

Activity

DÉPARTEMENT D'INFORMATIQUE – CENTRE UNIVERSITAIRE DE MILA
2019/2020

1. Activité Android

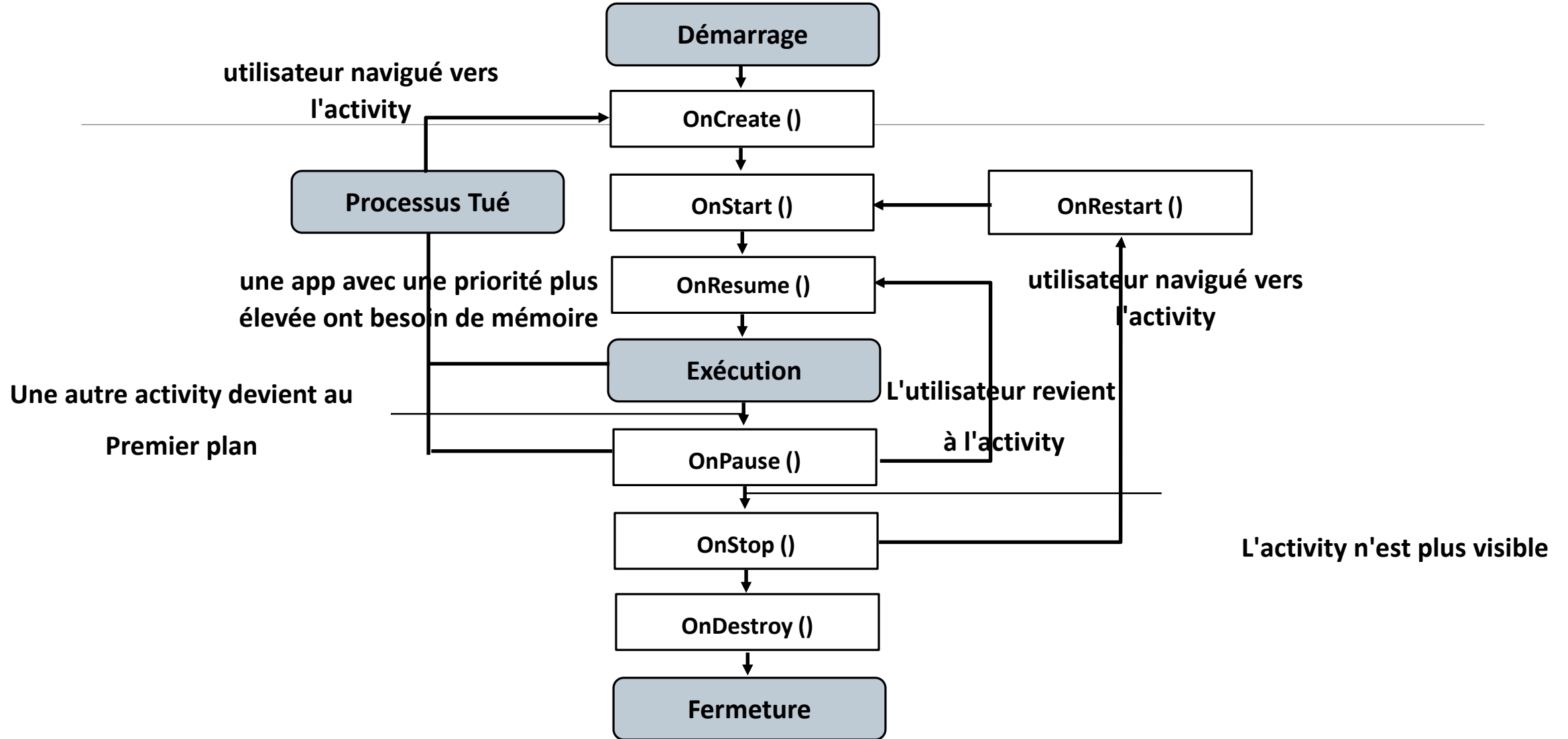
- Classe qui hérite d' Activity ou d'une classe dérivée d' Activity.
- On surcharge certaines méthodes qui sont appelées par Android pour définir le comportement (Callback méthodes) :

1. Activity Android

- onCreate: Ceci est le premier callback et appelé lorsque l'activité est créée pour la première fois.
- onStart :Ce callback est appelé lorsque l'activité devient visible pour l'utilisateur.
- onResume: Ceci est appelé lorsque l'utilisateur commence à interagir avec l'application.

1. Activity Android

- onPause: L'activité en pause ne reçoit aucune entrée de l'utilisateur et ne peut exécuter aucun code.
- onStop :Ce callback est appelé lorsque l'activité n'est plus visible.
- onRestart: Ce callback est appelé lorsque l'activité redémarre après avoir arrêté.
- onDestroy: Ce callback est appelé avant que l'activité soit détruite par le système.



Cycle de vie d'une activité

1.2. Squelette d'une activité

```
import android.app.Activity ;  
import android.os.Bundle ;  
public final class MainActivity extends Activity {  
    .....  
}
```

1.2. Squelette d'une activité

```
public void onCreate(Bundle
savedInstanceState) {
    Super.onCreate(savedInstanceState) ;
    // Placez votre code ici
}

public void onStart() {
    super.onStart() ;
    // Placez votre code ici
}
```

Squelette d'une activité

```
public void onRestart() {  
    super.onRestart() ;  
    // Placez votre code ici  
}  
  
public void onResume() {  
    super.onResume() ;  
    // Placez votre code ici  
}
```


Squelette d'une activité

```
public void onDestroy(){  
    // Placez votre code ici  
    super.onDestroy() ;  
}  
  
public void onStop(){  
    // Placez votre code ici  
    super.onStop() ;  
}
```

Squelette d'une activité

```
public void onPause(){
```

```
// Placez votre code ici
```

```
super.onPause() ;
```

```
}
```

```
public void onSaveInstanceState(Bundle savedInstanceState){
```

```
// Placez votre code ici
```

```
super.onSaveInstanceState(savedInstanceState) ;
```

```
}
```

Cycle de vie d'une activité

Android se réserve le droit de tuer le processus linux d'une activité s'il n'y a plus assez de ressources(mémoire)

Les règles sont les suivantes :

- Une activité en premier plan n'est tuée que si c'est elle qui consomme trop de ressources.
- Une activité en arrière plan ou non visible peut être tuée.
- Lorsqu'une activité a été tuée, si on revient dessus elle est relancée (onCreate)
- On peut sauvegarder l'état (c'est-à-dire les propriétés) d'une activité (dans onPause) pour le retrouver lorsqu'elle est recréée par le paramètre transmis à onCreate().

Cycle de vie d'une activité

Etats principaux :

- ✓ Active: Après l'exécution de `onResume`
- ✓ Suspendue : Après l'exécution de `onPause`
- ✓ Arrêtée: Après l'exécution de `onStop`
- ✓ Terminée: Après l'exécution de `onDestroy`

2. Intent

- Android utilise `<intent>` pour la communication.
- Intent est l'objet utilisé dans android pour le passage des informations entre les activités d'une même application et il peut assurer la communication entre les applications.

2.1 Utilité des intents dans android

- Android utilise les intents pour faciliter la communication entre les composants de ses applications tel que: les activités, les service et les récepteur (broadcast recievers).
 - Intent pour les activités: chaque écran d'une application android représente une activité. Pour passer à une autre activité il faut passer un objet de type intent au méthode *startActivity()*, ce qui permet de démarrer l'activité voulue en lui passe d'éventuelles informations.

2.1 Utilité des intents dans android

- Intent pour les services : les service travaillent dans le font et par la suite elle ne nécessitent aucun interface utilisateur. Intente peut lancer un service pour accomplir une tache à la fois comme le téléchargement de quelques fichiers.
- Afin de démarrer un service, intent utilise la méthode `startService()`.

2.2 Exemple d'Intent

- Par exemple: intent facilite le passage d'une activité à une autre en appelant `startActivity(intent)` comme suit:

```
Intent intent = new Intent ( getApplicationContext (),  
SecondActivity . class );
```

```
Intent.putExtra(" Name", "Guettiche");
```

```
Intent.putExtra(" Firstname", "Mourad");
```

```
startActivity ( intent );
```


2.2 Exemple d'Intent

Récupération des information

```
Intent intent = getIntent();
```

```
String Name = intent.getStringExtra("Name");
```

```
String Firstname = intent.getStringExtra("FirstName");
```

//d'autre type d'information:

```
Integer user_id = intent.getIntExtra("ID");
```

```
float user_rating = intent.getFloatExtra("RATING");
```



2.3 Intent et Bundle

Utilisation d'un objet de type Bundle

```
Bundle extras = new Bundle();  
extras.putString("NAME","GUETTICHE");  
extras.putInt("ID", 21);  
extras.putIntArray("tab", [1, 2, 3, 4, 5]);  
...
```

2.3 Intent et Bundle

```
Intent intent = getIntent();
```

```
Bundle extras = intent.getExtras();
```

```
//Extraction des informations
```

```
String nom = extras.getString("USER_NAME");
```

```
Integer ID= extras.getInt("ID");
```

```
Int tab[]=extra.getIntArray(tab);
```

2.4 Types d' Intents:

- Nous avons deux types d'intents: explicite et implicite:
 - Intent explicite: est utilisé pour la connexion interne c'est-à-dire la communication entre les éléments d'une même application. il assure donc la navigation et le passage de l'information interne.
 - Dans l' explicite intent on doit fournir explicitement le nom du composant ciblé par l'intent,
 - Dans l'implicite intent il ne faut pas mentionner le nom du composant ciblé pas l'intent mais plutôt l'action à achever en plus que cette action doit être performée par une autre application.

2.4.1 Implicite Intents

L' exemple suivant illustre un implicite intent pour l'ouverture d'une page web:

```
Intent intentObj = new Intent ( Intent . ACTION_VIEW );  
intentObj . setData ( Uri . parse ( "http://www.android.com" ));  
startActivity ( intentObj );
```

Lors de l' exécution de ce code, Android démarre automatiquement le navigateur pour l'ouverture de la page web demandée.

2.4.1 Implicite Intents

Un objet implicite intent contient les informations (champs) suivantes:

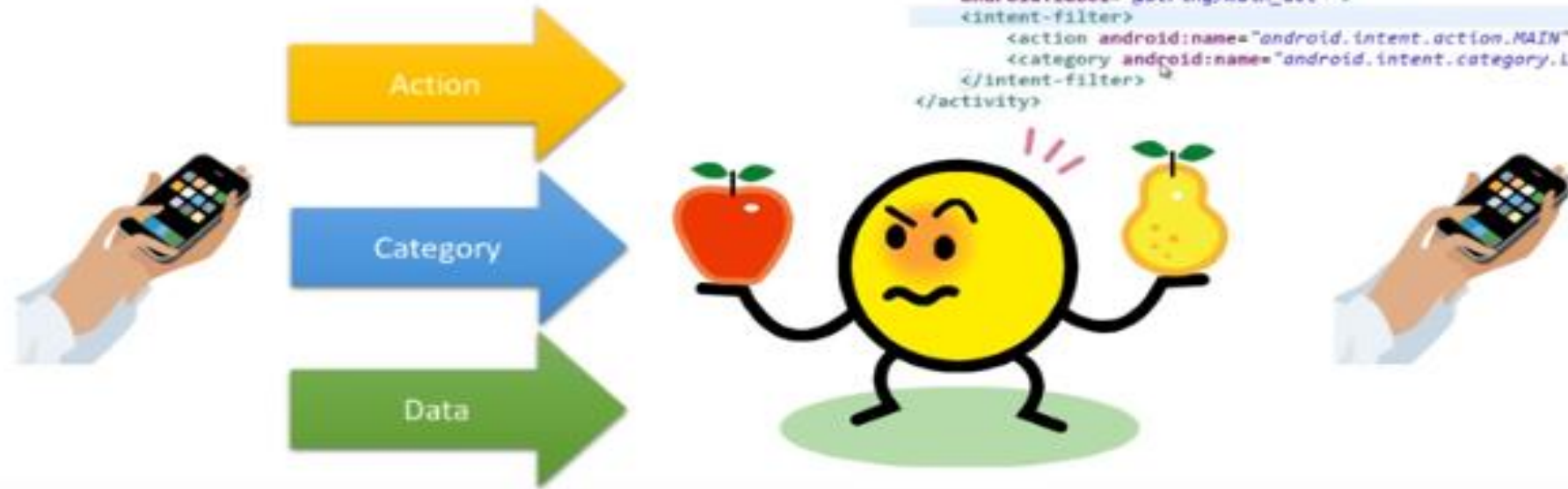
- L'action à réaliser, sous forme de chaîne de caractères
- Les données: contenu URL (Uniform Resource Locator) ou MIME (Multipurpose Internet Mail Extensions)
- Des données supplémentaires (extra) sous forme de paires clef/valeur.

2.4.2 Les actions

- L'action d'un intent est une chaîne de caractères qui symbolise le traitement à déclencher.
- De nombreuses constantes sont définies dans le SDK pour les actions nativement disponibles comme:
 - `Intent.ACTION_WEB_SEARCH` pour demander de réaliser une recherche sur Internet
 - `Intent.ACTION_CALL` pour passer un appel téléphonique.

IntentFilter

What is an IntentFilter?



```
<activity  
  android:name="slidenerd.vivz.MainActivity"  
  android:label="@string/main_act" >  
  <intent-filter>  
    <action android:name="android.intent.action.MAIN" />  
    <category android:name="android.intent.category.LAUNCHER" />  
  </intent-filter>  
</activity>
```


2.4.3 Exemple:

L' exemple suivant illustre un implicite intent pour la navigation en google Maps:

```
Intent intentObj = new Intent ( Intent . ACTION_VIEW );  
intentObj . setData ( Uri . parse ( "geo:36.462610,6.214816" ));  
startActivity ( intentObj );
```

Lors de l' exécution de ce code, Android démarre automatiquement le google Maps pour localiser l' endroit spécifier.

2.4.3 Example:

```
intent.setAction(Intent.ACTION_SEND);  
    String [] to={"guettiche.mourad@gmail.com", "Guet_mourad@yahoo.fr"};  
intent.putExtra(Intent.EXTRA_EMAIL,to);  
intent.putExtra(Intent.EXTRA_SUBJECT,"this was sent from my app");  
intent.putExtra(Intent.EXTRA_TEXT,"Hello, how are you?");  
    intent.setType("message/rfc822");  
    startActivity(intent);
```