

Chapitre III :

Cycle de vie d'une application Android

Programmation mobile – ISIMa 2021

Mme. Hajer BEN SALEM

Cycle de vie d'une application Android

- ⦿ Les composants d'une application Android ont un cycle de vie :
 - Un **début** quand Android les instancie pour répondre aux intents
 - Une **fin** quand les instances sont détruites
 - Entre les deux, ils peuvent être :
 - ⦿ **Actifs** ou **inactifs**
 - ⦿ **Visibles** ou **invisible**

Début

Active / inactive
Visible / Invisible

Fin

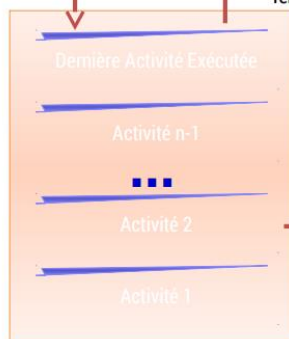
Pile d'activités

- ⊙ Les activités dans une application sont gérées sous forme de **pile**.
- ⊙ Quand une nouvelle activité démarre, elle est placée en haut de la pile et devient l'activité **en exécution**
 - L'activité précédente reste en dessous de la pile
 - Elle ne revient au premier plan que si la nouvelle activité soit fermée
- ⊙ Si l'utilisateur clique sur le bouton 'retour', l'activité suivante dans la pile devient active

Pile d'activités

Nouvelle Activité démarrée

Appui sur le bouton
Retour ou activité
fermée



Supprimée pour
libérer des
ressources

États d'une activité

Essentiellement 3 états :

○ **Active** ou **en exécution**

- Elle est au 1^{er} plan de l'écran
- C'est l'activité ciblée par les actions de l'utilisateur

○ **En pause**

- Elle a perdu focus, mais encore partiellement visible
- Une autre activité est en haut de pile, mais elle est soit transparente, soit ne couvre pas tout l'écran

○ **Arrêtée**

- Complètement recouverte par une autre activité
- Ses informations sont encore chargées, mais n'est plus visible
- Peut être tuée par le système

Événements d'un cycle de vie

○ Quand une activité passe d'un état à un autre, le Framework Android appelle les méthodes de transition correspondantes:

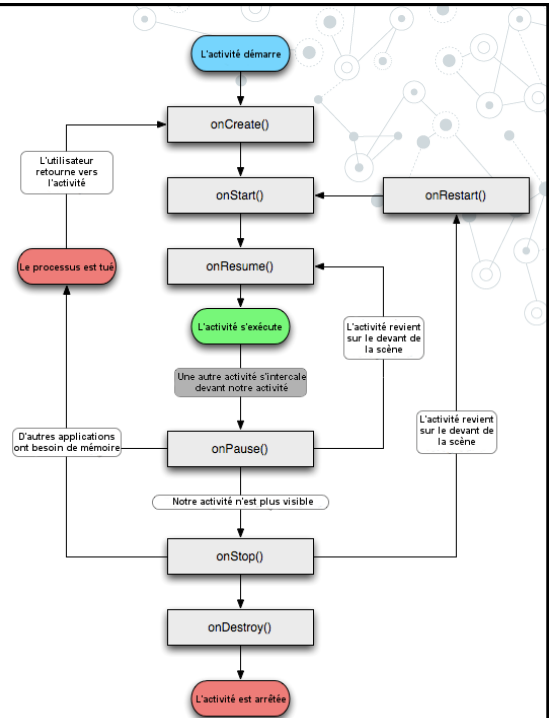
- void **onCreate(Bundle savedInstanceState)** : **OBLIGATOIRE**
- void **onStart()**
- void **onRestart()**
- void **onResume()**
- void **onPause()** : Recommandée
- void **onStop()**
- void **onDestroy()**

Cycle de vie d'une application Android

onCreate()

- Appelée à la création d'une activité
- Initialisation de tous les éléments
- Un bundle est passé à cette méthode, indiquant l'état précédent de l'activité

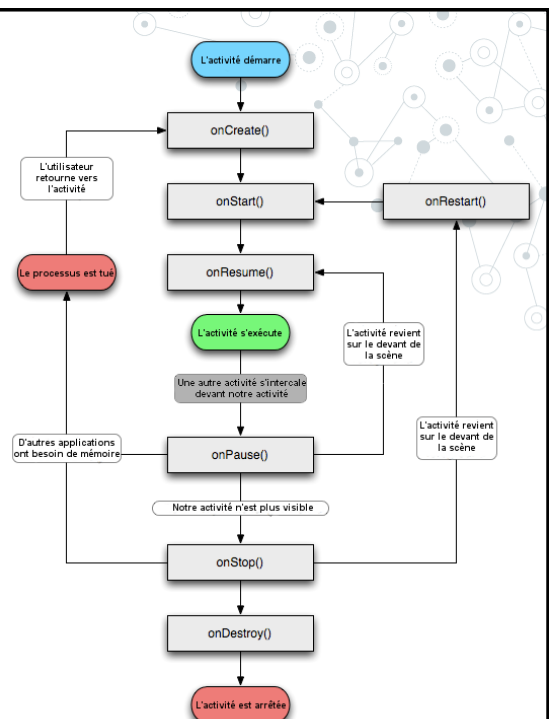
Toujours suivie de **onStart()**



Cycle de vie d'une application Android

onStart()

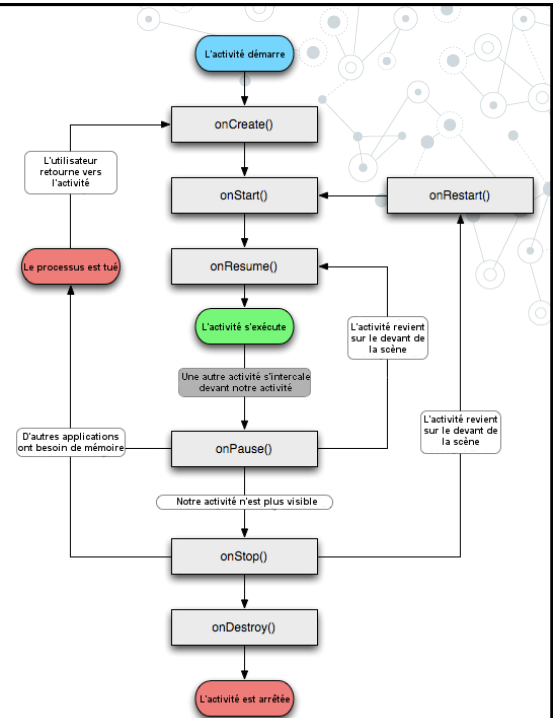
- Appelée juste avant que l'activité ne devienne visible, visible par :
 - **onResume** si l'activité revient en 1^{er} plan
 - **onStop** si l'activité est cachée



Cycle de vie d'une application Android

onRestart()

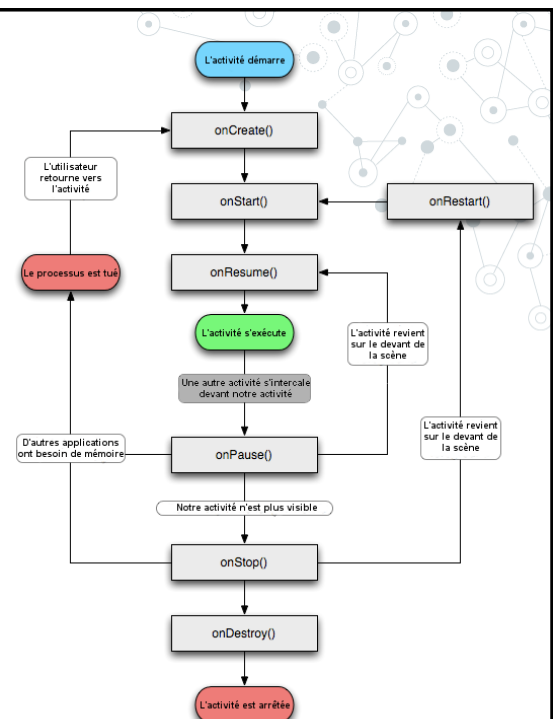
- Appelée quand l'activité va redémarrer après avoir été stoppée
- Toujours suivie de **onStart()**



Cycle de vie d'une application Android

onResume()

- Appelée juste avant que l'activité ne commence à interagir avec l'utilisateur
- À ce point, l'activité est en haut de pile
- Toujours suivie de **onPause()**



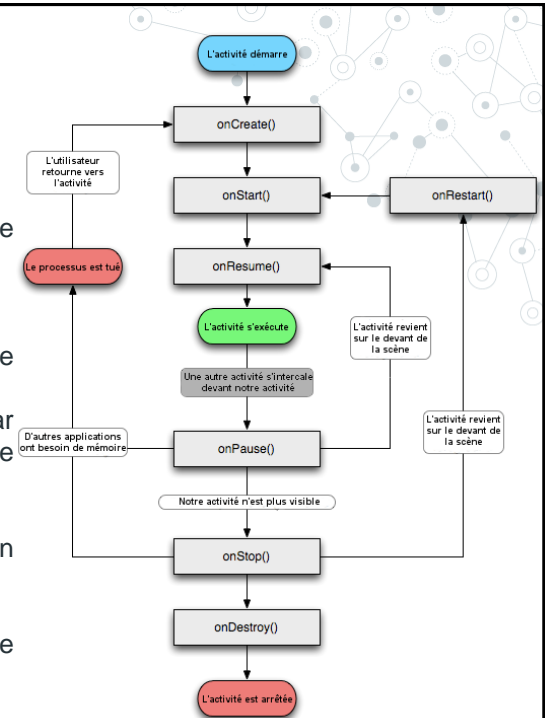
Cycle de vie d'une application Android

onPause()

- Appelée quand le système va démarrer une autre activité
- Utilisée typiquement pour
 - Consigner les données non sauvegardées
 - Arrêter les animations ou toute autre chose qui consomme la mémoire
- Ne doit pas consommer beaucoup de temps, car l'activité suivante ne va démarrer que si cette méthode retourne
- Suivie de :

● **onResume()** : si l'activité est rechargée en 1^{er} plan
 ● **onStop()** : si l'activité devient invisible

Dans cet état, l'activité peut être tuée par le système.



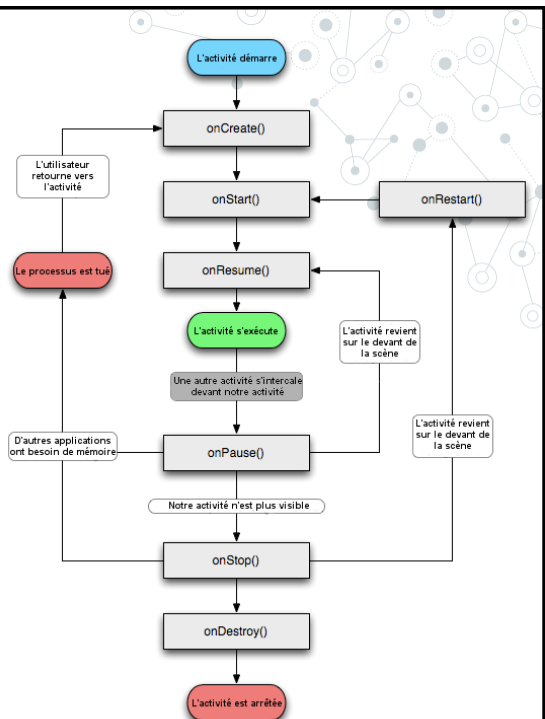
Cycle de vie d'une application Android

onStop()

- Appelée quand l'activité n'est plus visible à l'utilisateur
- Peut arriver si :
 - L'activité est détruite
 - Une autre activité a repris son exécution et l'a recouverte
- Suivie par :

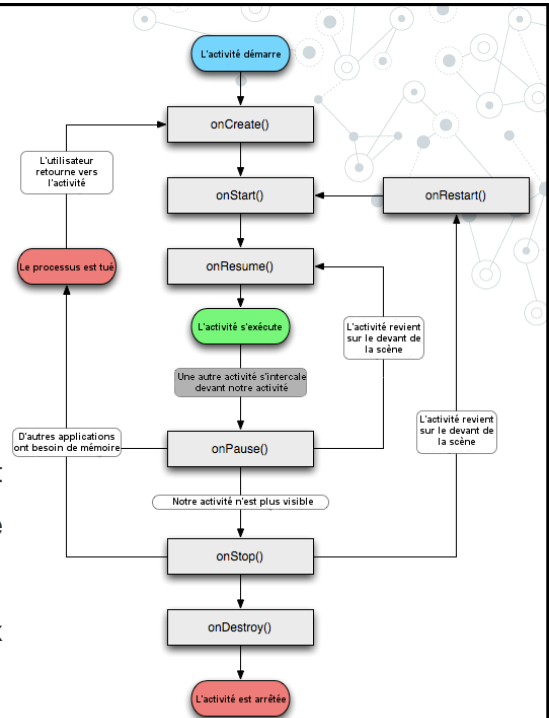
● **onRestart()** si l'activité recommence à interagir avec l'utilisateur
 ● **onDestroy()** si l'activité va disparaître

Dans cet état, l'activité peut être tuée par le système



Cycle de vie d'une application Android

- ⦿ `onDestroy()`
 - Appelée quand l'activité est détruite
 - Dernier appel que l'activité va recevoir
 - Peut intervenir si:
 - ⦿ L'activité se termine (appel de `finish`)
 - ⦿ Le système détruit temporairement cette instance pour gagner de l'espace
- ⦿ On peut distinguer entre ces deux scénarios avec la méthode `isFinishing()`



Démonstration

- ⦿ Création d'une nouvelle application
- ⦿ Afficher un 'Toast' pour chaque étape du cycle de vie