Android & Fragments

Jean-Marc Lecarpentier
Université de Caen

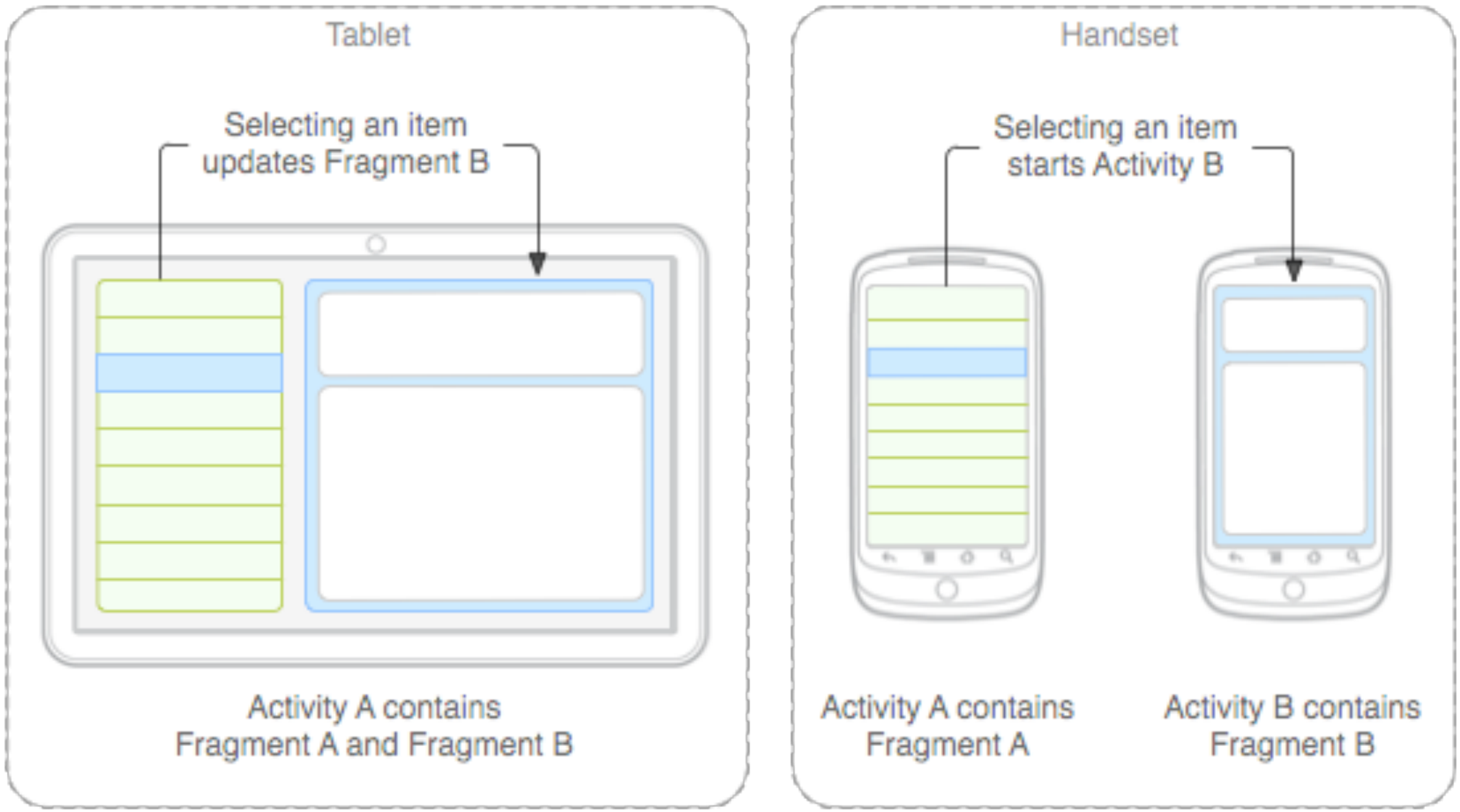
Au programme

- Fragment c’est quoi
- Principes des Fragments
- Fragments et layout XML
- Fragments @ run time
- Pile des Fragments
- Communication entre fragments
- Fragments et taille de l’écran
- Cycle de vie des Fragments/Activities

Principe

- Permettre la réutilisation de code
- Scinder les applications en parties indépendantes
- Gestion plus simple de la diversité des appareils
- Fragment ≃ Sub Activity

Exemple


Tablet
Selecting an item
updates Fragment B
Activity A contains
Fragment A and Fragment B
Handset
Selecting an item
starts Activity B
Activity A contains
Fragment A
Activity B contains
Fragment B

Créer un Fragment

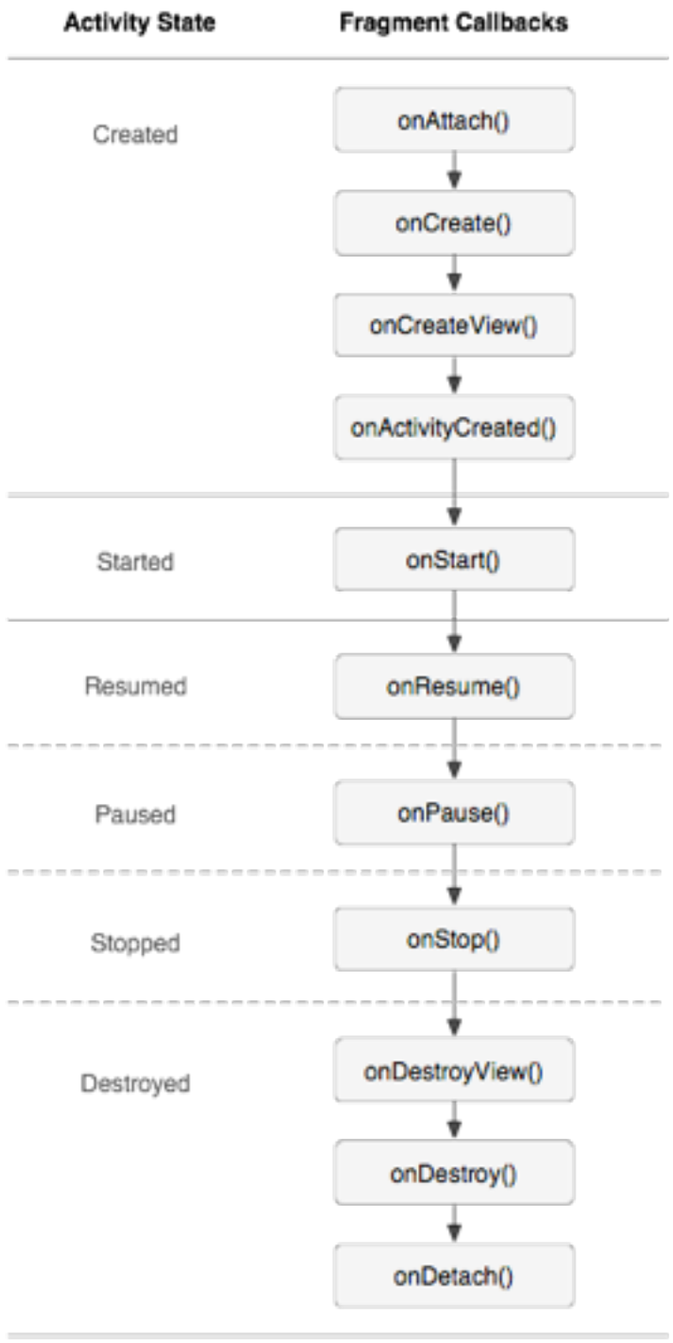
```
public class TitresFragment extends Fragment {

    public View onCreateView(LayoutInflater inflater, ViewGroup container,
            Bundle savedInstanceState) {
        // dire au fragment le layout à utiliser
        // paramètre container indique dans quel “parent” le fragment sera inclus
        View view = inflater.inflate(R.layout.fragment_titres, container, false);
        return view;
    }

    public void onAttach (Activity activity) {
        // callback appelé lorsque le fragment est rattaché à l’activité
    }
}
```

Cycle de vie

- Similaire à Activity
- 2 états en plus : onAttach, OnActivityCreated et onDetach
- onAttach est appelé avant onCreate



Fragments et Layout

- Attribut name indique le Fragment à instancier

```
<fragment
    android:id="@+id/contactsleft"
    android:name="fr.greyc.users.lecarpentier.demofragments.ContactsFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1.0"
    android:layout_below="@+id/images"
    android:layout_marginTop="5dp" />
```

- Fragment placé par le layout **ne peut pas** être remplacé *at runtime*

Fragment et Code

- Créer un fragment par programme
- Instancier le frafgment
- Utiliser le FragmentManager pour placer le fragment dans la vue

Fragment et Code

- Utilisation de FragmentManager et FragmentTransaction dans une Activity

```
// Instancier le fragment à utiliser
Fragment newFragment = new ExampleFragment();
// Utiliser le Manager de fragments et démarre une transaction
FragmentTransaction transaction = getFragmentManager().beginTransaction();

// Dire dans quel conteneur deux layout de l’activité il faut placer le fragment
transaction.add(R.id.fragment_container, newFragment);

// Indiquer de mettre ce changement dans la “pile” des fragments (cf. infra)
transaction.addToBackStack()

// faire un commit pour terminer
transaction.commit();
```

“Trouver” les fragments

- FragmentManager permet de “trouver” les fragments dans une Activity
  - getFragmentManager().findFragmentById(R.id.fid)
  - Possibilité d’avoir des fragments sans UI, gérés par des *tags*

Pile des fragments

- Similaire à la pile des Activity
- Mais au sein d'une Activity
- Permet la navigation entre "morceaux d'écran"
- Utilisation du bouton "Retour" de Android
- Pile gérée par le système

Gérer la pile des fragments

- Utiliser FragmentManager et FragmentTransaction

```
// Instancier le fragment à utiliser
Fragment newFragment = new ExampleFragment();
// Utiliser le Manager de fragments et démarre une transaction
FragmentTransaction transaction = getFragmentManager().beginTransaction();
// Dire dans quel conteneur deux layout de l'activité il faut placer le fragment
transaction.add(R.id.fragment_container, newFragment);
// Indiquer de mettre ce changement dans la "pile" des fragments (cf. infra)
transaction.addToBackStack()
// faire un commit pour terminer
transaction.commit();
```

- popBackStack() pour simuler un retour et revenir au fragment précédent

- addOnBackStackChangedListener() pour capter un changement dans la pile

Communication

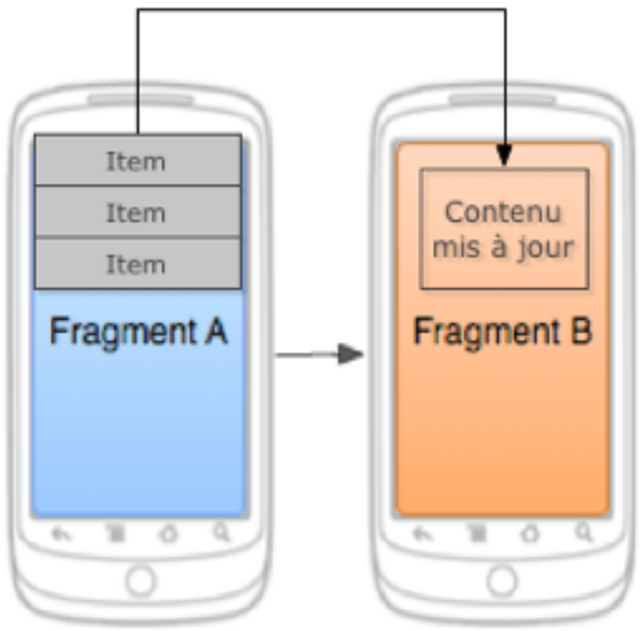
- Fragments **ne doivent pas** communiquer directement entre eux

- **Toujours passer par l'activité** : c'est elle qui doit gérer les interactions entre fragments

- Utilisation de callbacks entre les fragments et l'activité

- Activity doit implémenter les callbacks définis par les fragments

Fragments et évènements



Fragments et évènements

- Fragment A dérive de ListFragment

- Callback onListItemClick de ListFragment

- Transmettre à l'Activity l'info. "Item n°i cliqué"
  ➪ Utilisation d'interface et de callback

- Activity teste si Fragment B existe
  oui ➪ mettre le Fragment B à jour
  non ➪ lancer une Activity contenant le Fragment B
  et le mettre à jour

Factory pour Fragment

- Bonnes Pratiques : utiliser une Factory pour instance un Fragment
- Permet de passer les paramètres nécessaires pour la création du fragment
- Méthode statique newInstance
- Utiliser Callback onStart pour lancer des actions
- Voir App de démo du cours pour les détails de code

En bref

- Principes des fragments
- Création de fragment par layout
- Création de fragment par programme
- Communication fragments ↔ activité
- Fragments et gestion des divers tailles d’écran