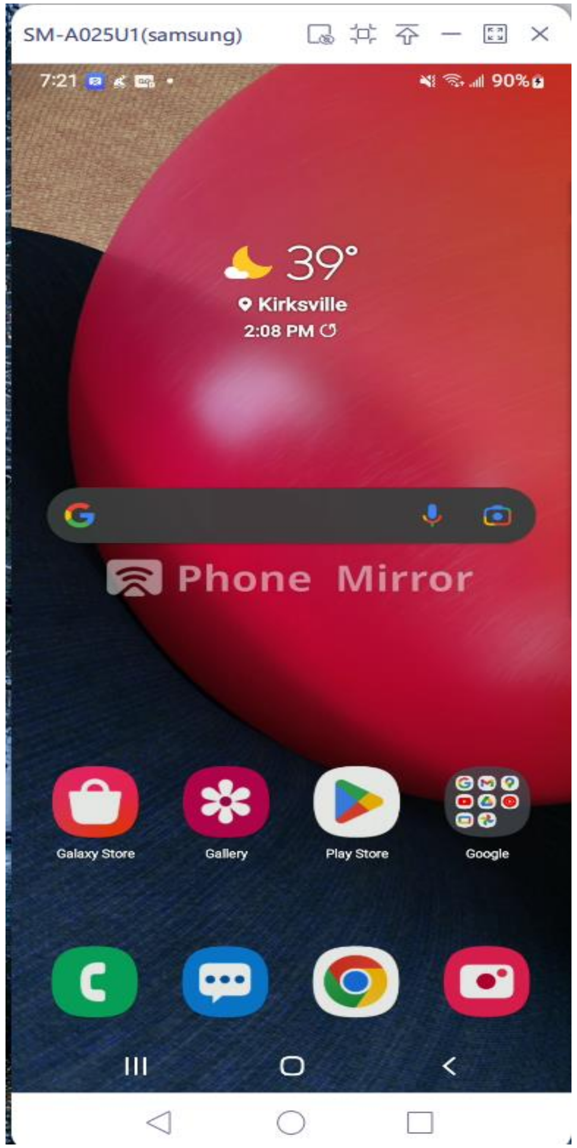
Android Programing 101 (Comments)

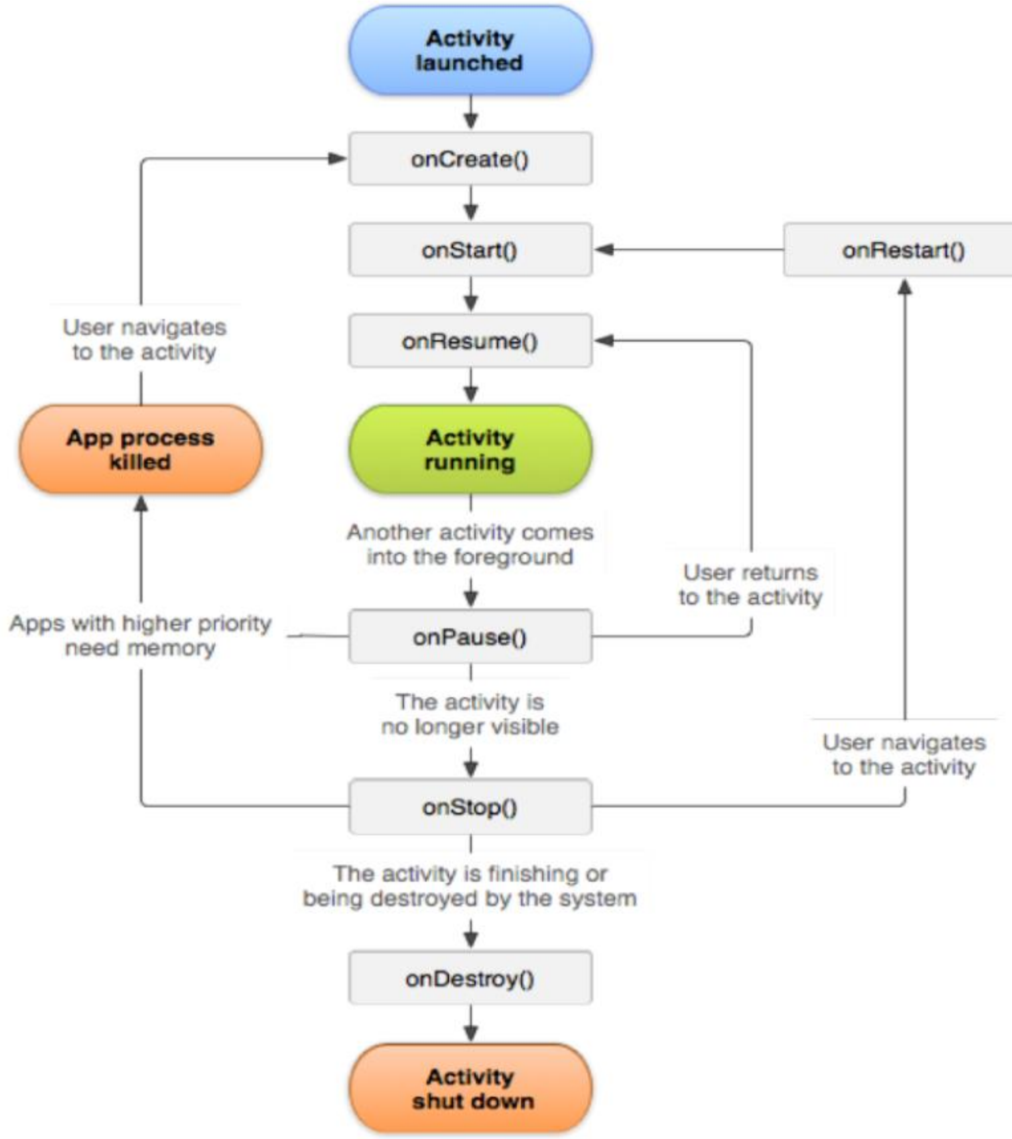
Dr. Charles Yu

- More about the Activity Lifecycle
- More about the Fragment Lifecycle

More about the Activity Lifecycle

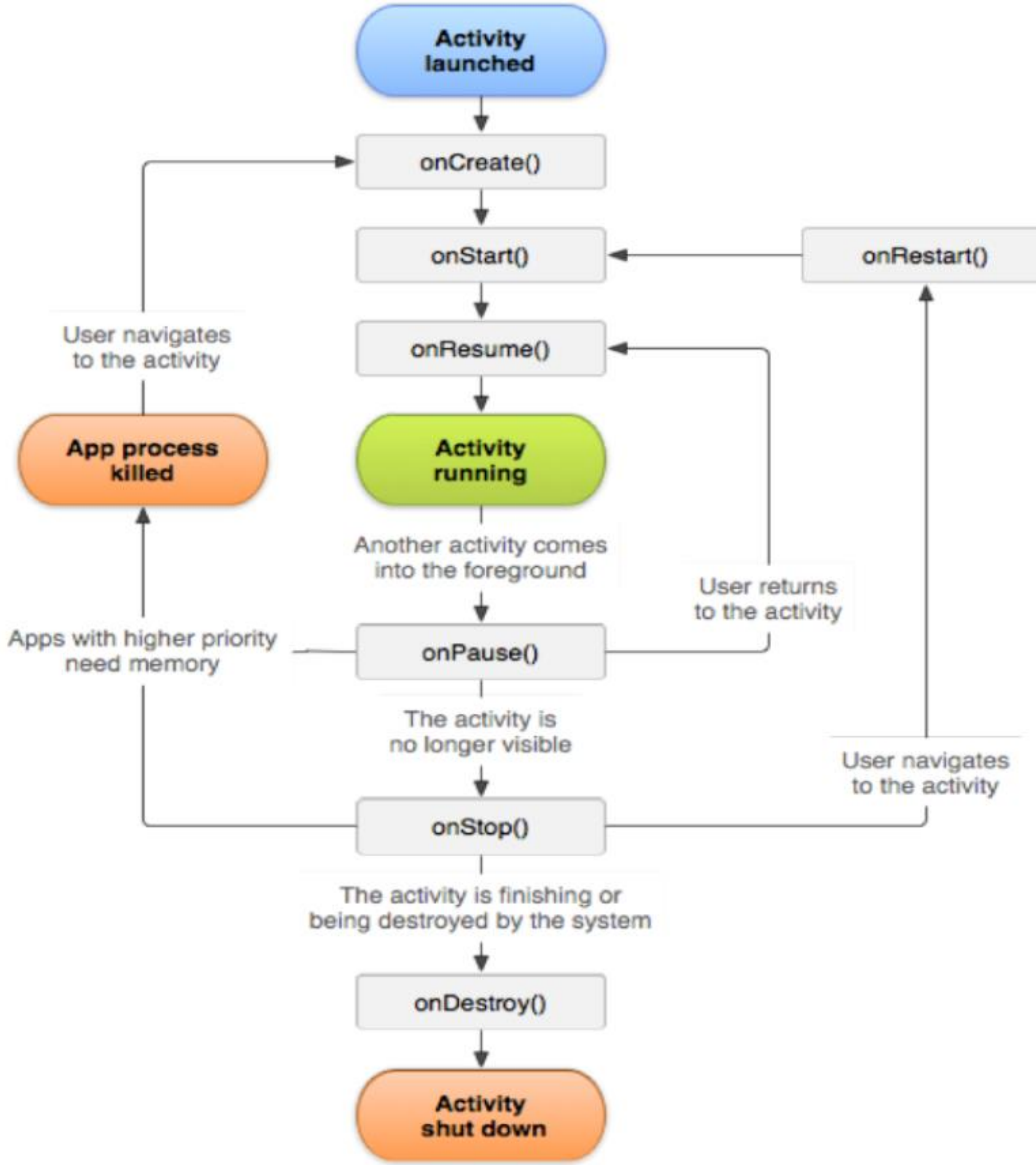
- In the Android phones, we have 3 buttons.
- This is the mainsteam of the design recently
- "<" for the Back key,
- "|||" for the list of App history
- "O" for the home screen




Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

More about the Activity Lifecycle

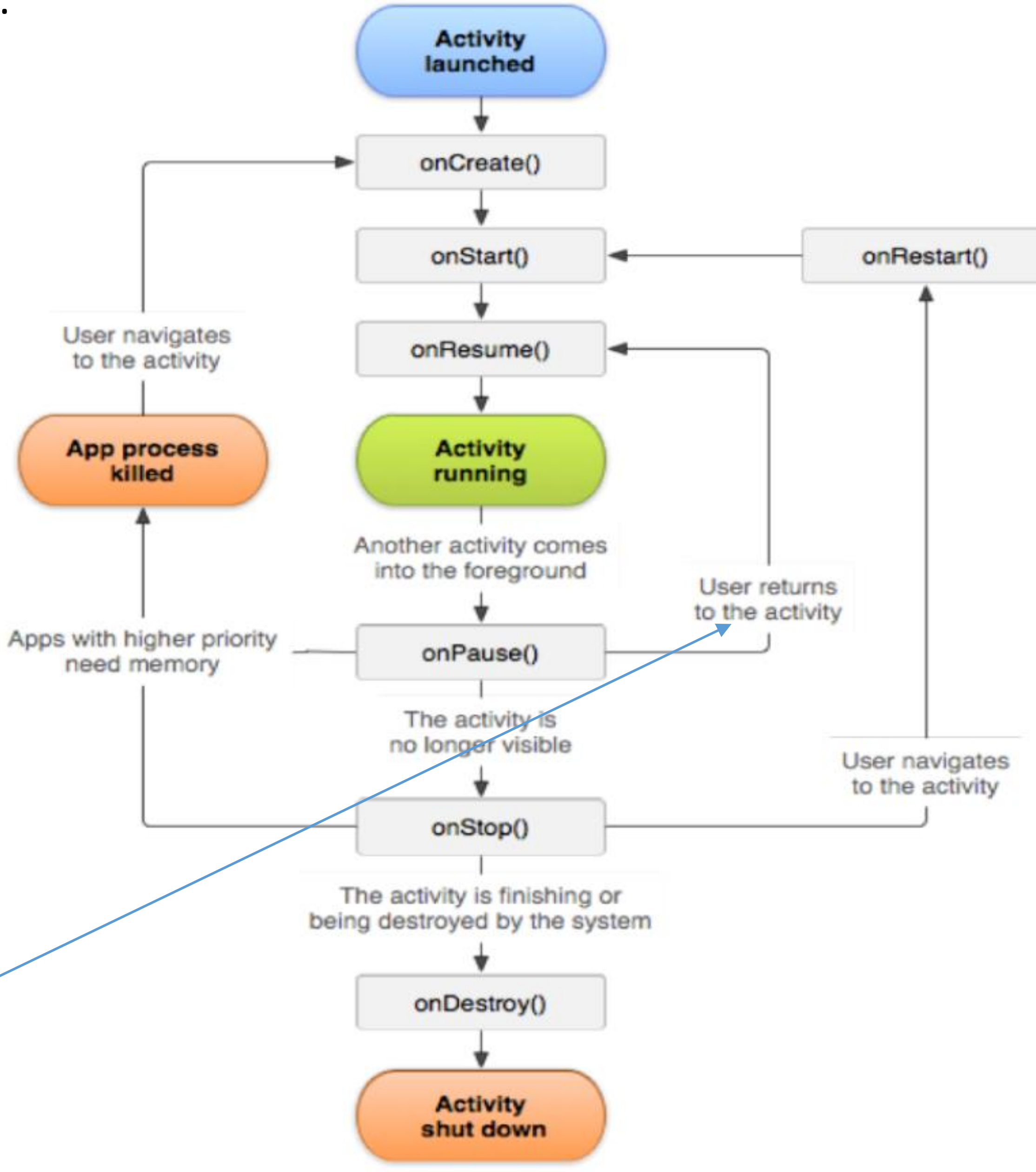
- If we press the **[Back]** key, **onPause()** will be triggered. Here is the detail:
    - Assuming there is an App, it has 2 activities in total. Now, Activity1→Activity2
    - We are now on **Activity2**
    - When **[Back]** is pressed, **Activity2** will run the onPause() and the screen will bring us back to the Activity1
    - Activity2 is now **no longer visible** (because Activity1 is **on top of that**)
    - Then, the Activity2 will wait for a short while, it will run into onStop()


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

More about the Activity Lifecycle

i.e. swipe out the snapshot in the home screen.

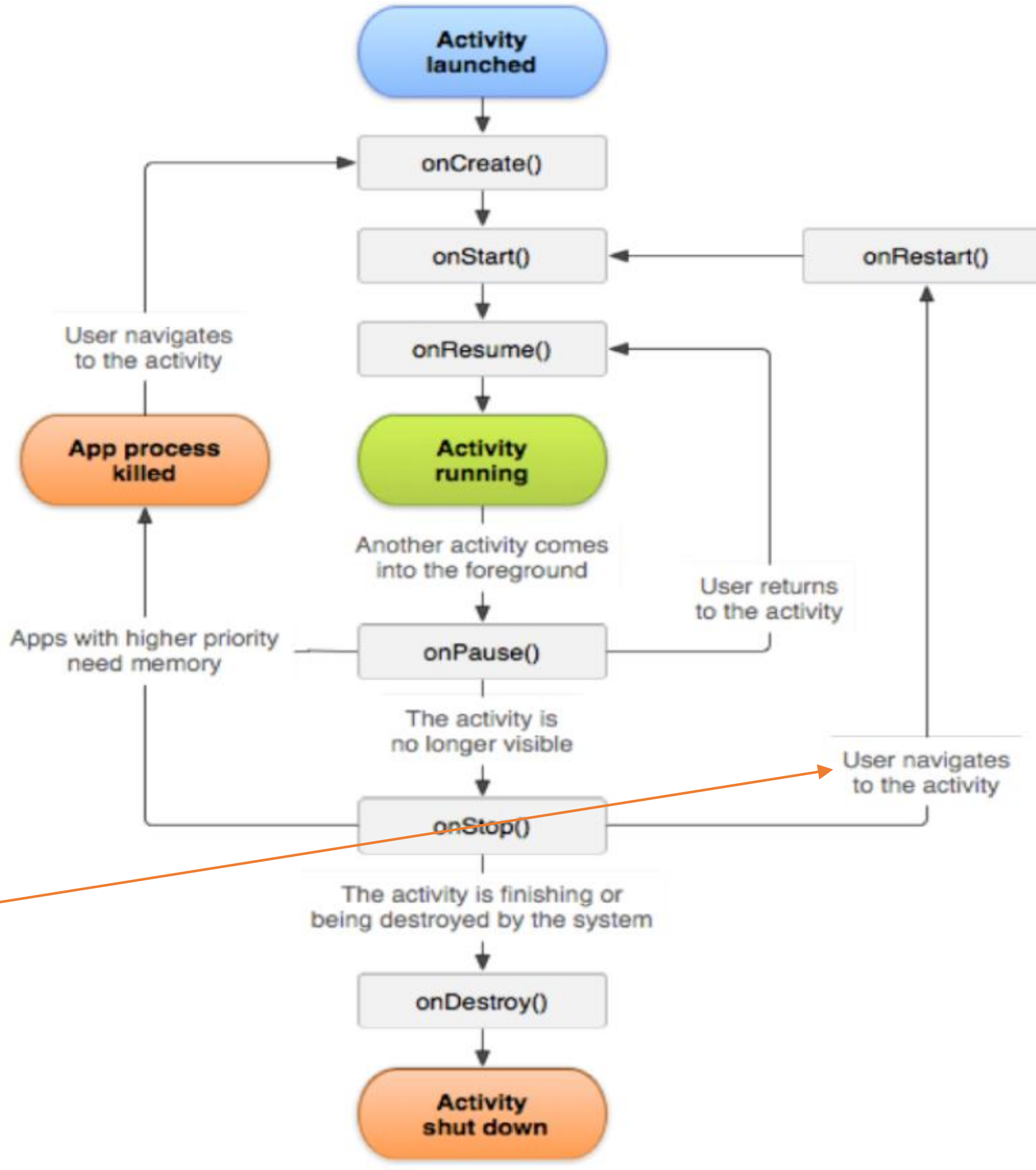
- Maybe for a little bit longer, either destroyed by user (**manually** or **programmatically**) or by Android OS, it will run into onDestroy(). Finally, It is heading to its FINAL DESTINATION.
- What if, when the **Activity2** is around the onPause() and re-trigger again by **Activity1,** by clicking some buttons like "NEXT" from **Activity1**?
  - The **Activity2** will call the onResume() by following the line "**user returns to the activity**"



Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

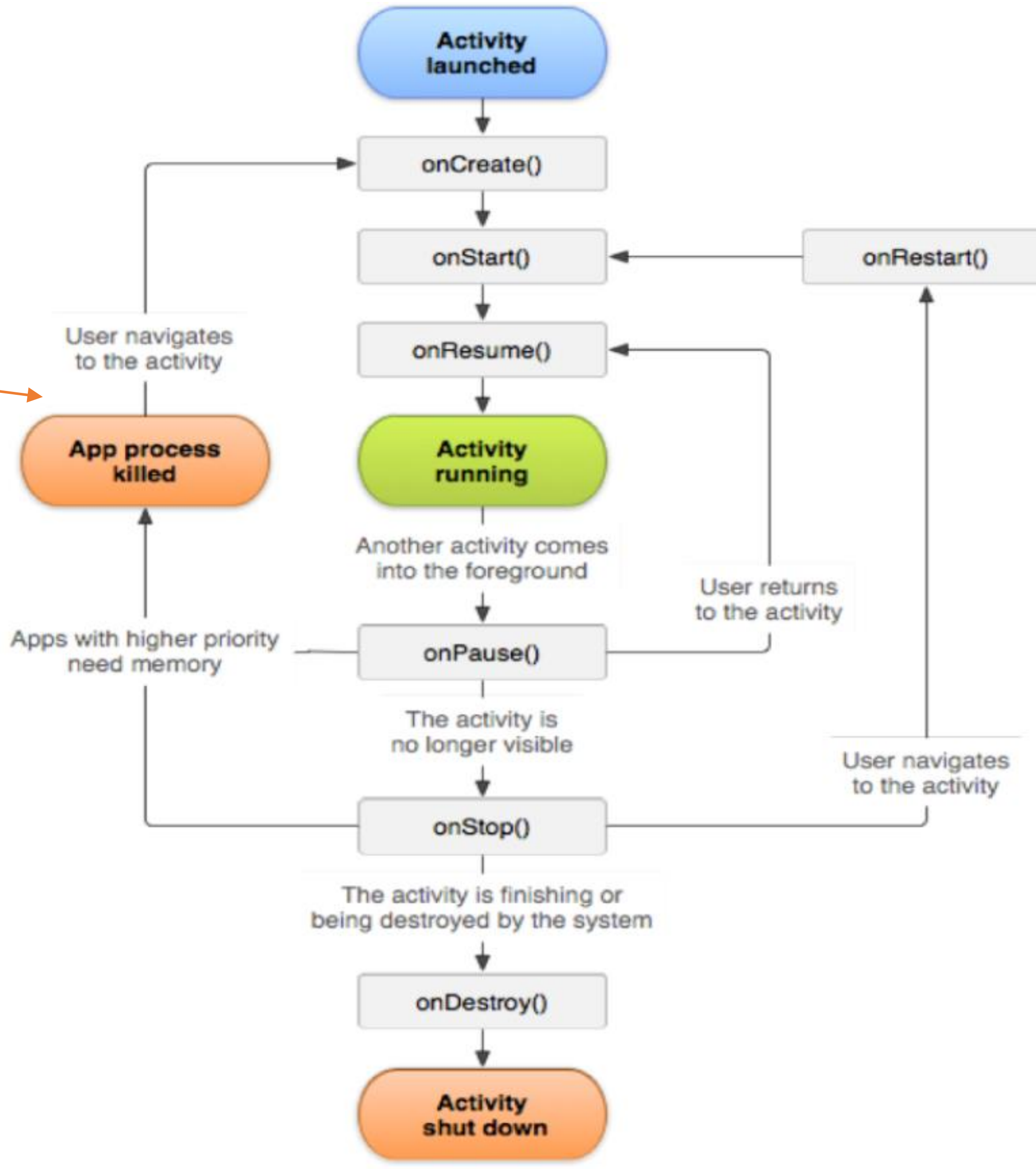
More about the Activity Lifecycle

- What If we press the **[Home]** key?
- You will quickly jump out of this App to the home screen and the Android OS will quickly understand that, maybe you planning not to use this App, at least, for a very short while.
  - So, the App will experience **onPause() and onStop()** calls quickly. It will stay here for a short while.
  - But what if the user quickly click the **icon** (in Android App drawer) for that App again? There will be a loopback by following the line "**User navigates to the activity**"


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

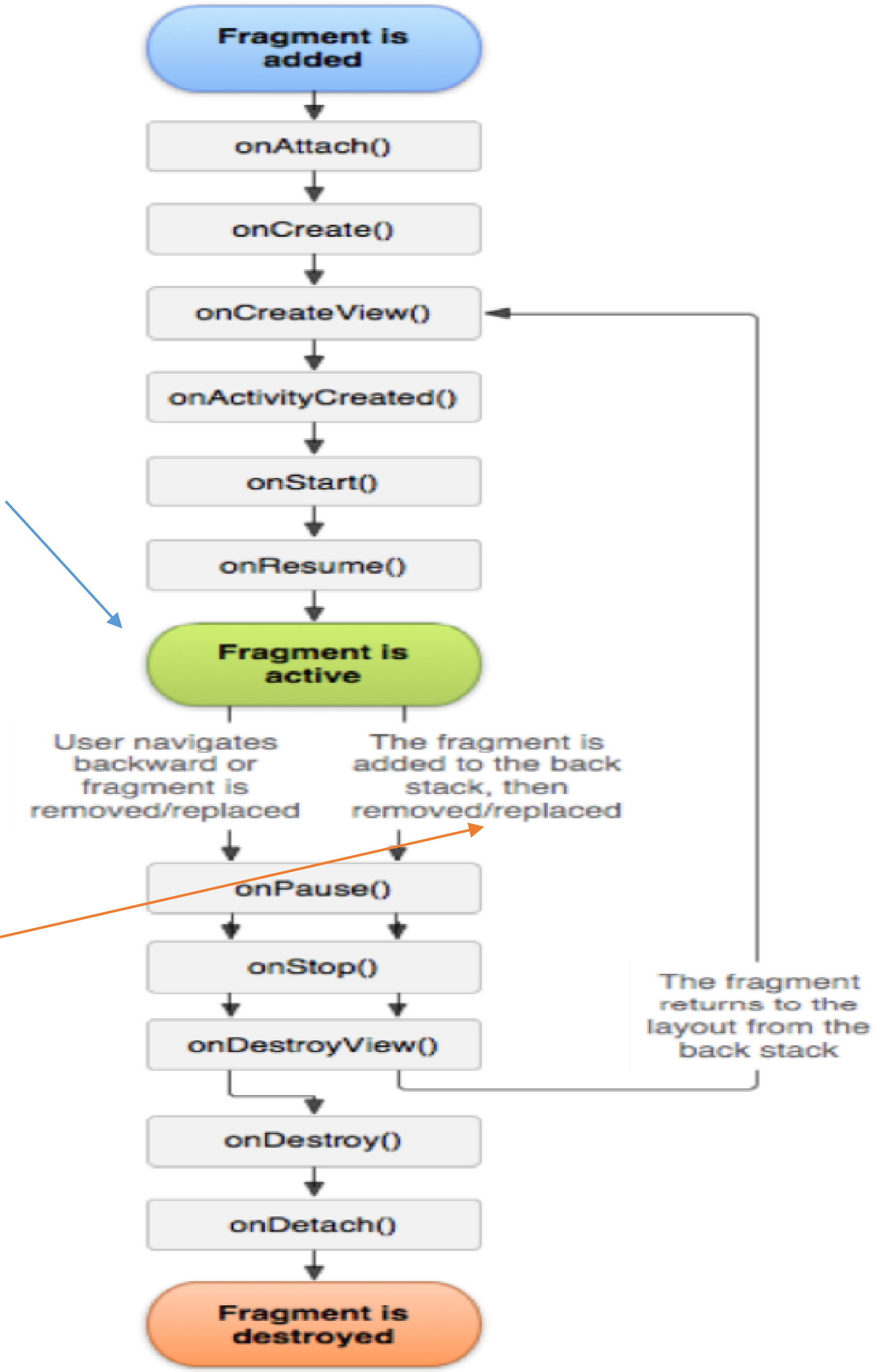
More about the Activity Lifecycle

- The app might get killed early in some cases!
- If some other Apps are extremely memory demanding, this poor app might get killed by following the 2 lines on the LHS after calling the onPause() and onStop(), respectively.
- After the death of the App, if the user click the icon of the poor App again, it will come back by calling onCreate(). No worries!


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

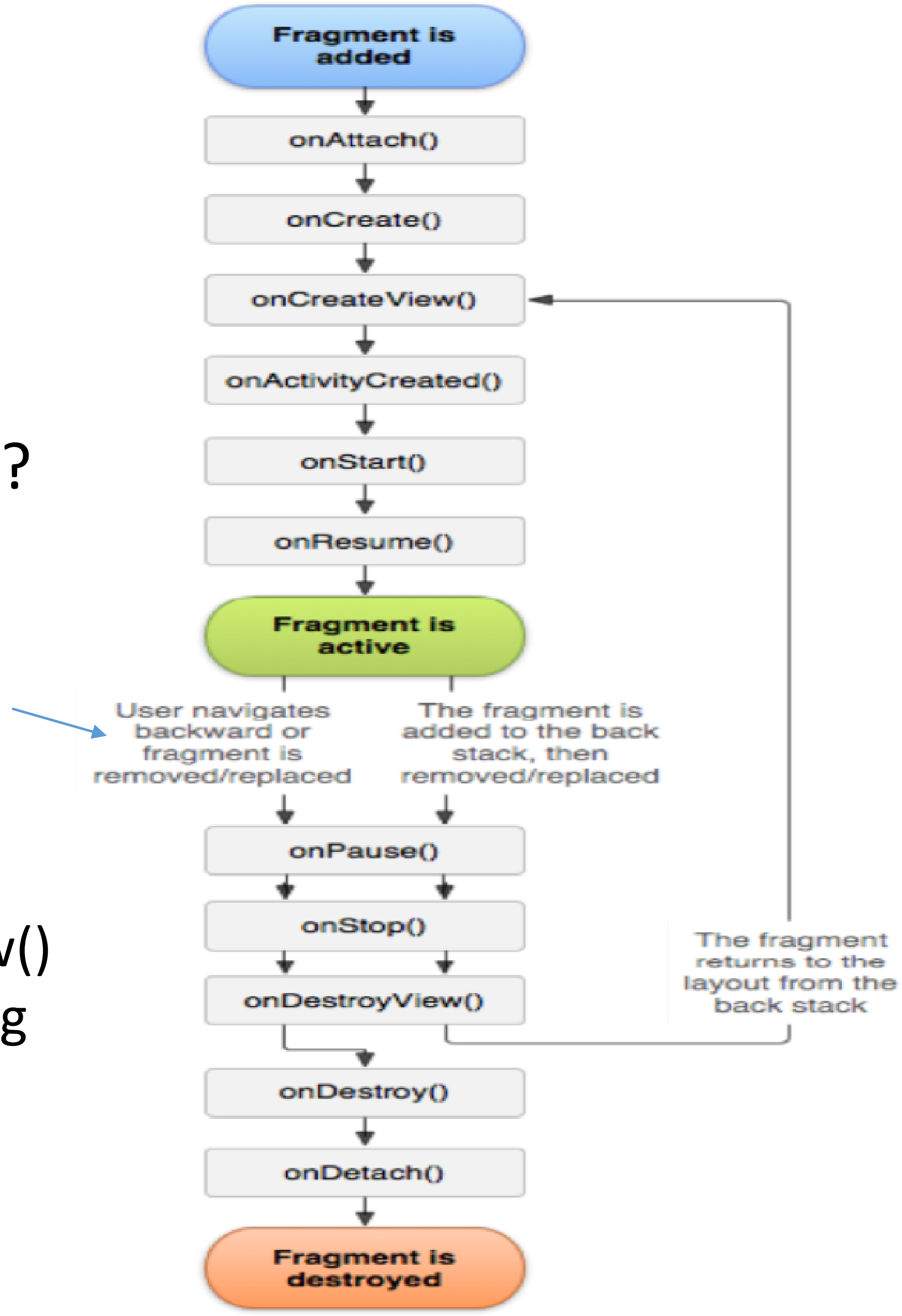
More about the Fragment Lifecycle

- After the fragment is coming into the foreground, as you can see --- the fragment is active
  - What if Fragment1→Fragment2 by clicking [Next] in Fragment1?
  - Fragment1 will be put to the **back stack**
  - Fragment1 is not visible now
  - Fragment2 is coming to the foreground
  - The back stack, it is just a list of entries of Fragments. The back stack is just a stack.
  - When this happens, Fragment1 will follow this line on the RHS
  - So, probably, Fragment1 is heading to its "Final Destination". But it is not necessarily!



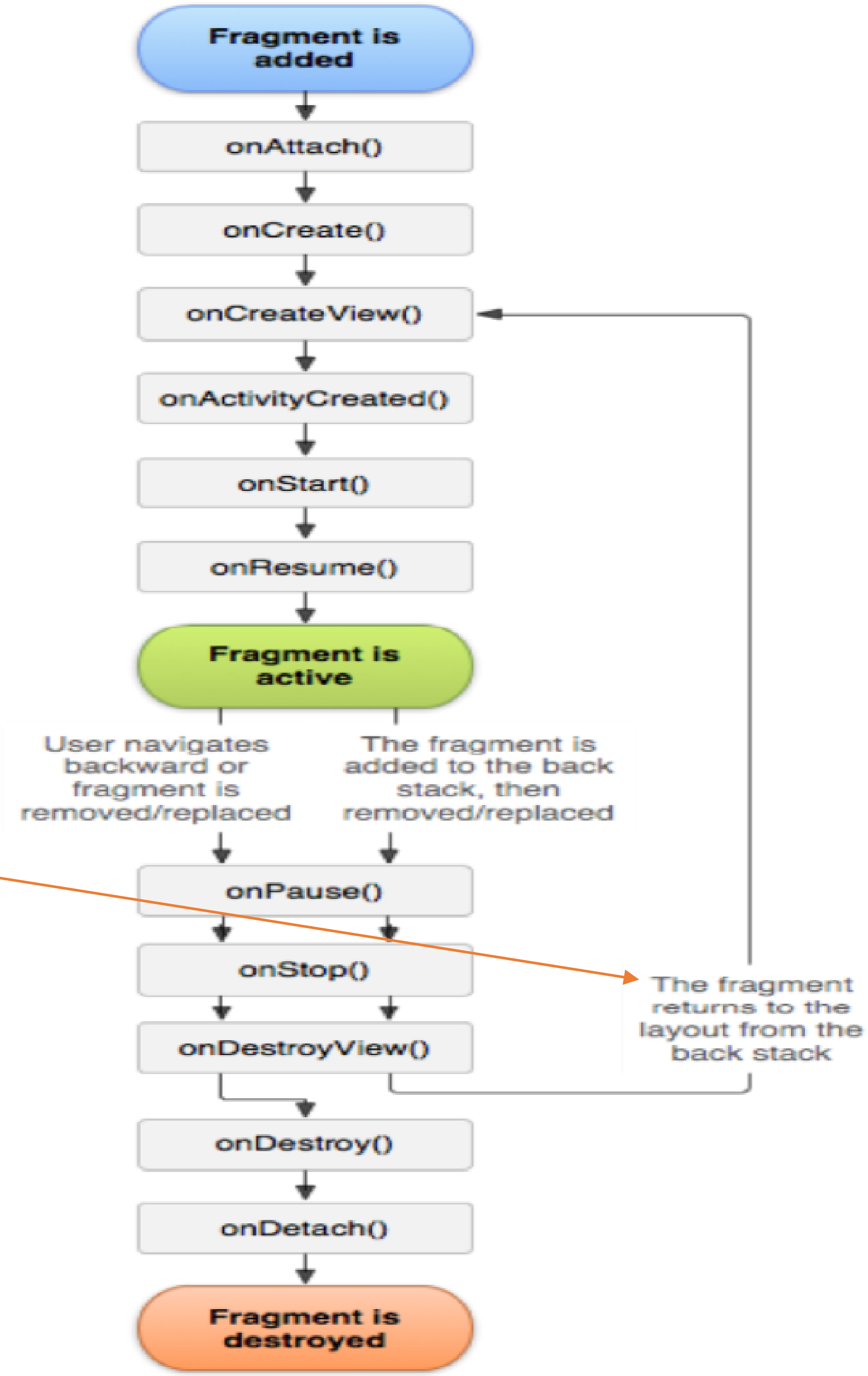
More about the Fragment Lifecycle

- From Fragment2's point of view, what if the user in Fragment2 and click the [Back] key?
  - Fragment2 will be place to the back stack and Fragment1 will be brought back.
  - First, the Fragment2 will follow this line on the LHS
    - Fragment2 will be put onto the back stack
  - Second, the Fragment1 will be brought back (from the back stack) after a call to onDestroyView()
  - Finally, the Fragment1 come back again by calling onCreateView()



More about the Fragment Lifecycle

- We can have a quick guess, for those “not using” fragments, if they are in the back stack **for a long while**, they will run the onStop() and stay here for a while, waiting for a call-back from the back stack



More about the Activity Lifecycle

- So, do the right thing,

in the right place!

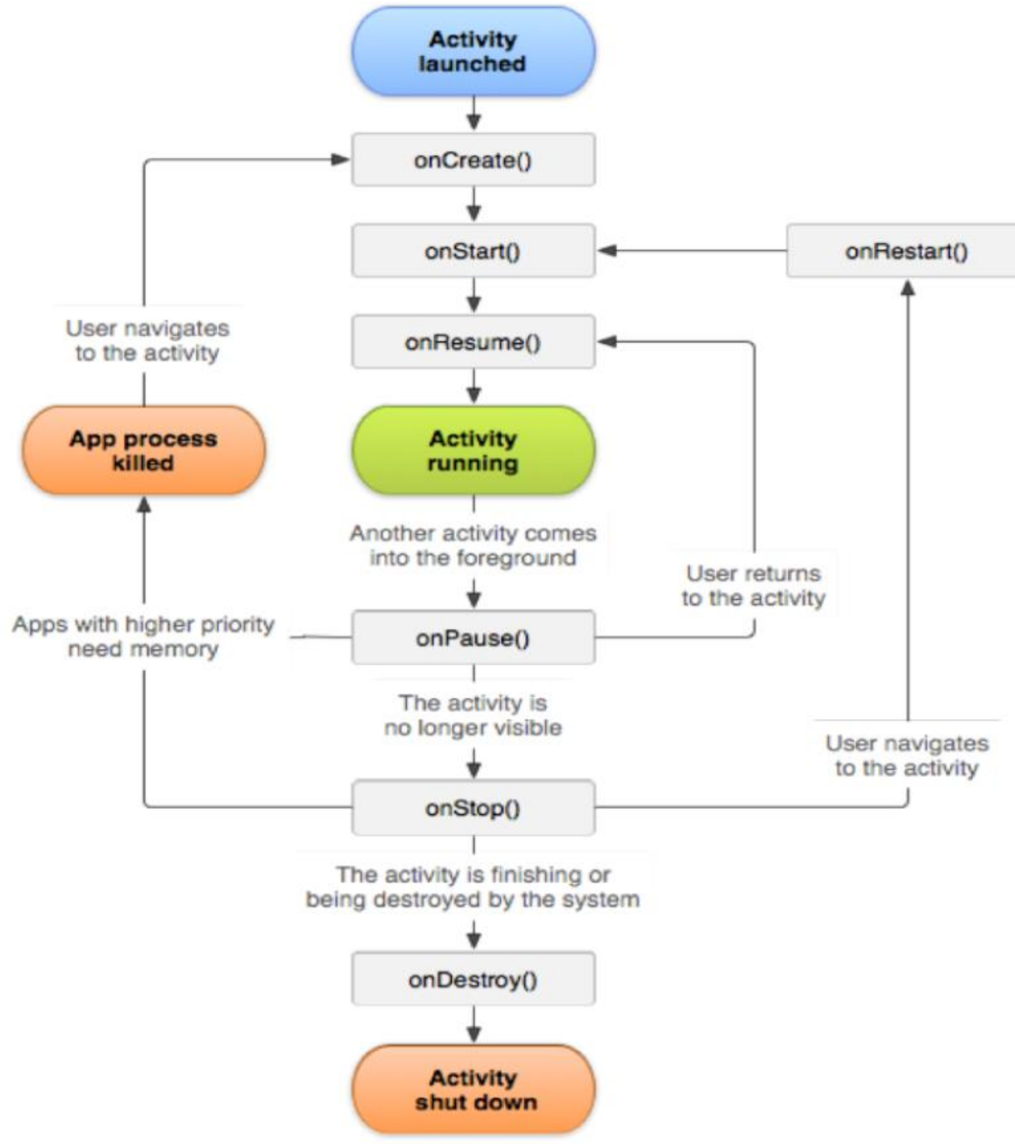
- Under such a kind of system mechanism,

what if I requested some system resources

(i.e. Bluetooth used in the App)

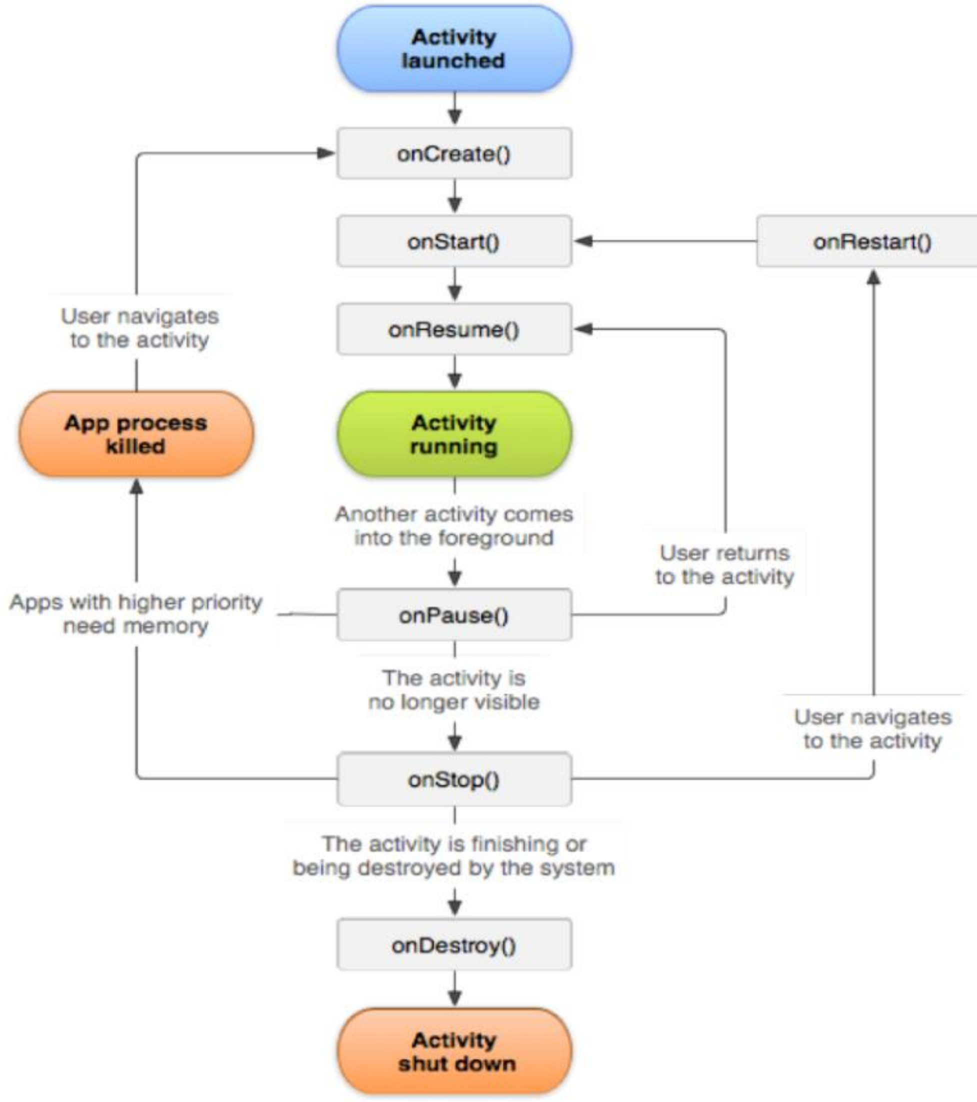
and I un-registered that, in the calls of?

  - onPause(): Too early! If you want to save the memory, it is OK. But you need to re-enable that in the onResume()


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

More about the Activity Lifecycle

- onStop(): Maybe it's a good time! But still, it depends on the "**design**" or "**purpose**" of this App. If this App is running in the background, and is kept doing something, it wouldn't be a good idea to un-register the system resource.
    - i.e. What if I installed a spy App to keep recording my lover's trajectory? ^_^
    - In this case, since this kind of App is definitely **running on the background**, it is not a good idea to un-register the system resource.
- onDestroy(): It is very safe to un-register everything!


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down