LẬP TRÌNH TRÊN THIẾT BỊ DI ĐỘNG

CHƯƠNG 4.1:

Giới thiệu về Fragments

- Giảng viên: Ths. Nguyễn Trung Hiếu
- Email:hieunt.tg@ptithcm.edu.vn
- Mobie: 0983051825

Mục tiêu

- Kết thúc bài học này bạn có khả năng
  - Hiểu rõ về Fragments

NỘI DUNG

- Giới thiệu về Fragments
  - Introduction to Fragments
  - LifeCycle of Fragments
  - Communicating Fragments
- Animation & Graphics
- Menus & Dialogs
- Media

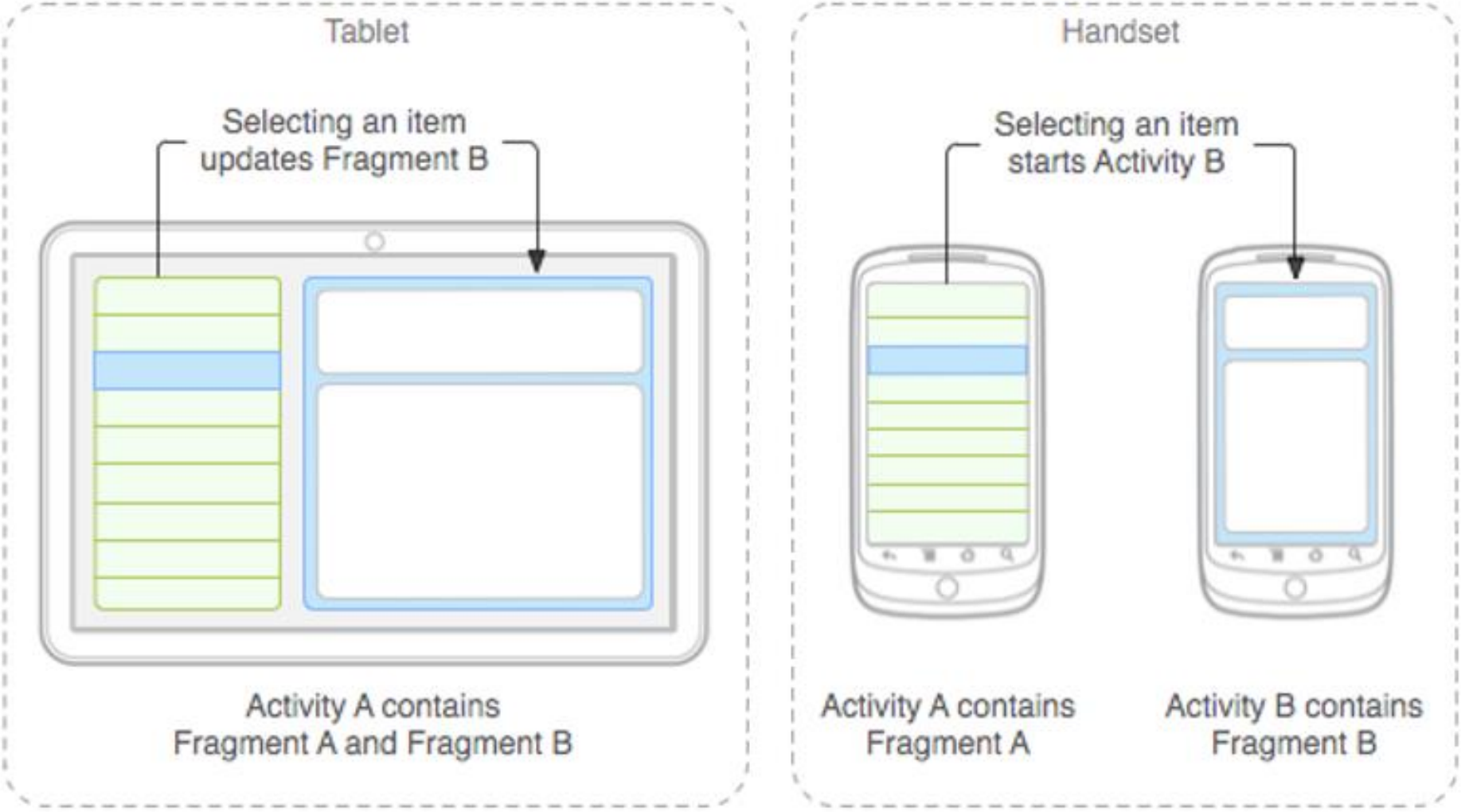
1

Giới thiệu về Fragments

Giới thiệu về Fragments

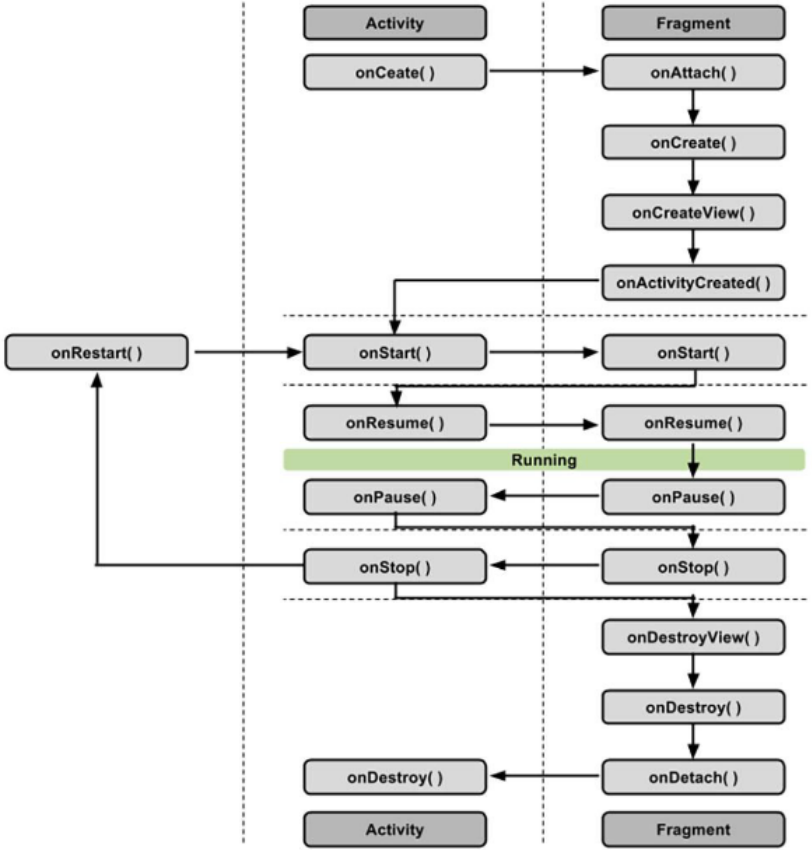
Một Fragment biểu diễn một hành vi hoặc một phần của giao diện người dùng trong một Activity.

- Có bố cục và vòng đời riêng
- Thêm, xóa động hoặc có nhiều Fragment
- Có thể có UI hoặc có thể không có Headless
- Được giới thiệu trong Honeycomb - API phiên bản 11


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

VÒNG ĐỜI CỦA FRAGMENT

- Fragment là một thành phần android độc lập có vòng đời và giao diện riêng được quản lý bởi một activity và hoạt động giống như một sub-activity
- Vòng đời của fragment bị ảnh hưởng trực tiếp bởi vòng đời của activity chứa nó. Tức là, khi activity bị tạm dừng thì tất cả các fragment được chứa bởi activity đó cũng tạm dừng, và khi activity bị hủy thì tất cả các fragment bên trong cũng bị hủy theo.



VÒNG ĐỜI CỦA FRAGMENT

- Khi Fragment được gắn vào Activity, các callback onAttach(), onCreate(), onCreateView(), onActivityCreated(), onStart(), onResume() lần lượt được gọi.
- Sau khi các callback trên được gọi, fragment lúc đó mới chính thức được xem là đang chạy.
- Sau đó, nếu người dùng bấm nút Back hay có bất kì thao tác gỡ/ thay thế fragment ra khỏi activity nào thì các callback onPause(), onStop(), onDestroyView(), onDestroy(), onDetach() sẽ được gọi. (Đây là trường hợp fragment chưa được thêm vào back stack. Ở phía dưới mình sẽ nói về trường hợp fragment được thêm vào back stack sau)

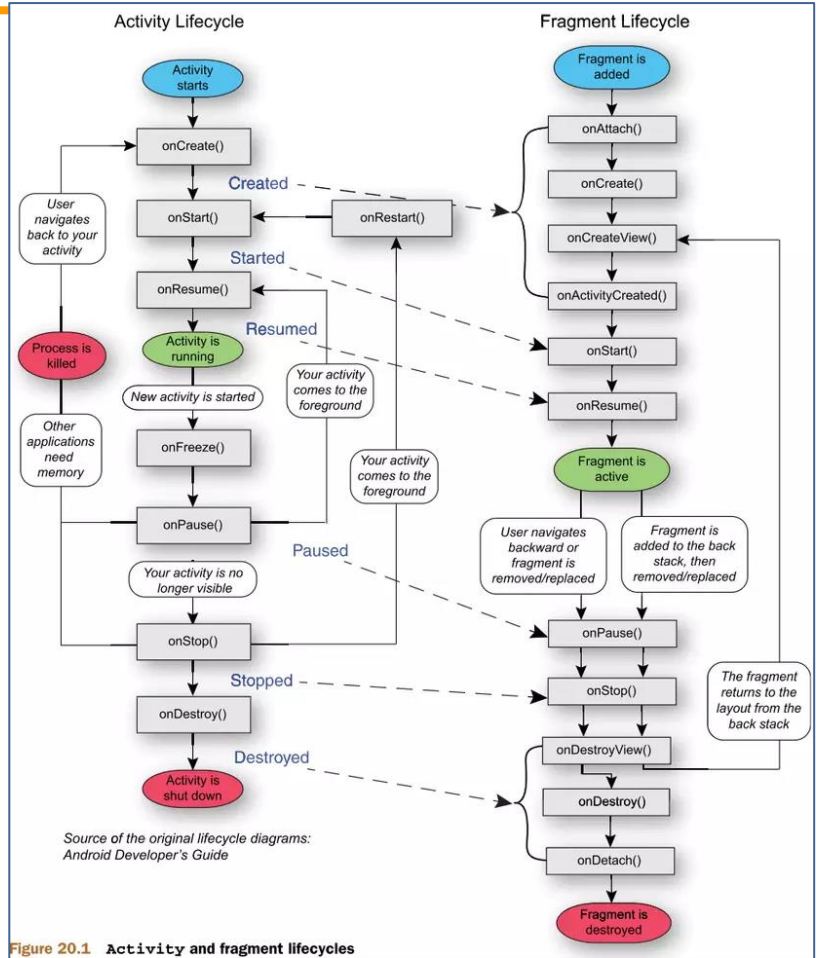


Figure 20.1 Activity and fragment lifecycles

VÒNG ĐỜI CỦA FRAGMENT

Giống như activity, fragment có thể tồn tại ở **3 trạng thái**:

•**Hoạt động (Resume):** Khi fragment được gắn vào activity, có thể nhìn thấy và có thể tương tác được.

•**Tạm dừng (Pause):** Nếu activity chứa fragment bị che lấp bởi 1 activity khác nhưng không bị che hoàn toàn, người dùng vẫn nhìn thấy được activity bị che lấp, chỉ là không tương tác được thì cả activity và fragment đều đi vào trạng thái tạm dừng.

•**Dừng (Stop):** Cũng giống như activity, fragment bị dừng khi bị thành phần nào đó che mất hoàn toàn. Ở trạng giai đoạn, các trạng thái của của fragment vẫn được giữ lại phòng trường hợp fragment được hiển thị trở lại. Và nếu nó không còn được hiển thị với người dùng thì fragment sẽ bị gỡ bỏ nếu activity bị hủy.

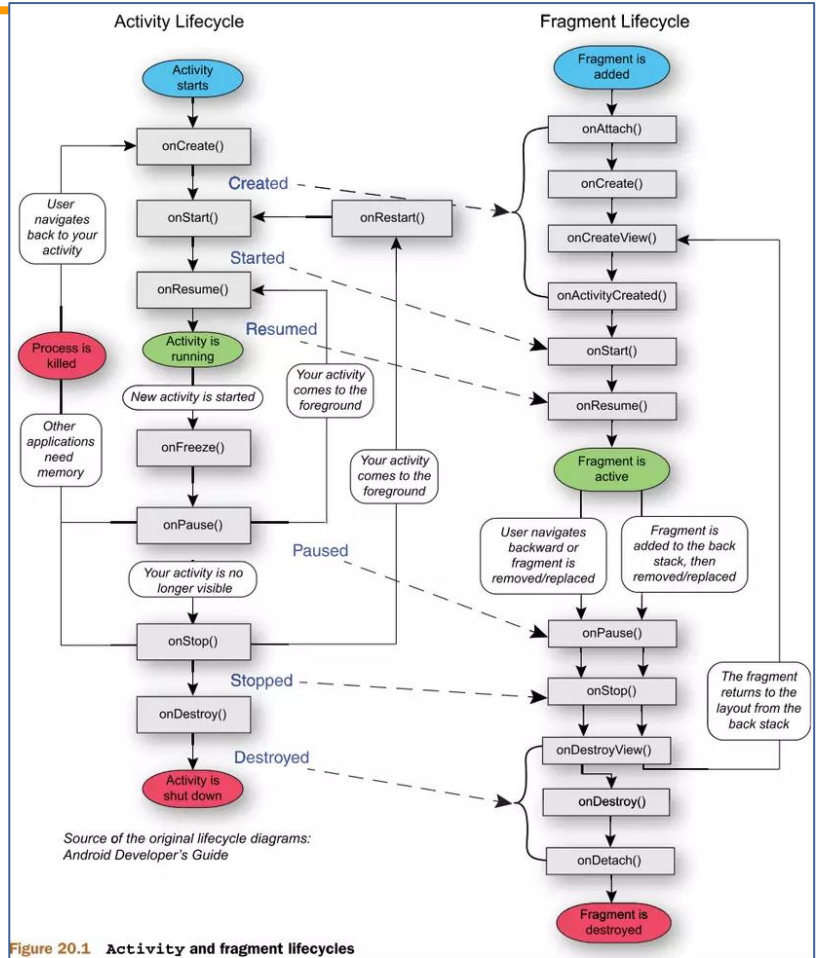
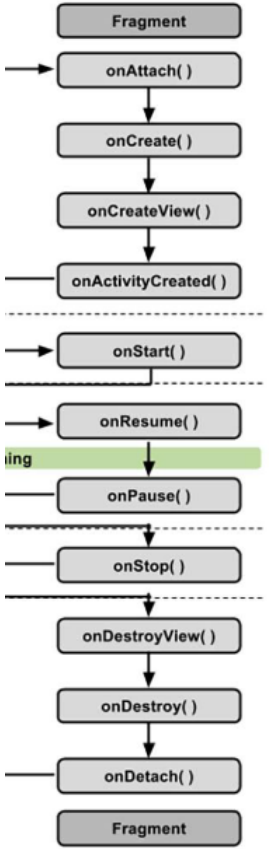


Figure 20.1 Activity and fragment lifecycles

VÒNG ĐỜI CỦA FRAGMENT

Các callback trong đời của fragment:

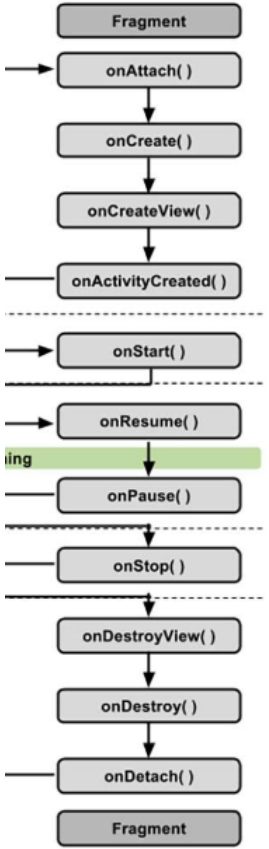
- **onAttach():** Callback này được gọi ngay khi fragment được gắn vào activity và được gọi **1 lần duy nhất** trong vòng đời của Fragment. Và ở giai đoạn này fragment đã biết được activity chứa nó rồi nên bạn có thể tận dụng callback này để kiểm tra sớm một số điều kiện nào đó.
- **onCreate():** Callback này được gọi khi fragment bắt đầu khởi dữ liệu đầu vào. Khác với activity, ở callback này fragment chưa thể khởi tạo giao diện cho màn hình, chúng ta cần phải đợi đến callback tiếp theo mới có thể khởi tạo giao diện. Callback này cũng được gọi một lần trong vòng đời fragment nên thường tận dụng để lấy dữ liệu từ bên ngoài truyền vào.
- **onCreateView()** Khi fragment bắt đầu vẽ UI lên màn hình, callback này được gọi nên chúng ta sẽ tận dụng callback này cho các thiết lập về giao diện. Theo như sơ đồ trên, thì callback này sẽ được gọi lại khi fragment được gỡ ra khỏi activity nhưng được đưa vào back stack, và được gọi lại hiển thị sau đó. Khi kết thúc callback này, hãy nhớ return một View. Lưu ý là chúng ta hoàn toàn có thể **return null nếu fragment không có UI.**



VÒNG ĐỜI CỦA FRAGMENT

Các callback trong đời của fragment:

- **onActivityCreated()** Ở callback này thì activity đã được khởi tạo hoàn toàn.
- **onStart()** Khi fragment bắt đầu được nhìn thấy bởi người dùng và chuẩn bị nhận tương tác.
- **onResume()** Người dùng hoàn toàn nhìn thấy và tương tác được với fragment
- **onPause()** khi activity đi vào **onPause()** (bị che mất 1 phần UI) thì fragment cũng đi vào **onPause()**. Chúng ta nên thực hiện việc sao lưu dữ liệu nếu cần thiết vì có thể người dùng sẽ rời khỏi fragment này. (ví dụ: App có thể là do bị kill ở hệ thống khi người dùng đang sử dụng thì nó cũng sẽ nhảy vào onPause trước đã)
- **onStop()** Fragment chính thức không còn được nhìn thấy nữa.
- **onDestroyView()** View sẽ bị hủy ở callback này. Nếu như fragment được add vào back stack thì khi được lấy lại sau đó, callback **onCreateView()** sẽ được gọi lại.
- **onDestroy()** Fragment sắp bị gỡ bỏ khỏi activity chứa. Khác với activity, fragment còn 1 callback cuối cùng để chính thức được xem như là đã chết **onDetach()**
- **onDetach()** Callback này gọi đến báo hiệu fragment sẽ được gỡ khỏi activity chứa nó và kết thúc vòng đời của fragment.



HIỂN THỊ THEO KIỂU ĐỘNG

Để thực hiện được việc add fragment vào FrameLayout mà bạn đã xây dựng sẵn bằng code java, chúng ta cần có 1 số lớp liên quan sau:

- **FragmentManager:**
  - Lớp dùng để quản lý các Fragment, lớp này được tích hợp vào trong mỗi activity để giúp các activity có thể dễ dàng để thêm (add), xóa (remove) hoặc thay thế (replace) các fragment ra khỏi một vùng không gian một cách linh động.
  - FragmentManager thêm, xóa, thay đổi các fragment dựa vào FragmentTransaction. Ta sử dụng FragmentTransaction thông qua phương thức beginTransaction() từ FragmentManager.
- **FragmentTransaction:**
  - add() – Khi FrameLayout còn rỗng phương thức này để add fragment vào cho FrameLayout đó. Nếu bạn tiếp tục thêm fragment khi đã tồn tại thì các fragment sẽ chạy song song nhưng fragment sau sẽ che mất view của fragment trước.
  - replace() – thay thế một fragment đang có sẵn ở FrameLayout bằng một fragment nào đó khác.
  - remove() – gỡ bỏ fragment ra khỏi một FrameLayout.
  - addToBackStack() – đưa fragment ở transaction hiện tại vào Back Stack khi đó fragment bị thay thế hay bị gỡ ra khỏi FrameLayout ở transaction này sẽ không bị xóa khỏi hệ thống mà vẫn còn được quản lý bên trong Back Stack.
  - commit() – muốn thực hiện được bằng add(), replace() hay remove() thì bạn cũng phải gọi commit(), để FragmentTransaction biết sẽ bắt đầu thực hiện các transaction mà bạn đã ra lệnh đó.

GIAO TIẾP GIỮA CÁC FRAGMENTS

- **Sử dụng Activity:**

- **Sử dụng Interface:**