Mobile Application Development
Fragments

Waterford Institute of Technology

October 7, 2016

John Fitzgerald

Activities

What is an Activity?

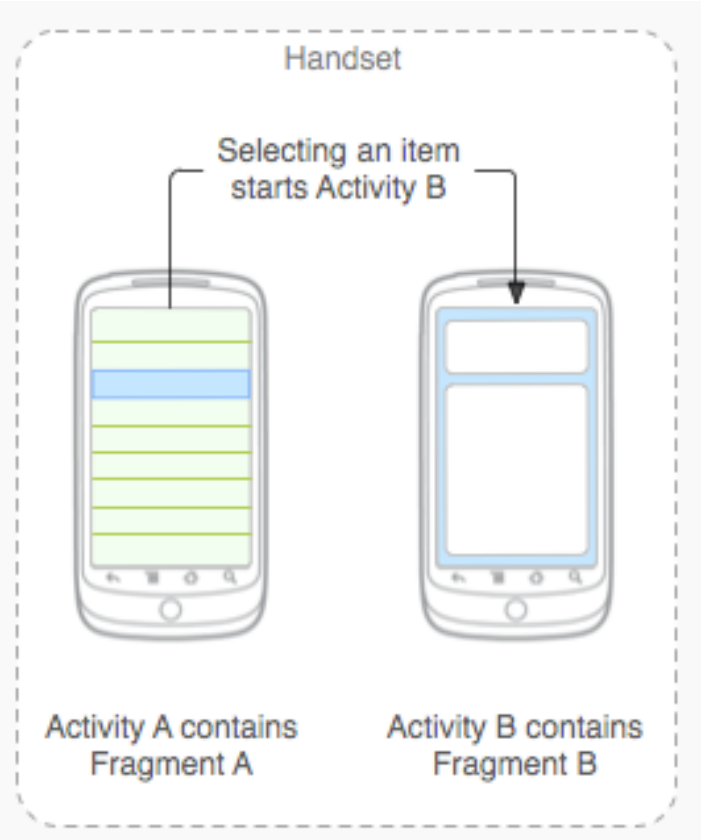
- Most visible element of an app
- Typically single screen individually displayed on device
- Application usually has many activities
- Prior to Honeycomb, the UI tightly bound to activity
- Subsequently modularization possible using Fragment class

```
public class ResidenceCameraActivity extends Activity
{
  ...
  @Override
  protected void onCreate(Bundle savedInstanceState)
  {
    super.onCreate(savedInstanceState);
    ...
  }
}
```

Fragments

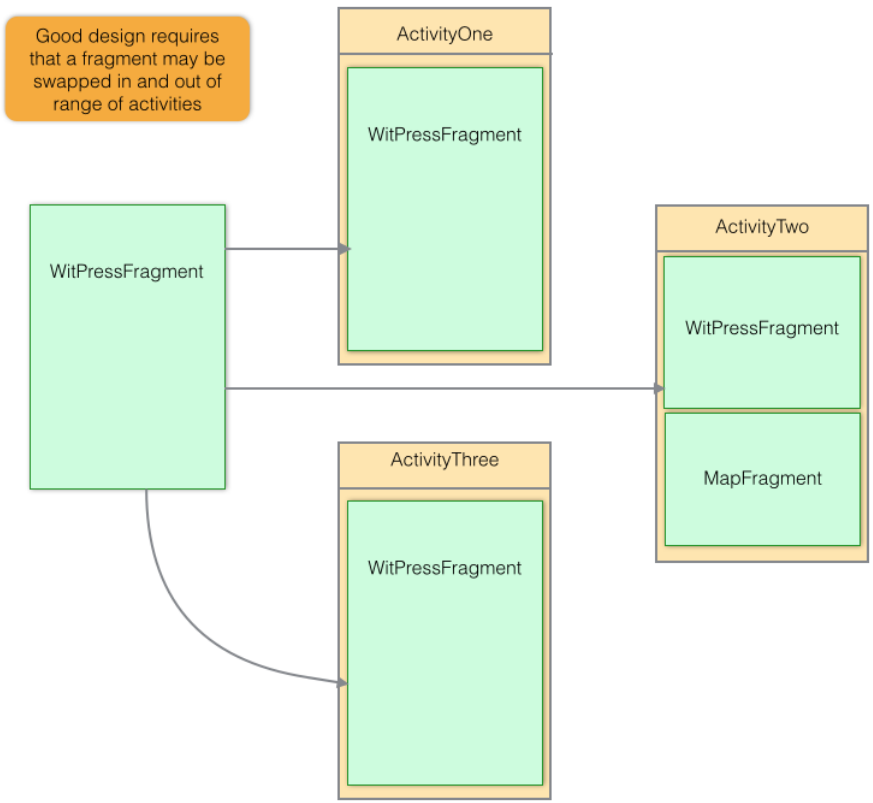
What is a Fragment?

- Controller object to perform tasks for activity
- Typically represents behaviour of portion of UI in activity
- Can be used without UI



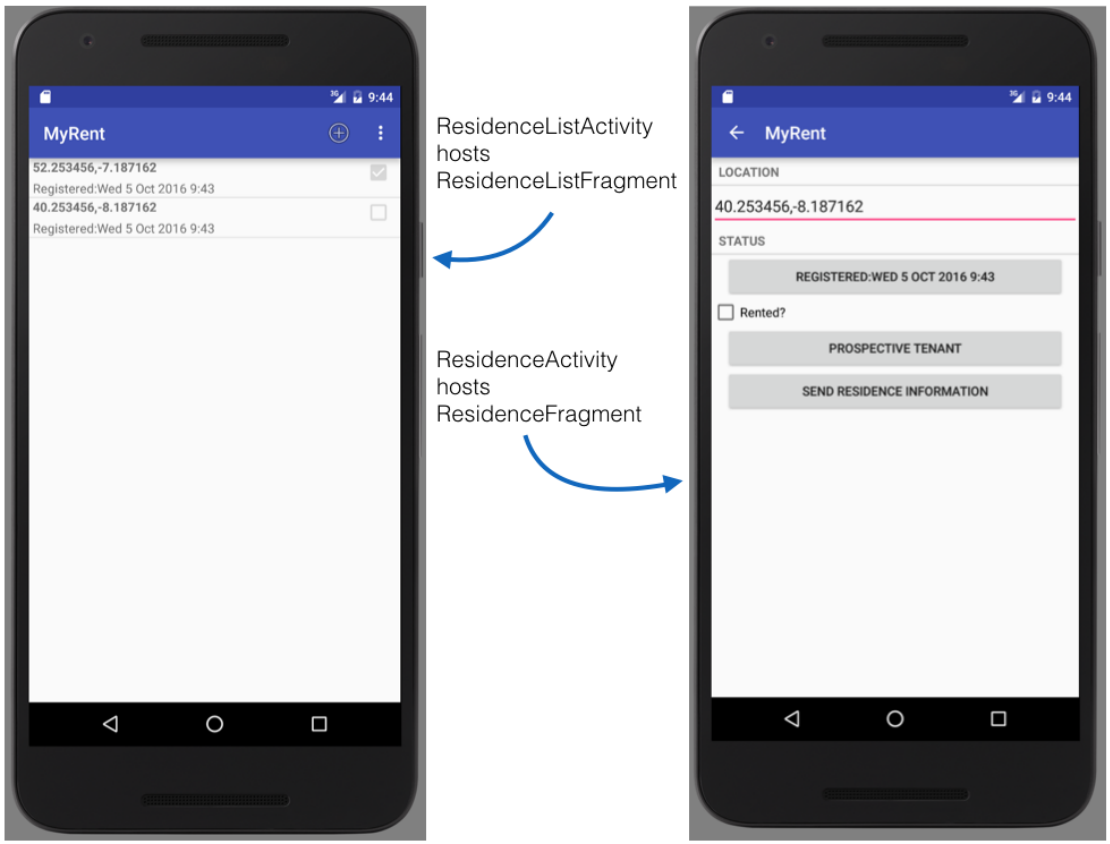
Fragments

Facilitate modularization



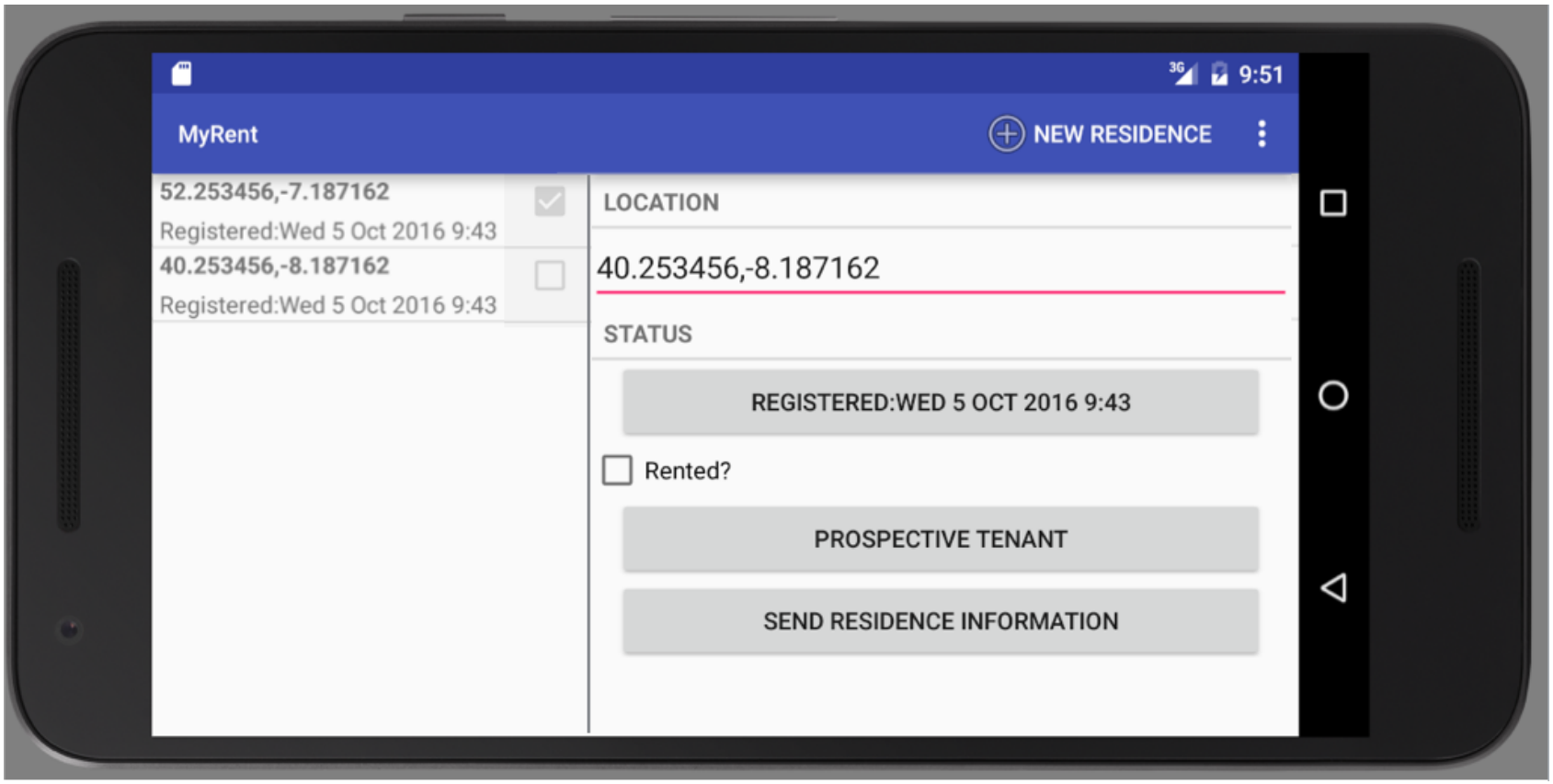
Fragments

Swap in and out of activities



Fragments

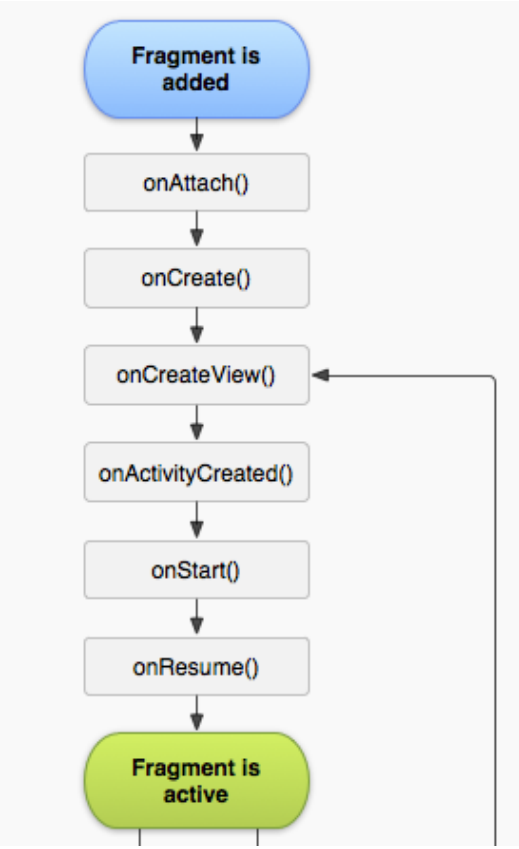
Swap in and out of activities



Fragments

Lifecycle

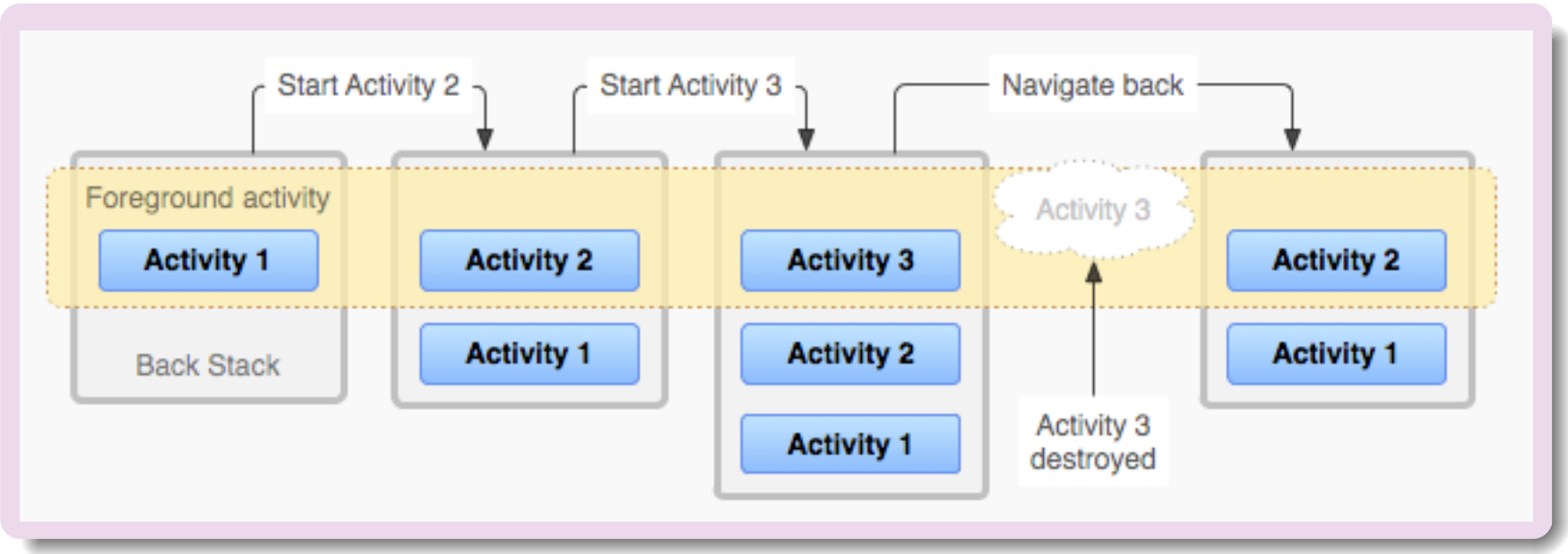
- Fragments have own lifecycle
- This directly affected by host activity
- Example when activity paused, fragment paused
- Fragment has some extra lifecycle functionality
  - Example: *onCreateView*



Fragments

Tasks & back stack

- **Task** : collection of activities
- Stored or arranged in **back stack**
- Can navigate these using back button
- Task list managed by Android O.S.



Fragments

Manipulating fragments

- FragmentManager responsible managing fragments
- Fragment transactions are used to add, remove, attach, detach, or replace fragments
- Can add remove fragment while activity running
- Can manipulate each fragment independently

Fragments

Baseline app

Begin with baseline app - no fragments

```
public class MainActivity extends AppCompatActivity
{

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
  }
}
```

Fragments

Convert to Activity-Fragment

- Introduce a fragment container (xml).
- Create a fragment class (java).
- Create a fragment layout (xml).
- Attach fragment to activity (in activity code).

Fragments

Introduce a fragment container

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/
    android"
        android:id="@+id/fragmentContainer"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>
```

Fragments

Create fragment class

```
public class Fragment_1 extends Fragment
{

  @Override
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

  }

  @Override
  public View onCreateView(LayoutInflater inflater, ViewGroup parent,
      Bundle savedInstanceState) {
    View v = inflater.inflate(R.layout.fragment_1, parent, false);
    return v;
  }
}
```

Fragments

Create a fragment layout

```
<RelativeLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   ...
   tools:context="ie.wit.twopane.MainActivity">
  <TextView
     android:layout_width="wrap_content"
     android:layout_height="wrap_content"
     android:textAppearance="?android:attr/textAppearanceLarge"
     android:text="Fragment 1"
     android:id="@+id/textViewFrag1"
     android:layout_marginTop="46dp"
     android:layout_alignParentStart="true"/>
</RelativeLayout>
```

Fragments

Inflating fragment layout

Fragment_1.java

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup parent, Bundle savedInstanceState) {
  View v = inflater.inflate(R.layout.fragment_1, parent, false);
  return v;
}
```

fragment_1.xml

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    ...
    tools:context="ie.wit.twopane.MainActivity">
  <TextView
      ...
      android:text="Fragment 1"
      android:id="@+id/textViewFrag1"
    ...
</RelativeLayout>
```

Fragments

Convert to Activity-Fragment

Attach fragment to activity

```
public void onCreate(Bundle savedInstanceState)
{
  super.onCreate(savedInstanceState);
  setContentView(R.layout.activity_fragment_container);

  FragmentManager manager = getSupportFragmentManager();
  Fragment fragment = manager.findFragmentById(R.id.
    fragmentContainer);
  if (fragment == null)
  {
    fragment = new Fragment_1();
    manager.beginTransaction().add(R.id.fragmentContainer, fragment).
     commit();
  }
}
```

Fragments

Convert to Activity-Fragment

MainActivity.java

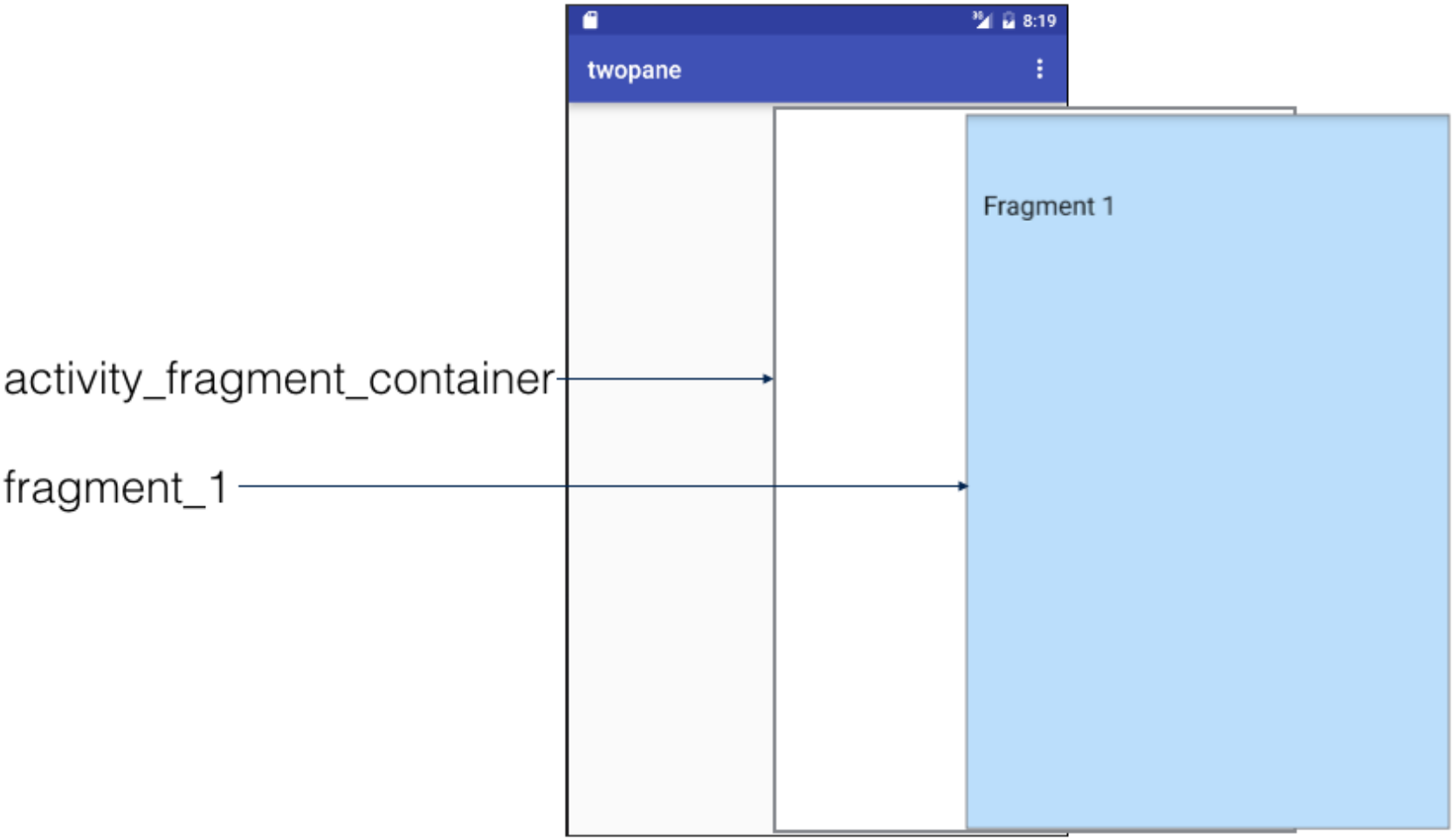
```
public class MainActivity extends AppCompatActivity  {
  public void onCreate(Bundle savedInstanceState)  {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_fragment_container);
    FragmentManager manager = getSupportFragmentManager();
    Fragment fragment = manager.findFragmentById(R.id.fragmentContainer);
    if (fragment == null)  {
      fragment = new Fragment_1();
      manager.beginTransaction()
             .add(R.id.fragmentContainer, fragment)
             .commit();
    }
  }
```

activity_fragment_container.xml

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/
android"
             android:id="@+id/fragmentContainer"
             android:layout_width="match_parent"
             android:layout_height="match_parent"/>
```

Fragments

FrameLayout potential container for different fragments



Fragments

Replace fragment with another

Activity now hosts different fragment.

```
private void swapFragments(Fragment replacementFragment) {
  FragmentManager manager = getSupportFragmentManager();
  FragmentTransaction transaction = manager.beginTransaction();

  // Replace whatever is in the fragment_container view with this fragment,
  // and add the transaction to the back stack
  transaction.replace(R.id.fragmentContainer, replacementFragment);
  transaction.addToBackStack(null);

  // Commit the transaction
  transaction.commit();
}
```

Fragments

Replace fragment with another

Activity now hosts different fragment.
Uses fluent programming (chaining).

```
private void swapFragments(Fragment replacementFragment) {;

  // Replace whatever is in the fragment_container view with this fragment,
  // add the transaction to the back stack and commit
  getSupportFragmentManager().beginTransaction()
      .replace(R.id.fragmentContainer, replacementFragment)
      .addToBackStack(null)
      .commit();
}
```

Fragments

Replace fragment with another

addToBackStack

- Add this transaction to the back stack.
- Transaction will be remembered after commit.
- Transaction will reverse its operation when popped off stack.
- Parameter string is optional transaction name.

```
FragmentTransaction addToBackStack (String name)
```

Referenced Material

1. Android Documentation: Fragments

`http://developer.android.com/guide/components/fragments.html`

[Accessed 2016-10-07]

2. Android Documentation: Tasks and Back Stack

`https://developer.android.com/guide/components/tasks-and-back-stack.html`

[Accessed 2016-10-07]