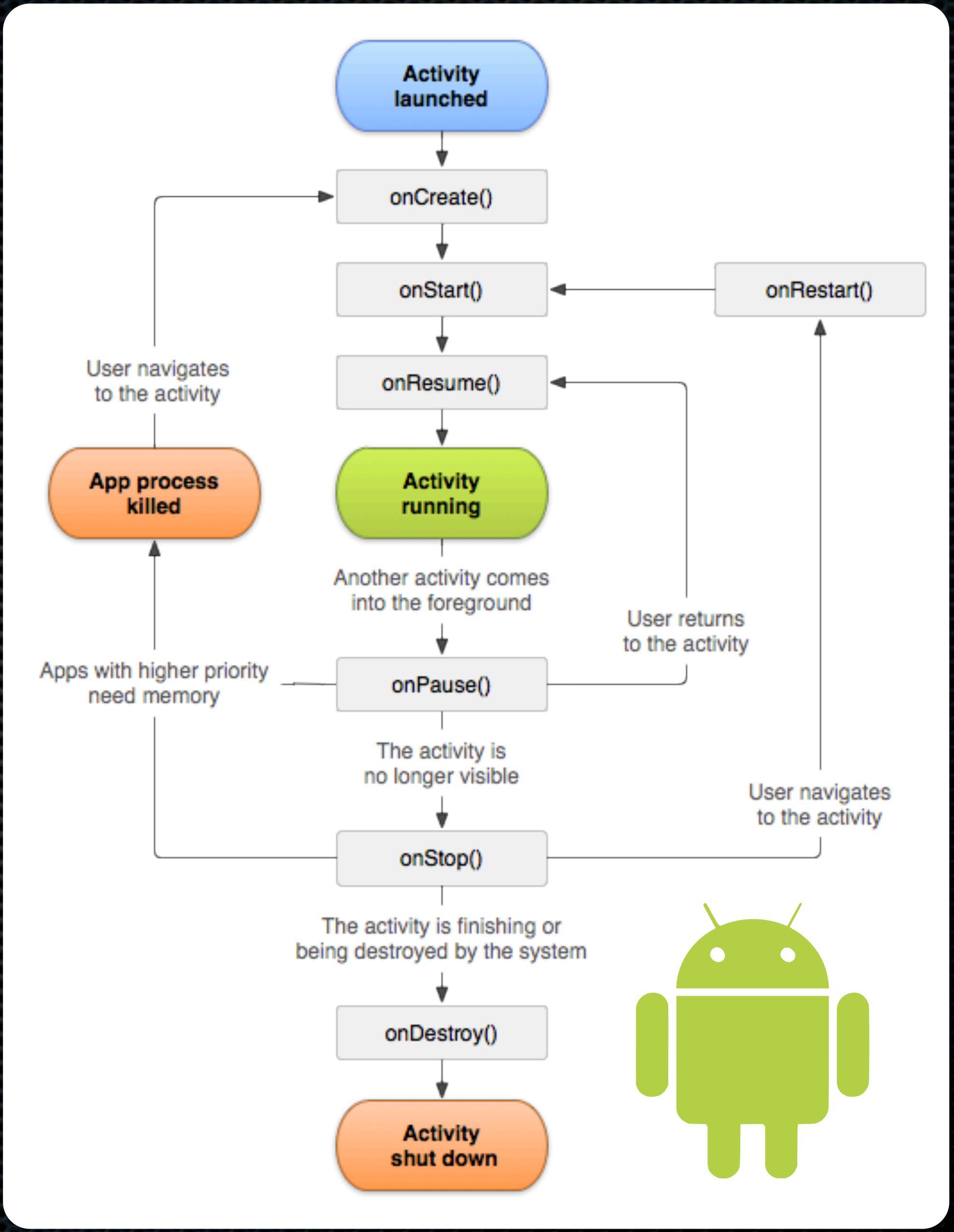
Mobile Application Programing: Android

View Persistence

Activities

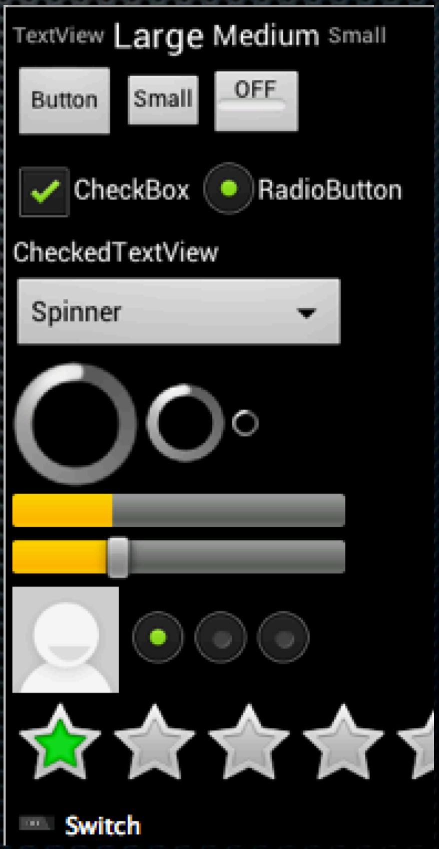
- Apps are composed of activities
- Activities are self-contained tasks made up of one screen-full of information
- Activities start one another and are destroyed commonly
- Apps can use activities belonging to another app



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

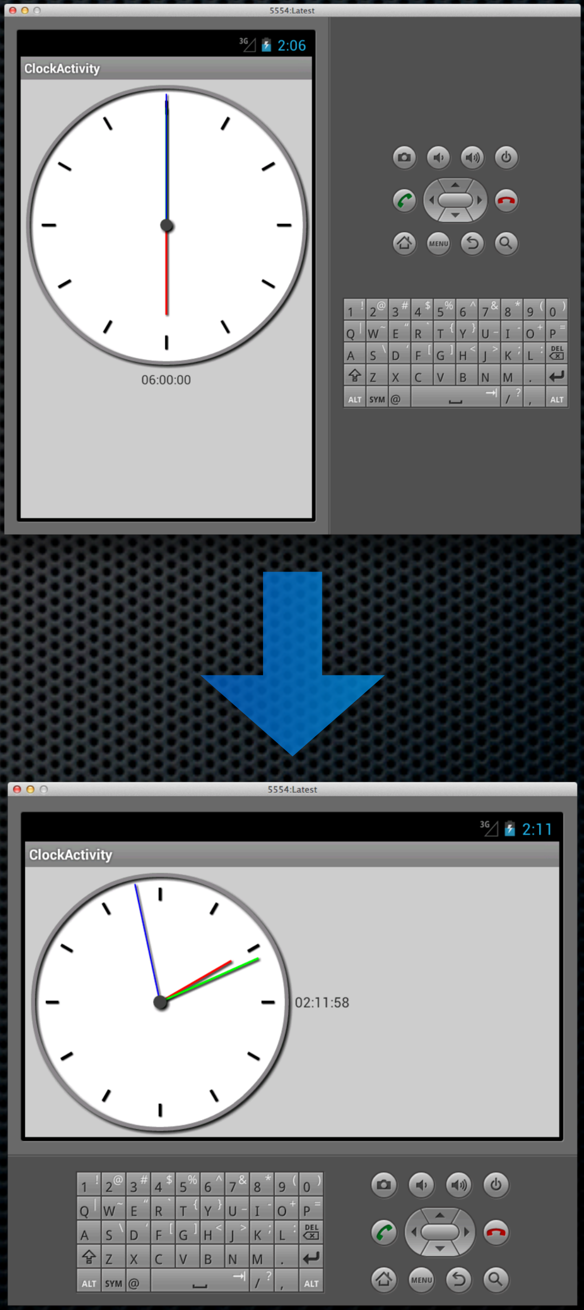
Creating a Custom Control

- Create subclass of View class
- Override:
  - onDraw(Canvas c)
  - onMeasure(int wMeasure, int hMeasure)
- Add listener interface and listener property for the interesting events the control generates and call on... methods when events occur



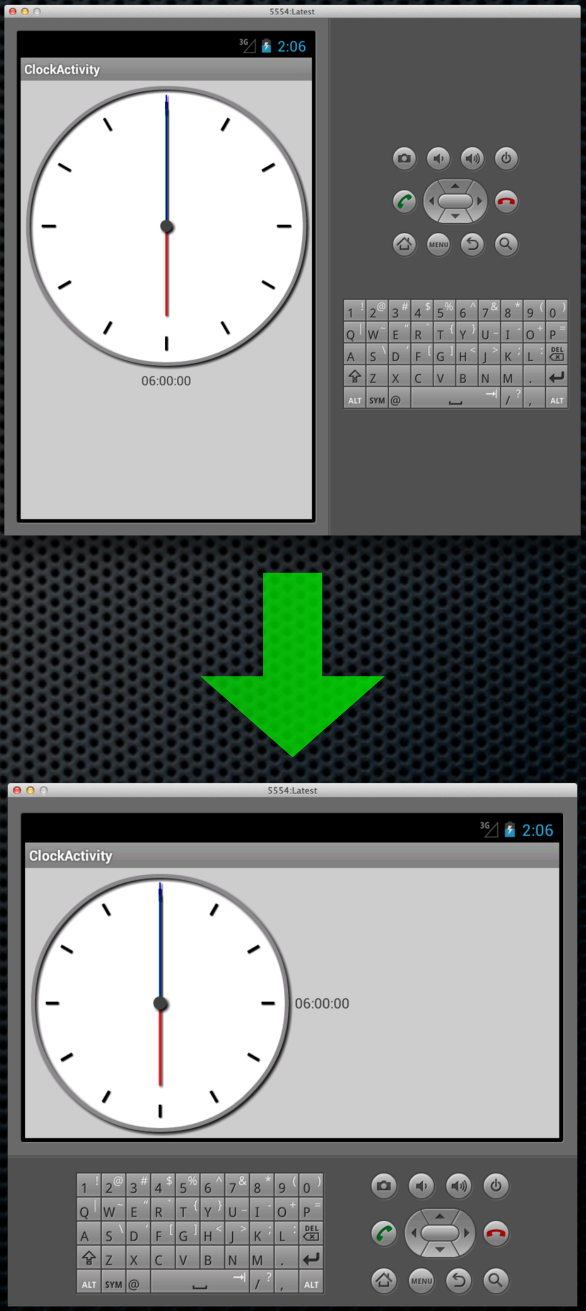
View Persistence

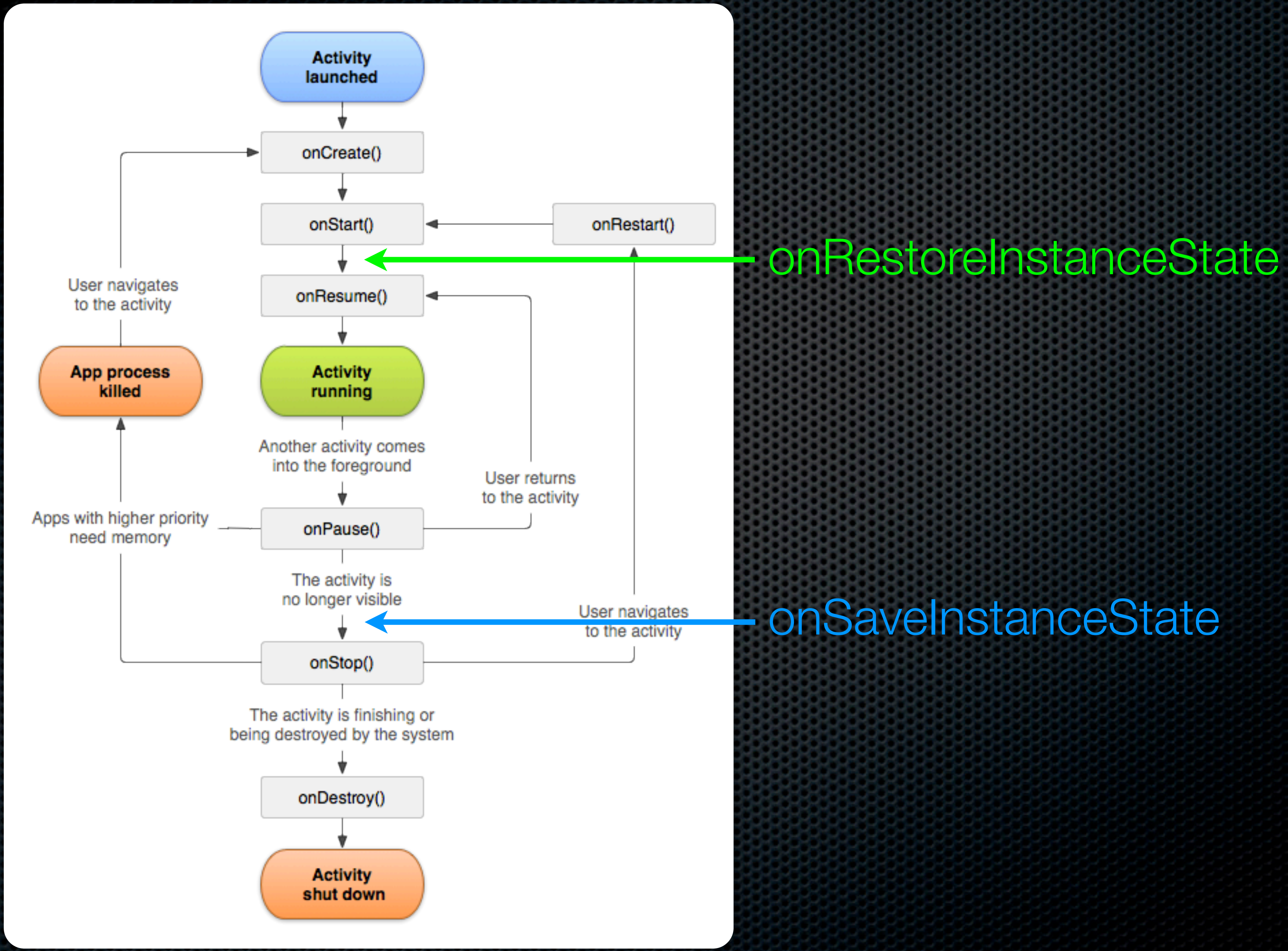
- Problem: Rotations rebuild activity
  - onCreate recreates view hierarchy
  - Data model restores state that has been committed, updating UI
    - (more on this in MVC lecture)
  - What about uncommitted UI state? eg. text in text box that has not yet been validated



View Persistence

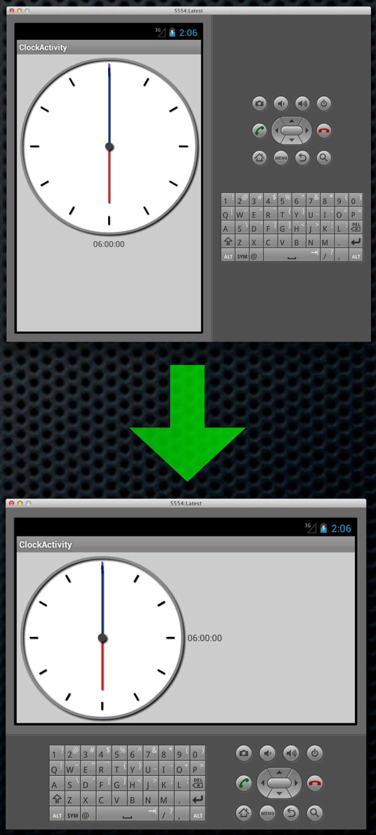
- Problem: Rotations rebuild activity
- Solution: Implement view-level persistence of transient state
  - Set ID property of custom view
    - Won't save state without one!
  - Override onSaveInstanceState
  - Override onRestoreInstanceState




Activity launched
onCreate()
onStart()
onRestart()
onRestoreInstanceState
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
onSaveInstanceState
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

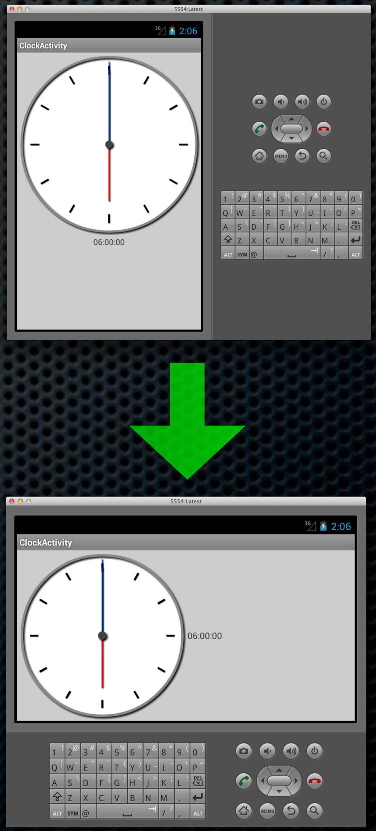
onSaveInstanceState

- Create a Parcelable to store state into
  - Use a custom subclass of BaseSaveState or an instance of Bundle
  - Call super.onSaveInstanceState to retrieve super class state and store in Parcelable
  - Store any non-reconstructable state in Parcelable



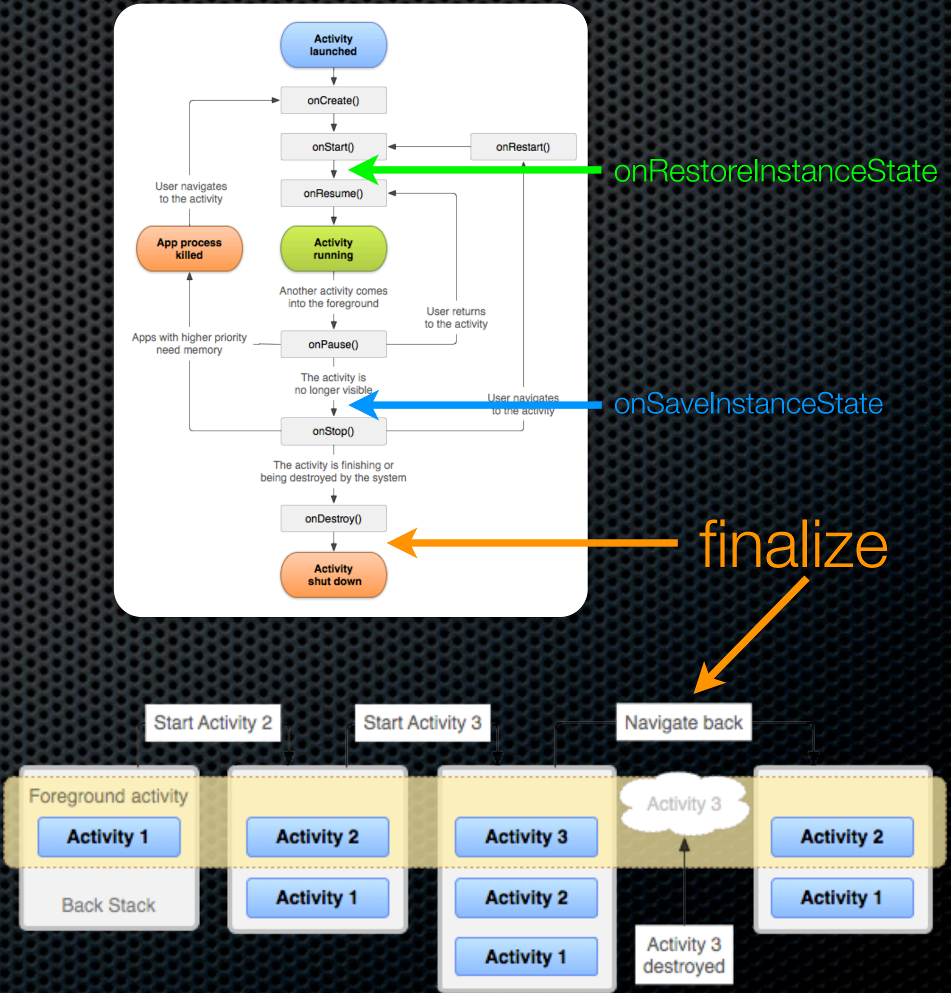
onRestoreInstanceState

- Cast Parcelable instance to whatever class was used in onSaveInstanceState
  - Use instanceof to ensure class is correct
- Retrieve super class state and call super.onRestoreInstanceState to restore it
- Retrieve any non-reconstructable state and restore it to the class instance



onSaveInstanceState vs. onPause

- onSaveInstanceState is meant to save transient activity state
- Instance state is deleted when activity is finalized
- Save application state in onPause for all non-transient state
- See MVC for details


Activity launched
onCreate()
onStart()
onRestart()
onResume()
User navigates to the activity
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
onRestoreInstanceState
onSaveInstanceState
finalize
Start Activity 2
Start Activity 3
Navigate back
Foreground activity
Activity 1
Back Stack
Activity 2
Activity 1
Activity 3
Activity 2
Activity 1
Activity 3
Activity 3 destroyed
Activity 2
Activity 1