



# Fragment



陳健文



# 什麼是 Fragment ?

---

Fragment(片段)類似 Activity, 可彈性地組織介面元件提供更好的操作體驗。

依附在 Activity 下且可為多個 Activity 使用

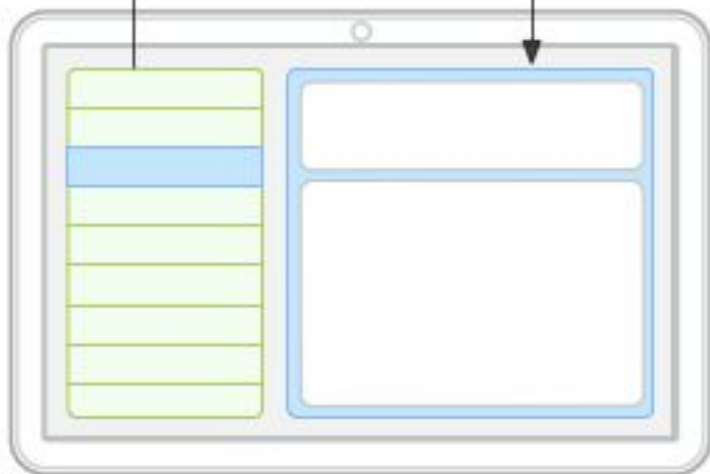
有自身的生命週期並可獨立接收事件

在 Android 3.0 之後加入

之前系統(API Level 11 以下)需安裝 android-support-v7 與 v4 才能支援

## Tablet

Selecting an item  
updates Fragment B



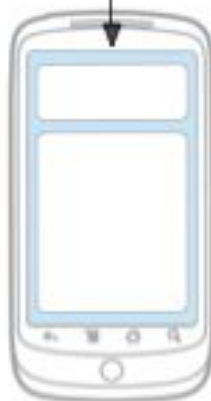
Activity A contains  
Fragment A and Fragment B

## Handset

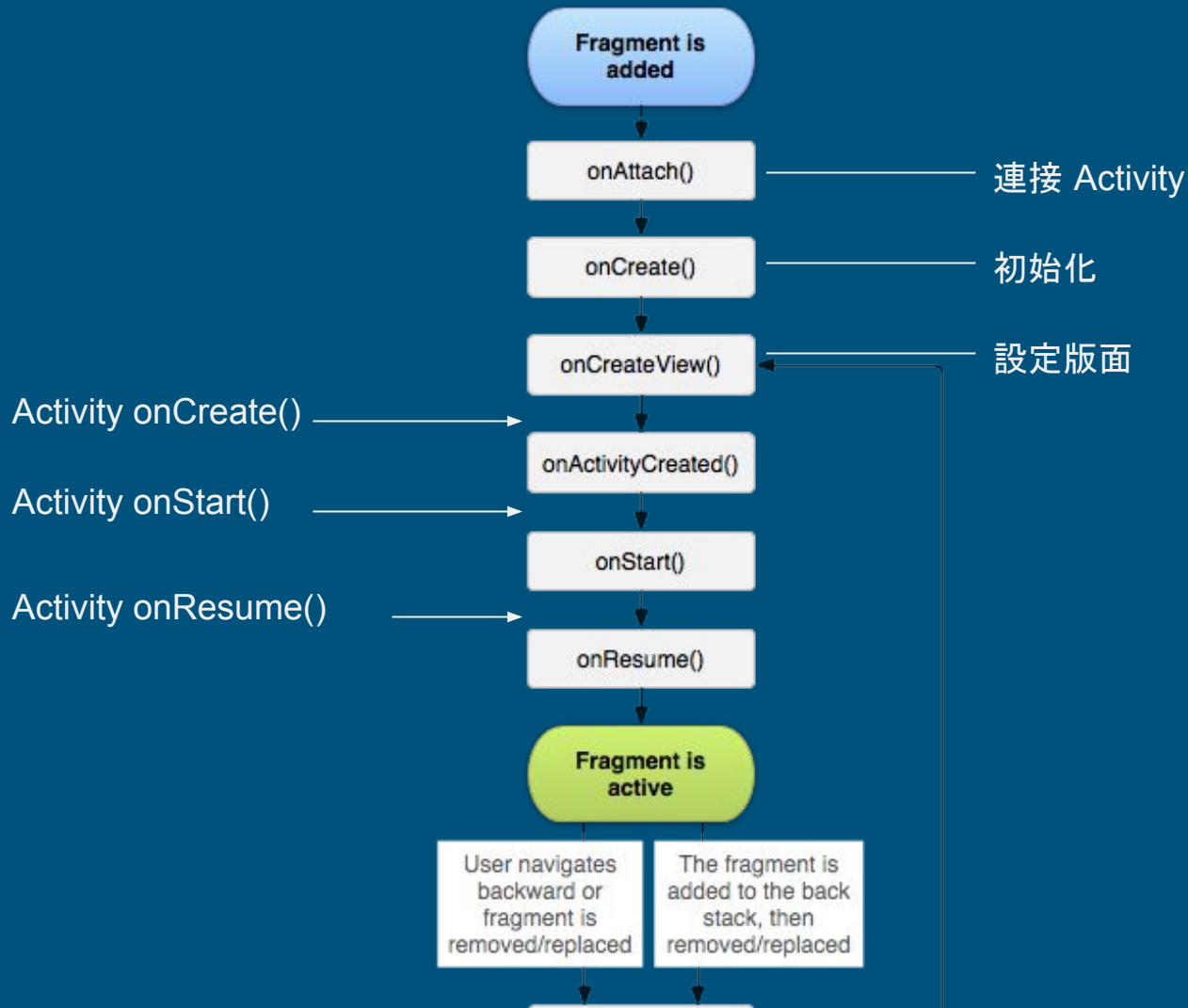
Selecting an item  
starts Activity B

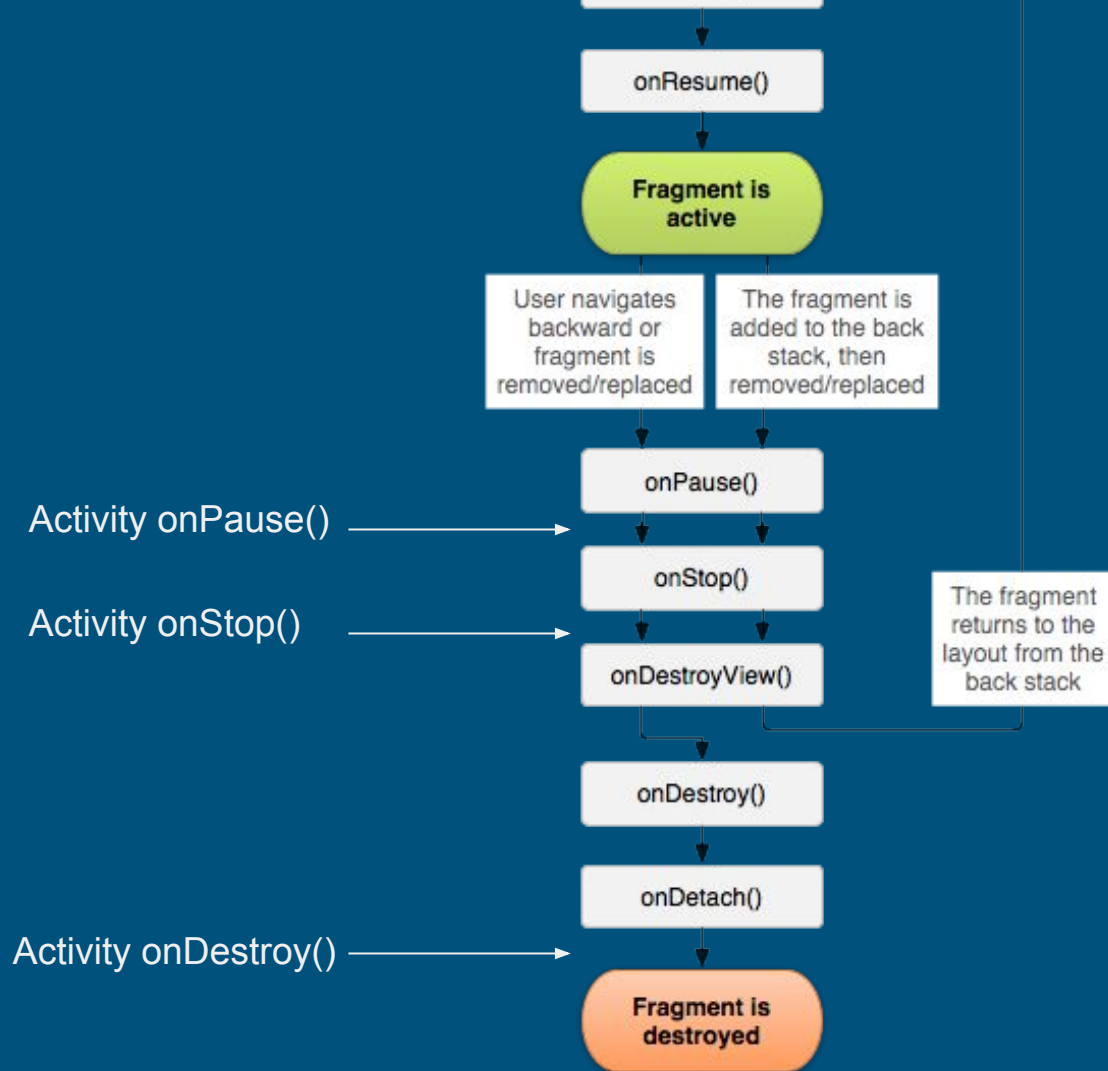


Activity A contains  
Fragment A



Activity B contains  
Fragment B





# 配置方式

---

## 靜態配置

在 XML 檔中宣告

可有許多個固定位置版塊

## 動態配置

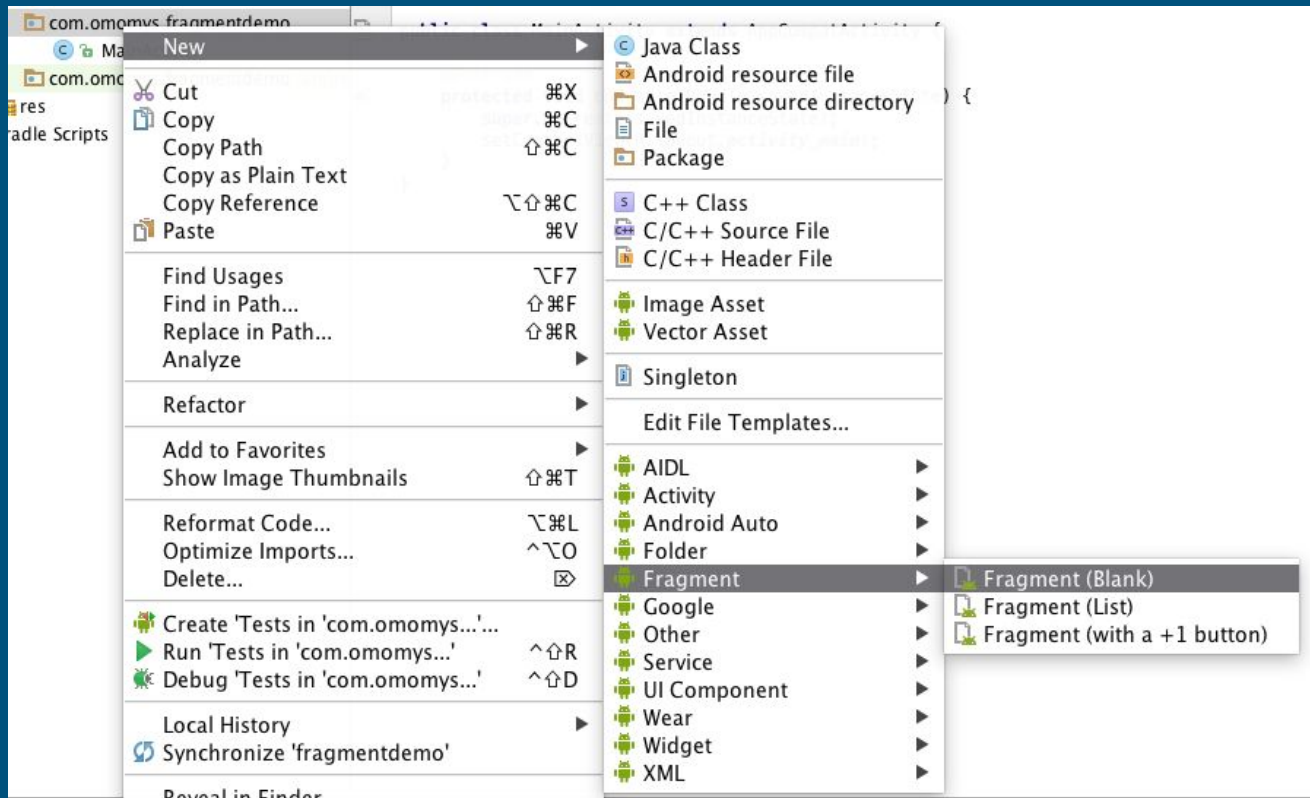
動態生成

數量與位置可動態調整

# 靜態配置

---

# 新增 Fragment 類別





# 設定版塊名稱與選項



Fragment (Blank)

Creates a blank fragment that is compatible back to API level 4.

Fragment Name:

FirstFragment



Create layout XML?

Fragment Layout Name:

fragment\_first

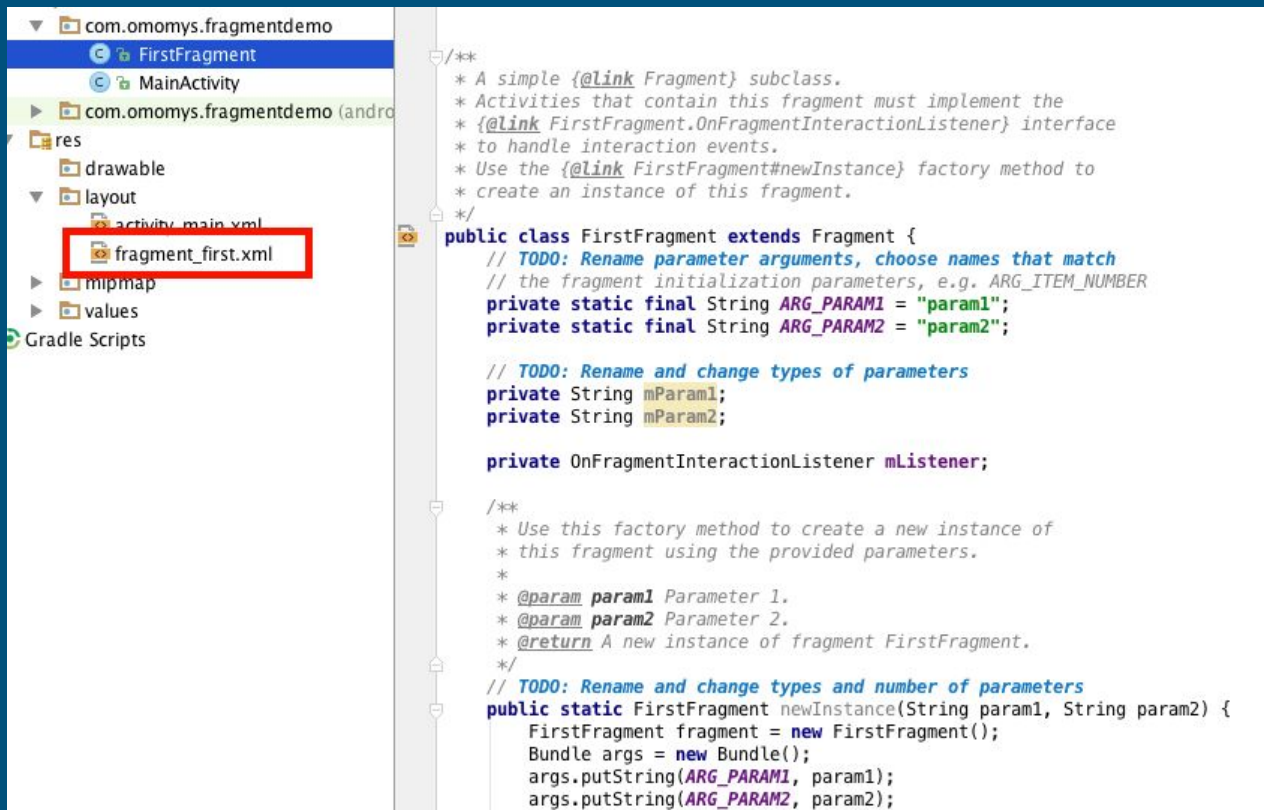


Include fragment factory methods?

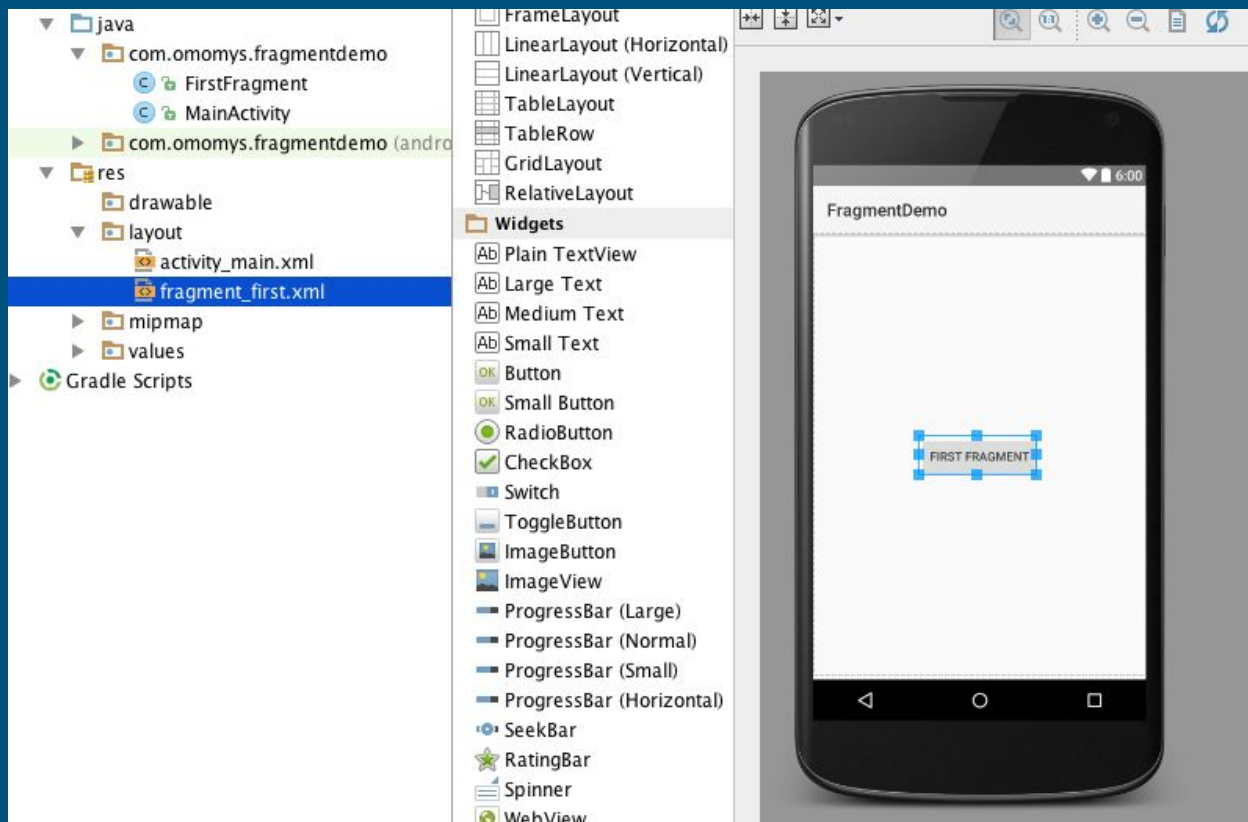


Include interface callbacks?

# 產生版塊類別檔與版面檔



# 設定片段 UI



# 在 Layout 檔中加入 XML fragment 項目

---

```
<fragment  
    android:name="com.omomys.fragmentdemo.FirstFragment"  
    android:id="@+id/fragment_first"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
/>
```

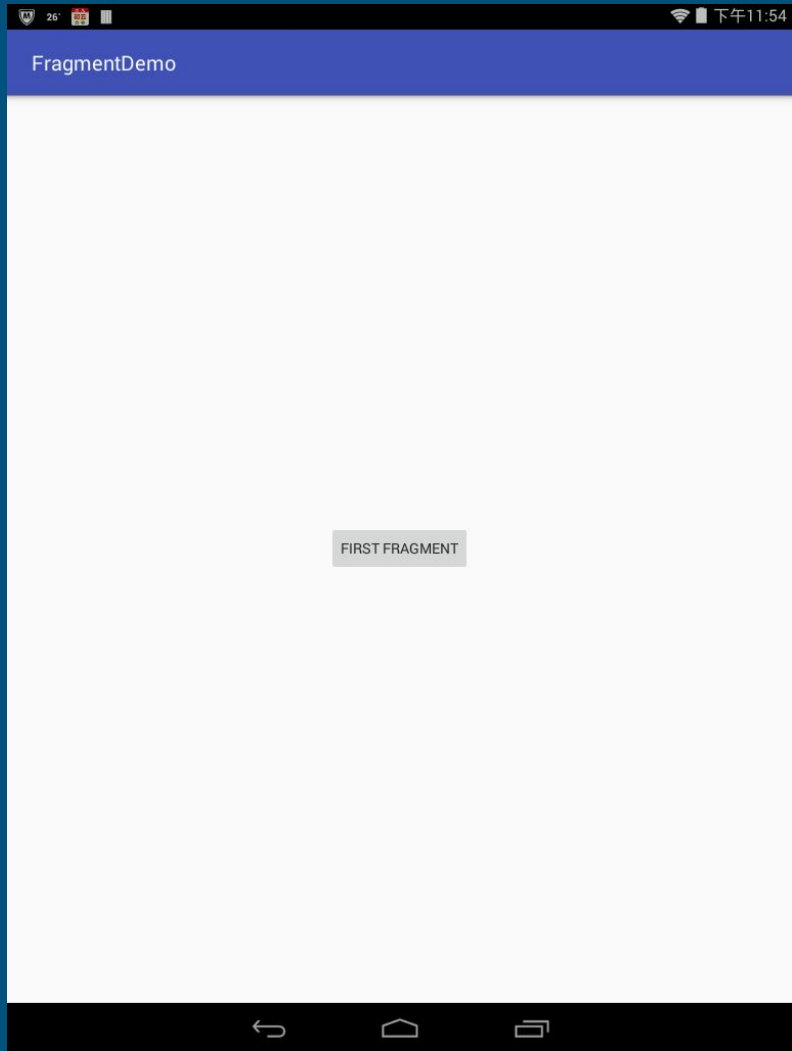
## 活動實作 OnFragmentInteractionListener 介面

```
public class MainActivity extends AppCompatActivity implements FirstFragment.OnFragmentInteractionListener {  
  
    private FragmentManager fragmentManager;  
    private FragmentTransaction fragmentTransaction;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        FirstFragment fragment = new FirstFragment();  
        fragmentManager = getFragmentManager();  
        fragmentTransaction = fragmentManager.beginTransaction();  
        fragmentTransaction.add(R.id.fragment_container, fragment);  
        fragmentTransaction.commit();  
    }  
  
    @Override  
    public void onFragmentInteraction(Uri uri){  
    }  
}
```

# 取得宿主 Activity

---

```
@Override
public void onAttachonAttach(Activity activity) {
    super.onAttach(activity);
    //Activity activity = context instanceof Activity ? (Activity) context : null;
    try{
        mListener = (OnFragmentInteractionListener) activity;
    }catch(ClassCastException e){
        throw new ClassCastException(activity.toString()
            + " must implement OnFragmentInteractionListener");
    }
}
```



## 用同樣的版塊類別配置二個版塊

```
<fragment android:id="@+id/fragment_first"  
    android:name="com.omomys.fragmentdemo.FirstFragment"  
    android:layout_height="wrap_content"  
    android:layout_width="match_parent"  
    android:layout_marginTop="50dp" />
```

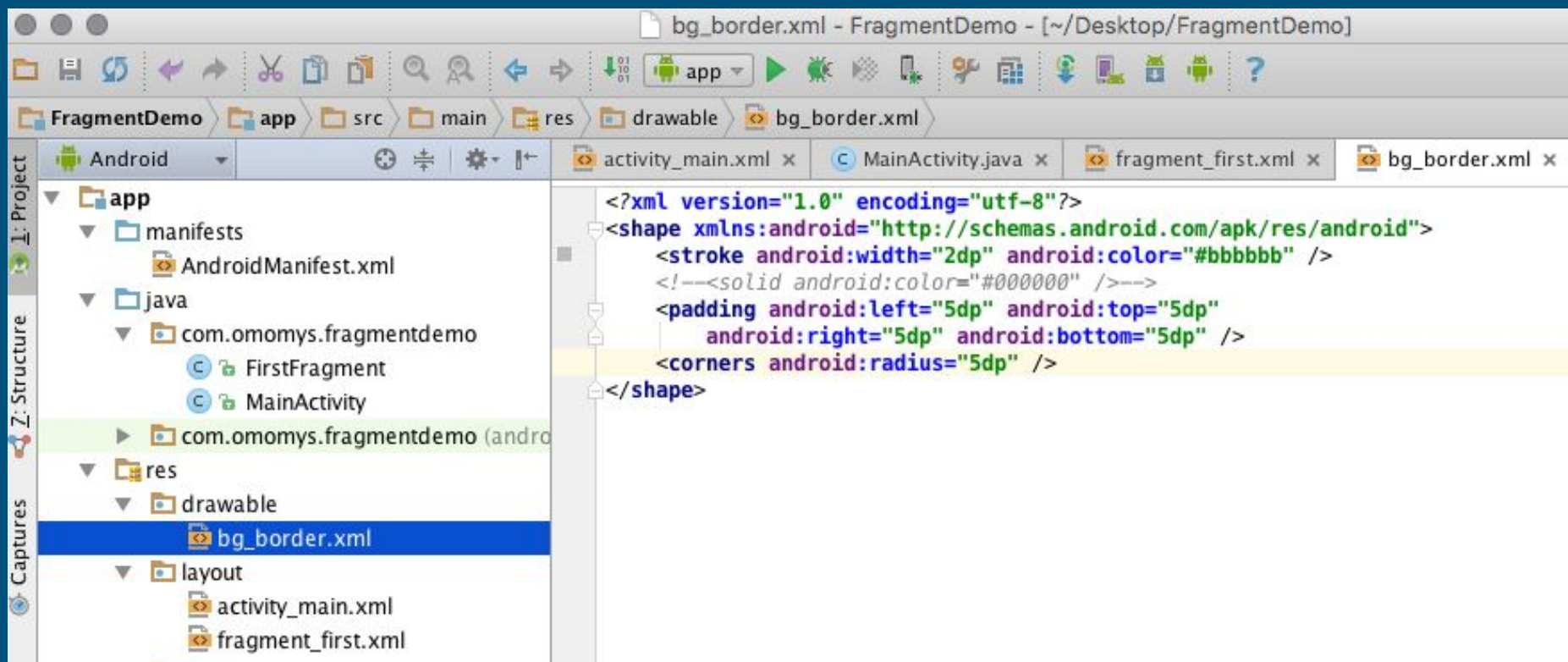
id 不同

```
<fragment android:id="@+id/fragment_first2"  
    android:name="com.omomys.fragmentdemo.FirstFragment"  
    android:layout_height="wrap_content"  
    android:layout_width="match_parent"  
    android:layout_below="@+id/fragment_first"  
    android:layout_marginTop="200dp" />
```

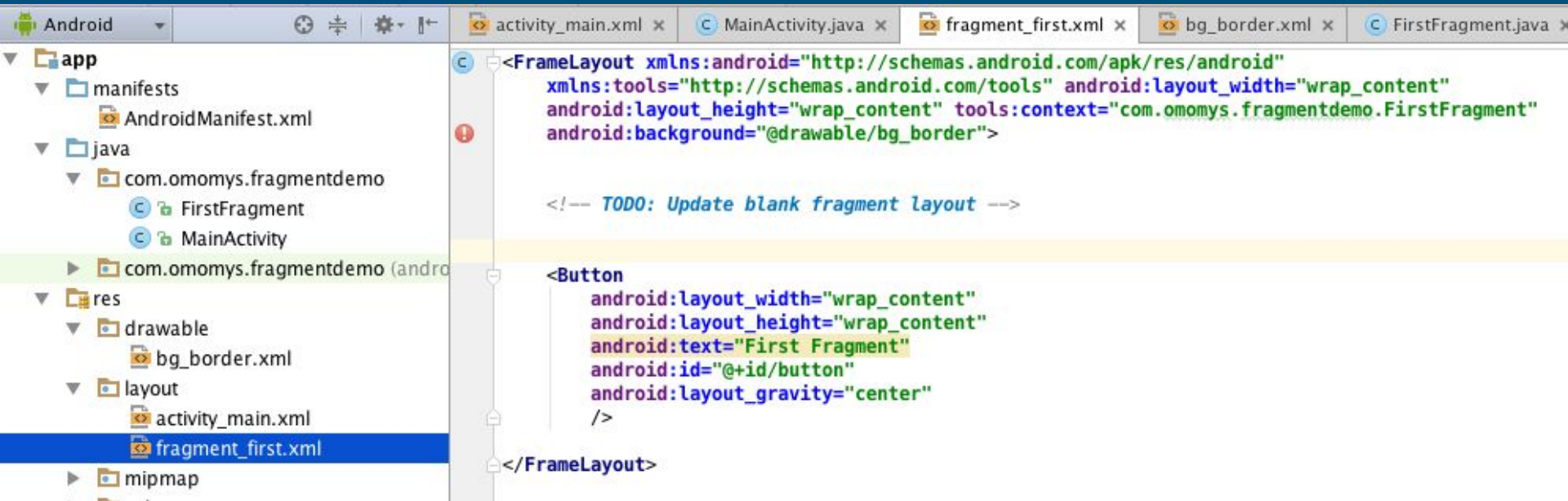
```
</RelativeLayout>
```



# 定義 Fragment 用的框線



# 為 Fragment 加上框線



半夜12:46

... 无SIM卡

## FragmentDemo

FIRST FRAGMENT

FIRST FRAGMENT

Button is clicked!

bingo

# 動態配置

---

# 動態配置的流程

---

自 FragmentManager 產生 FragmentTransaction 物件

透過 FragmentTransaction 物件進行操作：

add: 新增, replace: 置換

remove: 移除, attach: 附掛, detach: 卸除

addToBackStack: 加進堆疊, popBackStack: 自堆疊取回

commit 交付上述操作以執行

```
public void onClick(View view){
    FragmentTransaction fragTrans = getFragmentManager().beginTransaction();
    switch(whichFragment){
        case 1:
            fragTrans.replace(R.id.fragment_first, fragment1, "frag_first");
            whichFragment = 2;
            break;
        case 2:
            fragTrans.replace(R.id.fragment_first, fragment2, "frag_second");
            whichFragment = 1;
            break;
    }
    fragTrans.commit();
}
```

# 系統內建片段

---

ListFragment

WebViewFragment

DialogFragment

MapFragment

PreferenceFragment

...

# Demo

