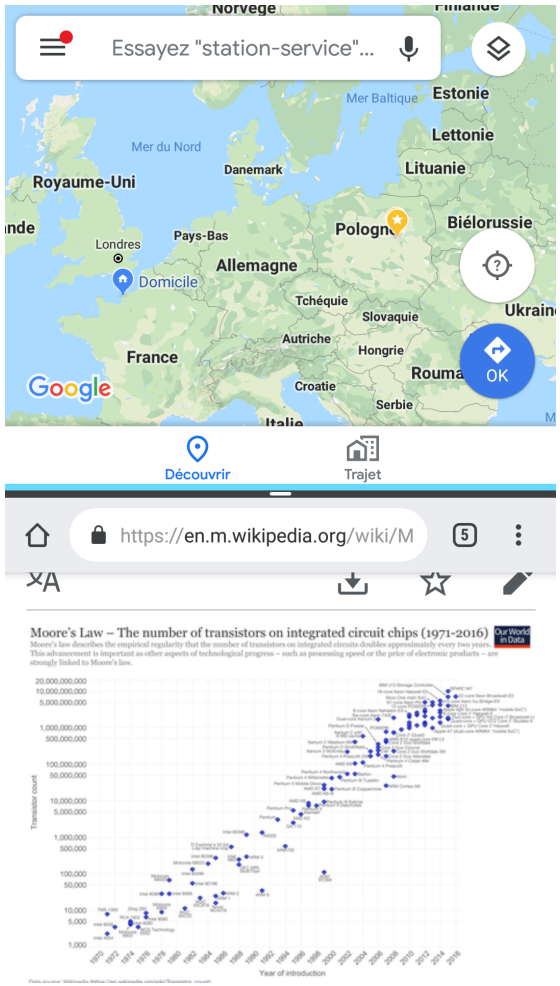
La pile des Activity



(http ://developer.android.com/guide/components/tasks-and-back-stack.html)

⇒ une seule Activity est exécutée à chaque instant (les autres processus en cours d'exécution sont des services)
⇒ Hé une minute !! Comment ça marche lorsque l'écran est divisé ?

Support Multi-Window



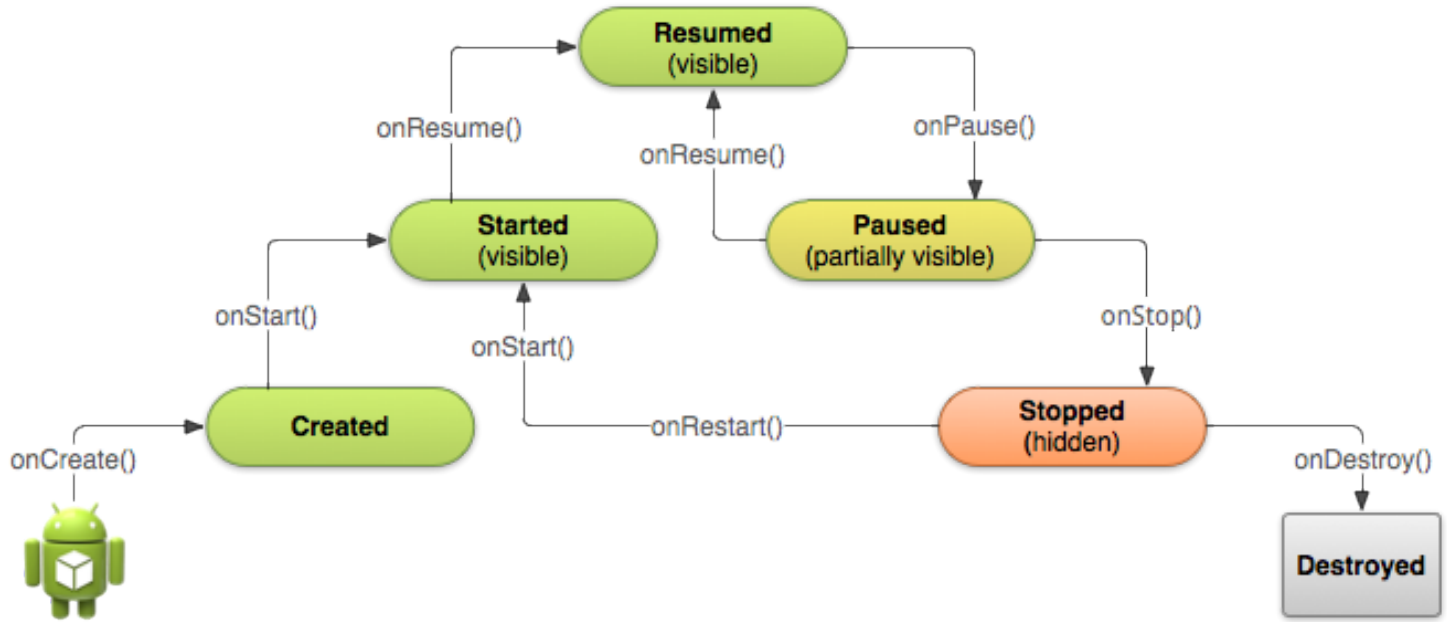
à partir d'Android 7.0 (API 24) vous pouvez observer deux apps exécutées côte-à-côte sur l'écran *split screen mode*

- *"Multi-window mode does not change the activity lifecycle"*[a].
- dans ce mode comme dans le mode classique, seule une Activity est active à un moment donné, celle avec laquelle l'utilisateur a interagit la dernière fois
- l'autre Activity est dans l'état `started` (cf. la suite)

a. extrait de developer.android.com/guide/topics/ui/multi-window.html

Activity - état d'une Activity

chaque Activity est caractérisée par son état



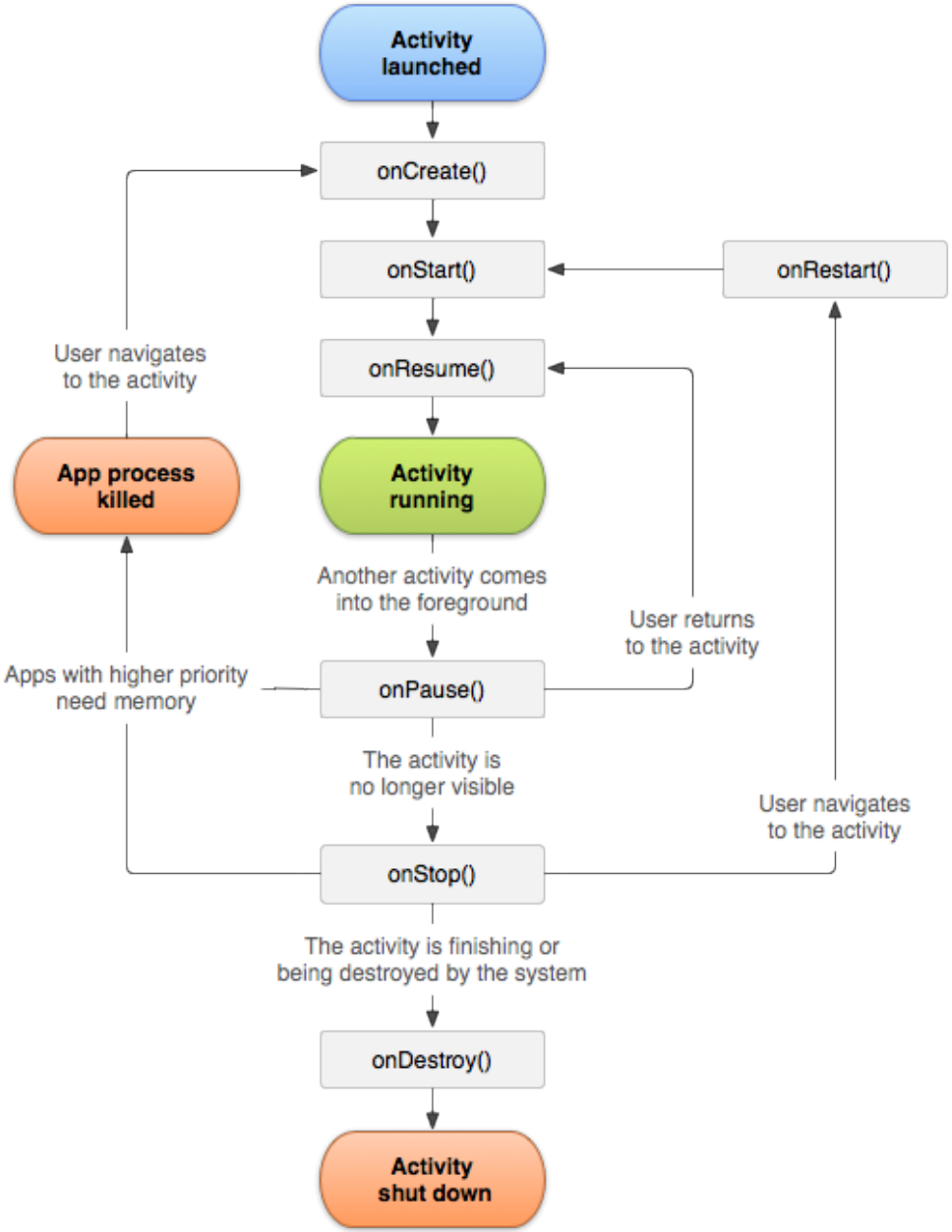
(http ://developer.android.com/training/basics/activity-lifecycle/starting.html)

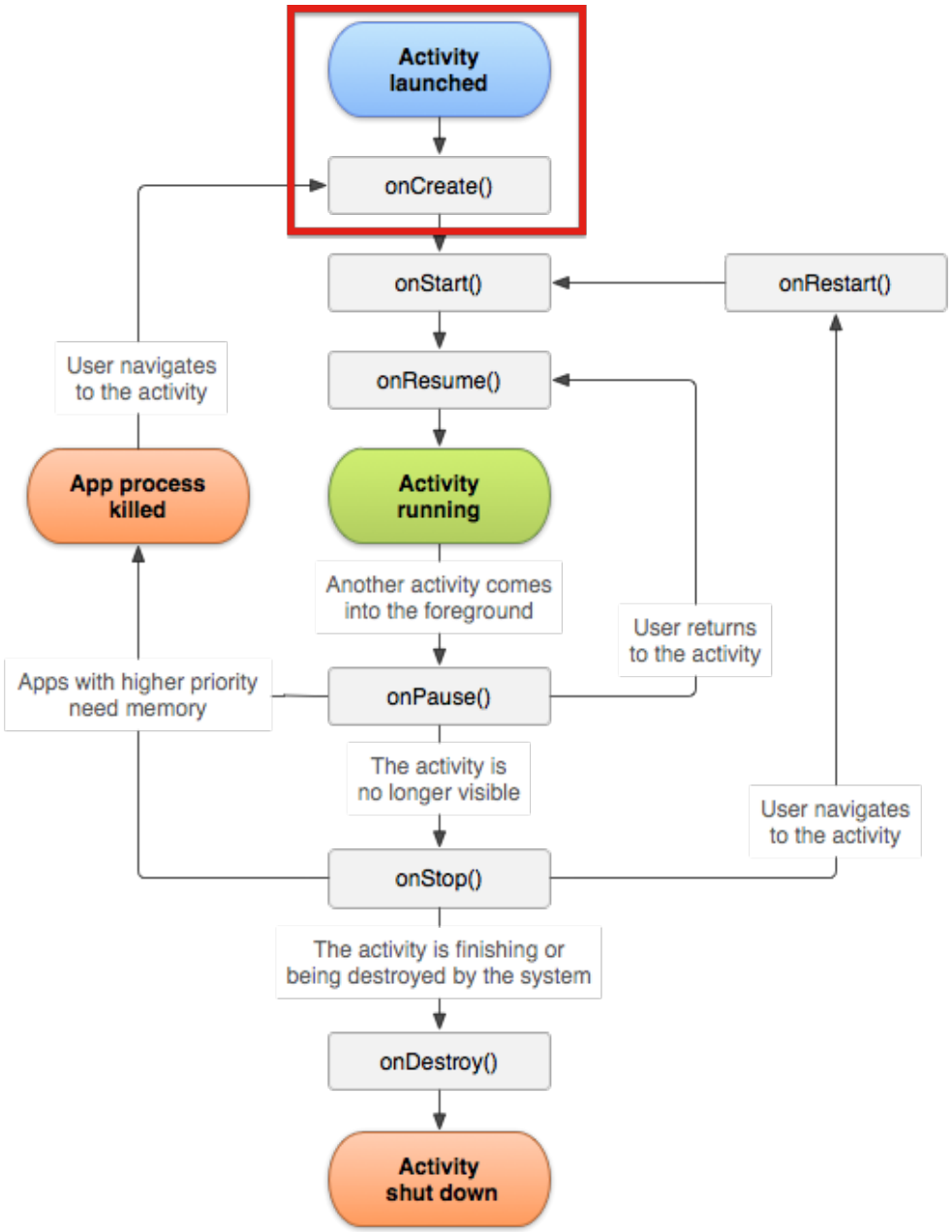
Callback Methods - méthodes de rappel (?)

chaque changement de l'état d'une Activity est notifiée à l'Activity elle-même au travers des *callback method* du cycle de vie de l'Activity

- onCreate(), [onRestart()], onStart(), onResume()
  $\rightarrow$ état Running (état Resumed)
  l'Activity a la main (user focus) et elle est visible
- onPause() $\rightarrow$ état Paused
  l'Activity est partiellement visible, aucun code n'est exécuté.
- onStop() $\rightarrow$ état Stopped
  l'Activity n'est plus visible, aucun code n'est exécuté.
- onDestroy() $\rightarrow$ R.I.P. (plus aucun état)

(source des images (page suivante) :
http ://developer.android.com/reference/android/app/Activity.html
http ://developer.android.com/guide/components/activities.html)

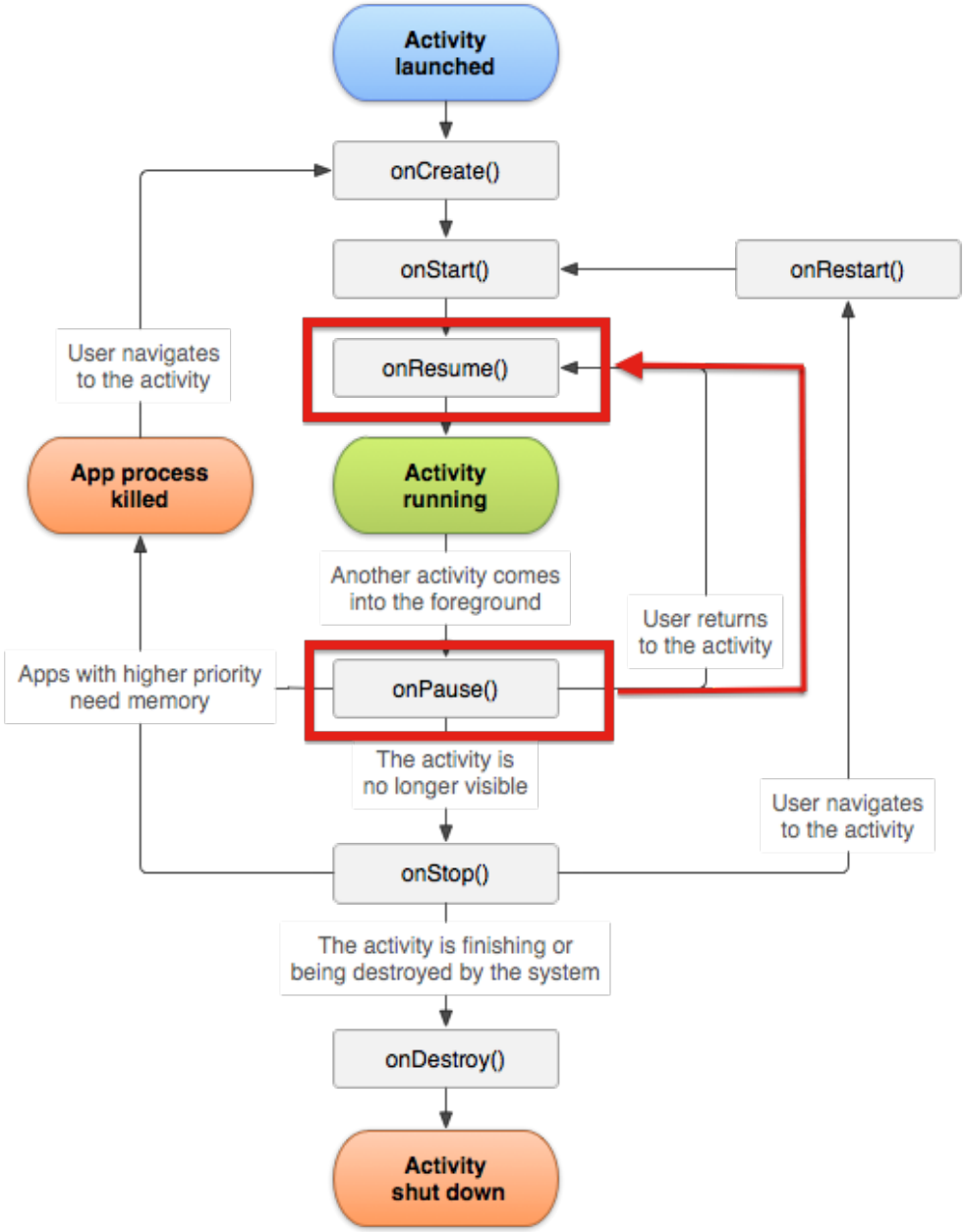

Activity launched
onCreate()
onStart()
onRestart()
onResume()
User navigates to the activity
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Activity - Callback Methods

onCreate()

- cette méthode est requise.
- c'est la première méthode qui est appelée par le système lorsque l'Activity est démarrée
- le code de *onCreate()* correspond à une sorte de phase d'initialisation, mise en place de l'interface utilisateur, définition des variables, etc.
- elle est également appelée lorsque l'Activity a été tuée par le système alors qu'elle était dans l'état *Paused* ou *Stopped*
- dans cette méthode est souvent appelée setContentView() qui permet la création de l'interface graphique par désérialisation à partir du contenu des fichiers XML


Activity launched
onCreate()
onStart()
onRestart()
onResume()
User navigates to the activity
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Activity - Callback Methods

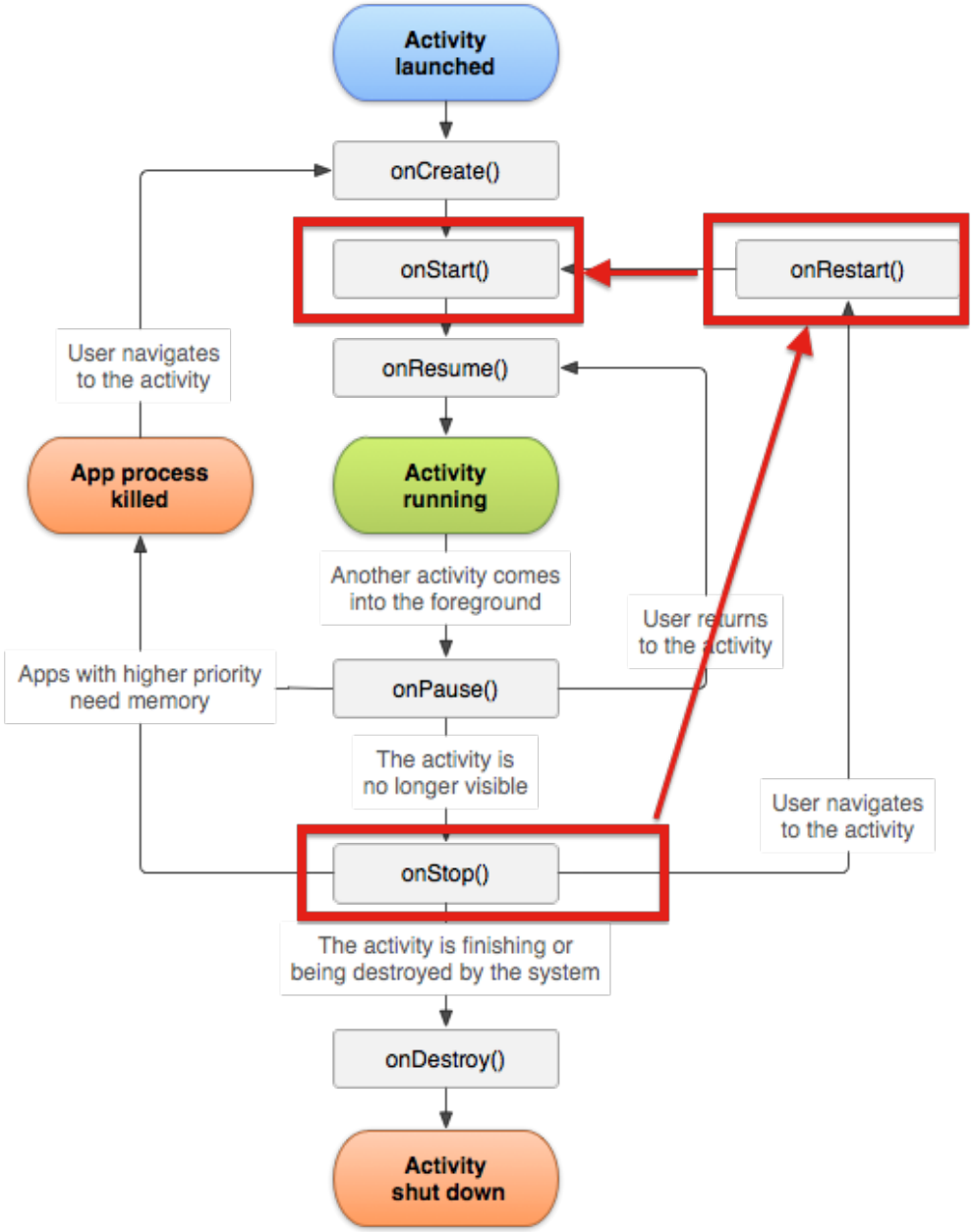
onPause()

- passage à l'état *Paused* lorsque l'Activity est partiellement visible (e.g. occultée en partie par un élément qui ne dépend pas de l'Activity), ou lorsqu'elle sera suivie par un état *Stopped*
- méthode utilisée pour libérer des ressources système : recepteurs de diffusion, gestionnaires de capteurs (e.g. GPS) et pour stopper les animations

état *Paused* $\rightarrow$ état *Running*

onResume()

- appelée chaque fois que l'Activity revient au premier plan
- la méthode doit réinitialiser les composants qui ont été mis en sommeil pendant l'appel à *onPause()*, redémarrer les animations (s'il y a lieu), etc.


Activity launched
onCreate()
onStart()
onRestart()
onResume()
User navigates to the activity
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Activity - Callback Methods

onStop()

- raisons du passage à l'état *Stopped* : appel téléphonique, changement d'app, lancement d'une nouvelle Activity, etc.
- état *Stopped* : UI n'est plus visible, aucun code exécuté mais l'Activity est toujours dans la mémoire du système
- onStop() : libération des ressources, opérations longues (écritures BD)

état *Stopped* $\rightarrow$ état *Running*

onRestart() $\rightarrow$ onStart()

- onRestart() est rarement utilisé, onStart() est utilisé
- passage de *Stopped* $\rightarrow$ *Running* : vérifications (GPS allumé ?), opérations effectuées dans onStart().

Activity - Cycle de vie

sauvegarde et restauration de l'état de l'instance de l'Activity

- dans certaines circonstances particulières, l'application peut être détruite et recréée
- pour éviter la perte des données (score en cours, valeurs temporaires, etc.), il est nécessaire de sauvegarder l'état de l'instance de votre Activity.

Activity - Cycle de vie

sauvegarde et restauration de l'état de l'instance de l'Activity

- une Activity détruite ne peut pas retrouver seule son état avant sa destruction

  $\Rightarrow$ il faut sauvegarder les données
- Android fournit des méthodes pour le faire :
  - sauvegarde avec la méthode onSaveInstanceState()
  - restauration avec la méthode onRestoreInstanceState()

Views

- l'état des *Views* est automatiquement sauvegardé à la condition qu'elles aient un identifiant ID (android :id)

Activity - Cycle de vie

onSaveInstanceState()

- appel : onSaveInstanceState(Bundle theBundle)

  le système fournit Bundle (une sorte de "sac à données")

  il peut être utilisé pour stocker les données selon un format de type clef-valeur :

  - String : theBundle.putString("nomdelachaine",valeur) ou
  - Integer : theBundle.putInt("nomdelentier",valeur)

onRestoreInstanceState()

- appel : onRestoreInstanceState(Bundle sacDeDonnees)
  - sacDeDonnees.getString("nomdelachaine")
  - sacDeDonnees.getInt("nomdelentier")

Activity - Cycle de vie - résumé

- Activity : un composant dont le but est de permettre l'interaction avec l'utilisateur
- Une Activity possède un cycle de vie qui commence avec un appel à la méthode *onCreate()* (méthode requise) et qui se termine avec la méthode *onDestroy()* (qui peut être omise)
- Une Activity affiche une GUI produite par la méthode *setContentView()* selon le principe de la désérialisation à partir de données stockées dans le répertoire res (ressources) dans des fichiers XML
- chaque changement dans la configuration de l'app $\rightarrow$ destruction $\rightarrow$ recréation

  $\Rightarrow$ les données doivent être sauvegardées en implémentant les méthodes *onSaveInstanceState()* et *onRestoreInstanceState()*

Références

- http ://developer.android.com/training/basics/activity-lifecycle/index.html
- http ://developer.android.com/reference/android/app/Activity.html
- http ://developer.android.com/training/basics/activity-lifecycle/starting.html
- http ://developer.android.com/training/basics/activity-lifecycle/pausing.html
- http ://developer.android.com/training/basics/activity-lifecycle/stopping.html
- http ://developer.android.com/training/basics/activity-lifecycle/recreating.html
- http ://developer.android.com/guide/topics/ui/notifiers/toasts.html