

SIL1

Programmation mobile sous Android

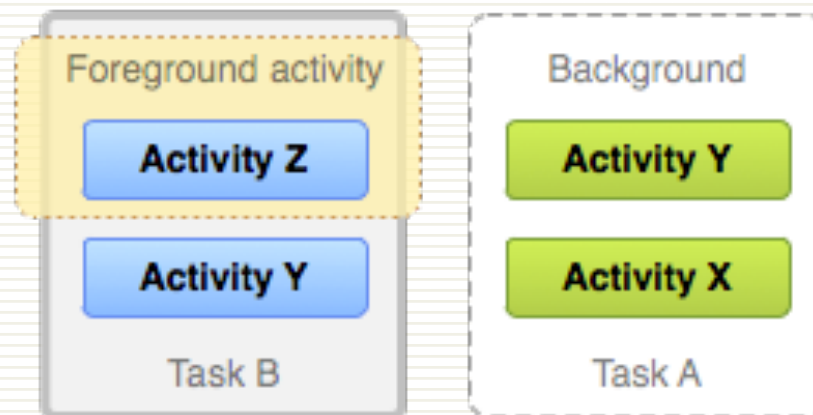
Tâches et piles d'activités

Intents explicites

Source : <http://developer.android.com/>

Application android

- ✓ Contient plusieurs activités liées entre elles
- ✓ Chaque application est indépendante :
 - Tourne dans son propre processus appelé **tâche**
 - Une tâche contient **une pile** pour gérer les activités



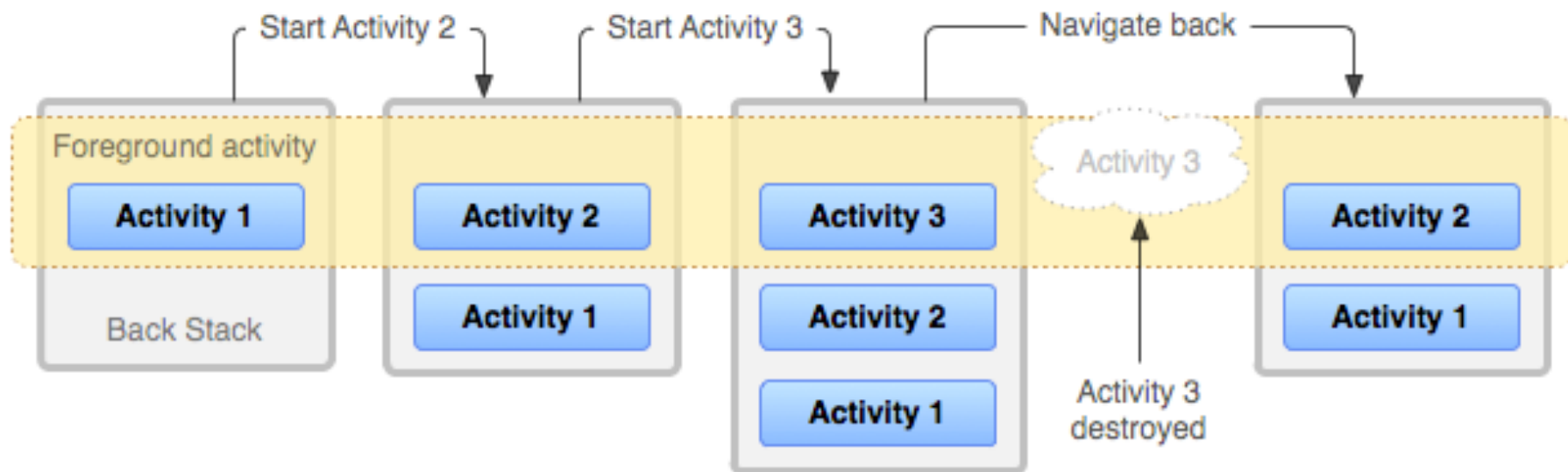
Tâche

- ✓ Lorsqu'une application est demandée par l'utilisateur :
 - Si la tâche existe, elle est placée en premier plan
 - Si la tâche n'existe pas, elle est créée et placée en premier plan
 - ◆ Le fichier AndroidManifest.xml contient l'activité principale à placer en premier dans la pile
 - ◆ Exemple :

```
<activity
    android:name=".MainActivity"
    android:label="Android : TP1" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

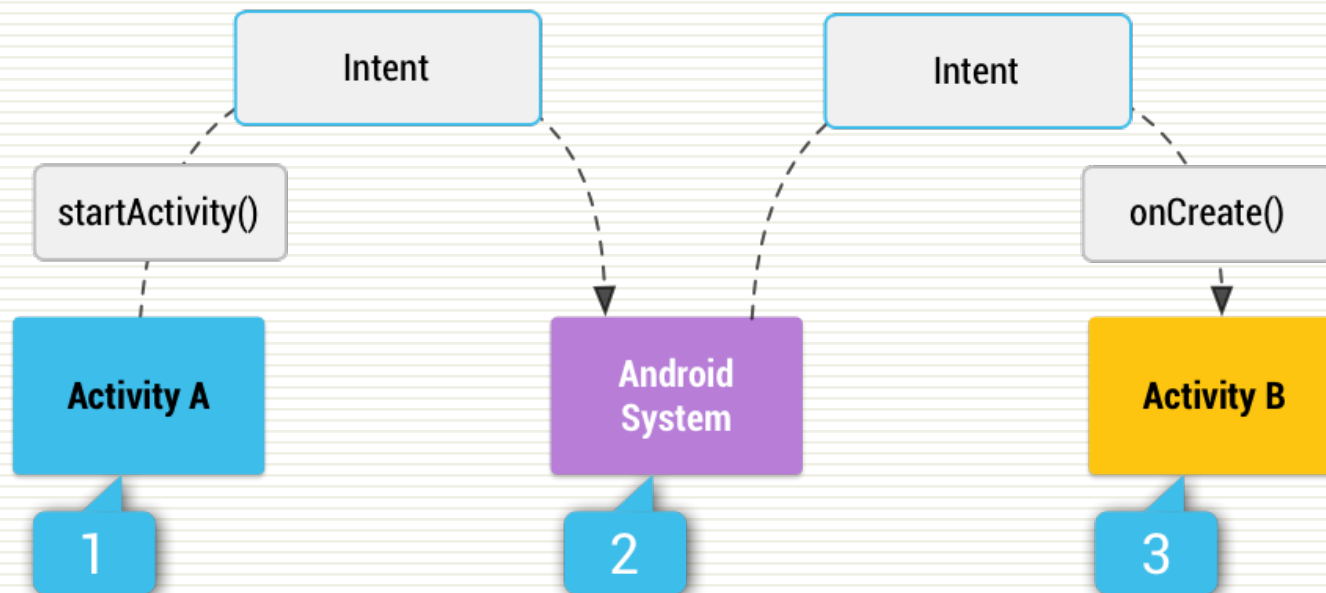
Pile d'activités

- ✓ Chaque fois qu'une activité est lancée, elle est placée sur la **pile** appelée aussi back stack



Créer une nouvelle activité

- ✓ Faire une demande d'intention au système
 - L'activité A demande la création d'une activité B



- Une intention – classe **Intent** – est :
 - ◆ Soit **explicite** : appel d'une activité de l'application
 - ◆ Soit **implicite** : demande d'un besoin proposé par une autre application (appel tél., appareil photo, etc.) ... *semaine prochaine*

Exemple d'intention explicite

- ✓ L'activité MainActivity demande la création de l'activité Exercice1Activity

```
// Création d'une intention concernant l'activité Exercice1Activity  
Intent intent = new Intent(this, Exercice1Activity.class);  
  
// Lancement de la demande de changement d'activité  
startActivity(intent);
```

Envoie de données entre activités (1/2)

- ✓ Utilisation de la méthode putExtra(...) de la classe Intent
 - Exemple : envoi de la chaîne prénom à l'activité HelloActivity

```
// Création d'une intention
Intent intent = new Intent(this, HelloActivity.class);

// Ajout de la chaîne prénom à l'intent
intent.putExtra(EXERCICE_4_PRENOM, prénom);
```

- Utilisation d'une constante pour définir le nom du paramètre d'envoi

```
public static final String EXERCICE_4_PRENOM = "prenom";
```

Envoie de données entre activités (2/2)

- ✓ Récupérer la valeur envoyée sur l'intent lors de la création de la nouvelle activité

```
public class HelloActivity extends ActionBarActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_hello_exercice4);  
  
        // Récupérer le prenom passé par l'intent  
        String prenom = getIntent().getStringExtra(Exercice4Activity.EXERCICE_4_PRENOM);  
    }  
}
```

Exercice4Activity.EXERCICE_4_PRENOM donne le nom du paramètre

Méthodes putExtra(...) et get*Type*Extra()

✓ Extrait de la classe Intent

- putExtra(String name, int value) /getIntExtra(String name, int defaultValue)
- putExtra(String name, int[] value)/getIntArrayExtra(String name)
- Float, double, String, etc.

✓ Plus de détails :

- <http://developer.android.com/reference/android/content/Intent.html>

Différentes façons de lancer une activité

- ✓ **startActivity(Intent)** : lance une nouvelle activité mais vous ne recevrez aucune information lorsque cette activité se terminera.
- ✓ **startActivityForResult(Intent, int resCode)** : lance une nouvelle activité dans le but d'obtenir un résultat. La méthode `onActivityResult(...)` est appelée en retour.

Obtenir un résultat (1/3)

- ✓ L'activité Exercice4Activity demande le lancement de l'activité HelloActivity mais souhaite un résultat lors de la fermeture de celle-ci
- ✓ Utilisation de la méthode startActivityForResult

```
// Création d'une intention
Intent intent = new Intent(this, HelloActivity.class);

// Ajout de la chaîne prenom à l'intent
intent.putExtra(EXERCICE_4_PRENOM, prenom);

// lancement de la demande de changement d'activity
startActivityForResult(intent, EXERCICE_4_HELLO_REQUEST);
```

- EXERCICE_4_HELLO_REQUEST permet de donner un numéro à la requête

```
public final static int EXERCICE_4_HELLO_REQUEST = 0;
```

Obtenir un résultat (2/3)

- ✓ Attendre le résultat dans l'activité Exercice4Activity :

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {

    // Vérification du retour à l'aide de notre identifiant
    if (requestCode == EXERCICE_4_HELLO_REQUEST) {

        // Vérification de l'état de retour
        if (resultCode == RESULT_OK) {

            // Envoyer une notification
            Toast.makeText(this, "retour ok", Toast.LENGTH_SHORT).show();

            // Effacer le prenom
            EditText editText = (EditText)findViewById(R.id.exercice4_editttext);
            editText.getText().clear();

        } else if (resultCode == RESULT_CANCELED) {

            // Envoyer une notification
            Toast.makeText(this, "retour cancel/back", Toast.LENGTH_SHORT).show();

        }

    }

}
```

Obtenir un résultat (3/3)

```
if (requestCode == EXERCICE_4_HELLO_REQUEST) {
```

- ✓ RequestCode permet de savoir quelle activité a été fermé (sorti de la pile)

- ✓ ResultCode permet de savoir :

```
if (resultCode == RESULT_CANCELED) {
```

- si le résultat était un choix de l'utilisateur avec le bouton back ↩

```
if (resultCode == RESULT_OK) {
```

- Ou de l'activité HelloActivity en retournant le result OK.
- ◆ Pour déclencher la fermeture d'une activité avec RESULT_OK

faire :

```
setResult(RESULT_OK);  
super.finish();
```

Les flags : maîtriser les intentions

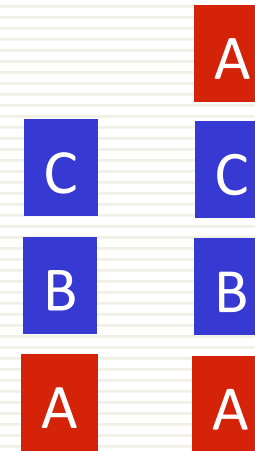
- ✓ Plusieurs flags pour différents besoins
- ✓ Exemple : revenir à l'activité principale MainActivity même si plusieurs activités sont devant dans la pile.
 - Intent.FLAG_ACTIVITY_CLEAR_TOP

```
Intent intent = new Intent(this, MainActivity.class);  
intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);  
startActivity(intent);
```

FLAG_ACTIVITY_CLEAR_TOP

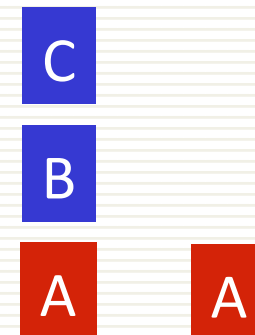
- ✓ Demande d'intention pour la création d'une activité A

NO_FLAG



FLAG_ACTIVITY_CLEAR_TOP

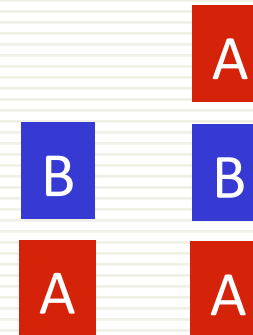
appel de la méthode onNewIntent() sur l'activité A déjà instanciée



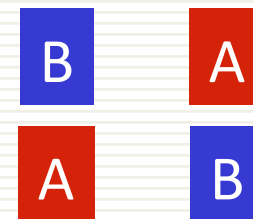
FLAG_ACTIVITY_REORDER_TO_FRONT

- ✓ Demande d'intention pour la création d'une activité A

NO_FLAG



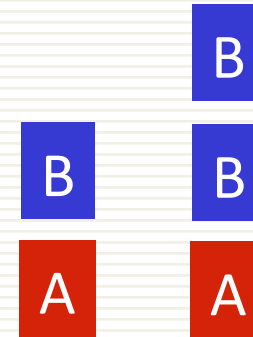
FLAG_ACTIVITY_REORDER_TO_FRONT



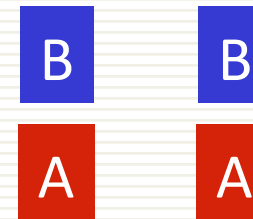
FLAG_ACTIVITY_SINGLE_TOP

- ✓ Demande d'intention pour la création d'une activité B

NO_FLAG



FLAG_ACTIVITY_SINGLE_TOP



Plus de détails

✓ Tâches et piles d'activité

- <http://developer.android.com/guide/components/tasks-and-back-stack.html>

✓ Intentions

- <http://developer.android.com/guide/components/intents-filters.html>
- <http://developer.android.com/training/basics/intents/result.html>
- <http://developer.android.com/training/basics/firstapp/starting-activity.html>

putExtra, getTypeExtr, flag :

- <http://developer.android.com/reference/android/content/Intent.html>