R4 Real 11

Programmation mobile sous Android

Introduction à la pile d'activités

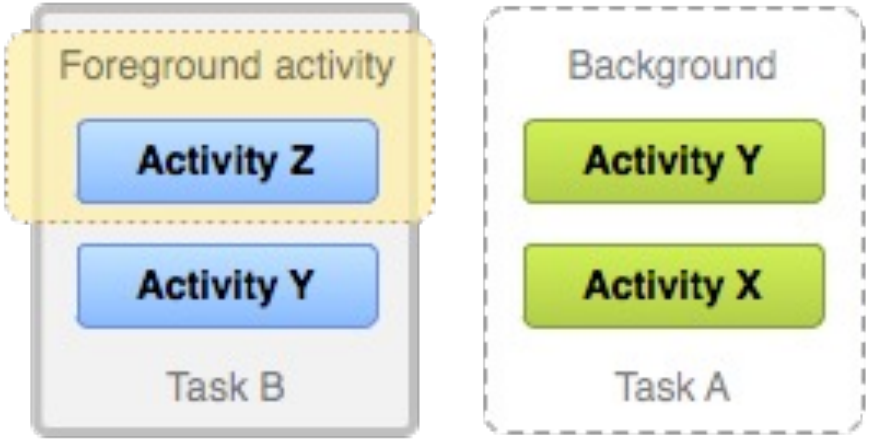
Introduction aux intents

03/02/2023

Pile d'activités (1/3)

- ✓ **Rappel** : une 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**



Source site android

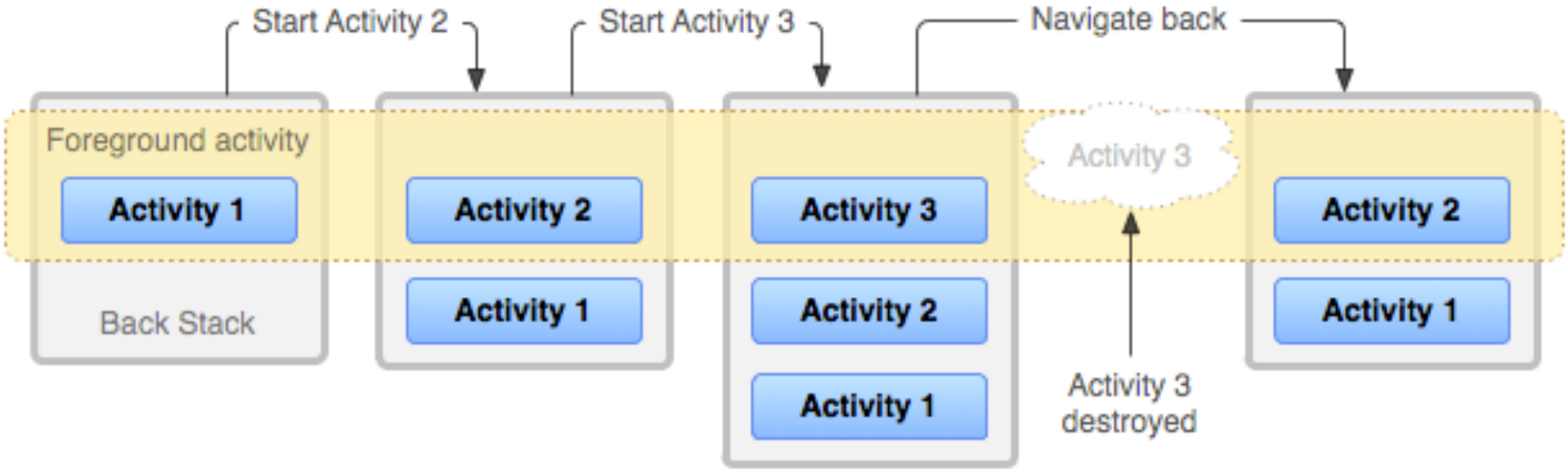
Pile d'activités (2/3)

- ✓ Lorsqu'une application est demandée :
  - Si la pile existe, elle est placée en premier plan
  - Si la pile 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:exported="true" >
   <intent-filter>
      <action android:name="android.intent.action.MAIN" />
      <category android:name="android.intent.category.LAUNCHER" />
   </intent-filter>
</activity>
```

Pile d'activités (3/3)

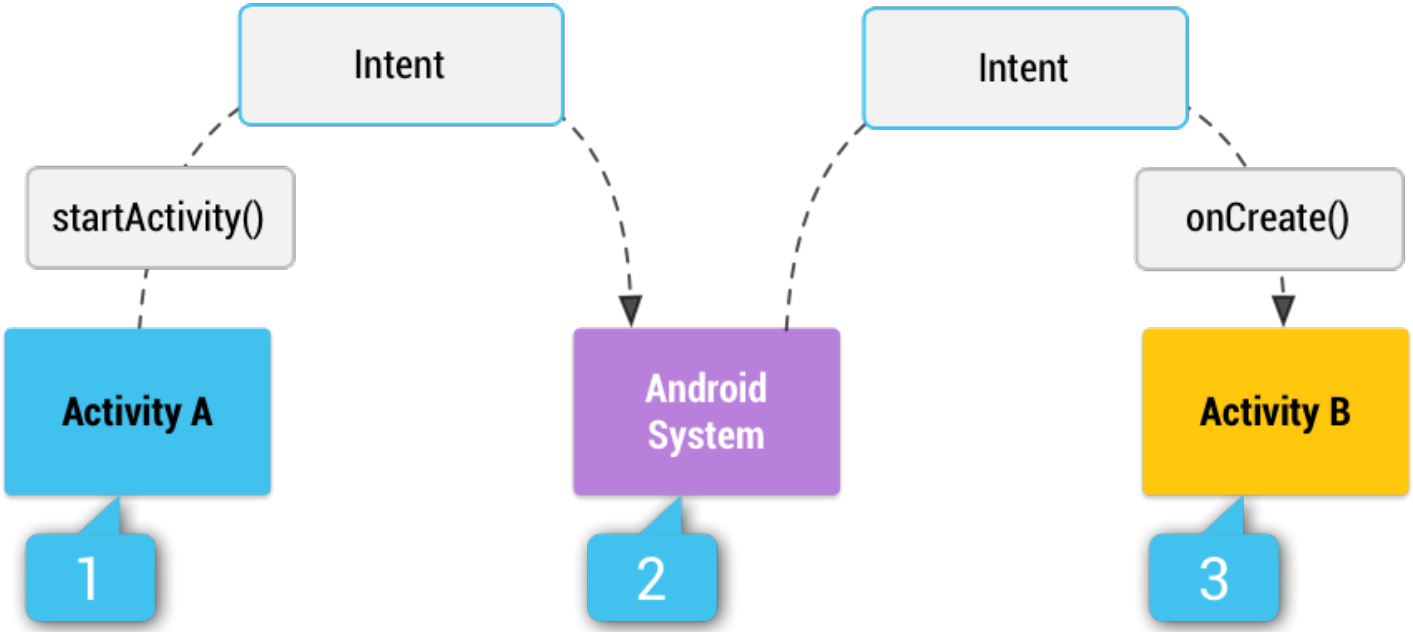
- ✓ Chaque fois qu'une activité est lancée, elle est placée sur la **pile** appelée aussi back stack



Source site android

Créer une nouvelle activité

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



Source site android

  - 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.) … ***dans un prochain cours***

Exemple d'intention explicite (1/2)

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

```
public void onExercice1(View view) {

   // Création d'une intention
   Intent Exercice1ViewActivityIntent = new Intent(MainActivity.this, Exercice1Activity.class);

   // Lancement de la demande de changement d'activité
   startActivity(Exercice1ViewActivityIntent);
}
```

Démo Exemple_2_1_AActivity

Exemple d'intention explicite (2/2)

- ✓ **Attention** lors de l'utilisation d'un listener spécifique (classe anonyme)

```
public class Exemple_2_2_AActivity extends AppCompatActivity {
…

// Traiter l'événement : abonnement, listener
allerALExemple.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View view) {

        // Création d'une intention
        Intent intent = new Intent(Exemple_2_2_AActivity.this, Exemple_2_2_BActivity.class);

        // Lancement de la demande de changement d'activité
        startActivity(intent);
}});
```

Pour avoir accès à l'objet `this` de la classe contenant le listener, on doit faire `NomDeLActivity.this` dans le listener.

Démo Exemple_2_2_AActivity

Envoi de données entre activités (1/2)

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

```
// Création d'une intention
Intent intent = new Intent(Exercice4Activity.this, HelloActivity.class);

// Ajoute la chaîne prenom à l'intent
intent.putExtra(HelloActivity.PRENOM_KEY, prenomView.getText().toString());
```

  - Utilisation d'une constante pour définir la clé (le nom) du paramètre d'envoie : à mettre dans l'activité devant recevoir l'information, ici `HelloActivity`

```
public static final String PRENOM_KEY = "prenom_key";
```

Envoi 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 AppCompatActivity {

    // Constante représentant la clé du paramètre de l'intent
    public static final String PRENOM_KEY = "prenom_key";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_hello);

        // Récupération du prénom passé par l'intent en utilisant la clé
        String prenom = getIntent().getStringExtra(PRENOM_KEY);
```

A refaire dans le TP2

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

Et un objet complexe ?

Démo dans exemple_2.data

- ✓ Rendre la classe de l'objet `Parcelable` !
  - C'est comme si on rendait l'objet sérialisable
  - Comment faire ?
    - `public class Resultat implements Parcelable {`
    - Et laissez faire Android Studio et sa proposition d'aide ! 

      « Implements method » puis « add Parcelable implementation »
  - Comment utiliser l'objet lors de l'intent ?
    - putExtra(String name, Parcelable value)
    - getParcelableExtra(String name)

Exemple de Parcelable

```
public class Resultat implements Parcelable {

    private int nombreVictoire=0;
    private int nombreEgalite=0;
    private int nombreDefaite=0;

    public Resultat() {
    }

    protected Resultat(Parcel in) {
        nombreVictoire = in.readInt();
        nombreEgalite = in.readInt();
        nombreDefaite = in.readInt();
    }

    @Override
    public int describeContents() {
        return 0;
    }

    @Override
    public void writeToParcel(@NonNull Parcel parcel, int i) {
        parcel.writeInt(nombreVictoire);
        parcel.writeInt(nombreEgalite);
        parcel.writeInt(nombreDefaite);
    }
```

Différentes façons de lancer une activité

- ✓ **`startActivity(Intent)`**: lance une nouvelle activité mais vous ne recevrez aucune information lorsque cette activité se terminera.

- ✓ **`ActivityResultLauncher<Intent>`**: lance une nouvelle activité et permet d'attendre un résultat à l'arrêt de l'activité lancée

Obtenir un résultat (1/4)

- ✓ Création d'un attribut dans l'activité A permettant d'attendre un résultat à l'arrêt de l'activité B demandée

Démo Exemple_2_4_AActivity

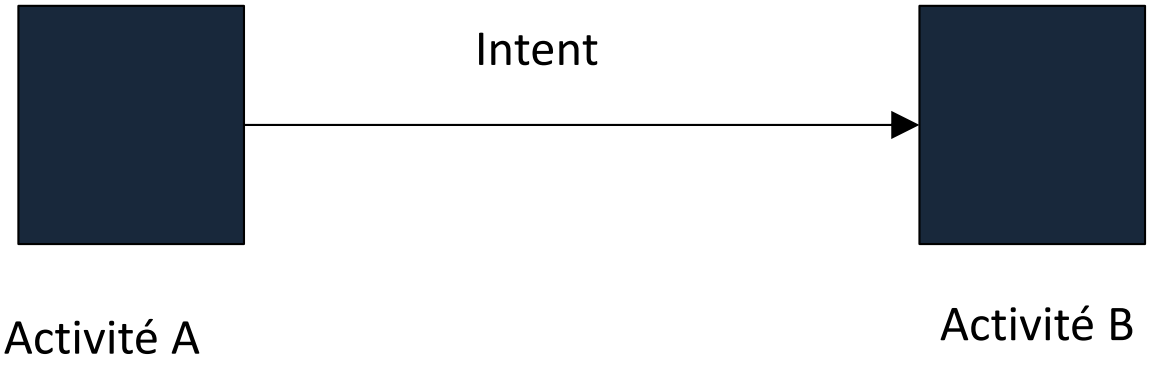
```
ActivityResultLauncher<Intent> activityResultLauncher =
registerForActivityResult(
        new ActivityResultContracts.StartActivityForResult(),
        new ActivityResultCallback<ActivityResult>() {

            @Override
            public void onActivityResult(ActivityResult result) {

               …
```

Obtenir un résultat (2/4)

Démo Exemple_2_4_AActivity

- ✓ Lancement de l'intent à l'aide de l'objet de type `ActivityResultLauncher`

```
Intent intent = new Intent(Activity_A.this, Activity_B.class);
activityResultLauncher.launch(intent);
```



Obtenir un résultat (3/4)

- ✓ À l'arrêt de l'activité B, le développeur décide de renvoyer de l'information à l'activité A : resultCode ou données (dans un intent)

Démo Exemple_2_4_BActivity

```
Intent intent=new Intent();
intent.putExtra(Activity_A.MESSAGE,"result OK");

// Envoi d'un result OK à l'activité A
setResult(RESULT_OK,intent);
```

Résultat possible : RESULT_OK ou RESULTAT_CANCELED
RESULTAT_CANCELED est aussi obtenu par le **bouton back** ↵

```
// Arrête une activité
finish();
```

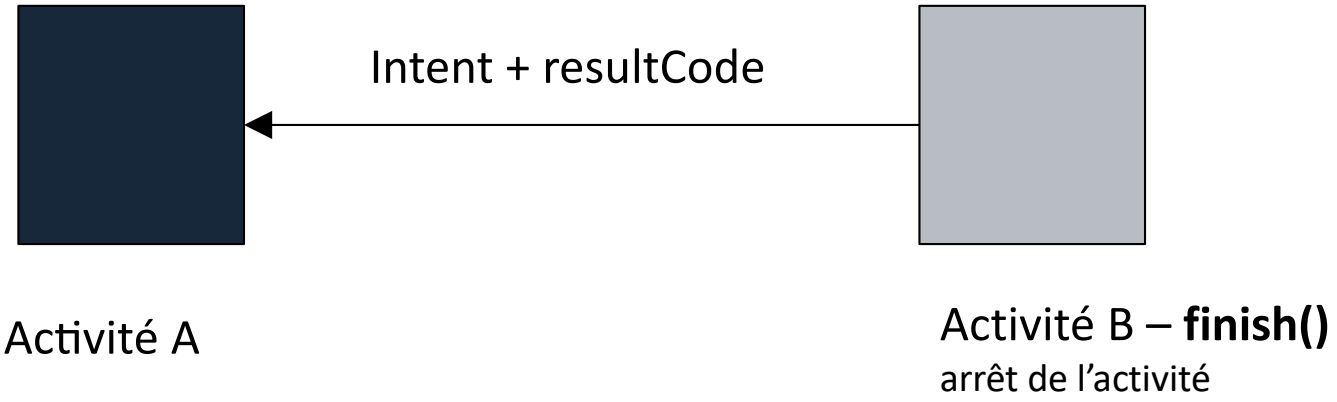
IMPORTANT : L'activité sera supprimée de la pile d'activités

Obtenir un résultat (4/4)

Démo Exemple_2_4_AActivity

✓ Récupération des informations côté activité A

```
@Override
public void onActivityResult(ActivityResult result) {

    if (result.getResultCode() == Activity.RESULT_OK) {

        // Récupération de l'intent et des données associées
        Intent intent = result.getData();
        String message = intent.getStringExtra(Activity_A.MESSAGE);

        //
        Toast.makeText(Activity_A.this,"De retour ! " + message, Toast.LENGTH_SHORT).show();
```



Les flags : maîtriser les intentions

- ✓ Plusieurs flags pour différents besoins

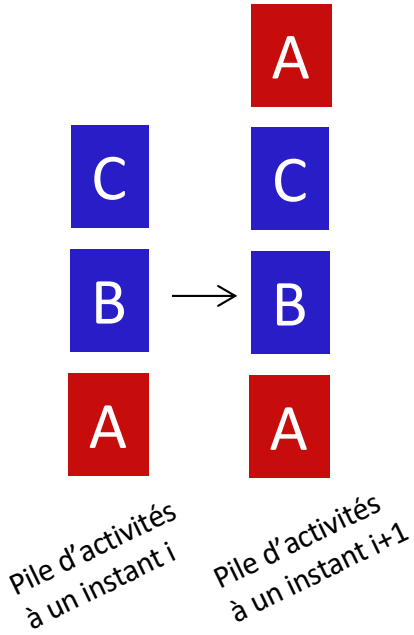
- ✓ Exemple : revenir à l'activité principale `MainActivity` même si plusieurs activités sont devant dans la pile.
  - `Intent.FLAG_ACTIVITY_CLEAR_TOP`

```
Intent intent = new Intent(this, Exemple_2_3_AActivity.class);
intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
startActivity(intent);
```

FLAG_ACTIVITY_CLEAR_TOP

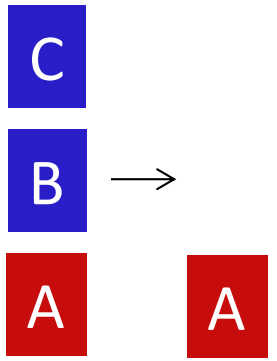
- ✓ Demande d'intention pour la création d'une activité A

NO_FLAG



FLAG_ACTIVITY_CLEAR_TOP

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

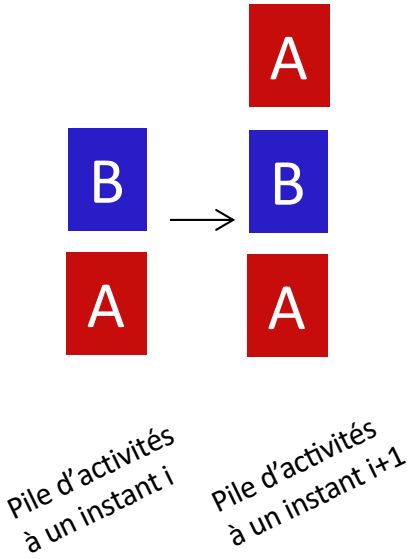


Démo Exemple_2_3_AActivity

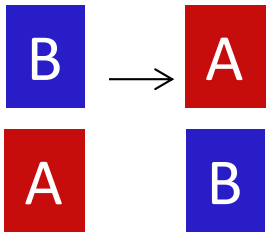
FLAG_ACTIVITY_REORDER_TO_FRONT

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

NO_FLAG



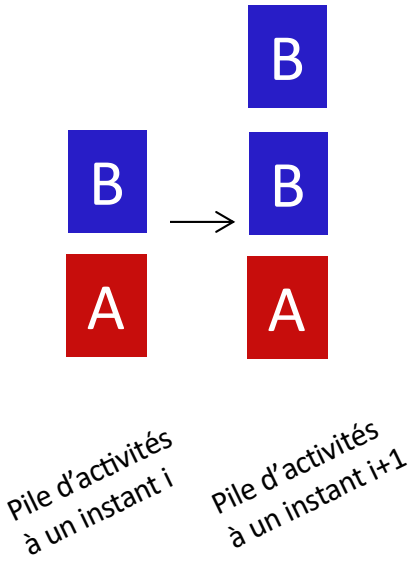
FLAG_ACTIVITY_REORDER_TO_FRONT



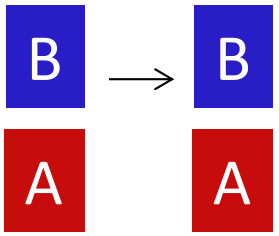
FLAG_ACTIVITY_SINGLE_TOP

- ✓ 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, getTypeExtra, flag :

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