

Fragmentek

Demkó Gábor

Mi a fragment?

A felhasználói felület egy önálló része.

- Egy activity tartalmazhat több fragmentet is (Fragmentek nem használhatók activity nélkül)
- UI modularitást biztosít, egységbe zárja a felület egy részének megvalósítását
- saját életciklusa van, könnyebb a bonyolult képernyők életciklus kezelése
- újrahasználató, egy adott Fragment több képernyő részeként is felhasználható



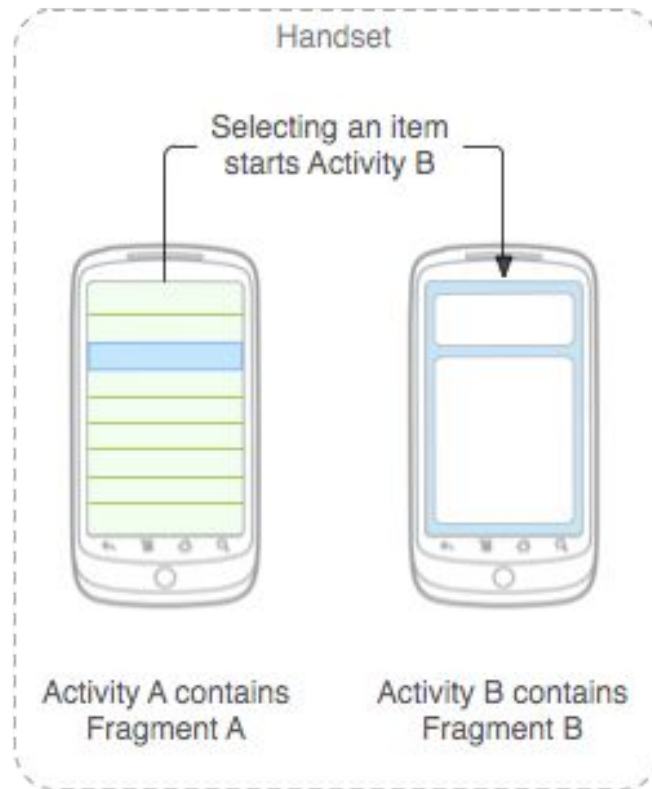
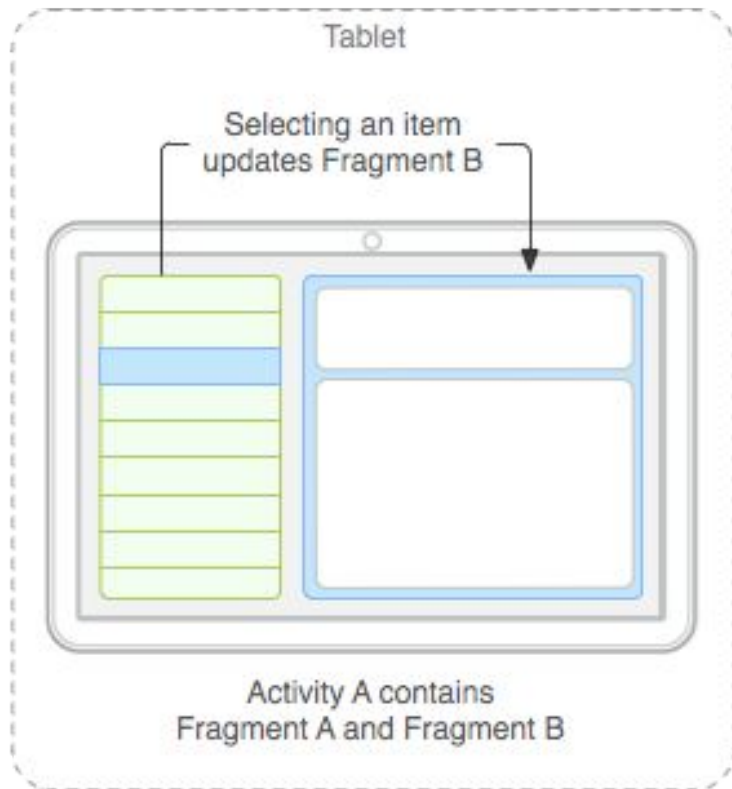
Tipikus felhasználások

Elsősorban a tabletek megjelenése ösztönözte a kifejlesztésüket

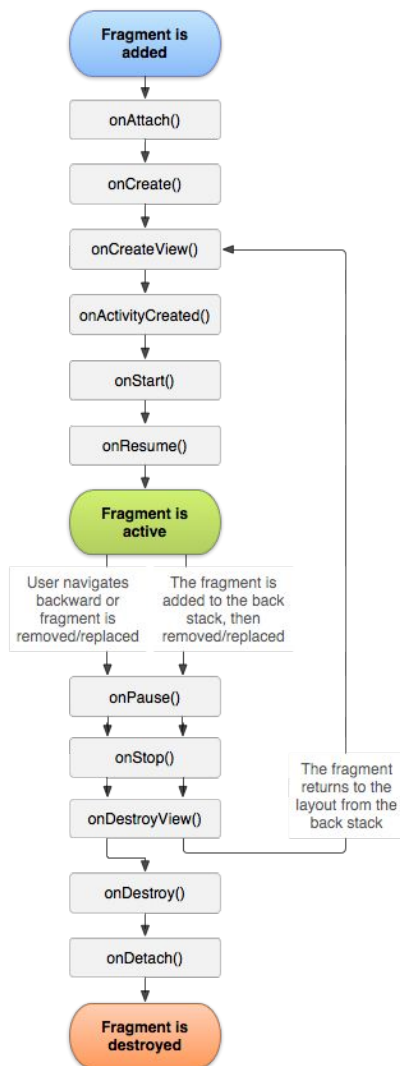
- Különböző képernyőméreteken különböző UI elrendezés alkalmazása
- Kód duplikáció elkerülése
- FragmentViewPager - Egy képernyőn belül lapozható tartalom sokkal egyszerűbb megvalósítása
- DialogFragment - A dialógusok életciklus kezelési hiányosságainak pótlása



Tablet példa



Életciklus



Egyes életciklusok megegyeznek az activity-vel

Ezen felül:

- fragment back stack eseményei
- activity-hez csatolás eseményei



Layoutból

```
public static class ExampleFragment extends Fragment {  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup container,  
                             Bundle savedInstanceState) {  
        // Inflate the layout for this fragment  
        return inflater.inflate(R.layout.example_fragment, container, false);  
    }  
}
```

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:orientation="horizontal"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
    <fragment android:name="com.example.news.ArticleListFragment"  
        android:id="@+id/list"  
        android:layout_weight="1"  
        android:layout_width="0dp"  
        android:layout_height="match_parent" />  
    <fragment android:name="com.example.news.ArticleReaderFragment"  
        android:id="@+id/viewer"  
        android:layout_weight="2"  
        android:layout_width="0dp"  
        android:layout_height="match_parent" />  
</LinearLayout>
```



Kódból

```
// Create new fragment and transaction
Fragment newFragment = new ExampleFragment();
FragmentManager transaction = getSupportFragmentManager().beginTransaction();

// Replace whatever is in the fragment_container view with this fragment,
// and add the transaction to the back stack
transaction.replace(R.id.fragment_container, newFragment);
transaction.addToBackStack(null);

// Commit the transaction
transaction.commit();
```

A fragmenteknek saját backstack-je van az activity-n belül!



Argumentumok

Ahogy az activity-nek intent extrákban adunk át paramétert, úgy a fragmenteknél argumentumokkal paraméterezünk.

Mikor fontos ez?

A rendszer nem csak az activity-ket állítja vissza, hanem a fragmenteket is (volt szó az activity-k visszaállításáról?)

- csak argumentumokkal lesz a fragment egyszerűen visszallítható az rendszer által
- csak argumentumokkal lesz egyszerűen újrahasználató a fragment áltaunk

Az argumentumok használata nem jön annyira triviálisan vagy a rendszer által kényszerítve mint az Activitynél az intent extrák, használatuk nem is kötelező. De erősen javaslom hacsak valaki nem a saját dolgát akarja nehezíteni később...



Tipikus hibák

Amire érdemes figyelni!

- a rendszer visszaállítja a fragmenteket, használjuk ezt ki, ne pedig küzdjünk ellene
- argumentumokkal paraméterezzünk, akkor is ha ez néha felesleges túlkomplikálásnak tűnik
- paraméter nélküli konstruktor (is) kell a rendszer általi visszaállításhoz
- fragmentek egymásba ágyazásakor, childFragmentManager-t használjuk!

Ha ezeket nem tarjuk be, akkor eleinte az alkalmazásunk teljesen jól működőnek fog tűnni. De a legalapabb eseteken kívül semmiféle életciklust nem kezeltünk le jól, az alkalmazás lépten nyomon látszólag véletlenszerűen crashelni, vagy rosszul fog működni!



```
git clone -b 07-fragments  
https://github.com/attrecto/Attrecto-Academy-Android.git
```

