Développement mobile

Aymeric Pete ( ?)

# Le principe de pile des activités (FIFO)

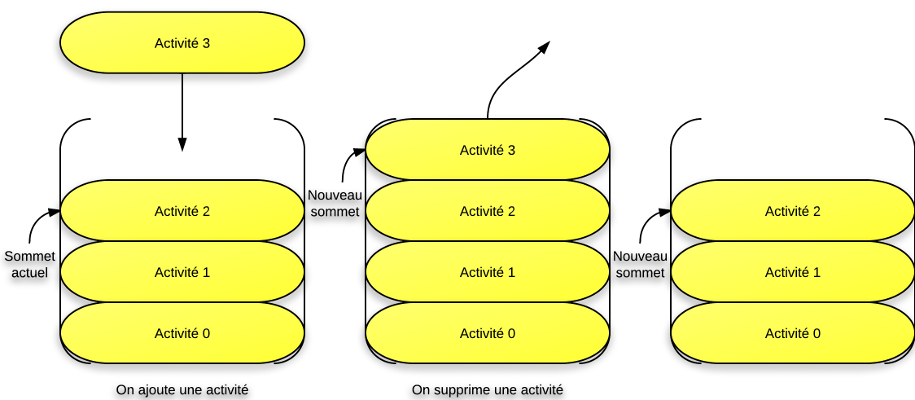


Schéma Principe des piles d'activités (fifo)

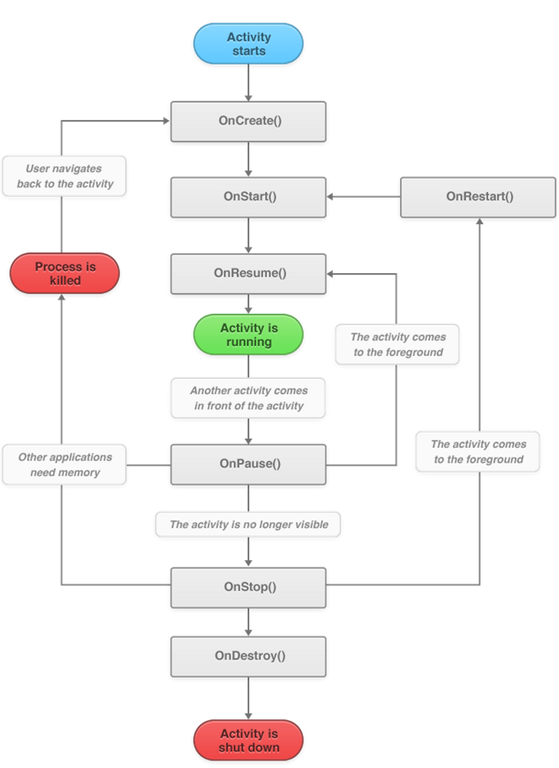


Schéma Cycle de vie d'une activité

L’évènement onResume() est appelé lorsque l’activité de l’application s’arrête inopinément (ex panne de réseau, appel, etc…).

Gradle**Moteur de production** fonctionnant sur la plateforme Java. Il permet de construire des projets en Java, Scala, Groovy voire C++. Gradle permet d’écrire des tâches de construction dans un fichier de construction en utilisant le langage Groovy. Il est possible d’importer des tâches standards qui permettent de construire des programmes utilisant un ou plusieurs langages (Java, Groovy, etc…) ou qui permettent d’exécuter des activités d’ingénierie logicielle telles qu’exécuter des tests unitaires, assurer la qualité du code (SonarQube, Checkstyle, etc…).

ManifesteLe manifeste est un **fichier XML** qui décrit les métadonnées de l’application (nom, icône, thème, etc…). Il est surtout là pour lister les différentes fonctionnalités requises au niveau matériel et droit d’accès de l’application. C’est également le fichier dans lequel sont listées les tailles d’écran qui seront supportées par l’application ainsi que la liste de toutes les activités présentes dans l’application. Ce fichier servira à la fois au téléphone pour savoir comment manipuler l’application et au Play Store pour savoir si l’appareil qui est en train de le parcourir est compatible avec l’application. Cela permettra aux applications d’être invisibles aux yeux des utilisateurs qui n’ont pas les fonctionnalités requises pour les faire fonctionner.

Java

Java est un langage de programmation orienté objet qui reprend en grande partie le langage C++ mais qui a été épuré des concepts les plus subtils du C++. Ce langage permet de développer des applications client-serveur. Côté client, les applets sont à l’origine de la notoriété du langage. Mais c’est surtout côté serveur que Java s’est imposé dans le milieu de l’entreprise grâce aux servlets, le pendant serveur des applets, et plus récemment les JSP (JavaServer Pages) qui peuvent se substituer à PHP, ASP et ASP.NET.

Drawable

Drawable est un répertoire créé par défaut dans lequel on peut stocker des ressources. On y trouve les images matricielles (de type PNG, JPEG ou encore GIF) ainsi que des fichiers XML qui permettent de décrire des dessins (ce qui donne des images vectorielles qui ne se dégradent pas quand on les agrandit).

Layout

Layout est également un répertoire créé par défaut dans lequel on peut stocker des ressources. On y trouve les fichiers XML

Fichier String = tout ce qui est texte

## Les interfaces graphiques

## Les composants graphiques

* TEXTVIEW
* IMAGEVIEW
* LINEARLAYOUT
* RELATIVELAYOUT
* SCROLLVIEW
* HORIZONTALSCROLLVIEW
* IMAGEBUTTON
* BUTTON

## Les principaux attributs des composants

Largeur & hauteur

android:layout\_width  
android:layout\_height  
android:layout\_weight

Positionnement

android:gravity  
android:layout\_gravity  
android:orientation  
android:padding=”10dp”  
android:layout\_margin=”0dp”

Java

Les attributs

Public = attribut accessible par tout

Private = attribut accessible uniquement par la classe

Protected = attribut accessible uniquement par la classe héritée

Soit des éléments de son layout, soit des variables.

Les différents types de variables sont : String (chaînes de caractères), int (numérique) et boolean (true ou false).

A quoi sert ‘*this*’ ?

*This*  correspond au pointer sur un objet courant (la classe dans laquelle je suis): ici, une activité car on hérite de AppCompatActivity.

Le *SetContentView*

Le *SetContentView* dans le *onCreate* sert à charger le layout correspondant : setContentView(R.layout.activity\_main);

Les imports

Les imports sont gérés automatiquement mais il faut les supprimer soit même si ils ne sont plus utilisés.

La gestion des évènements

Sur un bouton

A déclarer dans le onCreate de l’activité

p\_button.setOnClickListener(

new View.OnClickListener() {

public void onClick(View v) {

p\_activity.startActivity(new Intent(p\_activity, MainActivity.class));

}

}

);

Comment ouvrir un nouvel écran ?