ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Mobile Programing

Chapter 4.7. Fragment

Fragment

- A Fragment represents a reusable portion of your app's UI
- Fragment has its own lifecycle but cannot live on its own (must be hosted by an activity or other fragment)
- Phù hợp cho việc mô đun hóa giao diện thành từng phần khác nhau (có thể dùng lại được)

Create static Fragment

1. Thêm fragment vào project
2. Thêm khai báo fragment vào file xml của activity
3. Khai báo thuộc tính class hoặc name

```
<fragment
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/staticFragment"
    class="com.example.fragmentsample.FirstFragment">
```

Create Dynamic Fragment

1. Thêm lớp fragment vào project
2. Sử dụng FragmentContainerView trong activity layout để chứa fragment
3. Sử dụng Fragment Manager để thêm fragment vào activity

Fragment Manager

1. FragmentManager phụ trách thêm/xóa/thay thế fragment hoặc đưa fragment vào backstack
2. Mỗi Activity đã được Android gán sẵn với 1 FragmentManager → lấy ra bằng cách gọi hàm getFragmentManager()
3. Một Fragment cũng có thể chứa (host) các fragment khác. getChildFragmentManager() và getParentFragmentManager()

https://developer.android.com/guide/fragments/fragmentmanager

Fragment transaction

1. FragmentManager có thể thêm/xóa/thay thế fragment trong Fragment Container. Mỗi thao tác thay đổi trên gọi là 1 transaction.
2. Có thể group nhiều action vào 1 transaction
3. Lấy ra transaction bằng hàm:

```
FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();
```

4. Kết thúc transaction bằng commit

Commit là bất đồng bộ

1. Khi gọi commit thì transaction không được thực hiện ngay mà sẽ được lập lịch để gọi trong UI thread
2. commitNow không tương thích với addToBackStack
3. executePendingTransactions() tương thích với addToBackStack

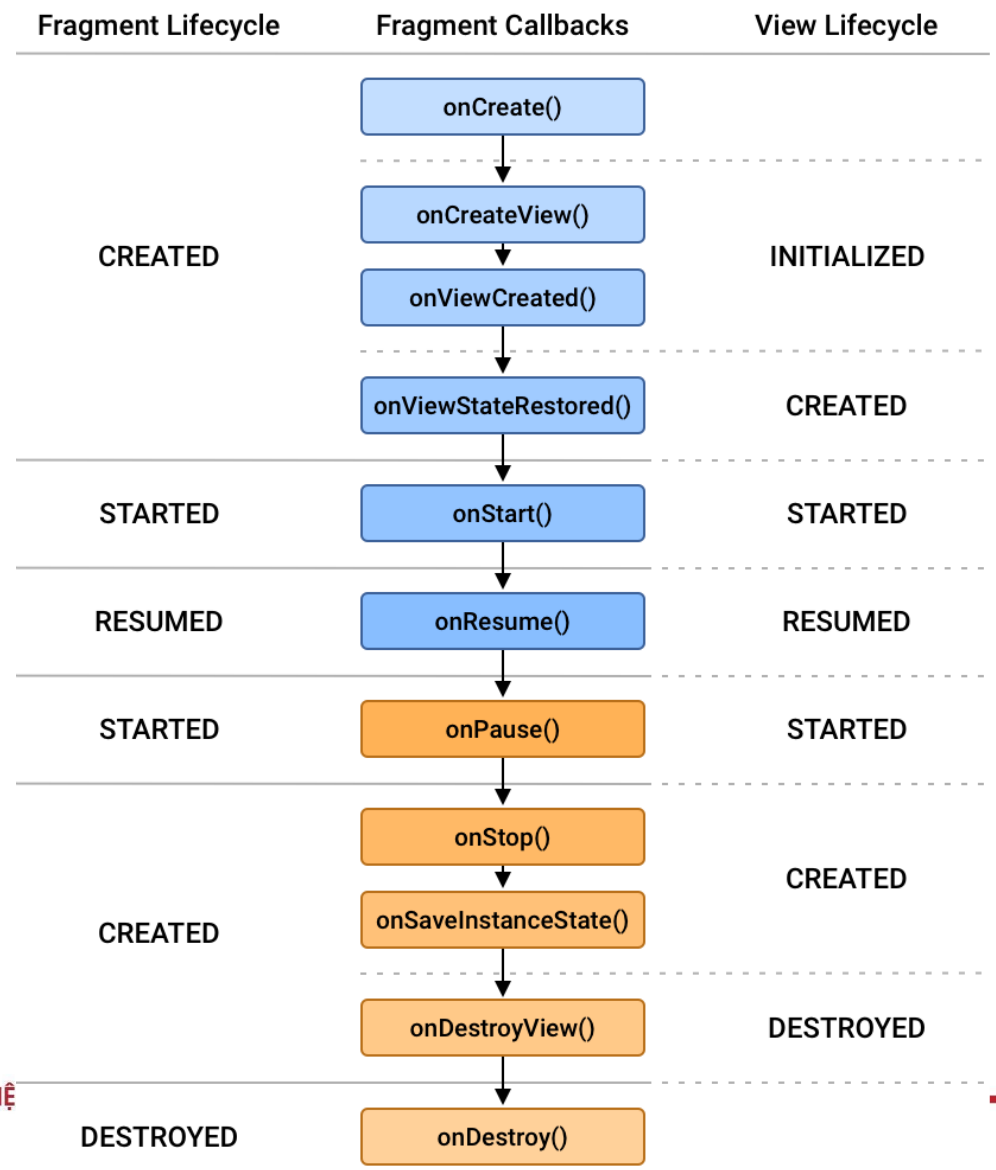
Fragment animation

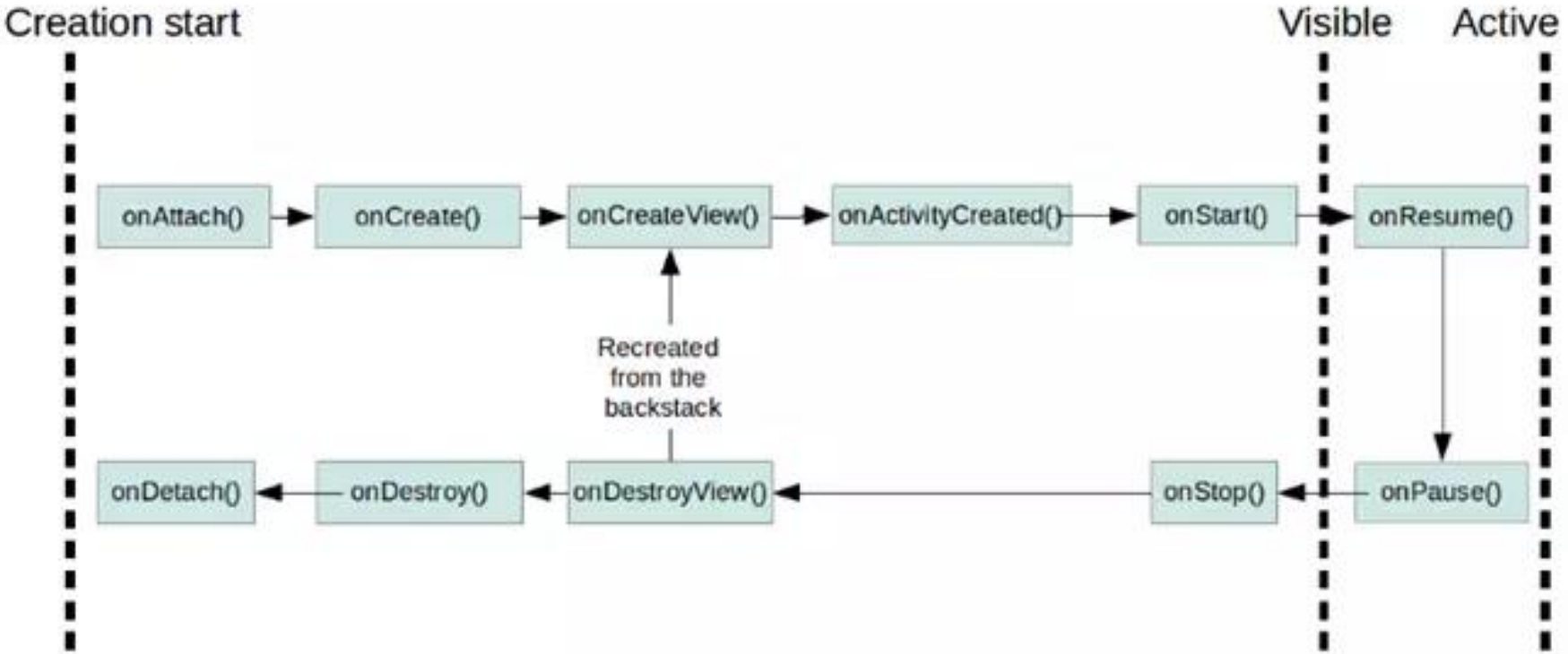
1. Tự đọc tài liệu

Fragment Life cycle

https://developer.android.com/guide/fragments/lifecycle

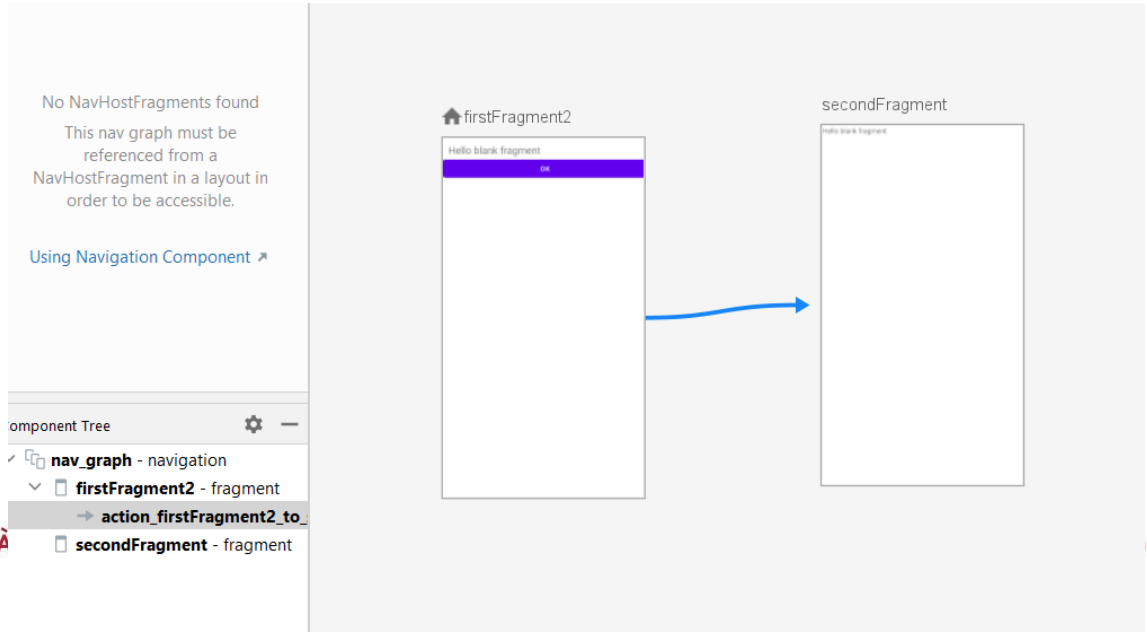
Fragment lifecycle




Creation start
Visible
Active
onAttach()
onCreate()
onCreateView()
onActivityCreated()
onStart()
onResume()
Recreated from the backstack
onDetach()
onDestroy()
onDestroyView()
onStop()
onPause()

Navigate giữa các Fragment bằng NavHostFragment

1. Tạo navigation navgraph
2. Kéo/khai báo các destination fragment cần tương tác vào nav_graph
3. Khai báo các action



Navigate giữa các Fragment bằng NavHostFragment

4. Trong activity_main layout khai báo navhost fragment

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity3">

    <fragment
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/fragment_container_view_tag"
        android:name="androidx.navigation.fragment.NavHostFragment"
        app:defaultNavHost = "true"
        app:navGraph="@navigation/nav_graph"></fragment>

</androidx.constraintlayout.widget.ConstraintLayout>
```

Navigate giữa các fragment

1. Mỗi NavHost có NavController tương ứng
2. Lấy ra NavController bằng hàm:
   - `NavHostFragment.findNavController(Fragment)`
   - `Navigation.findNavController(Activity, @IdRes int viewId)`
   - `Navigation.findNavController(View)`
3. Gọi hàm navigate của NavController với đầu vào là các action đã khai báo ở nav_graph

Giao tiếp giữa các fragment

1. Sử dụng ViewModel: truyền dữ liệu persistent data
2. Fragment result API: với các dữ liệu dùng 1 lần, có thể gói trong bundle
3. Global static class

Fragment result API