

Họ và tên: Phạm Duy Khánh

Mssv: 21522211

Lớp: Phát triển ứng dụng trên thiết bị di động - NT118.021

Report về quá trình gọi các phương thức trong vòng đời của 1 activity

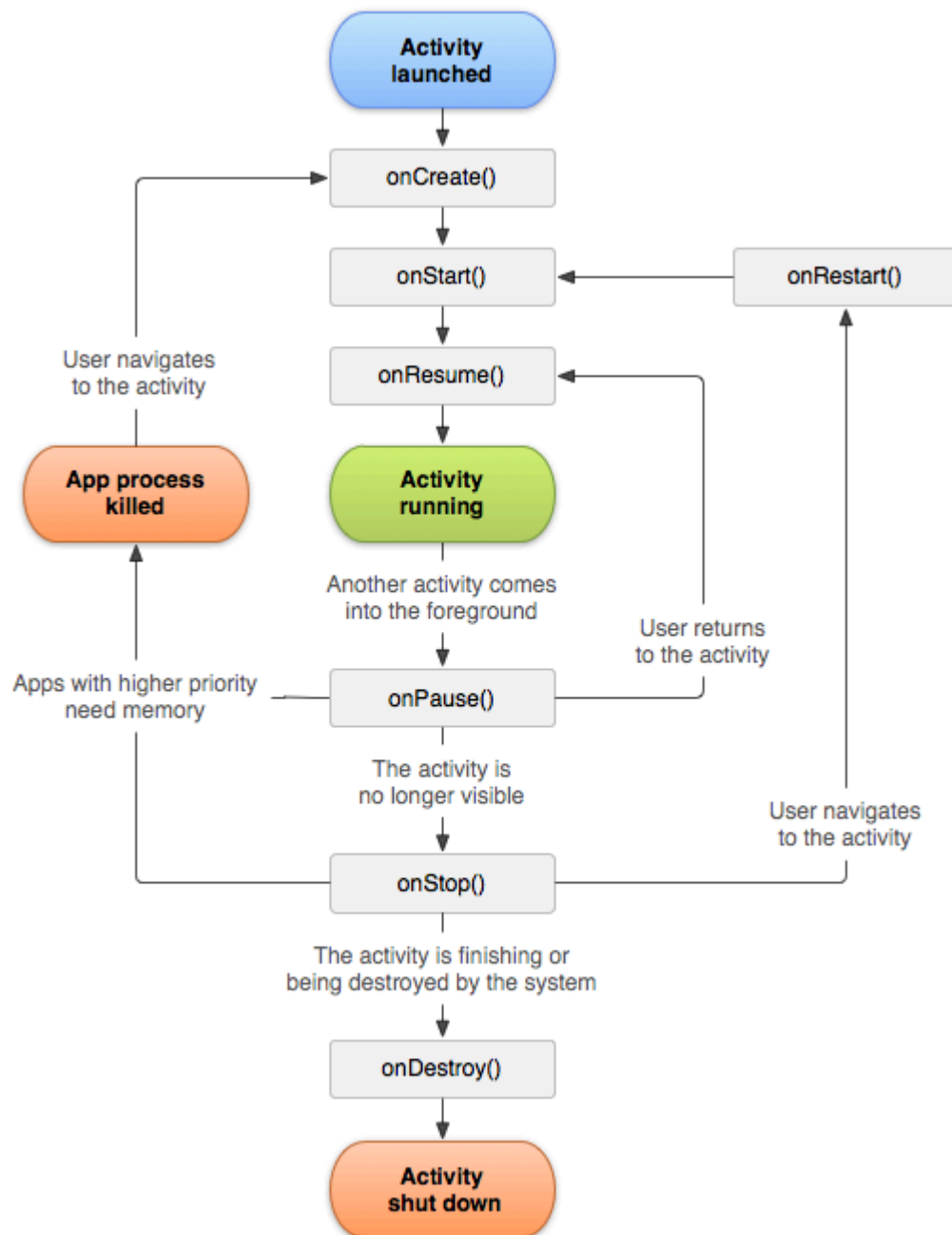
1. Activity

- Một **activity** thường đại diện cho một **single visual user interface** mà thông qua đây các **action** được tiến hành, thao tác. Các **activity** sẽ kết hợp với nhau để tạo nên một UI thống nhất. Mặc dù vậy nhưng mỗi **activity** lại thực sự hoạt động độc lập với nhau. Khi khởi động một ứng dụng thì sẽ có một main Activity được gọi và hiển thị màn hình giao diện của ứng dụng cho phép người dùng tương tác. Để việc chuyển từ một **activity** này sang một **activity** khác thì bạn sẽ phải start một **activity** thông qua một **activity** hiện thời thông qua **intents**.

2. Life Cycle States

- Một **activity** cơ bản có 3 states:
 - active or running : **activity** sẽ active hoặc running khi mà nó nằm ở phía trên của màn hình.
 - paused : nó sẽ ở trạng thái **paused** nếu activity ấy không được focus lên nữa nhưng vẫn được nhìn thấy bởi user. Ví dụ như có một activity khác nằm trên cùng và activity này là trong suốt hoặc activity này không chiếm full màn hình chẳng hạn. Trong trường hợp, paused activity vẫn hoàn toàn alive (nó duy trì tất cả các trạng thái và thông tin thành viên và vẫn duy trì liên kết tới cửa sổ quản lý), nhưng activity này có thể bị kill bởi hệ thống trong trường hợp có ít bộ nhớ.
 - stopped : activity bị stop hoàn toàn nếu nó bị cản trở bởi một activity khác. Nó vẫn có thể duy trì được tất cả các trạng thái và thông tin thành viên. Tuy nhiên, nó sẽ không thể quan sát được bởi user do màn hình cửa sổ của nó đã bị ẩn đi và nó thông thường sẽ bị kill bởi hệ thống trong trường hợp cần.

3. Life Cycle Events

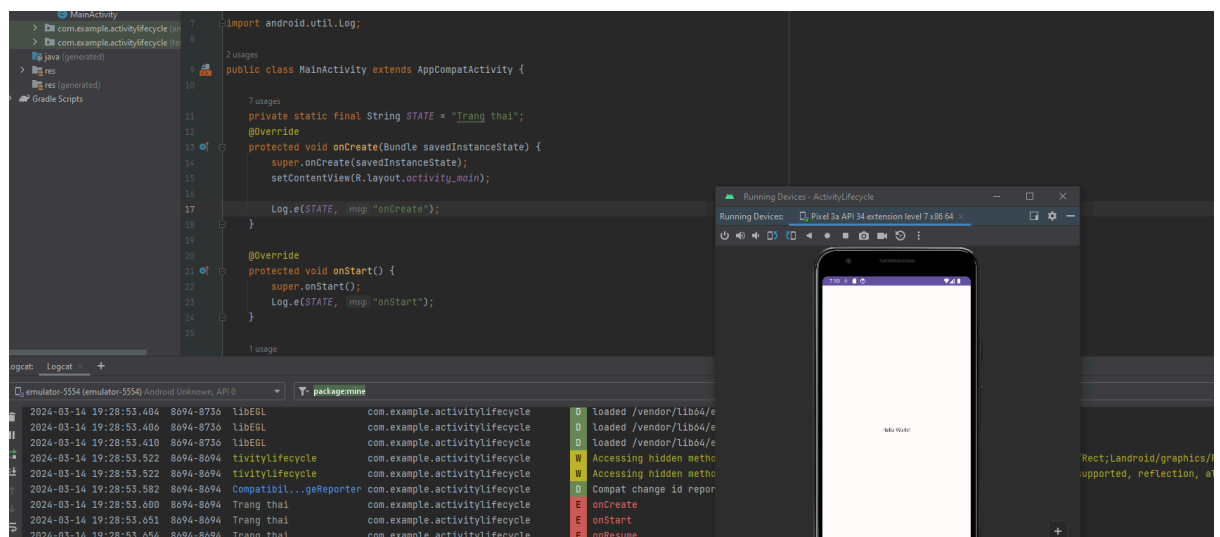
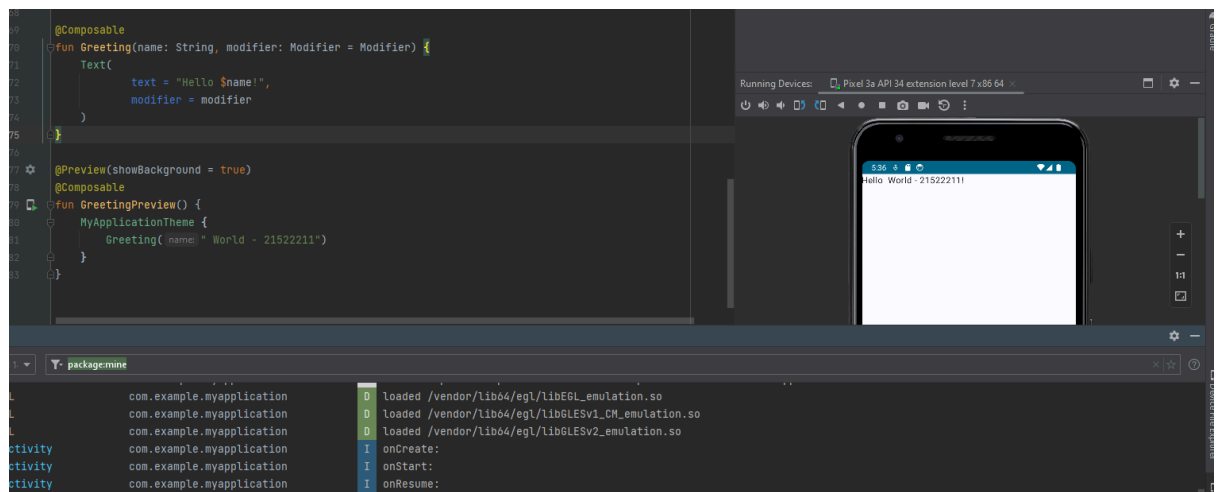


- Nếu một activity bị pause hoặc stop, hệ thống có thể drop nó khỏi bộ nhớ hoặc sẽ hỏi nó để kết thúc (thông qua phương thức `finish()`), hoặc đơn giản là kill process đó. Khi nó được hiển thị lại tới user, nó sẽ hoàn toàn được khởi động lại và khôi phục lại tới trạng thái trước đó của nó. Dưới đây là danh sách các phương thức chuyển trạng thái của activity :
 - `void onCreate(Bundle savedInstanceState)`
 - `void onStart()`
 - `void onRestart()`
 - `void onResume()`

- void onPause()
 - void onStop()
 - void onDestroy()
 - Tất cả những phương thức trên bạn đều có thể override để thực hiện công việc mình cần khi trạng thái thay đổi.
- BẮT BUỘC**
- Tất cả các activity cần phải chạy phương thức onCreate() để thực hiện bước setup đầu tiên khi đối tượng lần đầu tiên được khởi tạo.

4. Life Cycle Methods

**** Method **: onCreate()**



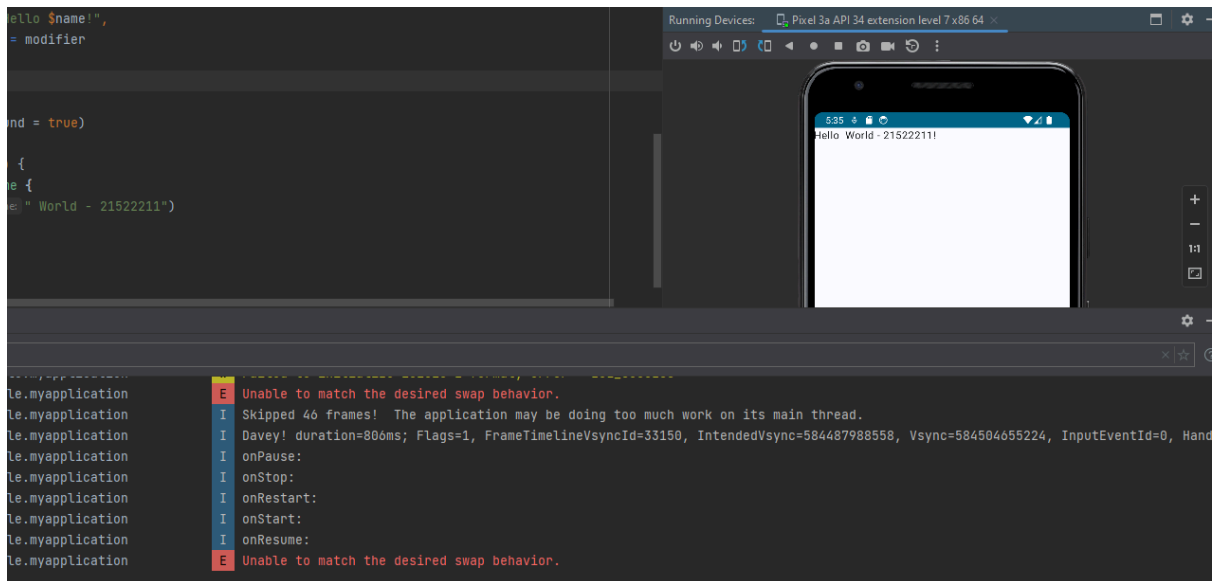
Được gọi khi lần đầu tiên activity được tạo

Đây là lúc bạn thiết lập tất cả các normal static như là tạo views, bind data vào list v.v...

Phương thức này truyền đối tượng Bundle chứa state trước đây của activity, trong trường hợp bắt được state này.

Tiếp theo sau sẽ là onStart()

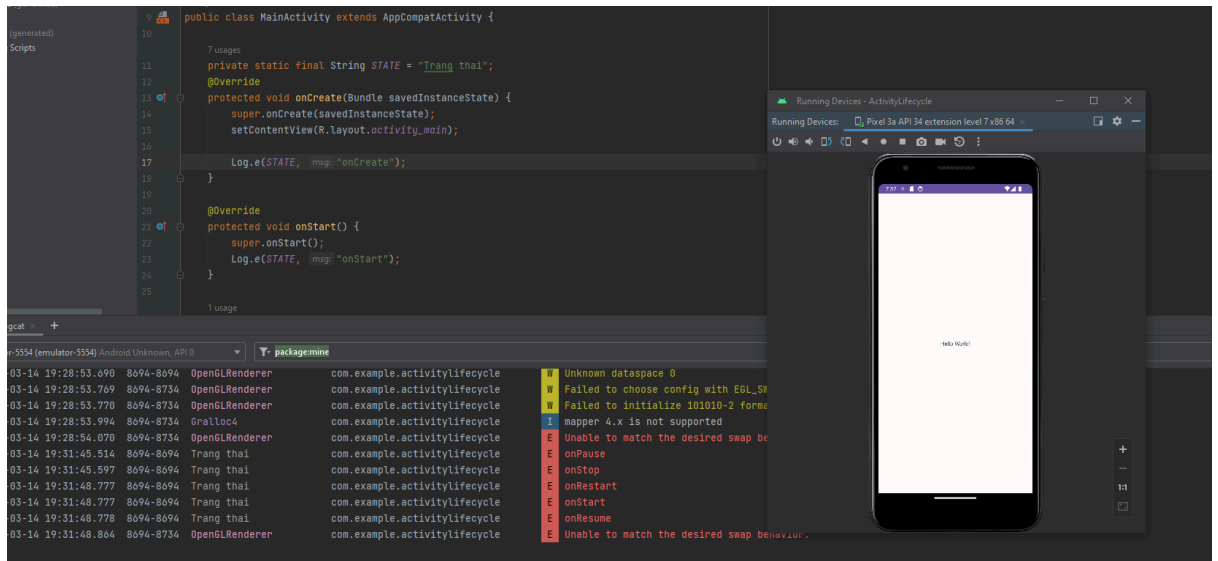
**** Method **** : onStart()



Gọi ngay khi activity bị stop hoặc ưu tiên để được start tiếp.

Luôn theo sau onStart()

**** Method **** : onStart()

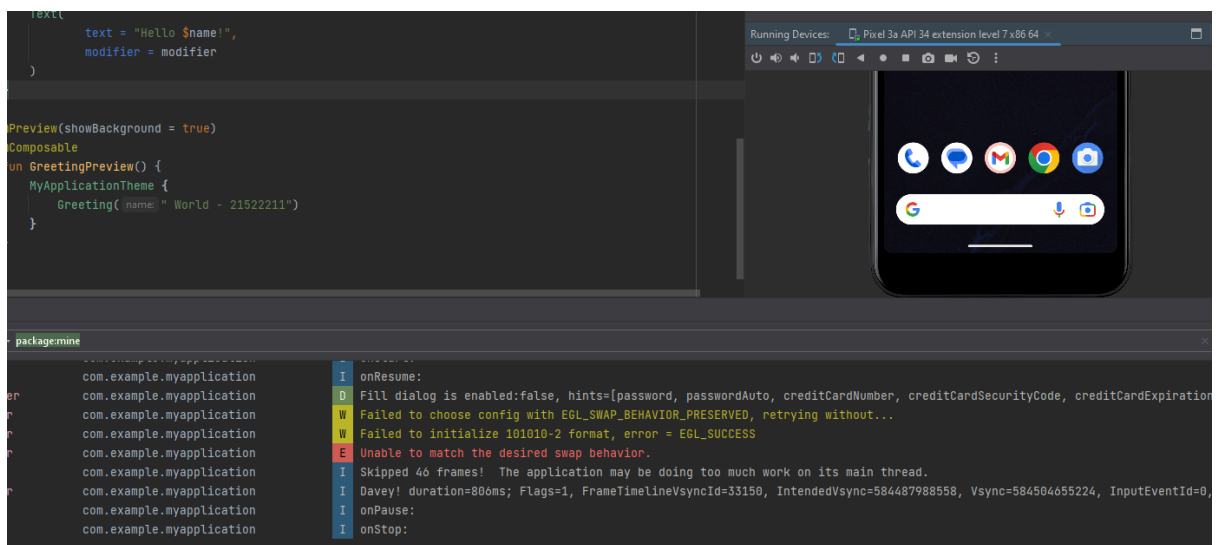


Gọi trước khi activity bắt đầu tương tác với user.

Tại lúc này, activity nằm trên cùng của stack

Luôn theo sau bởi onPause().

**** Method **: onPause()**



Gọi khi hệ thống chuẩn bị bắt đầu khôi phục activity khác

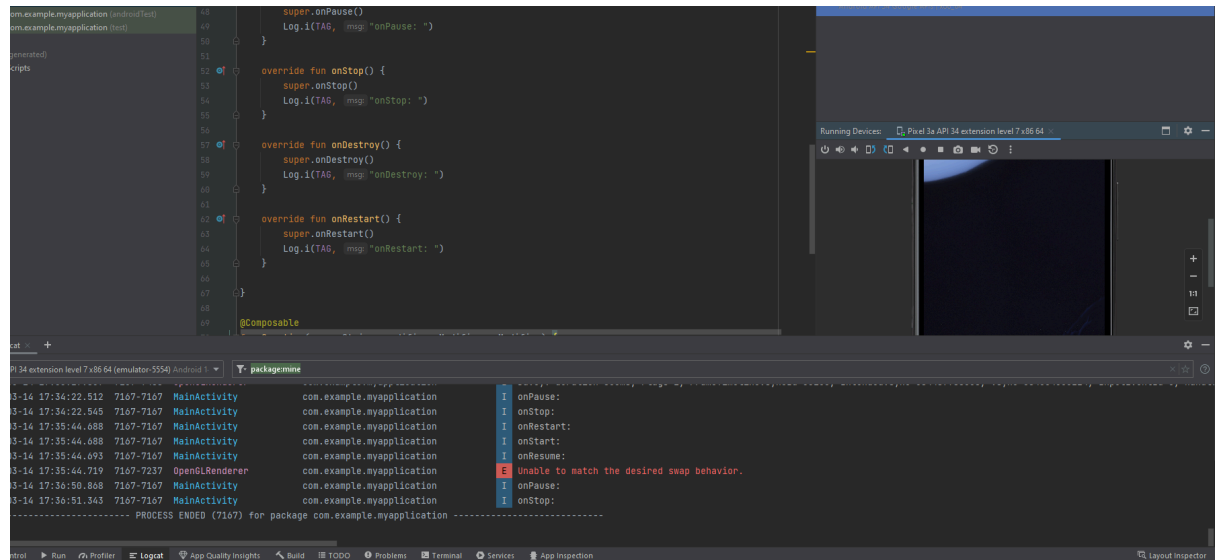
Phương thức này sẽ commit tất cả những thay đổi chưa được lưu và dữ liệu, dung các animations và các thứ khác tiêu tốn CPU.

Nó nên được thực hiện nhanh nhất có thể, bởi vì activity tiếp theo sẽ không thể resume được cho đến khi phương thức này trả giá trị về.

Là phương thức theo sau onResume() nếu activity được trả về từ trên cùng màn hình hoặc onStop() nếu nó trở nên ẩn đối với user.

Activity trong state này có thể bị kill bởi hệ thống.

**** Method **: onStop()**



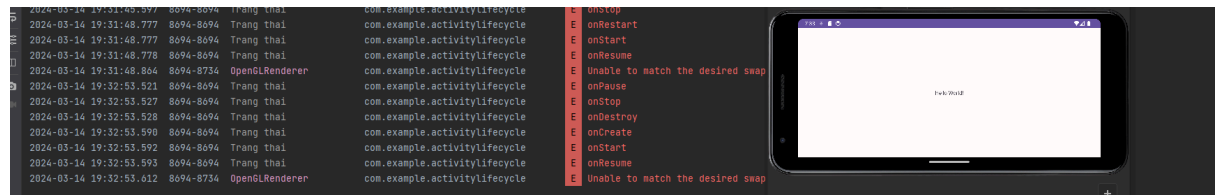
Được gọi khi activity không còn hiện lên đối với người dùng.

Phương thức này xảy ra bởi vì nó bị destroy hoặc bởi vì một activity khác resume lên và đè lên nó.

Xảy ra sau onRestart() nếu activity trở lại từ màn hình tương tác với user hoặc onDestroy() nếu activity này bị biến mất.

Activity trong state này có thể bị kill bởi hệ thống.

**** Method **: onDestroy()**



Được gọi khi acitivity bị destroy

Đây là phương thức cuối cùng mà activity sẽ được gọi

Nó có thể được gọi hoặc bởi vì activity đang kết thúc (có ai đó gọi `finish()`), hoặc bởi vì hệ thống đang tạm thời hủy các instance của activity này để tạo nên không gian mới.

Bạn có thể phân biệt giữa 2 trường hợp trên với phương thức `isFinishing()`.

Activity trong state này có thể bị kill bởi hệ thống.

5. Killable States

Một activities trong trạng thái killable có thể bị dừng bởi hệ thống tại bất kỳ thời điểm nào ngay sau khi phương thức trả về, mà không cần thực hiện thêm bất kỳ dòng code nào nữa.

Có 3 phương thức **`onPause()`**, **`onStop()`**, **`onDestroy()`** là killable

`onPause()` là phương thức duy nhất đảm bảo sẽ được gọi trước khi process bị kill - **`onStop()`** và **`onDestroy()`** có thể không được gọi.

Bạn nên sử dụng **`onPause()`** để lưu trữ dữ liệu tạm thời trong bộ nhớ.