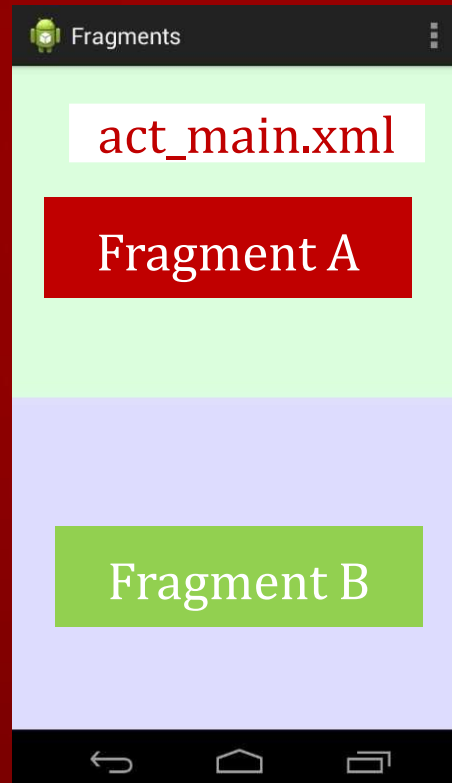


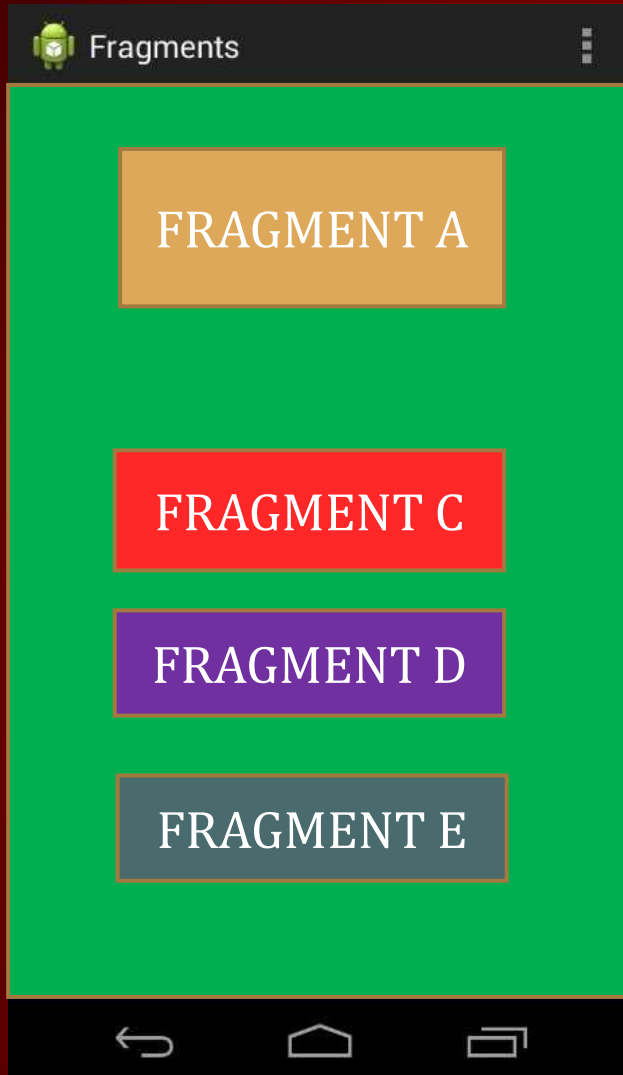
ANDROID FRAGMENTS

ANDROID FRAGMENTS

- ❑ Fragments are found inside activity
- ❑ One Activity can contain multiple Fragments



Fragment A – Add
Fragment B - Add



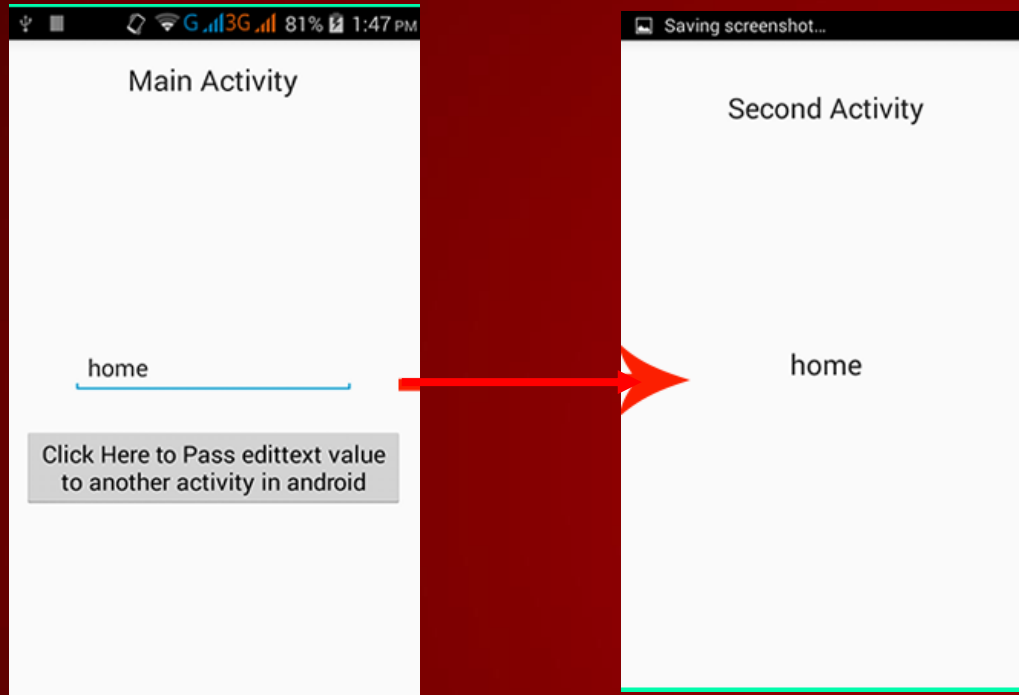
Fragment A – Add
Remove B - Remove
Fragment C – Add
Fragment D - Add
Fragment E – Add

- Acts as a small chunk of UI
- All Fragments are integral part of the `main_activity.java` (i.e its child elements)

FRAGMENTS

- ❑ Small Chunk of UI
- ❑ Has its own life cycle
- ❑ Can process its own events
- ❑ Can be added/removed while parent activity is running
- ❑ Introduced in API 11 – Honeycomb with the introduction of tablets
- ❑ Backward compatible upto API 7
- ❑ Act as a mini Activity

Activity

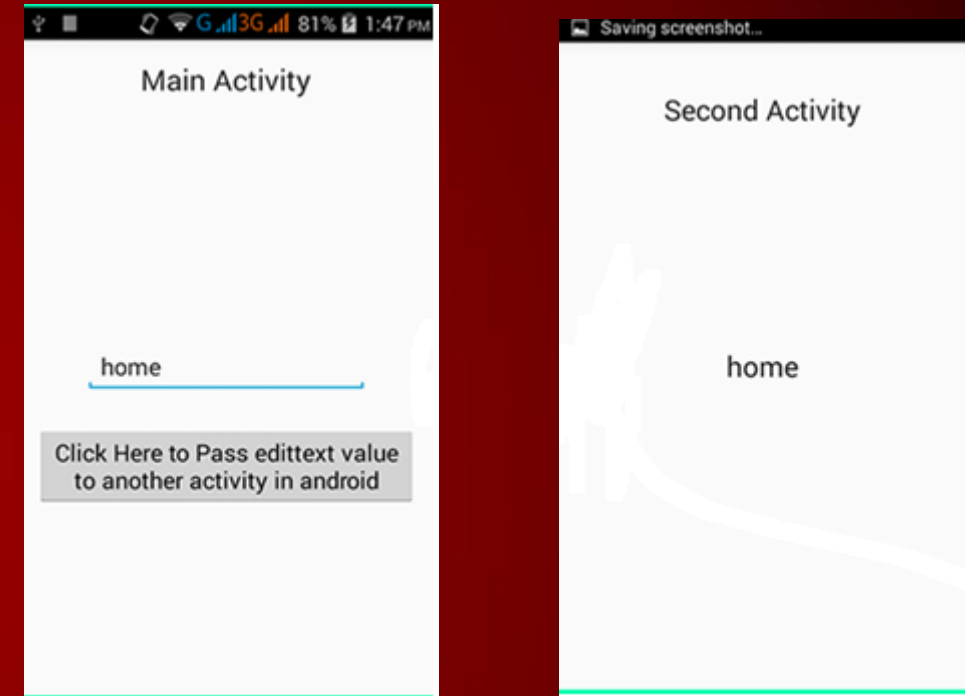


Main Activity

Second Activity

- ❑ Need two activities to display two kinds of layouts

Activity + Fragment



Main Activity

- ❑ Need only one Main activity to display two kinds of layouts

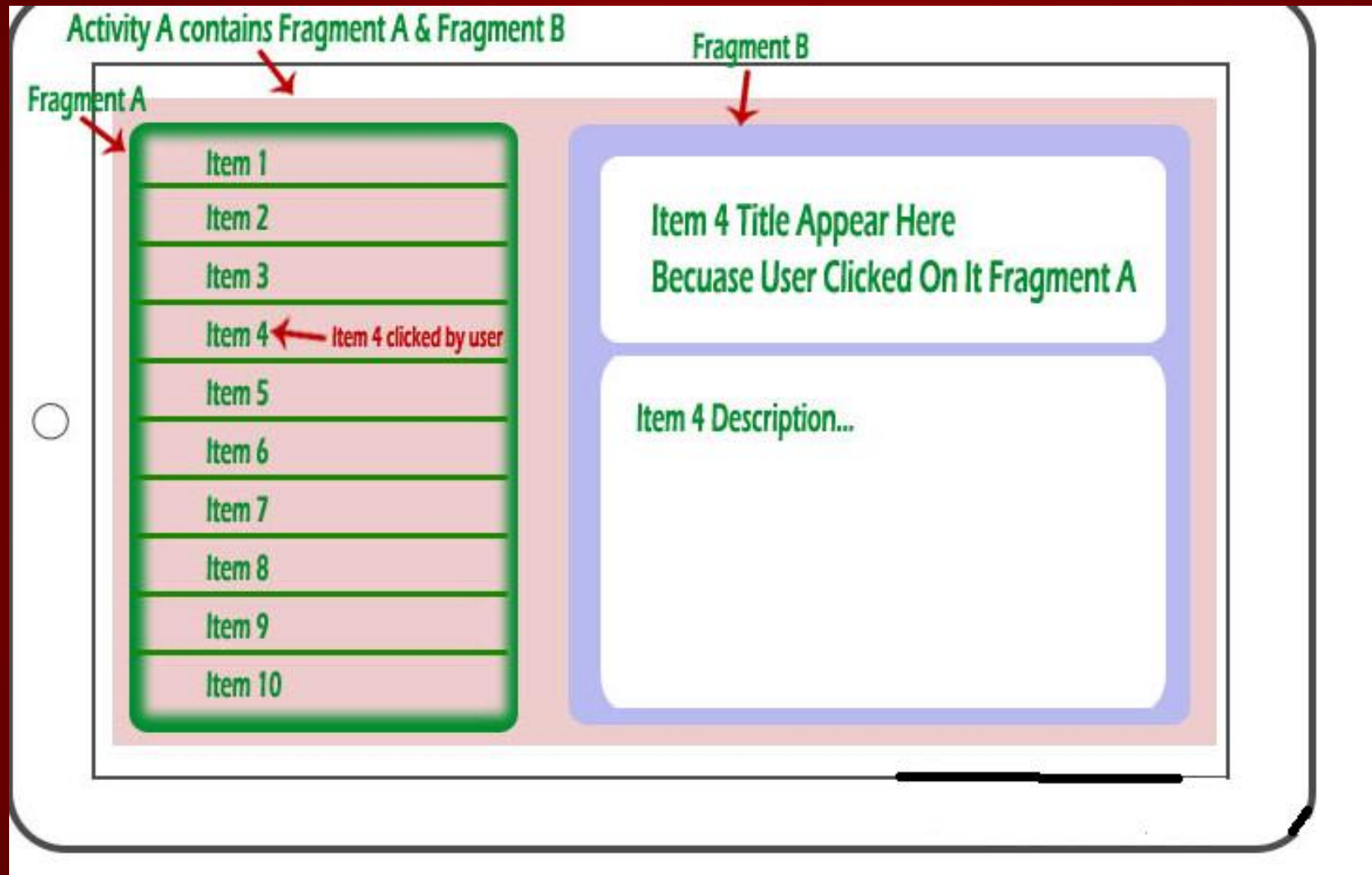
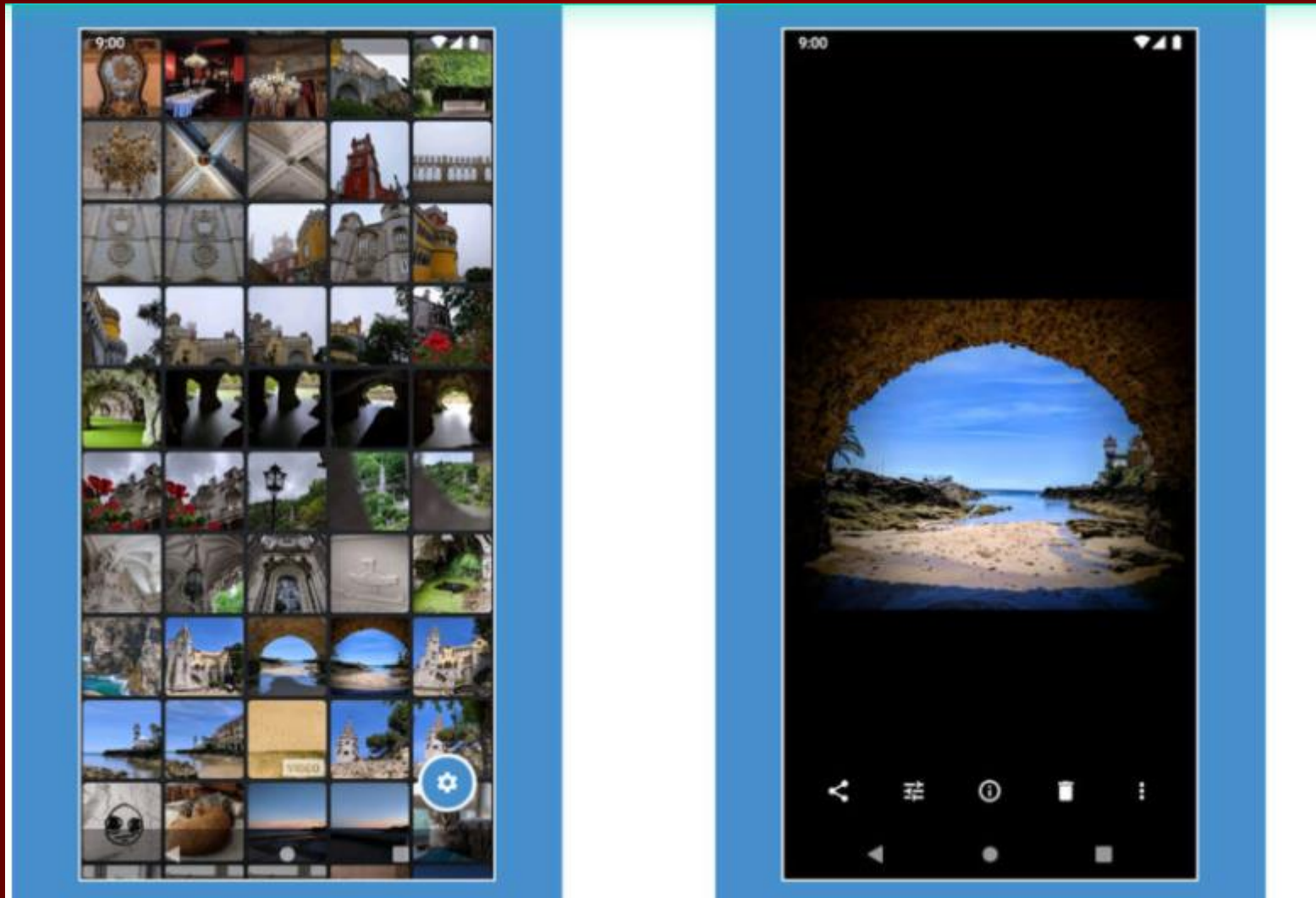


Image Gallery App



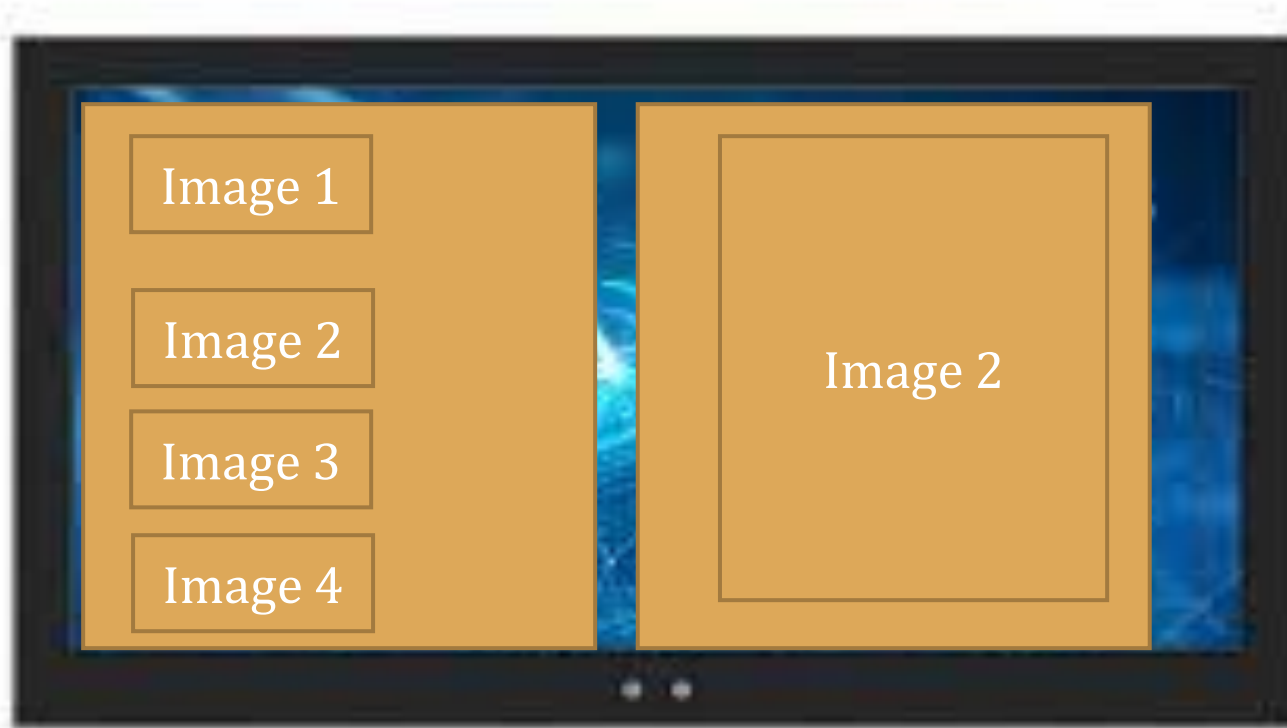
Main Activity

Second Activity

IMAGE GALLERY APP - TABLET

Major Advantage of Fragments
introduced with tablets

Enabling developers to effectively
use space in the tablets



Fragment A-List

Fragment B- Detailed View

Main Activity

Why Fragments?

- ❑ Combine several fragments in single activity
- ❑ Reuse same fragment across several activities
- ❑ Make better use of larger screen space in Tablets
- ❑ Support different layouts on portrait and landscape modes

Adding Fragments to Activity

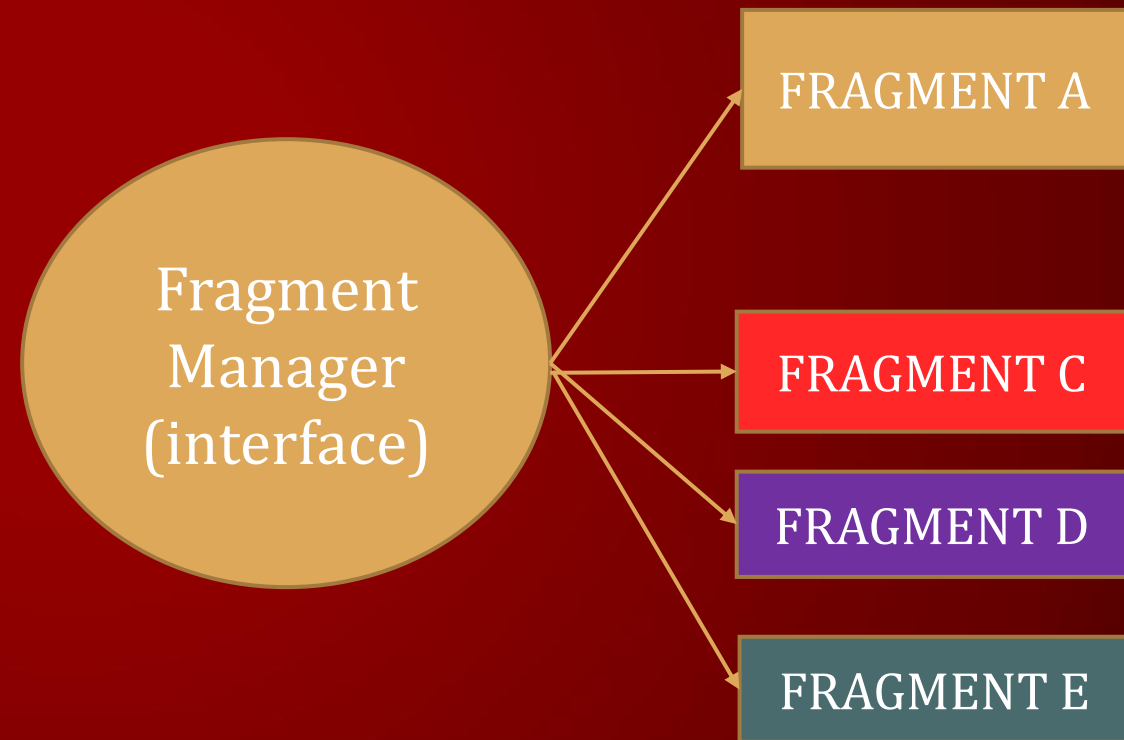
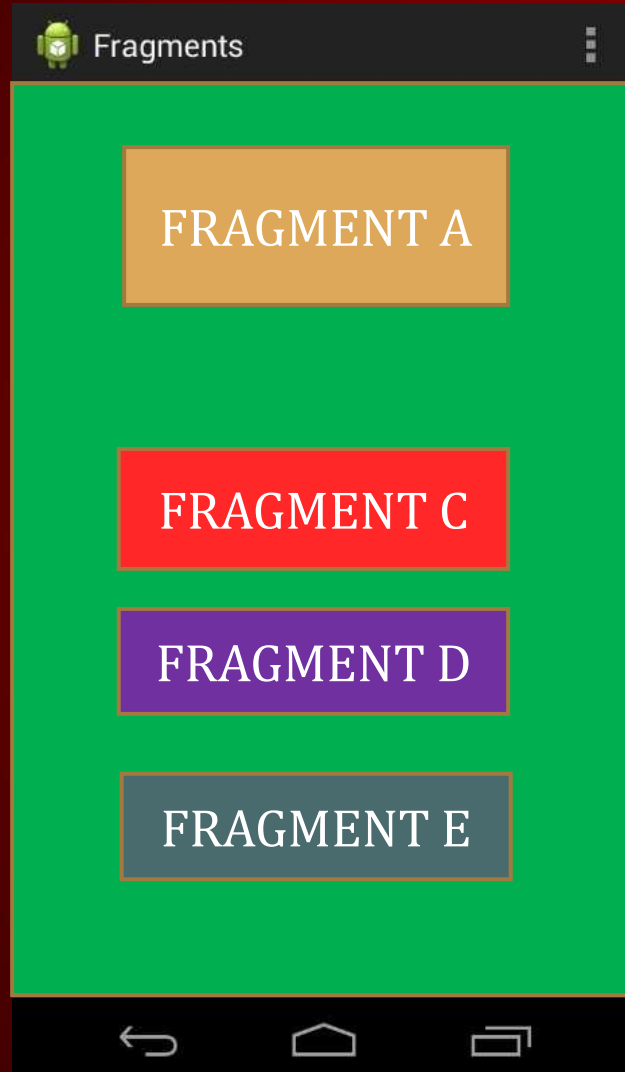
- ❑ Two Approaches
 - ❑ XML (less preferred)
 - ❑ Java / Programmatically (More Preferred)

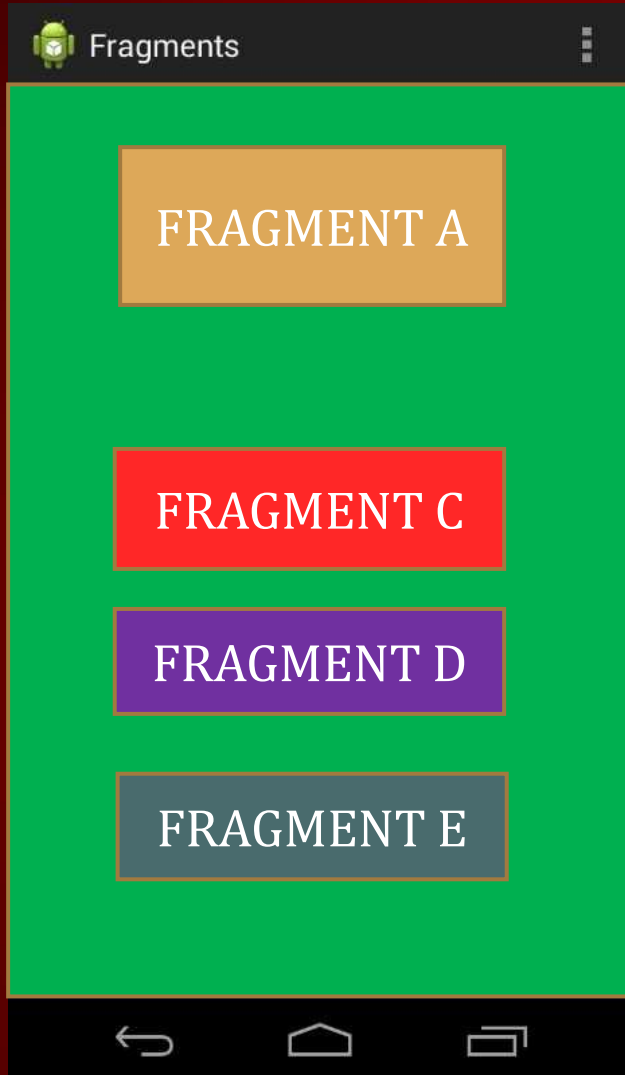
1. Steps to create Fragment by XML

- ❑ Create a subclass of Fragment
 - ❑ HelloFragment.java
- ❑ Create a layout for fragment
 - ❑ fragment hello.xml
- ❑ Link layout with Fagment subclass
 - ❑ override onCreateView()
- ❑ Place the Fragment inside an Activity
 - ❑ <fragment> inside activity_main.xml

Fragment Manager

- Interface to interact with fragment objects inside the Activity
- Handles all Fragments inside one activity





Fragment Transaction

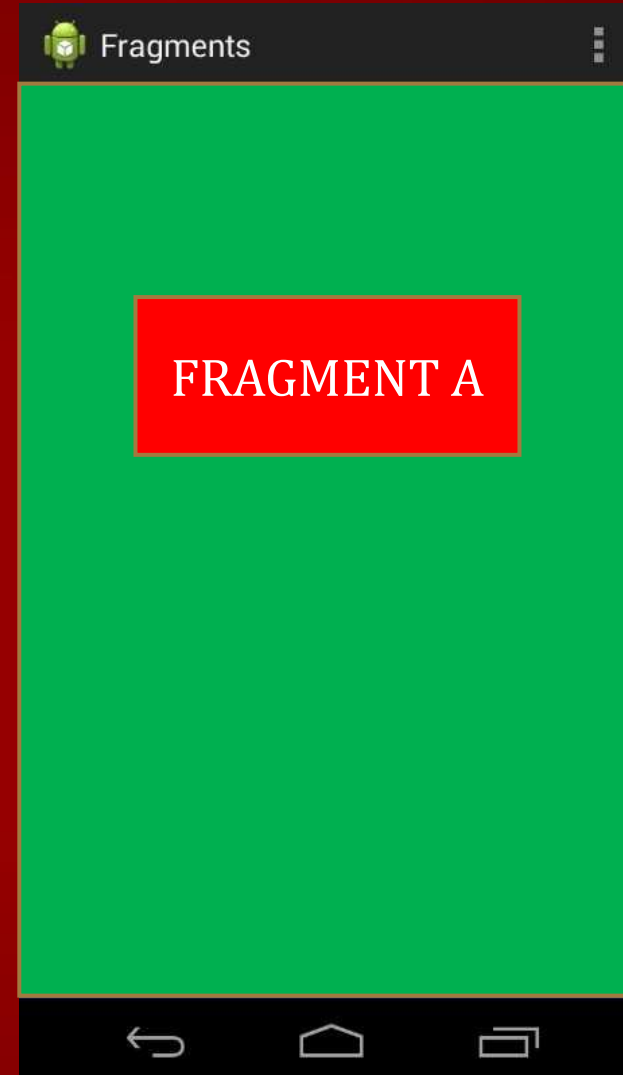
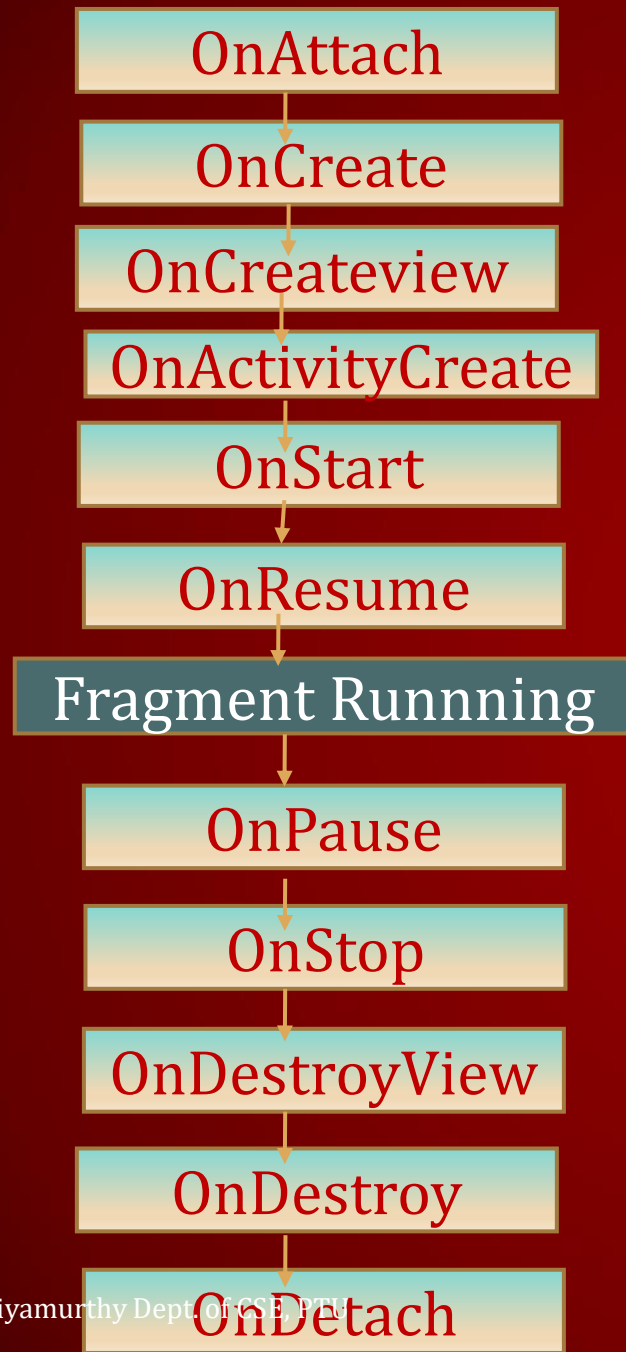
- Fragment Transaction takes place
Fragment A – Add
Remove B - Remove
Fragment C – Add
Fragment D - Add
Fragment E – Add
- Addition and removal of fragments are known as fragment Transactions
- Takes place in Main Activity with help of Fragment Transaction class.
- Fragment Transaction- API for performing a set of Fragment operations such as add, remove, replace.

2.Steps to create Fragments by Java/Programmatically

- ❑ Create a subclass of Fragment
 - ❑ HelloFragment.java
- ❑ Create a layout for fragment
 - ❑ fragment hello.xml
- ❑ Link layout with Fagment subclass
 - ❑ override onCreateView()
- ❑ Place the Fragment inside an Activity
 - ❑ Initialize Fragment Manager
 - ❑ Initialize Fragment Transaction
 - ❑ Start add/remove/replace operation
 - ❑ Commit the transaction

FRAGMENTS LIFECYCLE

FRAGMENTS LIFECYCLE



FRAGMENTS TRANSACTIONS

- ❑ Transaction Types
 - ❑ Add
 - ❑ Remove
 - ❑ Replace
 - ❑ Attach
 - ❑ Detach
 - ❑ Show
 - ❑ Hide

FRAGMENTS TRANSACTIONS

□ ADD

- Create a new Fragment
- Get the id of the container layout
- Use add() to add the fragment
 - The Fragment is added to the container layout

□ Remove

- Get the reference to the fragment to be removed
- Use remove() to remove the fragment
 - The fragment is destroyed
 - Detached from the host activity

FRAGMENTS TRANSACTIONS

- ❑ Replace
 - Get the id of the container layout
 - Get a reference to the new Fragment
 - Use `replace()` to replace the content fragment of the container layout
 - The new Fragment fills the container
 - The old fragment if any is destroyed

FRAGMENTS TRANSACTIONS

❑ Detach

- Get the reference to the fragment
- Use detach() to detach the fragment from Activity UI
 - The fragment's view is destroyed
 - The Fragment and its UI is detached from the host Activity
(object is present and it is not visible to the user)

❑ Attach

- Get the reference to the Detached fragment
- Use attach() to reattach the fragment
 - The fragment's view is recreated
 - The Fragment is re-attached to the activity and is visible

FRAGMENTS TRANSACTIONS

❑ Hide

- Get the reference to the fragment to hide
- Use `hide()` to hide the fragment from Activity UI
 - The fragment's view becomes invisible
 - The Fragment continue to be in Resumed state

❑ Show

- Get the reference to the Hidden fragment
- Use `show()` to make it visible
 - The view of the fragment is now visible
 - Remains in Resumed state as it was earlier

- Hide and show does not affect the states of the transactions