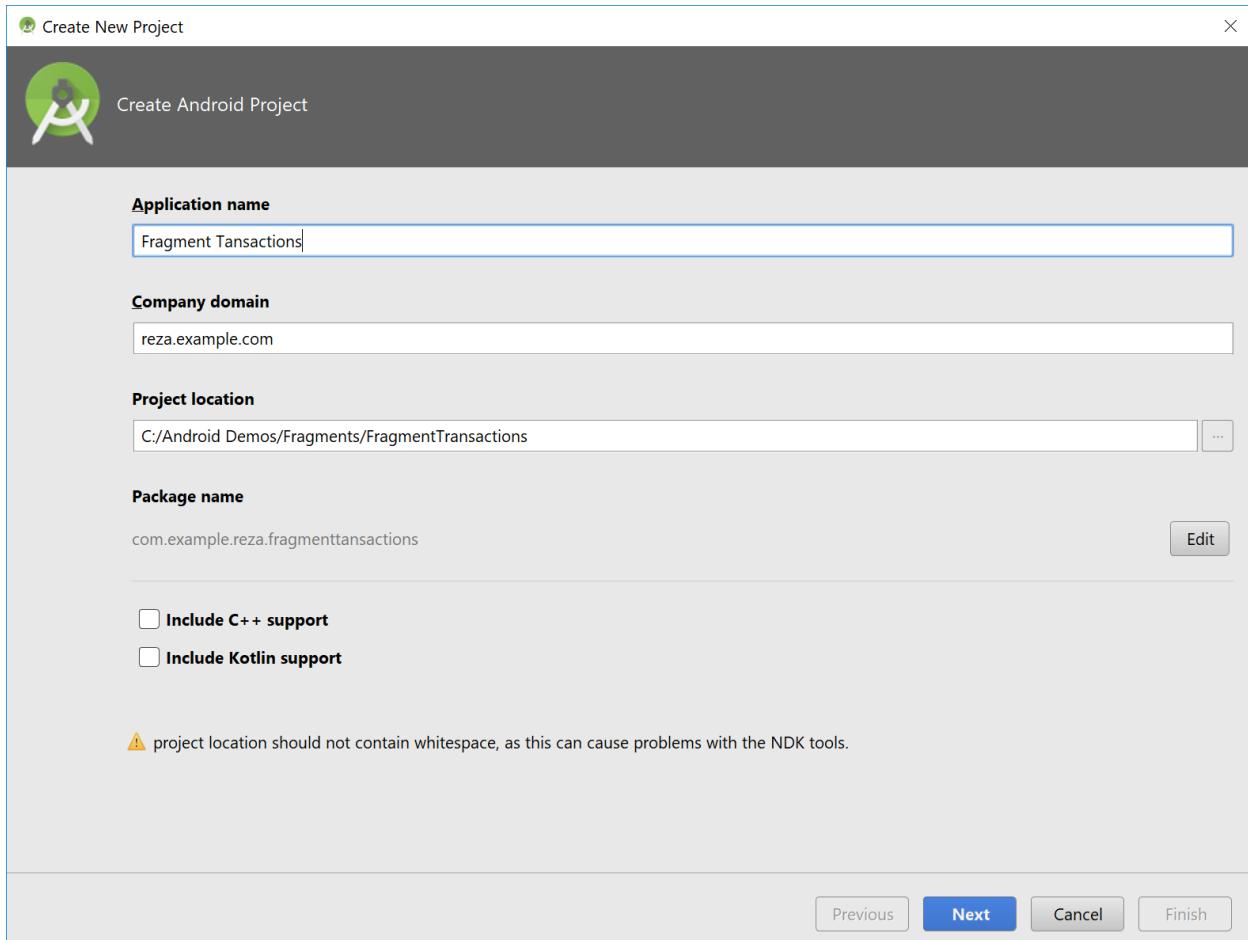


Fragment Transactions

Create a new project as shown:



Create New Project X

Target Android Devices

Select the form factors and minimum SDK

Some devices require additional SDKs. Low API levels target more devices, but offer fewer API features.

Phone and Tablet

API 15: Android 4.0.3 (IceCreamSandwich) ▼

By targeting **API 15 and later**, your app will run on approximately **100%** of devices. [Help me choose](#)

Include Android Instant App support

Wear

API 21: Android 5.0 (Lollipop) ▼

TV

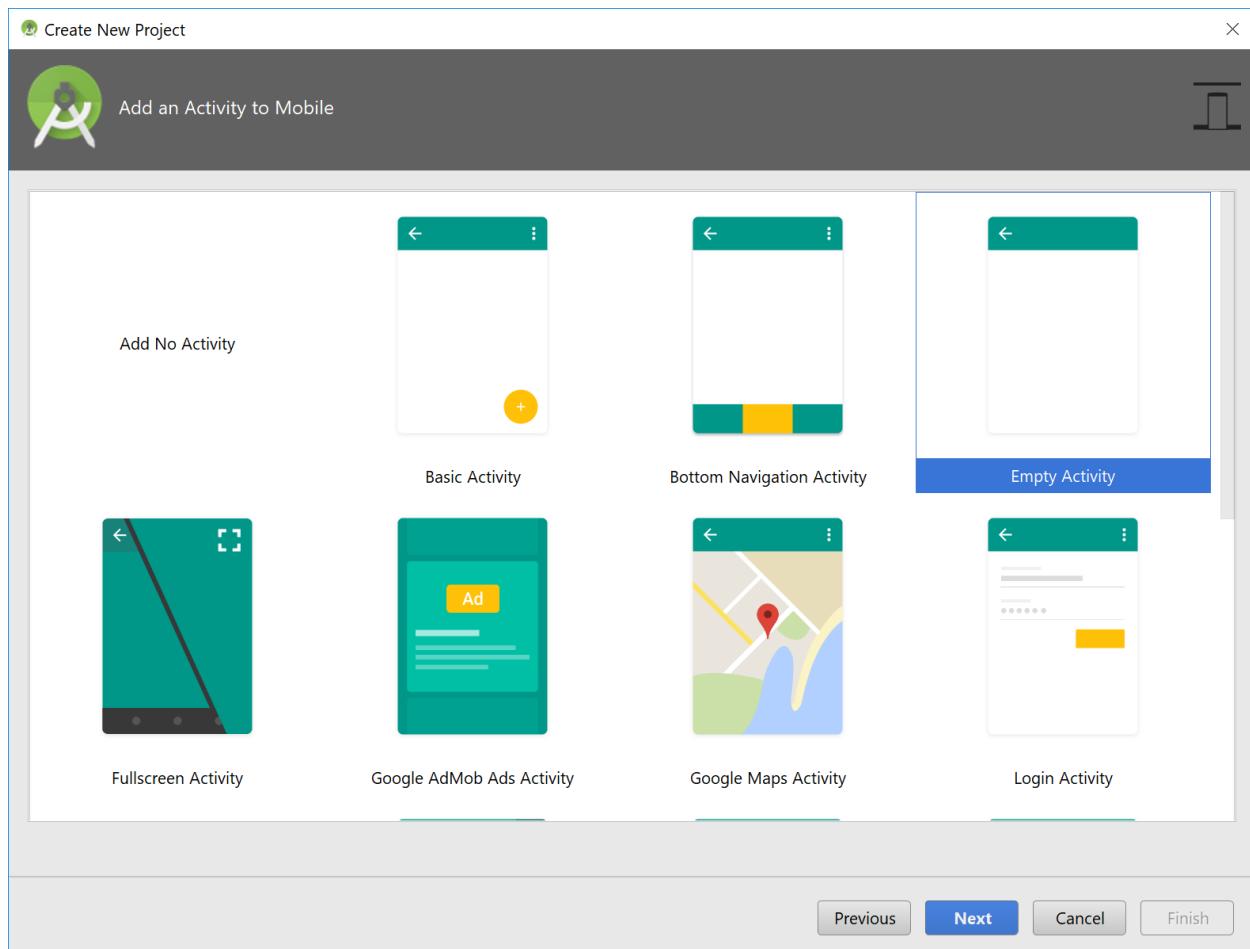
API 21: Android 5.0 (Lollipop) ▼

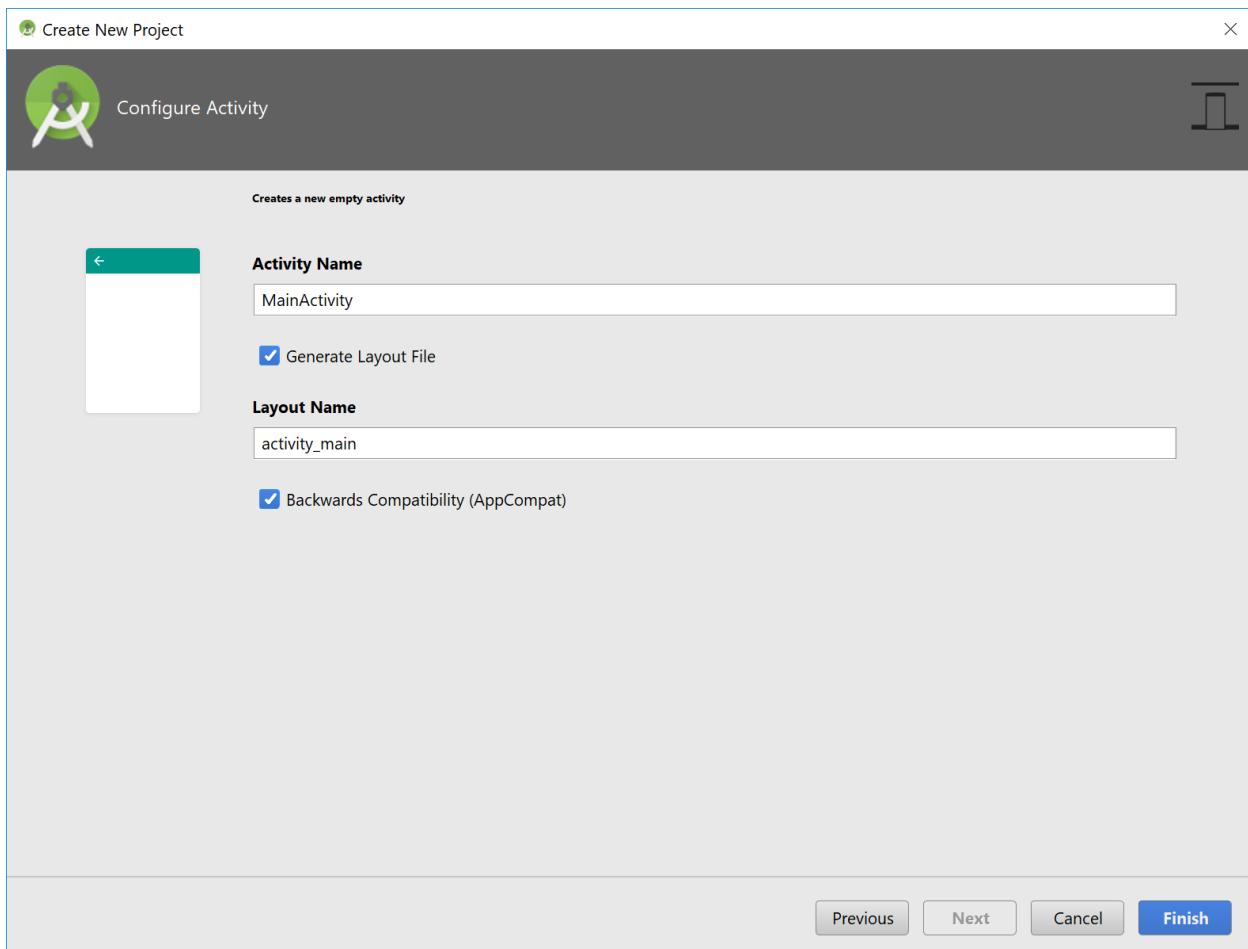
Android Auto

Android Things

API 24: Android 7.0 (Nougat) ▼

Previous Next Cancel Finish





The screenshot shows the Android Studio interface with the project 'FragmentTransactions' open. The left sidebar displays the project structure under the 'app' module, including 'manifests', 'java' (with 'MainActivity' selected), 'res', and 'values'. The main editor window shows the 'activity_main.xml' file, which contains the XML code for the main activity's layout. The bottom status bar indicates a successful Gradle build.

```
package com.example.reza.fragmenttransactions;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

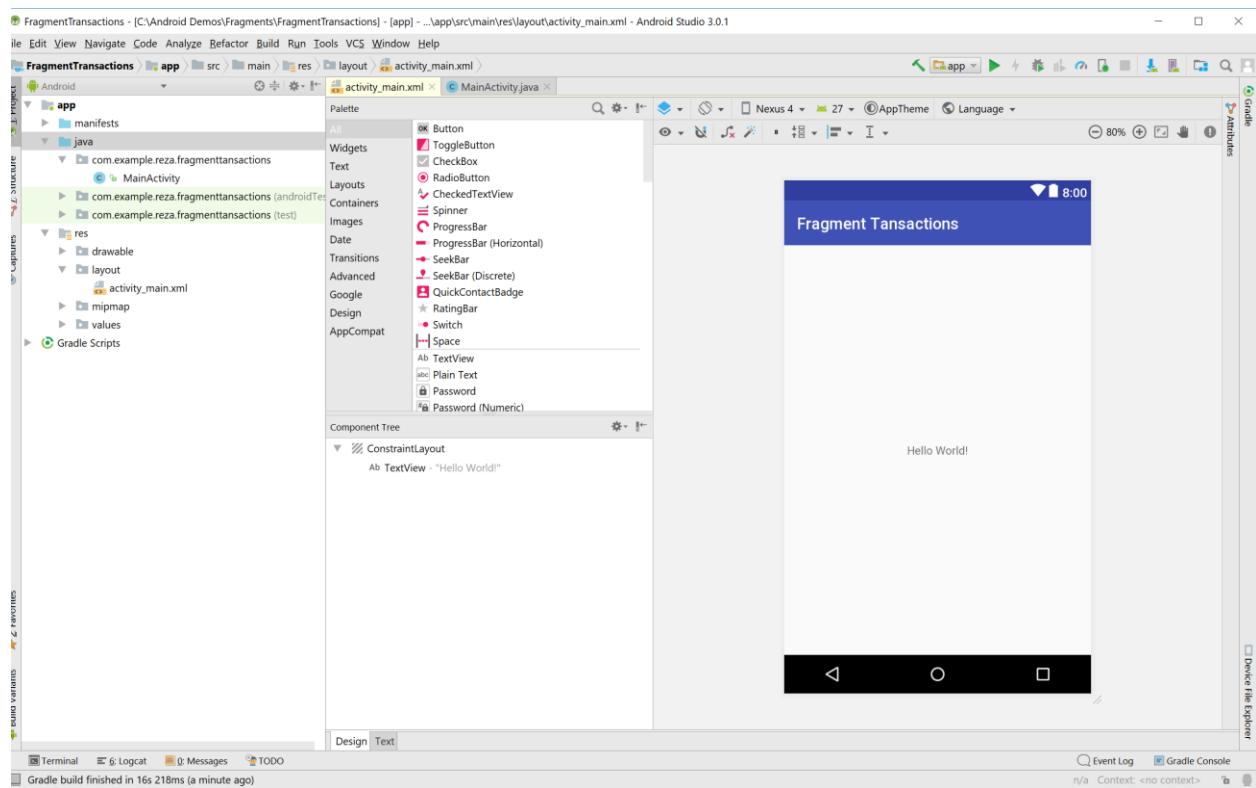
public class MainActivity extends AppCompatActivity {

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

The screenshot shows the Android Studio interface with the project 'FragmentTransactions' open. The left sidebar displays the project structure under the 'app' module, including 'manifests', 'java' (with 'MainActivity' selected), 'res', and 'values'. The main editor window shows the 'activity_main.xml' file, which contains the XML code for the main activity's layout. The bottom status bar indicates a successful Gradle build.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.reza.fragmenttransactions.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!">
        <!-- Layout constraints -->
        <app:layout_constraintBottom_toBottomOf="parent" />
        <app:layout_constraintLeft_toLeftOf="parent" />
        <app:layout_constraintRight_toRightOf="parent" />
        <app:layout_constraintTop_toTopOf="parent" />
    </TextView>
</android.support.constraint.ConstraintLayout>
```



Change the layout file (activity_main.xml) as shown. We are simply adding buttons for different Fragment transactions:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:id="@+id/linearLayout"
    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"
    android:background="#F0F0F0"
    android:orientation="vertical"
    tools:context="com.example.reza.fragmenttransactions.MainActivity">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="addFragmentA"
            android:text="Add A"/>

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="removeFragmentA"
            android:text="Rem A"/>

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="addFragmentB"
            android:text="Add B"/>

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="removeFragmentB"
            android:text="Rem B"/>
    </LinearLayout>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="replaceFragmentA"
            android:text="Replace By 'A'"/>

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="replaceByFragmentB"
            android:text="Replace By 'B'"/>
    </LinearLayout>
</LinearLayout>

```

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:id="@+id/linearLayout"
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="removeFragmentB"
    android:text="Rem B"/>
</LinearLayout>

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="replaceFragmentA"
            android:text="Replace By 'A'"/>

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="replaceByFragmentB"
            android:text="Replace By 'B'"/>
    </LinearLayout>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:onClick="attachFragmentA"
                android:text="Attach A"/>

            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:onClick="detachFragmentA"
                android:text="Detach A"/>

            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:onClick="attachFragmentB"
                android:text="Attach B"/>
        </LinearLayout>
</LinearLayout>

```

FragmentTransactions - [C:\Android Demos\Fragments\FragmentTransactions] - [app] - ...\\app\\src\\main\\res\\layout\\activity_main.xml - Android Studio 3.0.1

```

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
FragmentTransactions app src main res layout activity_main.xml
Android manifests java com.example.reza.fragmenttransactions MainActivity com.example.reza.fragmenttransactions (test) res drawable layout activity_main.xml mipmap values Gradle Scripts
activity_main.xml x MainActivity.java x
LinearLayout
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Attach A"
            android:onClick="attachFragmentA"/>
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Detach A"
            android:onClick="detachFragmentA"/>
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Show A"
            android:onClick="showFragmentA"/>
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hide A"
            android:onClick="hideFragmentA"/>
    </LinearLayout>
    <FrameLayout
        android:id="@+id/container"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="10dp"/>
</LinearLayout>

```

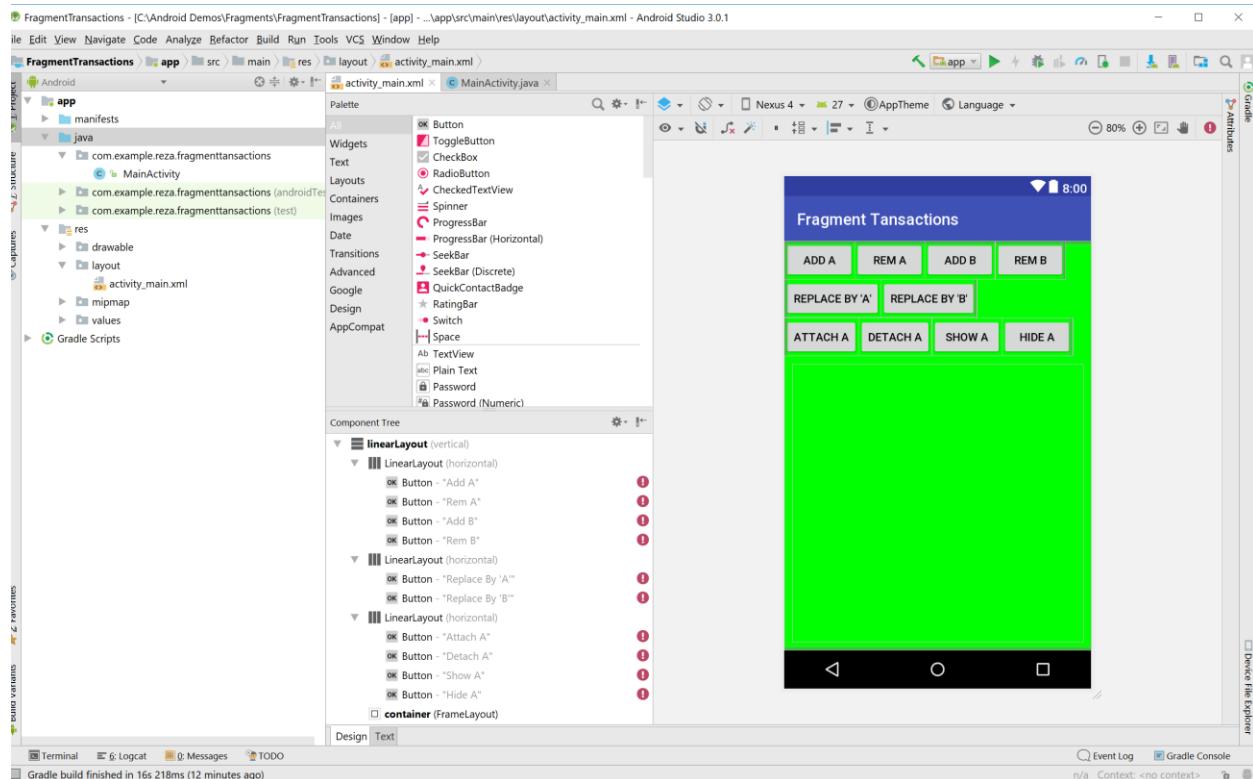
Design Text

Event Log Gradle Console

Gradle build finished in 16s 218ms (12 minutes ago)

15:42 CRLF: UTF-8 Context: <no context>

Here is how the user interface looks like:



In XML file, the red lines indicate that we don't have the onclick event handlers we have defined in our XML file so select them one by one and press Alt+Enter to have Android Studio to write the corresponding method for you in the corresponding Java file as shown:

```

<LinearLayout 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"
    android:background="#F0F0F0"
    android:orientation="vertical"
    tools:context="com.example.reza.fragmenttransactions.MainActivity">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="addFragmentA"
            android:text="Add A"/>

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="removeFragmentA"
            android:text="Rem A"/>

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="addFragmentB"
            android:text="Add B"/>

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="removeFragmentB"
            android:text="Rem B"/>
    </LinearLayout>
</LinearLayout>

```

FragmentTransactions - [C:\Android Demos\Fragments\FragmentTransactions] - [app] - ...\\app\\src\\main\\res\\layout\\activity_main.xml - Android Studio 3.0.1

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

FragmentTransactions app src main res layout activity_main.xml

MainActivity.java

```

    LinearLayout Button
        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"
        android:background="#00F0"
        android:orientation="vertical"
        tools:context="com.example.reza.fragmenttransactions.MainActivity">

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal">

            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:onClick="addFragmentA" />
        
```

Suppress: Add tool:ignore="OnClick" attribute

Create 'addFragmentA(View)' in 'MainActivity' > content

Create onClick event handler > p_content

Override Resource in Other Configuration... > p_content

Inject language or reference > fragmentA

```

            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:onClick="addFragmentB"
                android:text="Add B"/>

            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:onClick="removeFragmentB"
                android:text="Rem B"/>
        
```

```

    </LinearLayout>

```

Design Text

Event Log Gradle Console

Corresponding method handler "public void addFragmentA(android.view.View)" not found

20:37 CRLF: UTF-8 Context: <no context>

FragmentTransactions - [C:\Android Demos\Fragments\FragmentTransactions] - [app] - ...\\app\\src\\main\\java\\com\\example\\reza\\fragmenttransactions\\MainActivity.java - Android Studio 3.0.1

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

FragmentTransactions app src main java com example fragmenttransactions MainActivity

MainActivity.java

```

    package com.example.reza.fragmenttransactions;

    import android.support.v7.app.AppCompatActivity;
    import android.os.Bundle;
    import android.view.View;

    public class MainActivity extends AppCompatActivity {

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

        public void addFragmentA(View view) {
        }
    }

```

Event Log Gradle Console

Gradle build finished in 16s 218ms (14 minutes ago)

15:42 CRLF: UTF-8 Context: <no context>

FragmentTransactions - [C:\Android Demos\Fragments\FragmentTransactions] - [app] - ...\\app\\src\\main\\res\\layout\\activity_main.xml - Android Studio 3.0.1

```

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
FragmentTransactions app src main res layout activity_main.xml
Android
  manifests
  java
    com.example.reza.fragmenttransactions
      MainActivity
    com.example.reza.fragmenttransactions (androidTest)
    com.example.reza.fragmenttransactions (test)
  res
    drawable
    layout
      activity_main.xml
    mipmap
    values
  Gradle Scripts

```

activity_main.xml x MainActivity.java

```

LinearLayout LinearLayout Button
  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"
  android:background="#00F0"
  android:orientation="vertical"
  tools:context="com.example.reza.fragmenttransactions.MainActivity">

  <LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:onClick="addFragmentA"
      android:text="Add A"/>

    <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:onClick="removeFragmentA"
      android:text="Remove A"/>
  
```

Suppress: Add tools:ignore="OnClick" attribute

- Create onClick event handler > intent > content > "
- Create onClick event handler > intent > content > "
- Override Resource in Other Configuration... > "
- Inject language or reference > "

```

    <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:onClick="removeFragmentB"
      android:text="Rem B"/>
  
```

```

</LinearLayout>
<LinearLayout
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:orientation="horizontal">

```

Design Text

Event Log Gradle Console

Corresponding method handler "public void removeFragmentA(android.view.View)" not found

26.39 CRLF: UTF-8 Context: <no context>

FragmentTransactions - [C:\Android Demos\Fragments\FragmentTransactions] - [app] - ...\\app\\src\\main\\java\\com\\example\\reza\\fragmenttransactions\\MainActivity.java - Android Studio 3.0.1

```

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
FragmentTransactions app src main java com example reza fragmenttransactions MainActivity
Android
  manifests
  java
    com.example.reza.fragmenttransactions
      MainActivity
    com.example.reza.fragmenttransactions (androidTest)
    com.example.reza.fragmenttransactions (test)
  res
    drawable
    layout
      activity_main.xml
    mipmap
    values
  Gradle Scripts

```

activity_main.xml x MainActivity.java

```

MainActivity removeFragmentA()
package com.example.reza.fragmenttransactions;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class MainActivity extends AppCompatActivity {

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

    public void addFragmentA(View view) {
    }

    public void removeFragmentA(View view) {
    }
}

```

Event Log Gradle Console

Gradle build finished in 16s 218ms (15 minutes ago)

18:45 CRLF: UTF-8 Context: <no context>

The following screen captures show no red lines meaning that we have created all the onClick event handlers in our Java file:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:id="@+id/linearLayout"
    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"
    android:background="#F0F0F0"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="addFragmentA"
            android:text="Add A"/>

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="removeFragmentA"
            android:text="Remove A"/>

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="addFragmentB"
            android:text="Add B"/>

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:onClick="removeFragmentB"
            android:text="Remove B"/>
    </LinearLayout>
</LinearLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:id="@+id/linearLayout"
    android:layout_height="wrap_content"
    android:onClick="removeFragmentB"
    android:text="Remove B"/>
</LinearLayout>

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Replace By A"
        android:onClick="replaceByFragmentA"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Replace By B"
        android:onClick="replaceByFragmentB"/>
    </LinearLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Attach A"
        android:onClick="attachFragmentA"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Detach A"
        android:onClick="detachFragmentA"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Attach Fragment A"
        android:onClick="attachFragmentA"/>
    </LinearLayout>
```

```

<LinearLayout
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Attach A"
        android:onClick="attachFragmentA"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Detach A"
        android:onClick="detachFragmentA"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show A"
        android:onClick="showFragmentA"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hide A"
        android:onClick="hideFragmentA"/>

</LinearLayout>

<FrameLayout
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="10dp"/>

</LinearLayout>

```

And here is the Java file with all those event handlers (they have no bodies so you have to implement them)

```

>MainActivity hideFragmentA()
{
}

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

public void addFragmentA(View view) {
}

public void removeFragmentA(View view) {
}

public void addFragmentB(View view) {
}

public void removeFragmentB(View view) {
}

public void replaceByFragmentA(View view) {
}

public void replaceByFragmentB(View view) {
}

public void attachFragmentA(View view) {
}

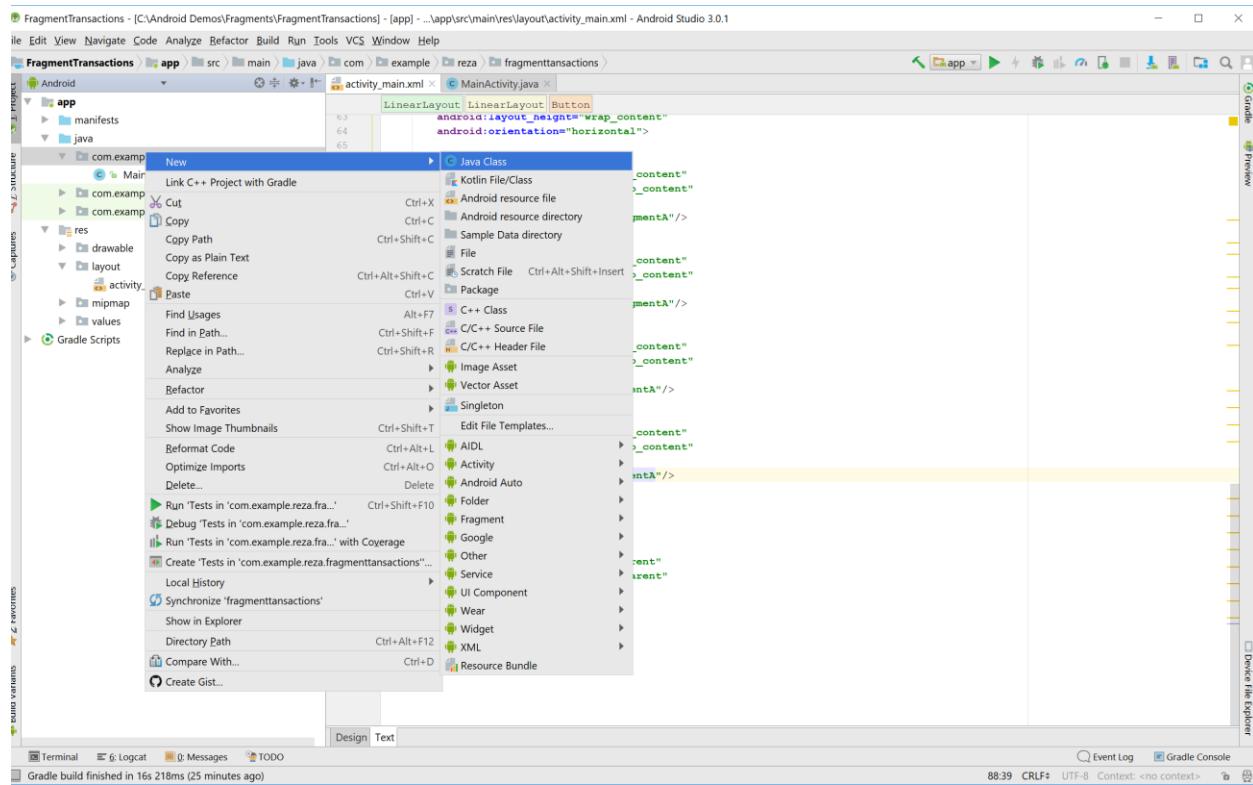
public void detachFragmentA(View view) {
}

public void showFragmentA(View view) {
}

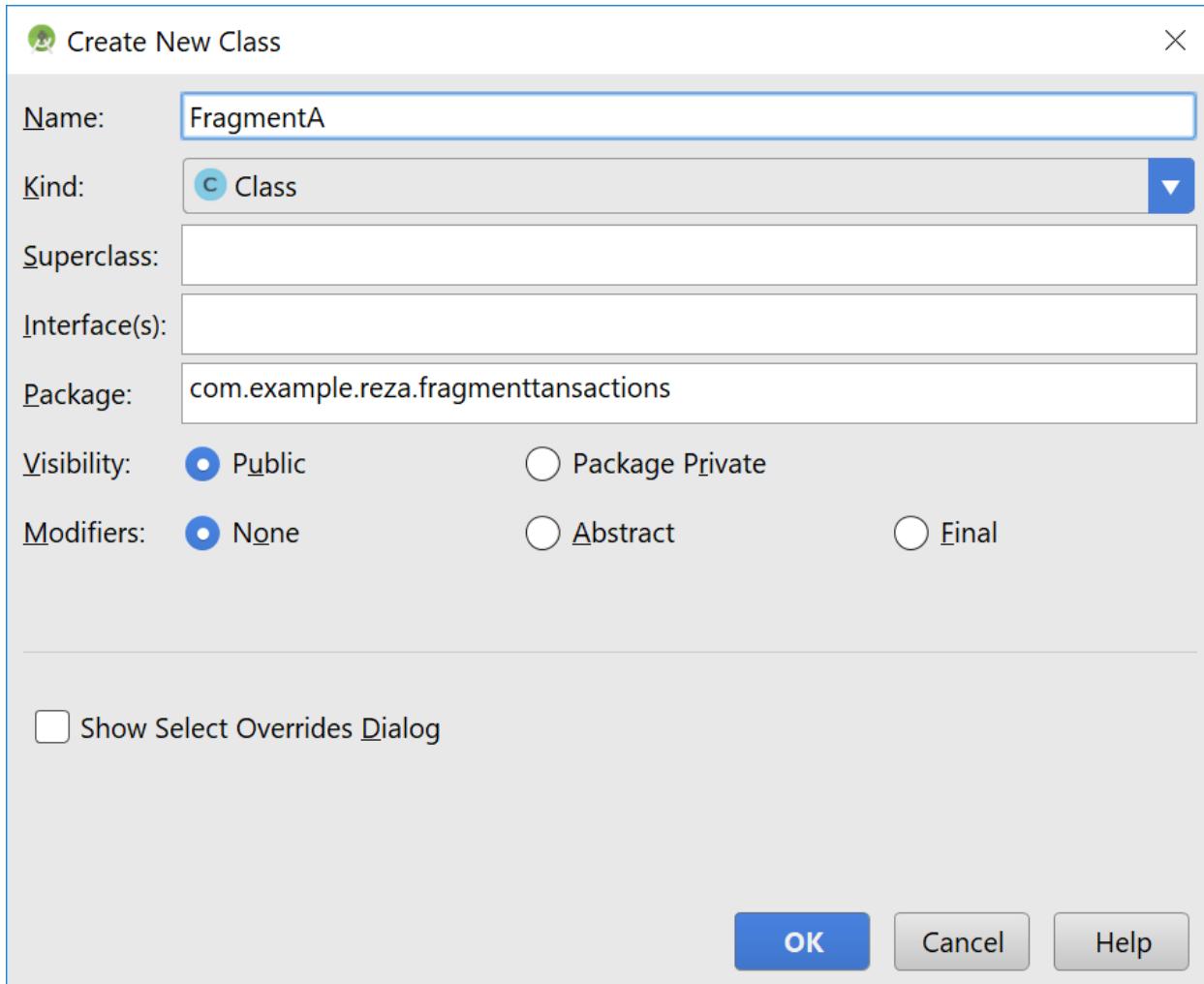
public void hideFragmentA(View view) {
}

```

Next we are going to create our fragments. Our fragments are subclasses of Fragment class so first we create a new Java class:



Call it FragmentA:



FragmentTransactions - [C:\Android Demos\Fragments\FragmentTransactions] - [app] - ...\\app\\src\\main\\java\\com\\example\\reza\\fragmenttransactions\\FragmentA.java - Android Studio 3.0.1

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

FragmentTransactions app src main java com example reza fragmenttransactions FragmentA

activity_main.xml FragmentA.java MainActivity.java

Android Project Captures Device File Explorer

Build Variants Favorites

Terminal Logcat Messages TODO Event Log Gradle Console

7:23 CRLF: UTF-8 Context: <no context>

Class 'FragmentA' is never used

```
FragmentA
package com.example.reza.fragmenttransactions;

/*
 * Created by Reza on 2018-03-09.
 */
public class FragmentA {
```

FragmentTransactions - [C:\Android Demos\Fragments\FragmentTransactions] - [app] - ...\\app\\src\\main\\java\\com\\example\\reza\\fragmenttransactions\\FragmentA.java - Android Studio 3.0.1

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

FragmentTransactions app src main java com example reza fragmenttransactions FragmentA

activity_main.xml FragmentA.java MainActivity.java

Android Project Captures Device File Explorer

Build Variants Favorites

Terminal Logcat Messages TODO Event Log Gradle Console

Identifier expected

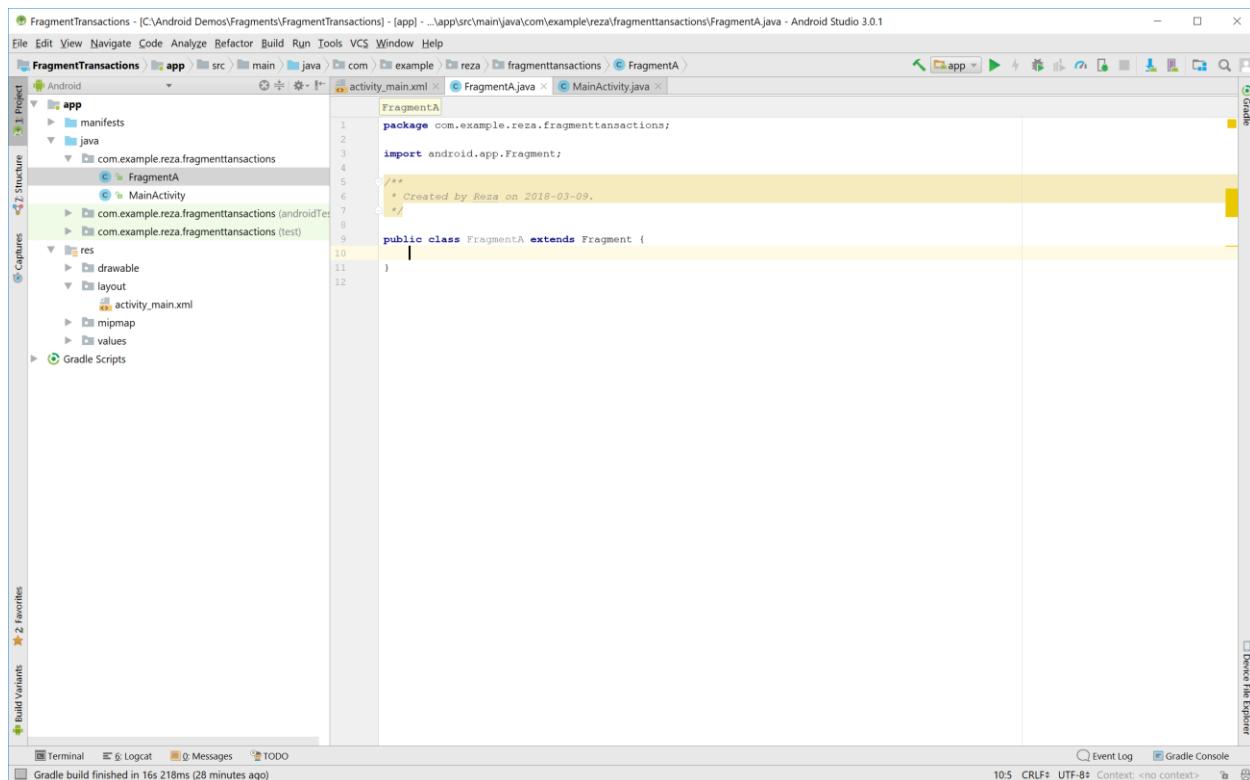
7:34 CRLF: UTF-8 Context: <no context>

```
FragmentA
package com.example.reza.fragmenttransactions;

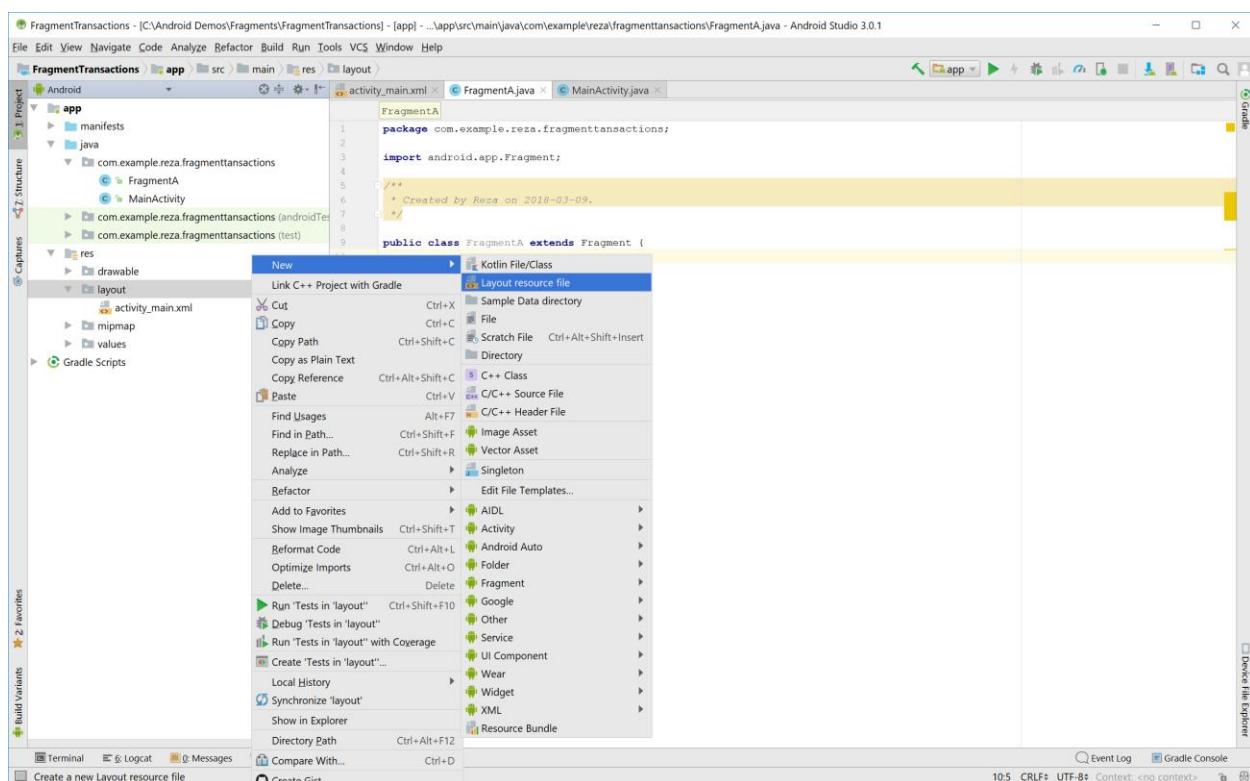
/*
 * Created by Reza on 2018-03-09.
 */
public class FragmentA extends Fr{}
```

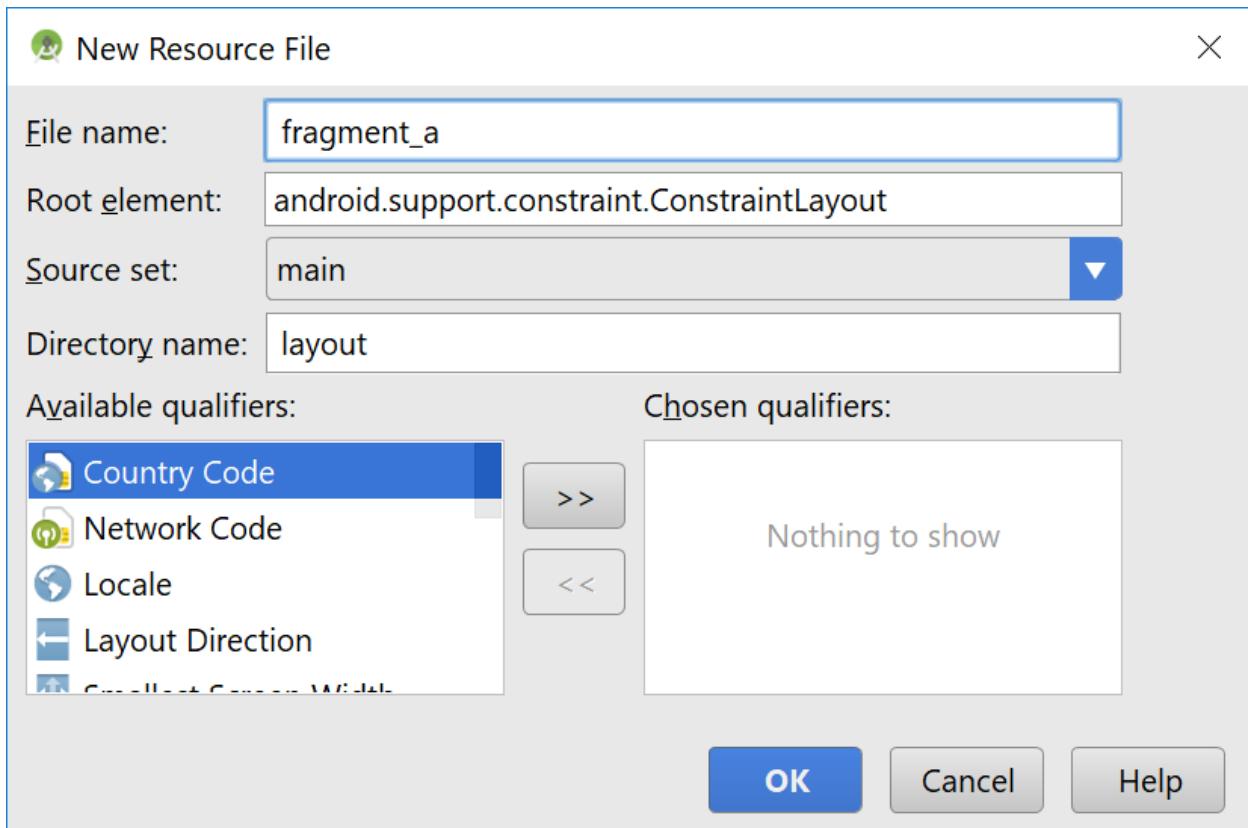
The code completion dropdown shows the following suggestions:

- ' FragmentTransaction (android.app)
- ' Fragment (android.app)
- ' Fragment (android.support.v4.app)
- ' FragmentTransaction (android.support.v4.app)
- ' FragmentManager (android.app)
- ' FragmentManager (android.support.v4.app)
- ' FractionBuilder (android.text.style.TtsSpan)
- ' FragmentContainer (android.app)
- ' FragmentController (android.app)
- ' FragmentHostCallback<E> (android.app)
- ' FragmentManager (android.app)

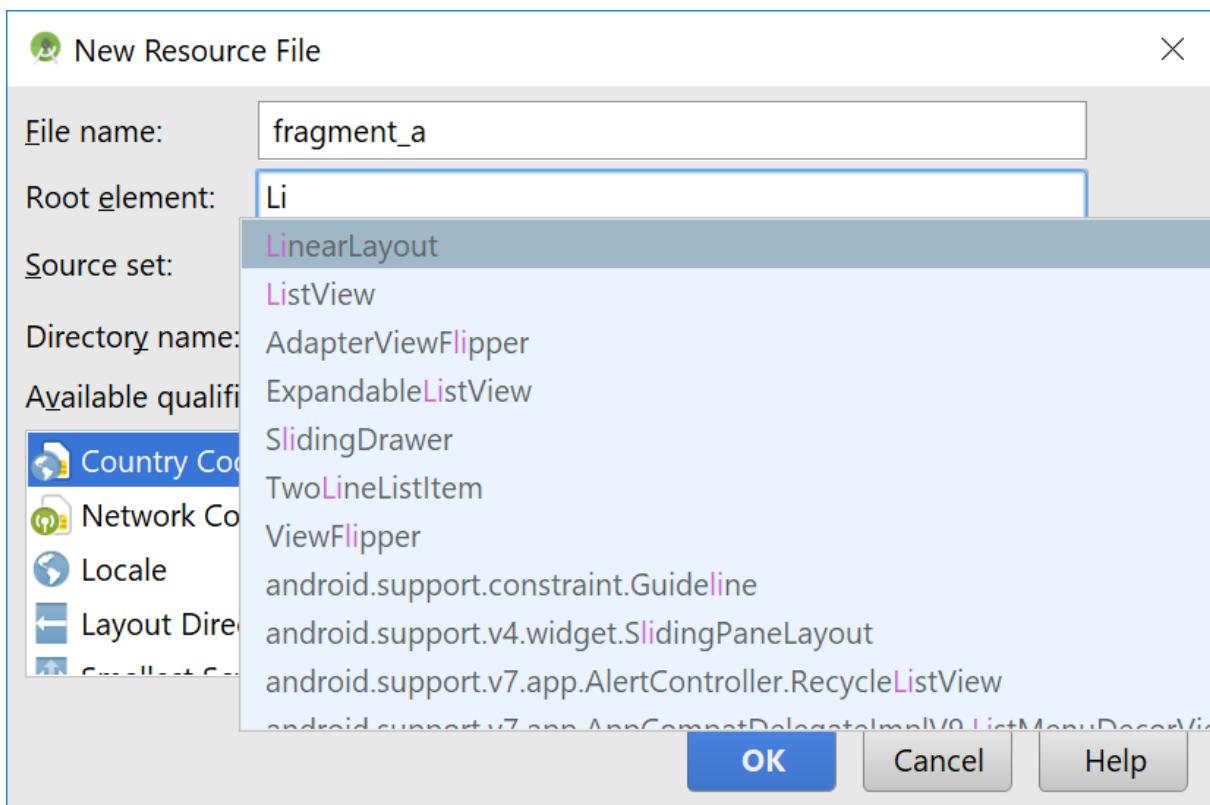
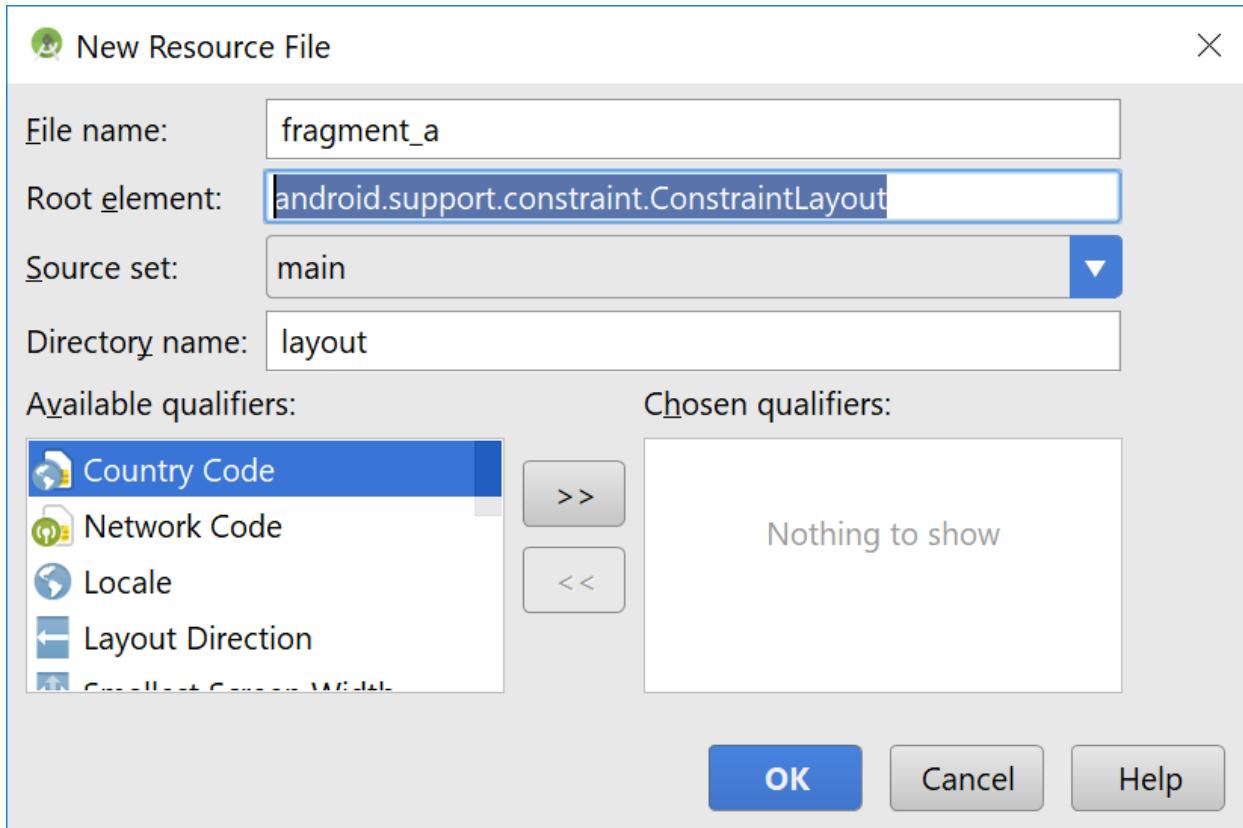


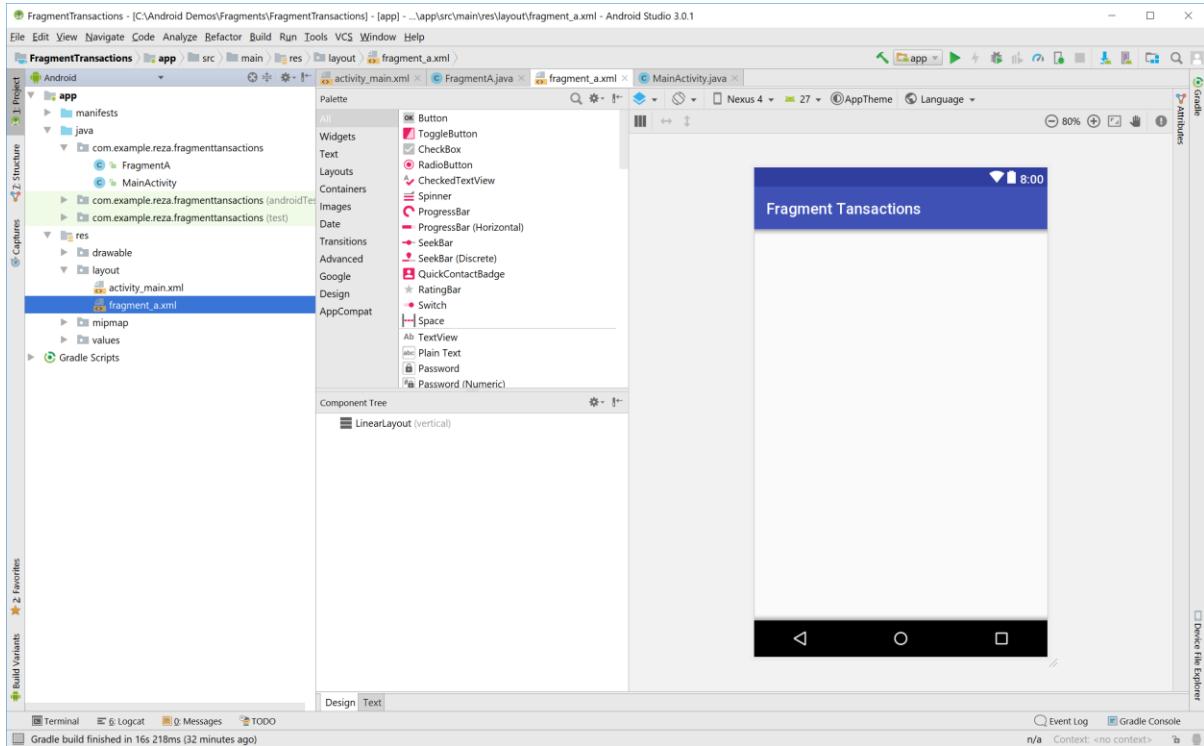
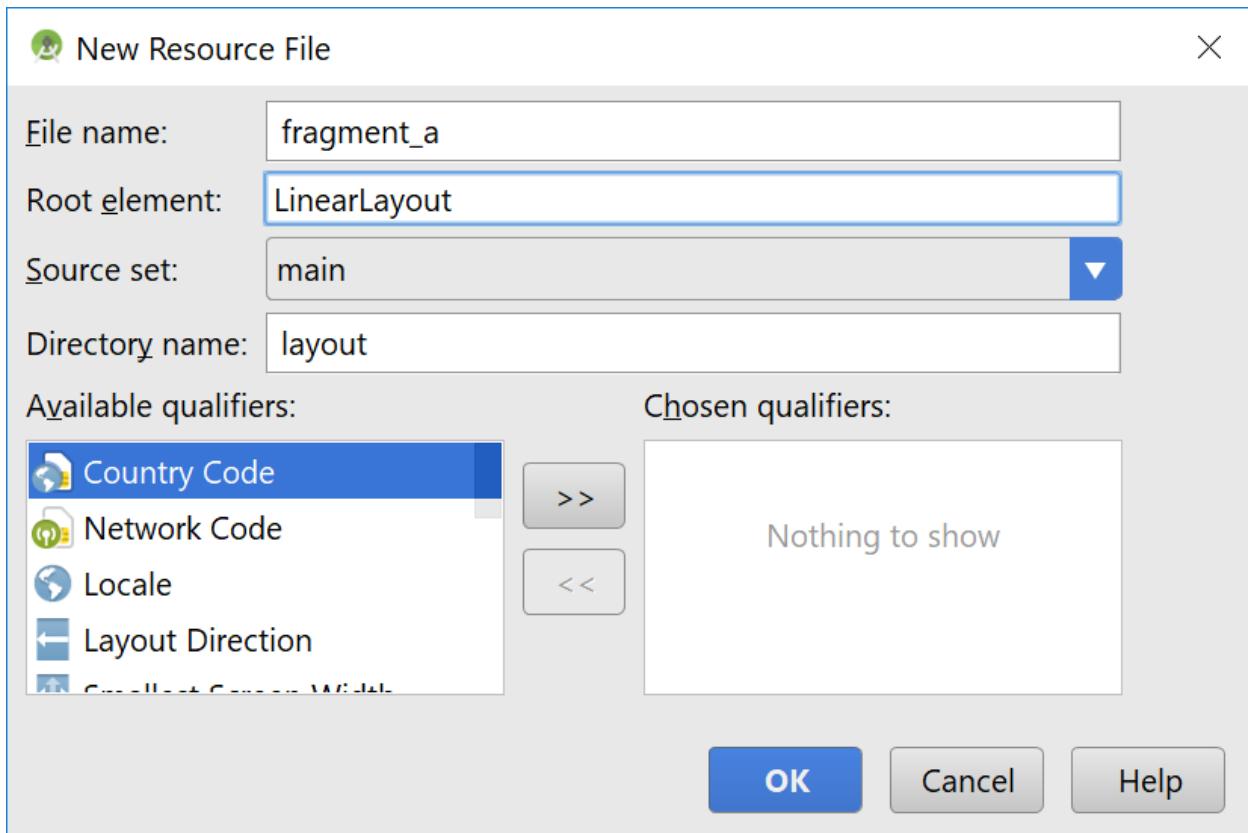
Next create a layout for your fragment class as shown:





We are going to change the root layout to a LinearLayout as shown:





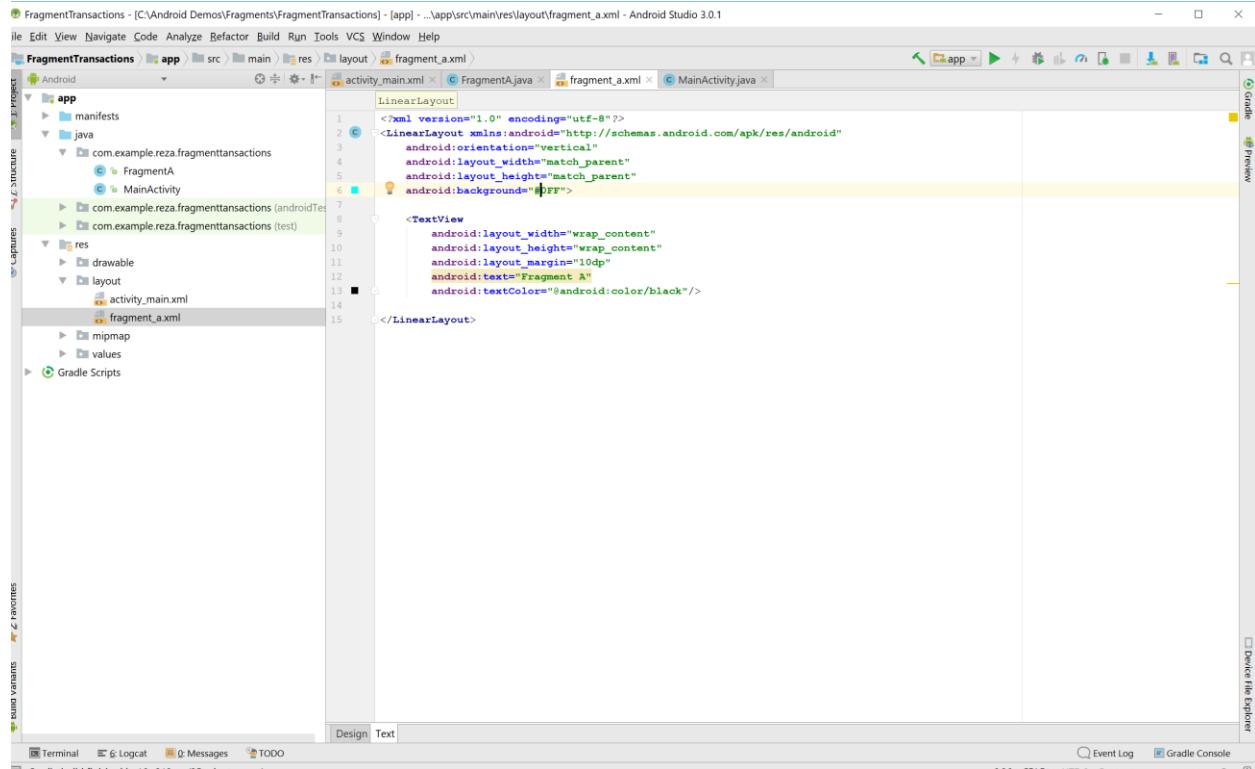
Here is the layout:

The screenshot shows the Android Studio interface with the project 'FragmentTransactions' open. The 'fragment_a.xml' file is selected in the layout editor. The code view shows the XML structure:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">
```

The project structure on the left includes 'app', 'java' (with 'com.example.reza.fragmenttransactions' containing 'FragmentA' and 'MainActivity'), 'res', 'drawable', and 'layout' (containing 'activity_main.xml' and 'fragment_a.xml'). The bottom status bar indicates 'Gradle build finished in 16s 218ms (32 minutes ago)'.

Add an element and set the background color so it is easy to see the fragment when displayed in the activity:

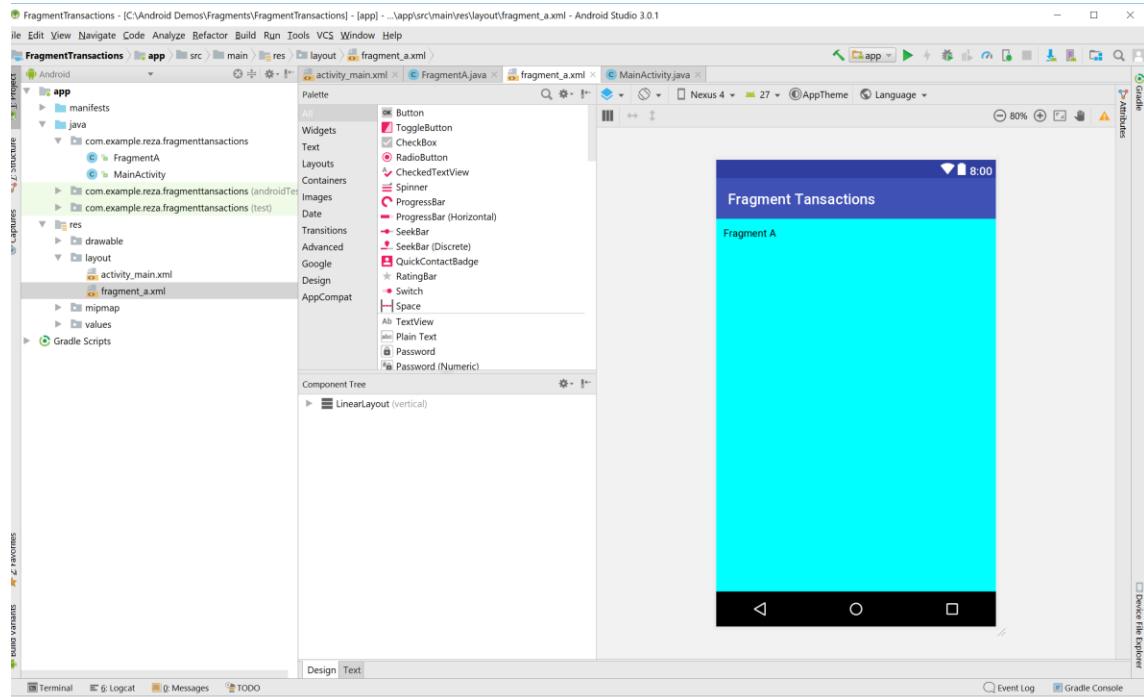


The screenshot shows the Android Studio interface with the project 'FragmentTransactions' open. The 'fragment_a.xml' file is selected in the layout editor. The XML code defines a linear layout with a white background and a text view with black text and a black background:

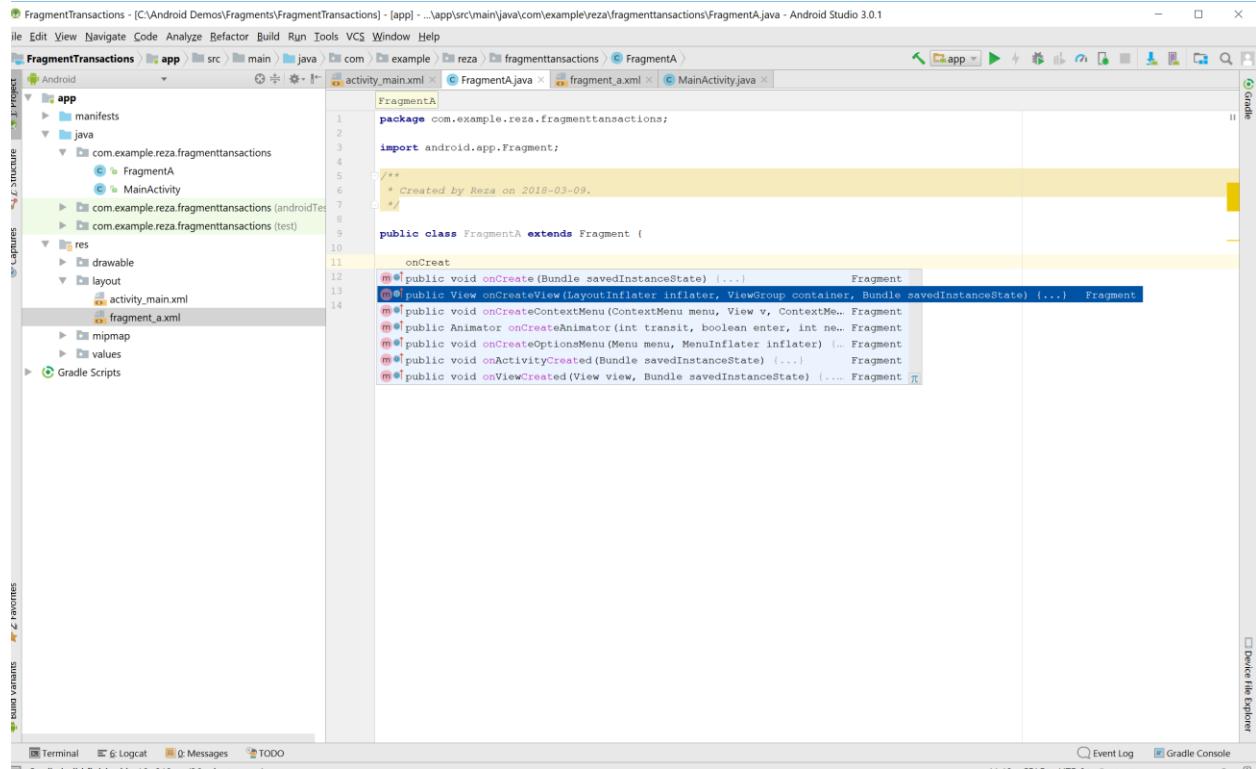
```

<LinearLayout
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFF">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="Fragment A"
        android:textColor="@android:color/black"/>
</LinearLayout>

```



Now go back to FragmentA.java code and use the onCreate () method to create the connection between the java code and its layout file.



```
FragmentTransactions - [C:\Android Demos\Fragments\FragmentTransactions] - [app] - ...app\src\main\java\com\example\reza\fragmenttransactions\FragmentA.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
FragmentTransactions app src main java com example reza fragmenttransactions FragmentA
activity_main.xml FragmentA fragment_a.xml MainActivity.java
Android
PROJECT app manifests java com.example.reza.fragmenttransactions FragmentA MainActivity
com.example.reza.fragmenttransactions (androidTest)
com.example.reza.fragmenttransactions (test)
LAYOUTS res drawable layout activity_main.xml fragment_a.xml
mipmap values
Gradle Scripts
1 package com.example.reza.fragmenttransactions;
2
3 import android.app.Fragment;
4
5 /**
6  * Created by Reza on 2018-03-09.
7  */
8
9 public class FragmentA extends Fragment {
10
11     @Override
12     public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
13         return inflater.inflate(R.layout.fragment_a, container, false);
14     }
}
```

You need to replace the highlighted code as shown:

The screenshot shows the Android Studio interface with the FragmentTransactions project open. The FragmentA.java file is displayed in the main editor window. The code is as follows:

```
FragmentA onCreateView()
{
    package com.example.reza.fragmenttransactions;
    import android.app.Fragment;
    import android.os.Bundle;
    import android.support.annotation.Nullable;
    import android.view.LayoutInflater;
    import android.view.View;
    import android.view.ViewGroup;

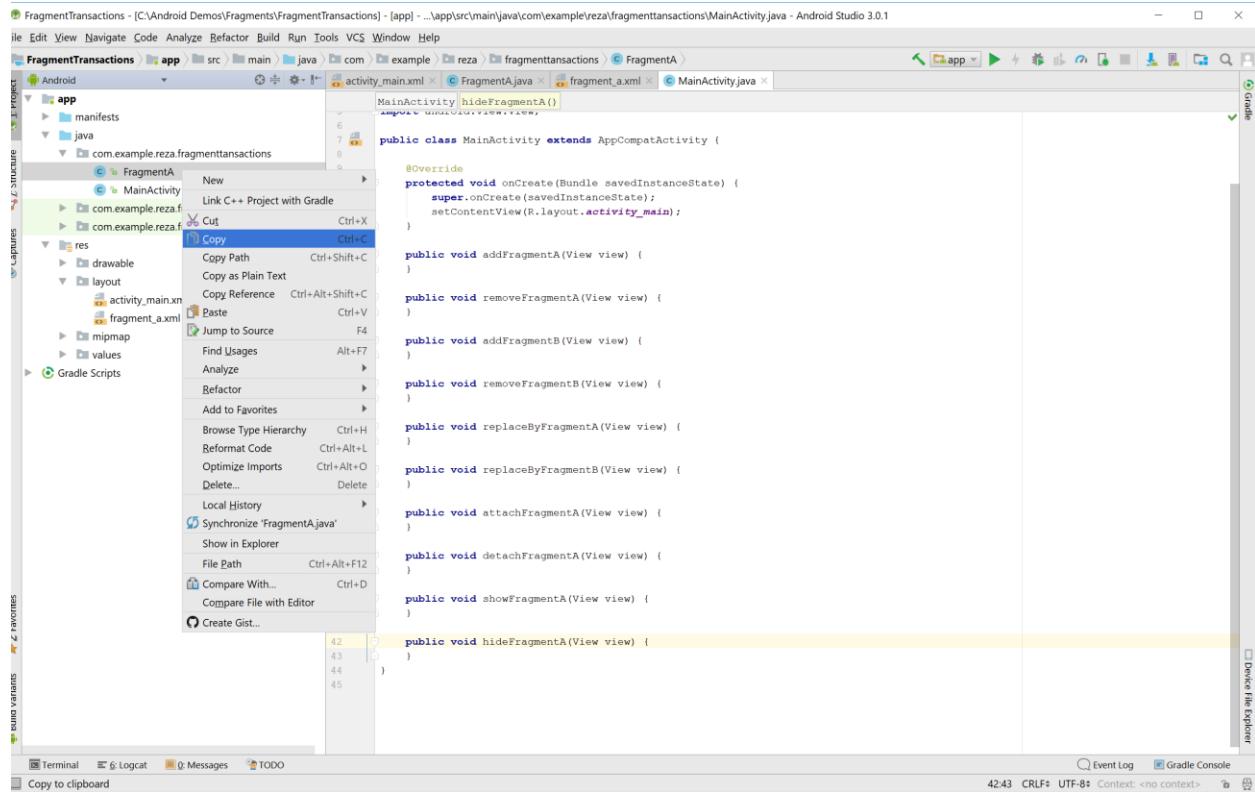
    /**
     * Created by Reza on 2018-03-09.
     */
    public class FragmentA extends Fragment {
        @Nullable
        @Override
        public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, Bundle savedInstanceState) {
            return super.onCreateView(inflater, container, savedInstanceState);
        }
    }
}
```

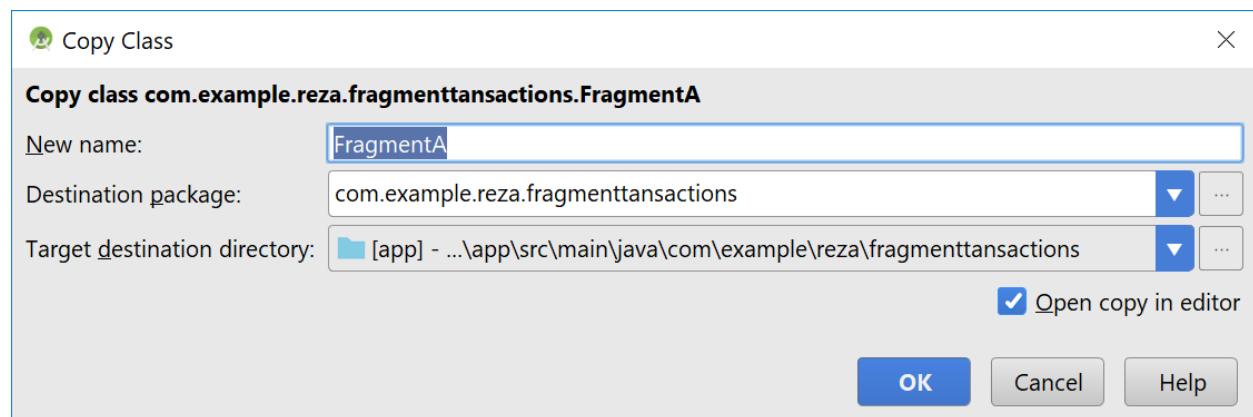
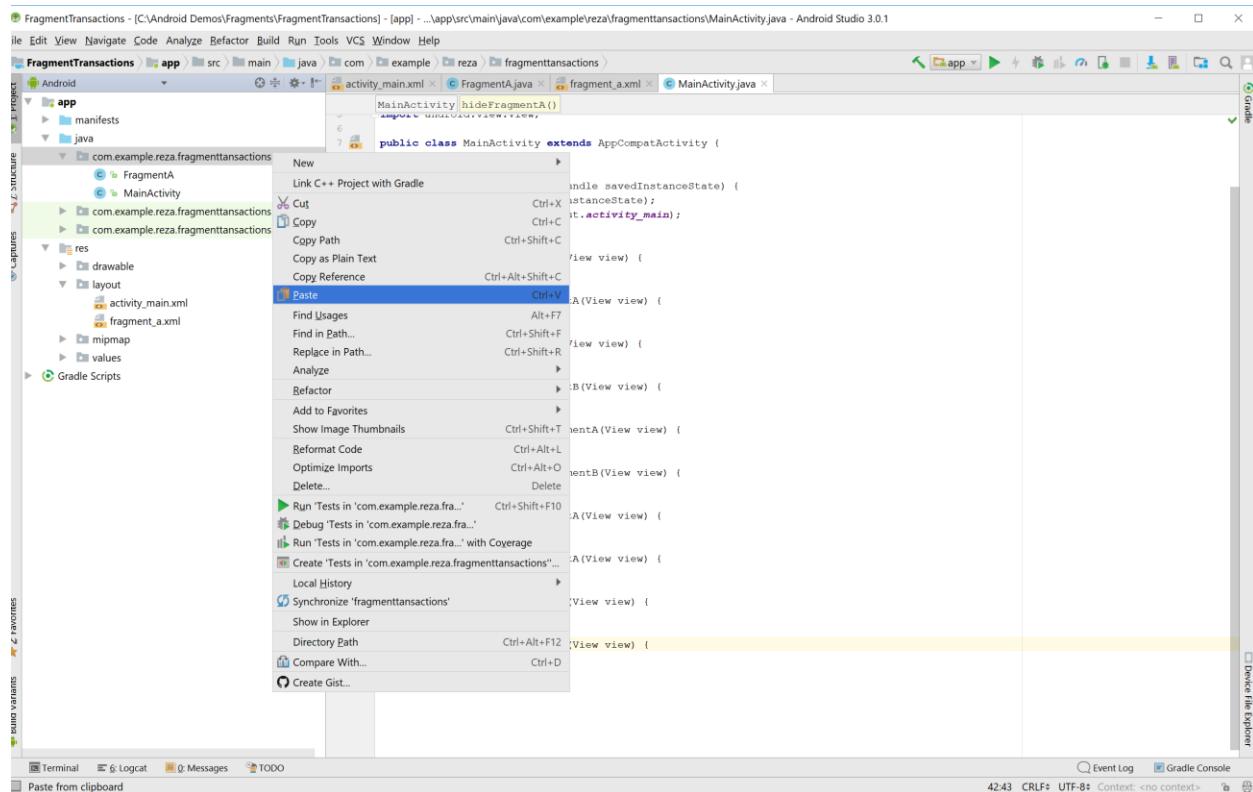
The screenshot shows the same Android Studio interface with the FragmentA.java file. The code has been modified to inflate a specific layout resource:

```
FragmentA onCreateView()
{
    package com.example.reza.fragmenttransactions;
    import android.app.Fragment;
    import android.os.Bundle;
    import android.support.annotation.Nullable;
    import android.view.LayoutInflater;
    import android.view.View;
    import android.view.ViewGroup;

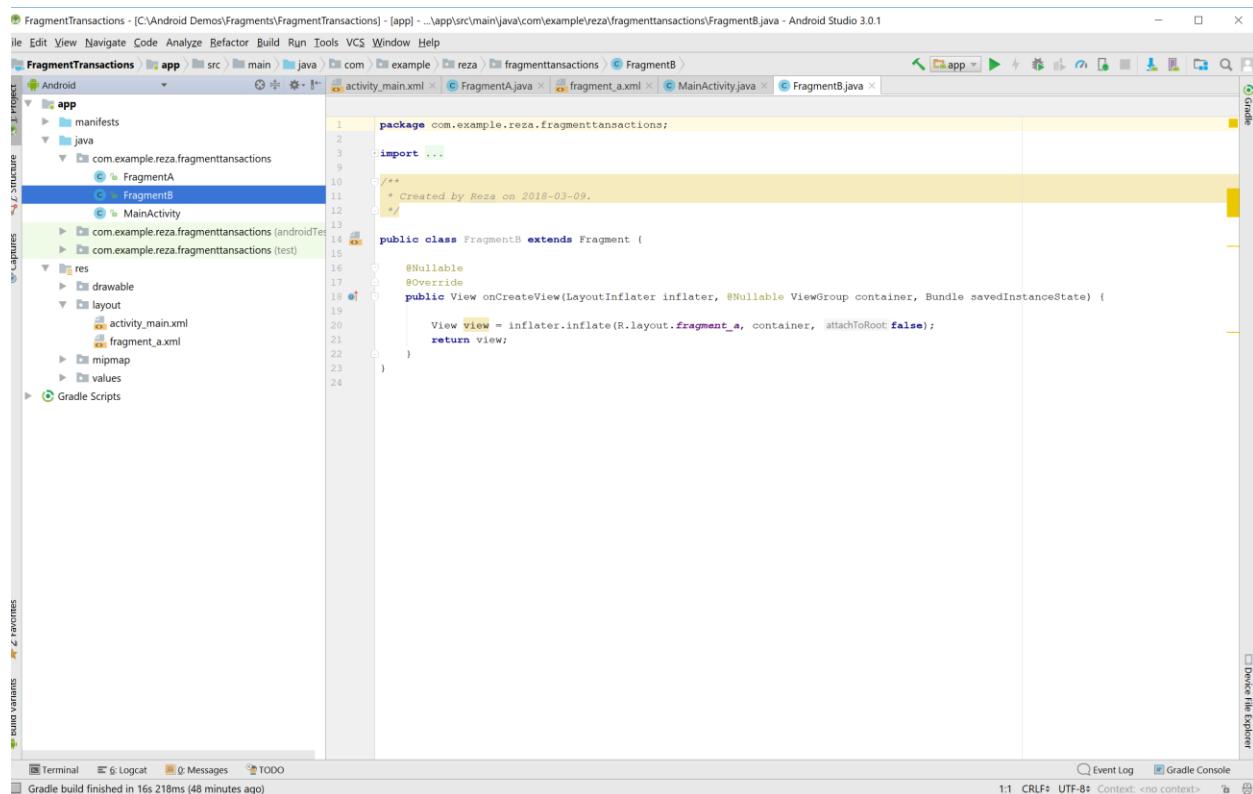
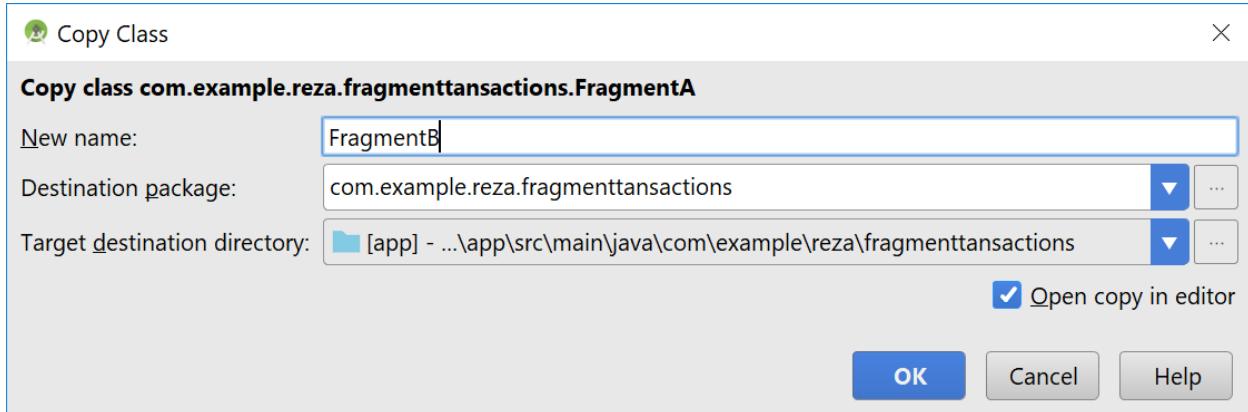
    /**
     * Created by Reza on 2018-03-09.
     */
    public class FragmentA extends Fragment {
        @Nullable
        @Override
        public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, Bundle savedInstanceState) {
            View view = inflater.inflate(R.layout.fragment_a, container, false);
            return view;
        }
    }
}
```

Now for FragmentB we can just copy FragmentA and paste it inside the package as shown:

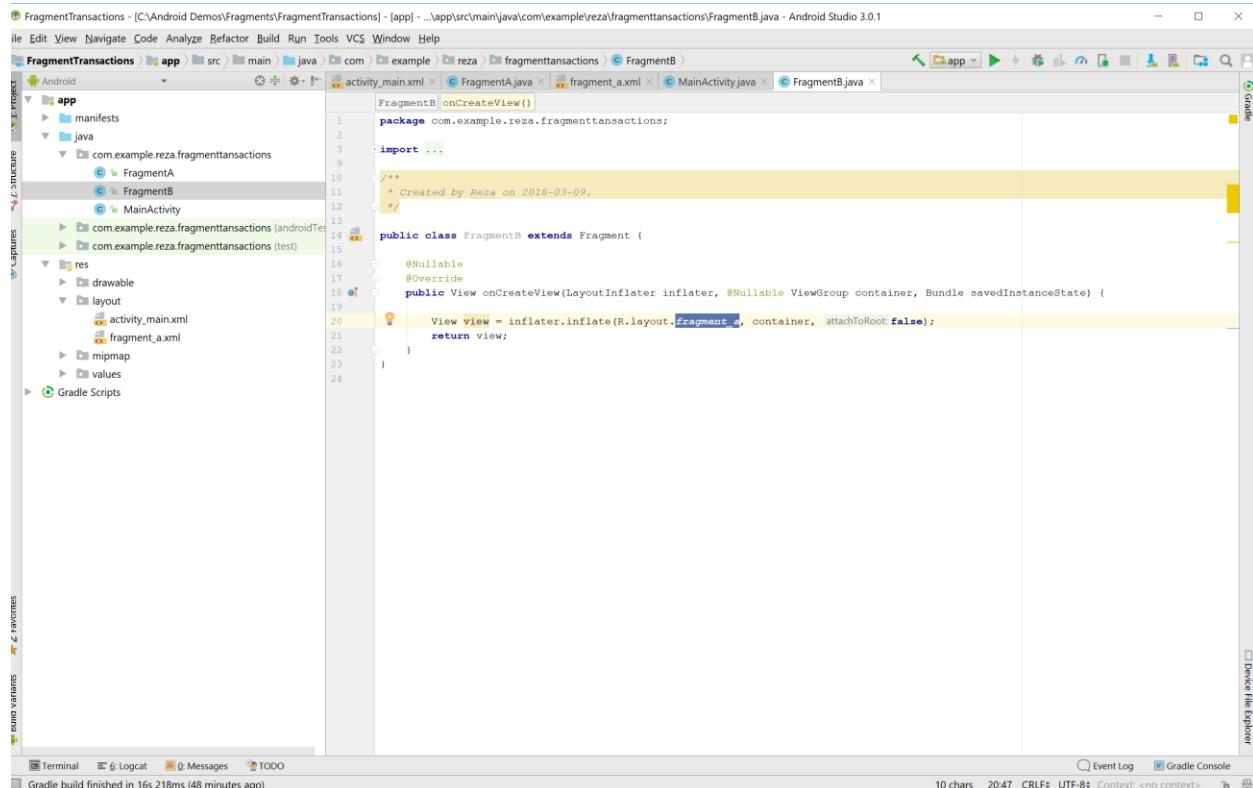




Change the name to FragmentB and click Ok as shown:



We need to change fragment_a to fragment_b in the highlighted code as shown:

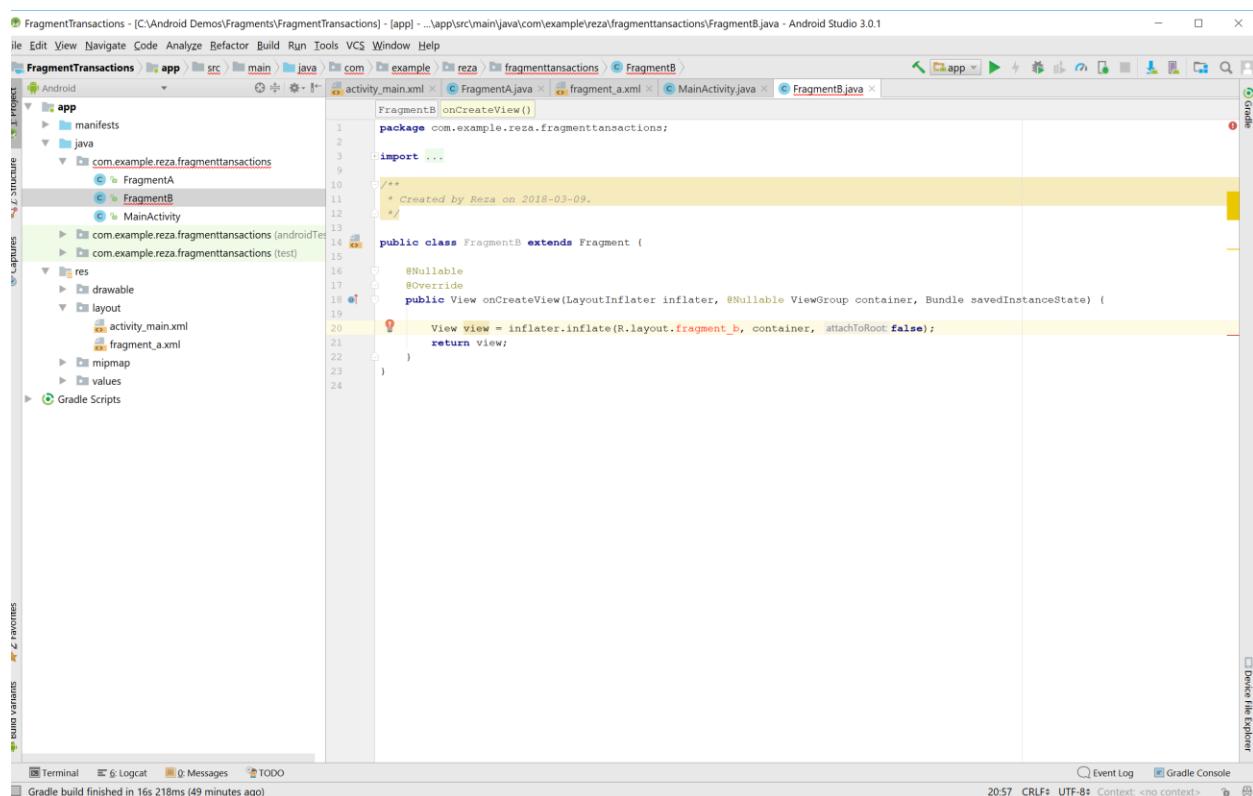


```
FragmentTransactions - [C:\Android Demos\Fragments\FragmentTransactions] - [app] - ...\\app\\src\\main\\java\\com\\example\\reza\\fragmenttransactions\\FragmentB.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
FragmentTransactions app src main java com example reza fragmenttransactions FragmentB
Android app manifests java com.example.reza.fragmenttransactions FragmentA FragmentB MainActivity
com.example.reza.fragmenttransactions (androidTest) com.example.reza.fragmenttransactions (test)
res drawable layout activity_main.xml fragment_a.xml MainActivity.java FragmentB.java
mipmap values
Gradle Scripts

activity_main.xml FragmentA.java fragment_a.xml MainActivity.java FragmentB.java

public class FragmentB extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_a, container, attachToRoot: false);
        return view;
    }
}

Event Log Gradle Console
Terminal Logcat Messages TODO
Gradle build finished in 16s 218ms (48 minutes ago) 10 chars 20:47 CRLF: UTF-8 Context: <no context>
Device file explorer
```



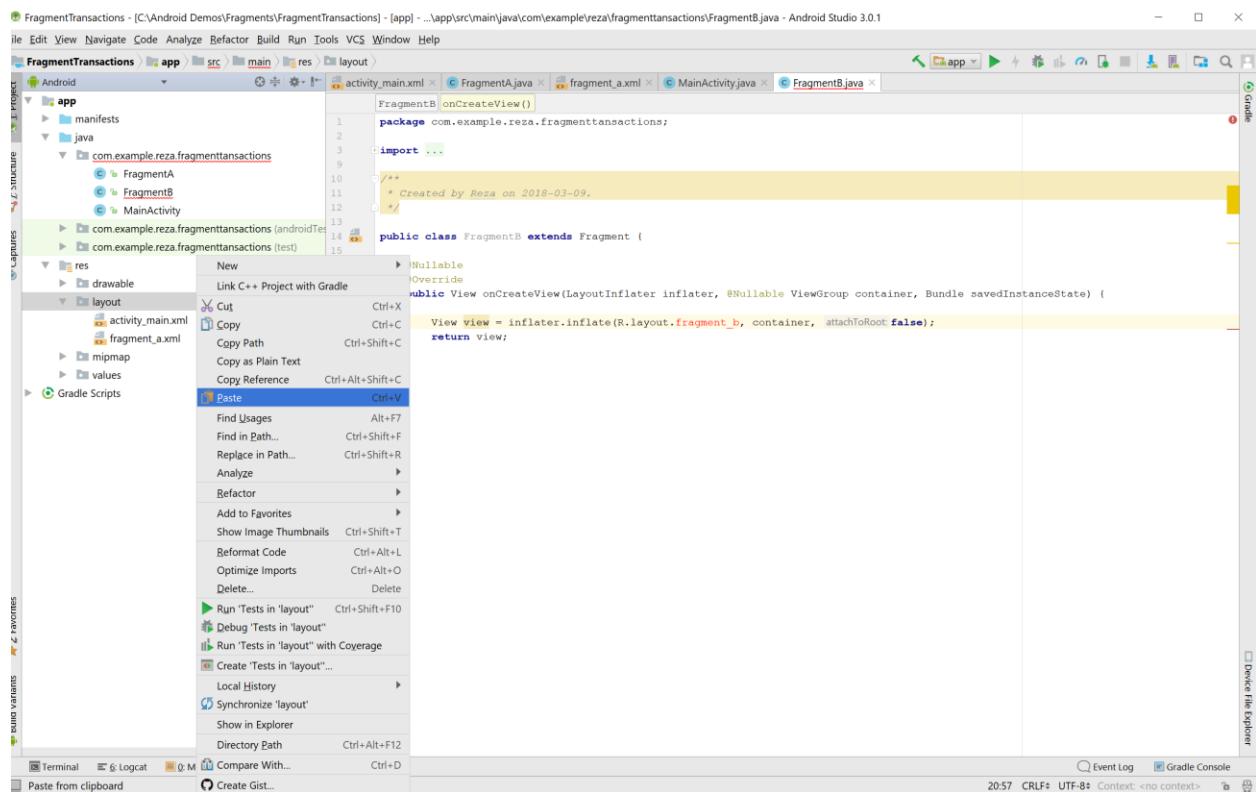
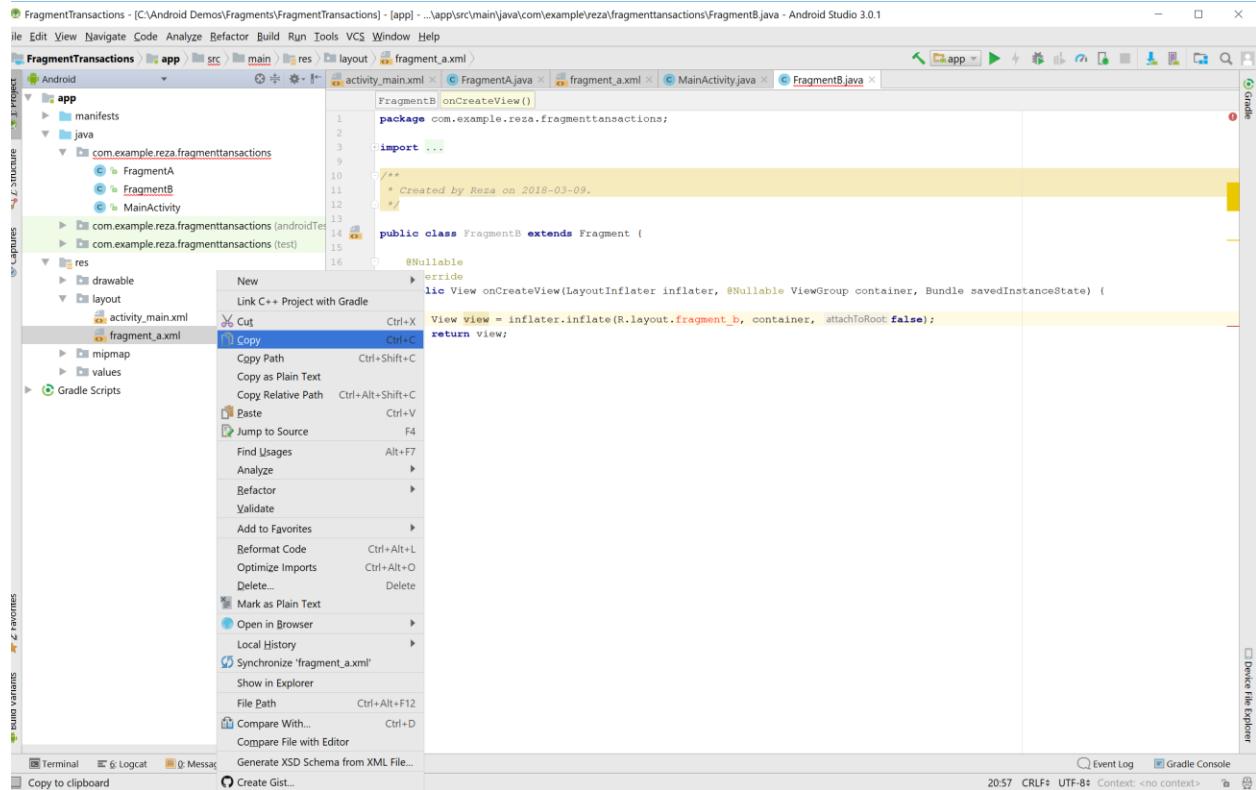
```
FragmentTransactions - [C:\Android Demos\Fragments\FragmentTransactions] - [app] - ...\\app\\src\\main\\java\\com\\example\\reza\\fragmenttransactions\\FragmentB.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
FragmentTransactions app src main java com example reza fragmenttransactions FragmentB
Android app manifests java com.example.reza.fragmenttransactions FragmentA FragmentB MainActivity
com.example.reza.fragmenttransactions (androidTest) com.example.reza.fragmenttransactions (test)
res drawable layout activity_main.xml fragment_a.xml MainActivity.java FragmentB.java
mipmap values
Gradle Scripts

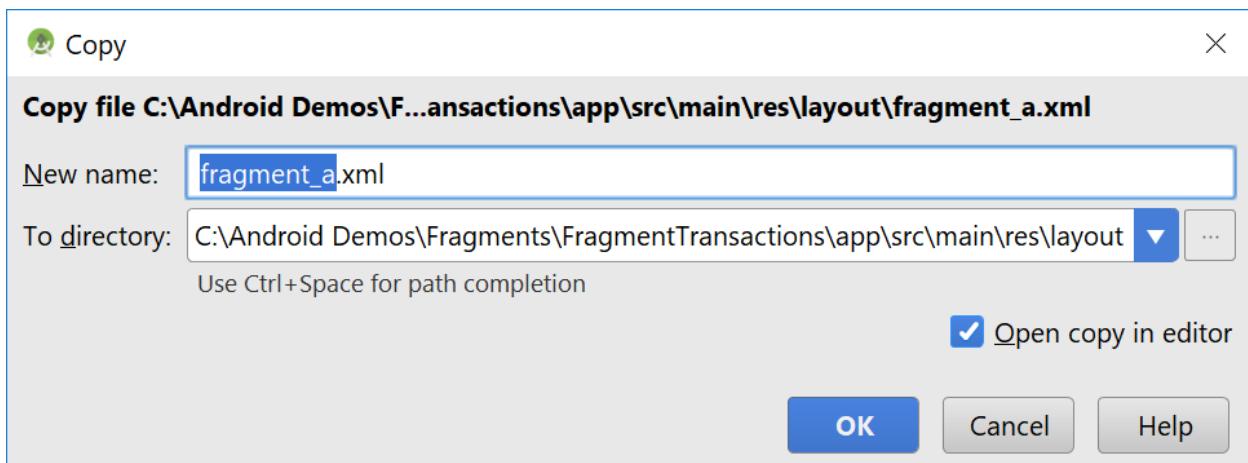
activity_main.xml FragmentA.java fragment_a.xml MainActivity.java FragmentB.java

public class FragmentB extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_b, container, attachToRoot: false);
        return view;
    }
}

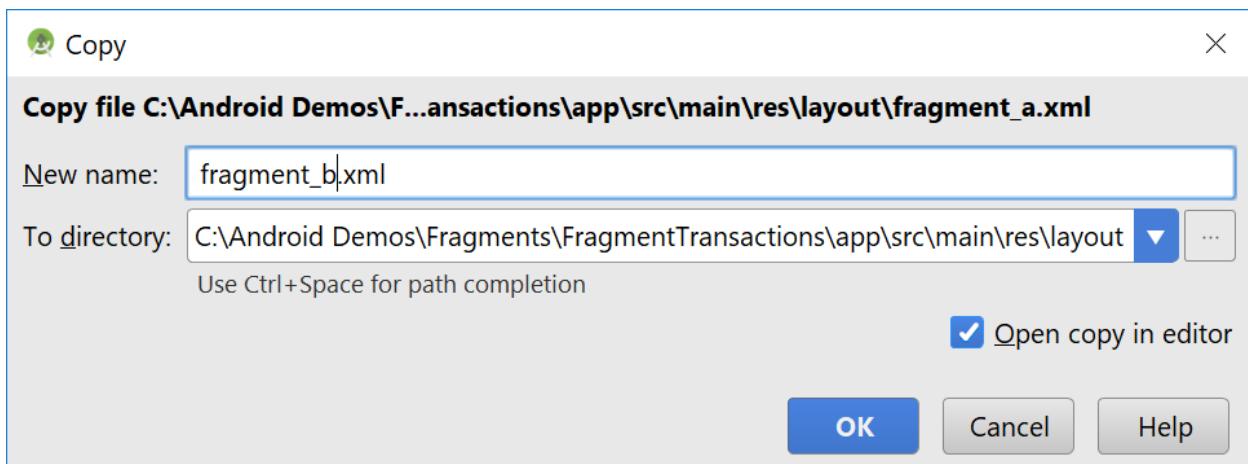
Event Log Gradle Console
Terminal Logcat Messages TODO
Gradle build finished in 16s 218ms (49 minutes ago) 20:57 CRLF: UTF-8 Context: <no context>
Device file explorer
```

It appears as red since we don't have a fragment_b.xml file so copy fragment_a.xml and paste it in the layout folder as shown:





Change the name from fragment_a.xml to fragment_b.xml as shown:



The screenshot shows the Android Studio interface with the project 'FragmentTransactions' open. The layout editor is displaying the XML code for 'fragment_b.xml'. The code defines a vertical linear layout with a white background and a black text view containing the string 'Fragment A'. The text view has a 10dp margin on the left.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFF">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="Fragment A"
        android:textColor="@android:color/black"/>

</LinearLayout>
```

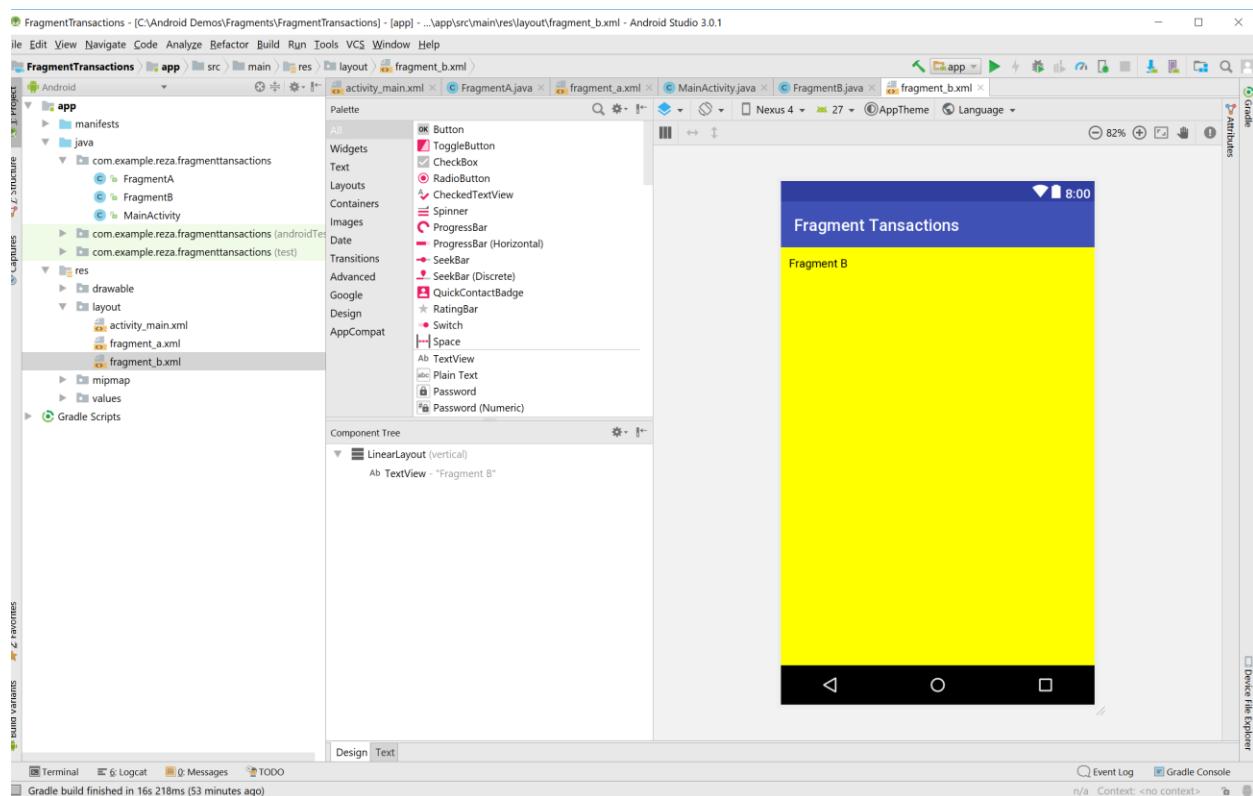
Change the background color and the text as shown:

The screenshot shows the Android Studio interface with the project 'FragmentTransactions' open. The layout editor is displaying the XML code for 'fragment_b.xml'. The background color has been changed to #FF0000 (red), and the text has been changed to 'Fragment B'.

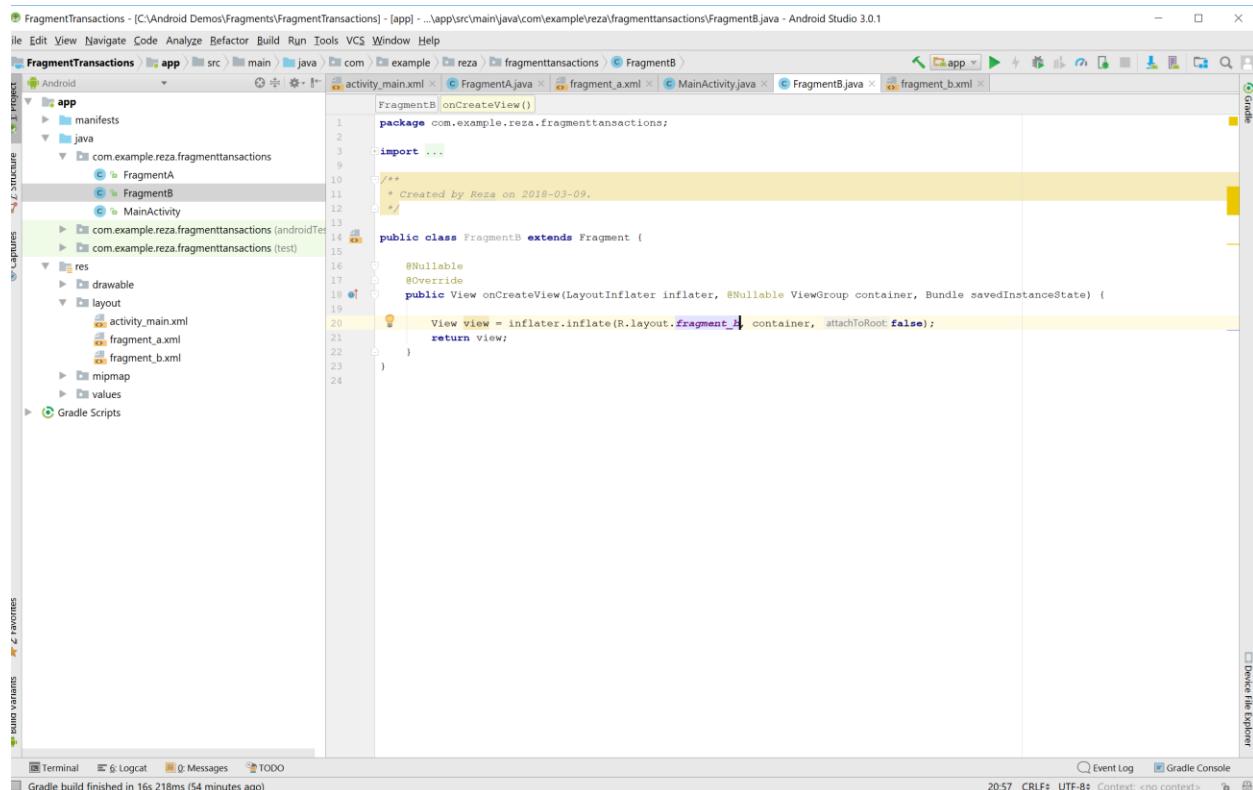
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FF0">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="Fragment B"
        android:textColor="@android:color/black"/>

</LinearLayout>
```



And here is the FragmentB.java and the “fragment_b” is not red anymore:



```
FragmentTransactions - [C:\Android Demos\Fragments\FragmentTransactions] - [app] - ...\\app\\src\\main\\java\\com\\example\\reza\\fragmenttransactions\\FragmentB.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
FragmentTransactions app src main java com example reza fragmenttransactions FragmentB
Android manifests java com.example.reza.fragmenttransactions FragmentA FragmentB MainActivity
com.example.reza.fragmenttransactions (androidTest) com.example.reza.fragmenttransactions (test)
res drawable layout activity_main.xml fragment_a.xml fragment_b.xml
mipmap values
Gradle Scripts
```

```
FragmentB onCreateView()
1 package com.example.reza.fragmenttransactions;
2 import ...
3 /**
4 * Created by Reza on 2018-03-09.
5 */
6 public class FragmentB extends Fragment {
7     @Nullable
8     @Override
9     public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, Bundle savedInstanceState) {
10         View view = inflater.inflate(R.layout.fragment_b, container, attachToRoot: false);
11         return view;
12     }
13 }
```

Terminal 0: Logcat 0: Messages 0: TODO
Gradle build finished in 16s 218ms (54 minutes ago)

Event Log Gradle Console

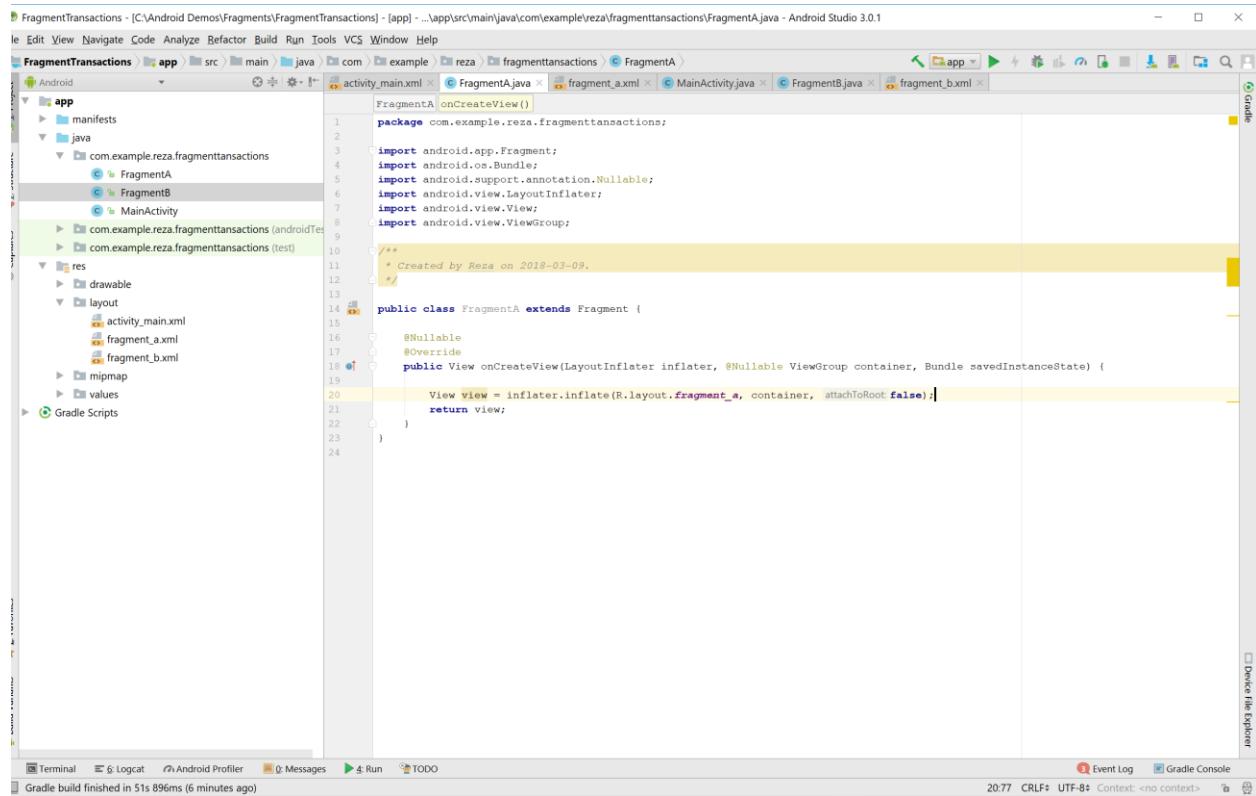
20:57 CRLF: UTF-8 Context: code context

Here is the app running in the emulator at this point:



At this point, add the life cycle methods so we can trace the life cycle method calls during the program execution as shown:

This is how our Fragment.java looks like now:



The screenshot shows the Android Studio interface with the FragmentTransactions project open. The FragmentA.java file is selected in the editor. The code implements the onCreateView() method, which inflates a layout named fragment_a.xml. The code is annotated with a copyright notice: /* * Created by Reza on 2018-03-09. */

```
FragmentA|onCreateView()
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
package com.example.reza.fragmenttransactions;
import android.app.Fragment;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
/*
 * Created by Reza on 2018-03-09.
 */
public class FragmentA extends Fragment {
    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_a, container, false);
        return view;
    }
}
```

Modify the file as shown:

```

FragmentTransactions - [C:\Android Demos\Fragments\FragmentTransactions] - [app] - ...\\app\\src\\main\\java\\com\\example\\reza\\fragmenttransactions\\FragmentA.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
FragmentTransactions app src main java com example reza fragmenttransactions FragmentA
Android manifests java com.example.reza.fragmenttransactions FragmentA FragmentB MainActivity
res drawable layout activity_main.xml fragment_a.xml fragment_b.xml
mipmap values
Gradle Scripts
FragmentA onActivityCreated()
1 package com.example.reza.fragmenttransactions;
2
3 import android.app.Fragment;
4 import android.content.Context;
5 import android.os.Bundle;
6 import android.support.annotation.Nullable;
7 import android.util.Log;
8 import android.view.LayoutInflater;
9 import android.view.View;
10 import android.view.ViewGroup;
11
12 /**
13 * Created by Reza on 2018-03-09.
14 */
15
16 public class FragmentA extends Fragment {
17
18     private static final String TAG = FragmentA.class.getSimpleName();
19
20     @Override
21     public void onAttach(Context context) {
22         super.onAttach(context);
23         Log.i(TAG, msg "onAttach()");
24     }
25
26     @Override
27     public void onCreate(Bundle savedInstanceState) {
28         super.onCreate(savedInstanceState);
29         Log.i(TAG, msg "onCreate()");
30     }
31
32     @Nullable
33     @Override
34     public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, Bundle savedInstanceState) {
35
36         View view = inflater.inflate(R.layout.fragment_a, container, attachToRoot: false);
37
38         Log.i(TAG, msg "onCreateView()");
39         return view;
40     }
41
42     @Override
43     public void onActivityCreated(Bundle savedInstanceState) {
44         super.onActivityCreated(savedInstanceState);
45         Log.i(TAG, msg "onActivityCreated()");
46     }
47
48     @Override
49     public void onStart() {
50         super.onStart();
51         Log.i(TAG, msg "onStart()");
52     }
53
54     @Override
55     public void onResume() {
56         super.onResume();
57         Log.i(TAG, msg "onResume()");
58     }
59
60     @Override
61     public void onPause() {
62         super.onPause();
63         Log.i(TAG, msg "onPause()");
64     }
65
66     @Override
67     public void onStop() {
68         super.onStop();
69         Log.i(TAG, msg "onStop()");
70     }
71
72     @Override
73     public void onDestroyView() {
74         super.onDestroyView();
75         Log.i(TAG, msg "onDestroyView()");
76     }
77
78 }

```

Device File Explorer

Terminal Logcat Android Profiler Messages Run TODO Event Log Gradle Console

Grade build finished in 51s 896ms (16 minutes ago)

44:53 CRLF: UTF-8 Context: <no context>

```

FragmentTransactions - [C:\Android Demos\Fragments\FragmentTransactions] - [app] - ...\\app\\src\\main\\java\\com\\example\\reza\\fragmenttransactions\\FragmentA.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
FragmentTransactions app src main java com example reza fragmenttransactions FragmentA
Android manifests java com.example.reza.fragmenttransactions FragmentA FragmentB MainActivity
res drawable layout activity_main.xml fragment_a.xml fragment_b.xml
mipmap values
Gradle Scripts
FragmentA onActivityCreated()
31
32
33
34     @Override
35     public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, Bundle savedInstanceState) {
36
37         View view = inflater.inflate(R.layout.fragment_a, container, attachToRoot: false);
38
39         Log.i(TAG, msg "onCreateView()");
40         return view;
41     }
42
43     @Override
44     public void onActivityCreated(Bundle savedInstanceState) {
45         super.onActivityCreated(savedInstanceState);
46         Log.i(TAG, msg "onActivityCreated()");
47     }
48
49     @Override
50     public void onStart() {
51         super.onStart();
52         Log.i(TAG, msg "onStart()");
53     }
54
55     @Override
56     public void onResume() {
57         super.onResume();
58         Log.i(TAG, msg "onResume()");
59     }
60
61     @Override
62     public void onPause() {
63         super.onPause();
64         Log.i(TAG, msg "onPause()");
65     }
66
67     @Override
68     public void onStop() {
69         super.onStop();
70         Log.i(TAG, msg "onStop()");
71     }
72
73     @Override
74     public void onDestroyView() {
75         super.onDestroyView();
76         Log.i(TAG, msg "onDestroyView()");
77     }
78
79 }

```

Device File Explorer

Terminal Logcat Android Profiler Messages Run TODO Event Log Gradle Console

Grade build finished in 51s 896ms (16 minutes ago)

44:53 CRLF: UTF-8 Context: <no context>

The screenshot shows the Android Studio interface with the FragmentTransactions project open. The code editor displays FragmentA.java, which contains the following lifecycle methods:

```
FragmentA.onCreate()
    Log.i(TAG, msg: "onStart()");

@Override
public void onResume() {
    super.onResume();
    Log.i(TAG, msg: "onResume()");

}

@Override
public void onPause() {
    super.onPause();
    Log.i(TAG, msg: "onPause()");

}

@Override
public void onStop() {
    super.onStop();
    Log.i(TAG, msg: "onStop()");

}

@Override
public void onDestroyView() {
    super.onDestroyView();
    Log.i(TAG, msg: "onDestroyView()");

}

@Override
public void onDestroy() {
    super.onDestroy();
    Log.i(TAG, msg: "onDestroy()");

}

@Override
public void onDetach() {
    super.onDetach();
    Log.i(TAG, msg: "onDetach()");

}
```

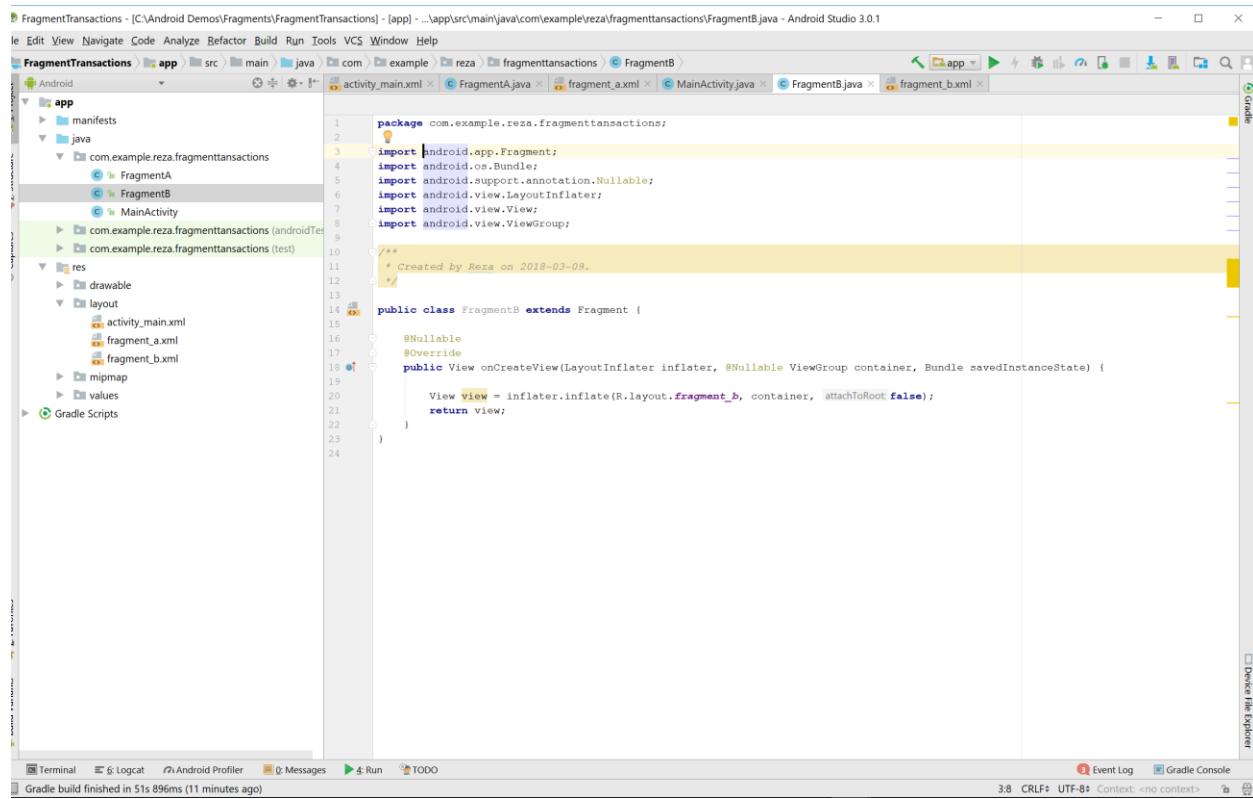
The code editor shows several red error markers (exclamation points) scattered throughout the code, indicating syntax or compilation errors.

Now do the same for FragmentB.java as shown.

Note that methods are identical except that instead of Log.i we are using Log.e so for FragmentB lifecycle methods, the color would be red (e is for error).

It is just to make it easier to distinguish between the two fragments' life cycle methods.

Here is how FragmentB.java looks like now:



```
package com.example.reza.fragmenttransactions;
import android.app.Fragment;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

/**
 * Created by Reza on 2018-03-09.
 */
public class FragmentB extends Fragment {

    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_b, container, false);
        return view;
    }
}
```

Change it as shown:

Notice the red words as I am modifying the code. To add the import statement (the lack of the import statement is causing the error and the red color) select the word and press Alt+Enter and Android Studio adds the required import statement hence adding the library:

The screenshot shows the Android Studio interface with the FragmentTransactions project open. The FragmentB.java file is the active editor. The code defines a FragmentB class that extends Fragment. It includes overridden methods for onAttach, onCreate, and onCreateView. The onCreateView method inflates a layout from R.layout.fragment_b. The code uses annotations like @Override, @Nullable, and @NonNull. A yellow highlight covers the entire code block.

```
FragmentTransactions - [C:\Android Demos\Fragments\FragmentTransactions] - [app] - ...\\app\\src\\main\\java\\com\\example\\reza\\fragmenttransactions\\FragmentB.java - Android Studio 3.0.1

package com.example.reza.fragmenttransactions;

import android.app.Fragment;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

/**
 * Created by Reza on 2018-03-09.
 */

public class FragmentB extends Fragment {

    private static final String TAG = FragmentB.class.getSimpleName();

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        Log.e(TAG, "onAttach()");
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Log.e(TAG, "onCreate()");
    }

    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_b, container, false);
        return view;
    }
}
```

This screenshot is similar to the first one, but it shows a code completion tooltip for the 'Context' parameter in the onAttach method. The tooltip displays the fully qualified class name 'android.content.Context'. The code completion feature is being used to resolve the symbol 'Context'.

```
FragmentTransactions - [C:\Android Demos\Fragments\FragmentTransactions] - [app] - ...\\app\\src\\main\\java\\com\\example\\reza\\fragmenttransactions\\FragmentB.java - Android Studio 3.0.1

package com.example.reza.fragmenttransactions;

import android.app.Fragment;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

/**
 * Created by Reza on 2018-03-09.
 */

public class FragmentB extends Fragment {

    private static final String TAG = FragmentB.class.getSimpleName();

    @Override
    public void onAttach(@android.content.Context context) {
        super.onAttach(context);
        Log.e(TAG, "onAttach()");
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Log.e(TAG, "onCreate()");
    }

    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_b, container, false);
        return view;
    }
}
```

FragmentTransactions - [C:\Android Demos\Fragments\FragmentTransactions] - [app] - ...\\app\\src\\main\\java\\com\\example\\reza\\fragmenttransactions\\FragmentB.java - Android Studio 3.0.1

```
FragmentB.onAttach()
package com.example.reza.fragmenttransactions;

import android.app.Fragment;
import android.content.Context;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

/**
 * Created by Reza on 2018-03-09.
 */

public class FragmentB extends Fragment {

    private static final String TAG = FragmentB.class.getSimpleName();

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        Log.e(TAG, "onAttach()");
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Log.e(TAG, "onCreate()");
    }

    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_b, container, attachToRoot: false);
        return view;
    }
}
```

FragmentTransactions - [C:\Android Demos\Fragments\FragmentTransactions] - [app] - ...\\app\\src\\main\\java\\com\\example\\reza\\fragmenttransactions\\FragmentB.java - Android Studio 3.0.1

```
FragmentB.onAttach()
package com.example.reza.fragmenttransactions;

import android.app.Fragment;
import android.content.Context;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

/**
 * Created by Reza on 2018-03-09.
 */

public class FragmentB extends Fragment {

    private static final String TAG = FragmentB.class.getSimpleName();

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        Log.e(TAG, msg "onAttach()");
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Log.e(TAG, msg "onCreate()");
    }

    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_b, container, attachToRoot: false);
        return view;
    }
}
```

Continue to add the life cycle methods. The following shows the FragmentB.java file in its entirety once it has been completed:

FragmentTransactions - [C:\Android Demos\Fragments\FragmentTransactions] - [app] - ...\\app\\src\\main\\java\\com\\example\\reza\\fragmenttransactions\\FragmentB.java - Android Studio 3.0.1

```

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
FragmentTransactions > app > src > main > java > com > example > reza > fragmenttransactions > FragmentB
activity_main.xml < FragmentA.java < fragment_a.xml < MainActivity.java < FragmentB.java < fragment_b.xml <
FragmentB
1 package com.example.reza.fragmenttransactions;
2
3 import android.app.Fragment;
4 import android.content.Context;
5 import android.os.Bundle;
6 import android.support.annotation.Nullable;
7 import android.util.Log;
8 import android.view.LayoutInflater;
9 import android.view.View;
10 import android.view.ViewGroup;
11
12 /**
13 * Created by Reza on 2018-03-09.
14 */
15
16 public class FragmentB extends Fragment {
17
18     private static final String TAG = FragmentB.class.getSimpleName();
19
20     @Override
21     public void onAttach(Context context) {
22         super.onAttach(context);
23         Log.e(TAG, msg: "onAttach()");
24     }
25
26     @Override
27     public void onCreate(Bundle savedInstanceState) {
28         super.onCreate(savedInstanceState);
29         Log.e(TAG, msg: "onCreate()");
30     }
31
32     @Nullable
33     @Override
34     public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, Bundle savedInstanceState) {
35
36         View view = inflater.inflate(R.layout.fragment_b, container, attachToRoot false);
37         Log.e(TAG, msg: "onCreateView()");
38
39         return view;
40     }
41     @Override
42     public void onActivityCreated(Bundle savedInstanceState) {
43         super.onActivityCreated(savedInstanceState);
44         Log.e(TAG, msg: "onActivityCreated()");
45     }
46
47     @Override
48     public void onStart() {
49         super.onStart();
50         Log.e(TAG, msg: "onStart()");
51     }
52
53     @Override
54     public void onResume() {
55         super.onResume();
56         Log.e(TAG, msg: "onResume()");
57     }
58
59     @Override
60     public void onPause() {
61         super.onPause();
62         Log.e(TAG, msg: "onPause()");
63     }
64
65     @Override
66     public void onStop() {
67         super.onStop();
68         Log.e(TAG, msg: "onStop()");
69     }
70
71     @Override
72     public void onDestroyView() {
73         super.onDestroyView();
74         Log.e(TAG, msg: "onDestroyView()");
75     }

```

Terminal Logcat Android Profiler Messages Run TODO Event Log Gradle Console

Gradle build finished in 4s 422ms (14 minutes ago) 81.6 CRLF: UTF-8 Context: <no context>

FragmentTransactions - [C:\Android Demos\Fragments\FragmentTransactions] - [app] - ...\\app\\src\\main\\java\\com\\example\\reza\\fragmenttransactions\\FragmentB.java - Android Studio 3.0.1

```

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
FragmentTransactions > app > src > main > java > com > example > reza > fragmenttransactions > FragmentB
activity_main.xml < FragmentA.java < fragment_a.xml < MainActivity.java < FragmentB.java < fragment_b.xml <
FragmentB
31
32     @Nullable
33     @Override
34     public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, Bundle savedInstanceState) {
35
36         View view = inflater.inflate(R.layout.fragment_b, container, attachToRoot false);
37         Log.e(TAG, msg: "onCreateView()");
38
39         return view;
40     }
41     @Override
42     public void onActivityCreated(Bundle savedInstanceState) {
43         super.onActivityCreated(savedInstanceState);
44         Log.e(TAG, msg: "onActivityCreated()");
45     }
46
47     @Override
48     public void onStart() {
49         super.onStart();
50         Log.e(TAG, msg: "onStart()");
51     }
52
53     @Override
54     public void onResume() {
55         super.onResume();
56         Log.e(TAG, msg: "onResume()");
57     }
58
59     @Override
60     public void onPause() {
61         super.onPause();
62         Log.e(TAG, msg: "onPause()");
63     }
64
65     @Override
66     public void onStop() {
67         super.onStop();
68         Log.e(TAG, msg: "onStop()");
69     }
70
71     @Override
72     public void onDestroyView() {
73         super.onDestroyView();
74         Log.e(TAG, msg: "onDestroyView()");
75     }

```

Terminal Logcat Android Profiler Messages Run TODO Event Log Gradle Console

Gradle build finished in 4s 422ms (14 minutes ago) 81.6 CRLF: UTF-8 Context: <no context>

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left sidebar shows the project structure under the "app" module. It includes "Manifests", "java" (containing FragmentA, FragmentB, and MainActivity), "res" (with drawable, layout, and values folders), and "Gradle Scripts".
- Code Editor:** The main editor window displays the Java code for FragmentB. The code is annotated with Log.e statements to track lifecycle events.
- Toolbars and Menus:** The top menu bar includes File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, and Help. The toolbar below the menu has icons for Terminal, Logcat, Android Profiler, Messages, Run, and TODO.
- Bottom Status Bar:** The status bar at the bottom indicates "Gradle build finished in 4s 422ms (15 minutes ago)" and shows the current file path: "FragmentTransactions - [C:\Android Demos\Fragments\FragmentTransactions] - [app] - ...\\app\\src\\main\\java\\com\\example\\reza\\fragmenttransactions\\FragmentB.java - Android Studio 3.0.1".

```
public void onStart() {
    super.onStart();
    Log.e(TAG, msg: "onStart()");
}

@Override
public void onResume() {
    super.onResume();
    Log.e(TAG, msg: "onResume()");
}

@Override
public void onPause() {
    super.onPause();
    Log.e(TAG, msg: "onPause()");
}

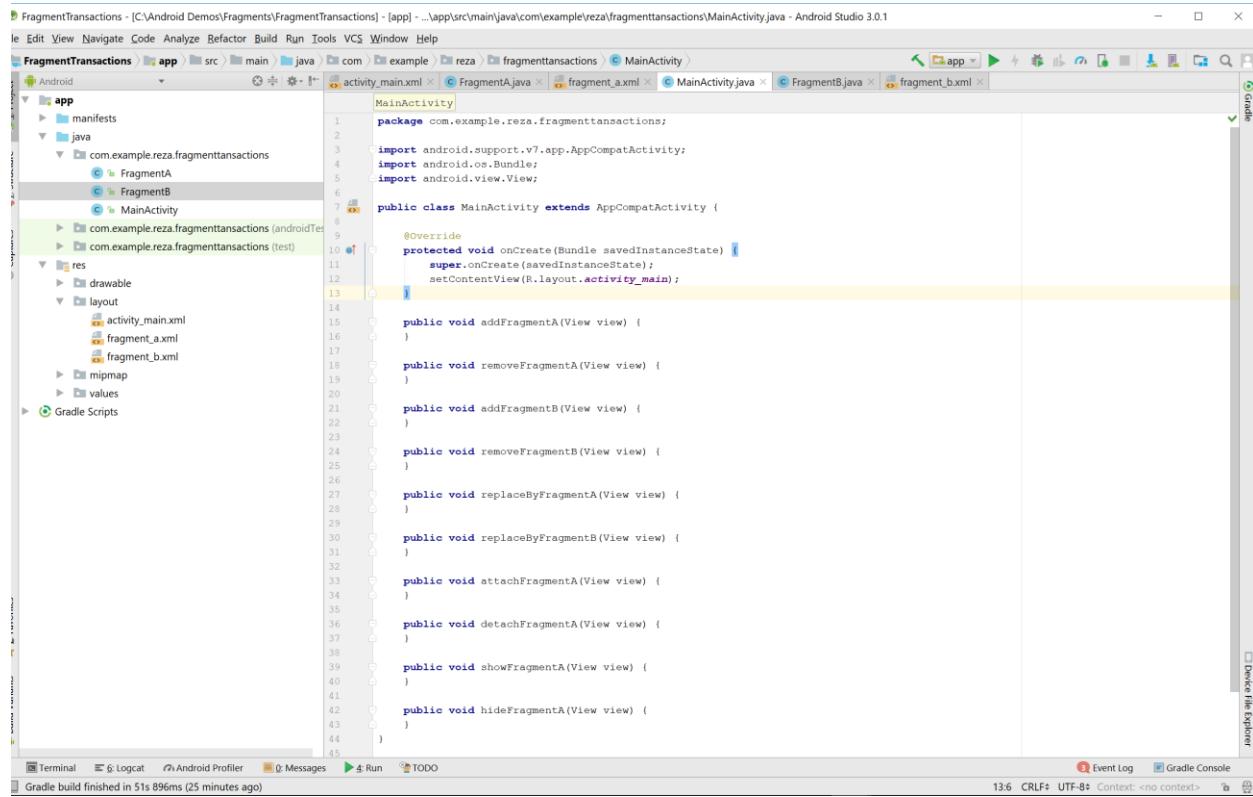
@Override
public void onStop() {
    super.onStop();
    Log.e(TAG, msg: "onStop()");
}

@Override
public void onDestroyView() {
    super.onDestroyView();
    Log.e(TAG, msg: "onDestroyView()");
}

@Override
public void onDestroy() {
    super.onDestroy();
    Log.e(TAG, msg: "onDestroy()");
}

@Override
public void onDetach() {
    super.onDetach();
    Log.e(TAG, msg: "onDetach()");
}
```

And here is the code for our MainActivity.java :



The screenshot shows the Android Studio interface with the file 'MainActivity.java' open in the editor. The code implements an AppCompatActivity with several methods for managing fragments:

```
package com.example.reza.fragmenttransactions;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class MainActivity extends AppCompatActivity {

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

    public void addFragmentA(View view) {
    }

    public void removeFragmentA(View view) {
    }

    public void addFragmentB(View view) {
    }

    public void removeFragmentB(View view) {
    }

    public void replaceByFragmentA(View view) {
    }

    public void replaceByFragmentB(View view) {
    }

    public void attachFragmentA(View view) {
    }

    public void detachFragmentA(View view) {
    }

    public void showFragmentA(View view) {
    }

    public void hideFragmentA(View view) {
    }
}
```

The code editor shows syntax highlighting for Java and XML files. The bottom status bar indicates a successful Gradle build.

At this point we can start writing our code for the click events of our buttons that invoke different fragment transaction methods.

Let's start with addFragmentA() button code:

Now when we want to add a fragment, these are the steps that we need to perform:

Create a new Fragment

Get the id of the container layout (inside which we are going to add our fragment)

Use add() method to add the fragment to our container layout

Now going back to MainActivity.java file and going back to addFragmentA() method, the first thing we need to do is to get our FragmentManger object since we are going to create and add our fragment programmatically (the second method we described in the previous demos) so add the following code for addFragmentA() method:

The screenshot shows the Android Studio interface with the project 'FragmentTransactions' open. The left sidebar shows the project structure with files like activity_main.xml, FragmentA.java, fragment_a.xml, and MainActivity.java. The main editor window displays the MainActivity.java code. A yellow highlight is applied to the line of code where the transaction is committed:

```

public void addFragmentA(View view) {
    FragmentA fragmentA = new FragmentA();
    FragmentManager manager = getFragmentManager();
    FragmentTransaction transaction = manager.beginTransaction();
    transaction.add(R.id.container, fragmentA, "fragA");
    transaction.commit();
}

```

Here is the code with a larger font. Note that “container” is the id of our FrameLayout in the activity_main.xml file and in that frame we will be displaying our fragment.

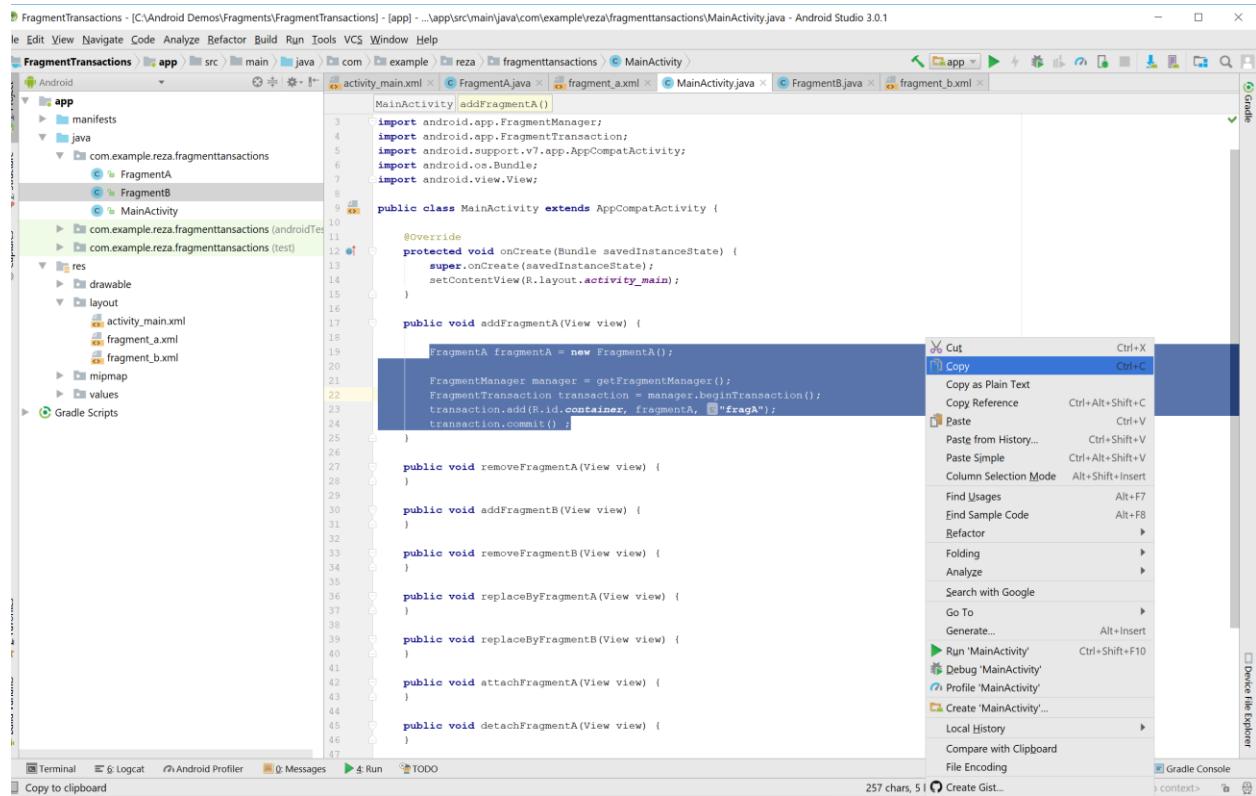
Also “fragA” is the string (the tag) we gave to this transaction so we can find our fragment later on. The code below creates an object of FragmentA class and with the help of the transaction object, adds this fragment to the host container (the frame layout here) which has the id “container”:

```

public void addFragmentA(View view) {
    FragmentA fragmentA = new FragmentA();
    FragmentManager manager = getFragmentManager();
    FragmentTransaction transaction = manager.beginTransaction();
    transaction.add(R.id.container, fragmentA, "fragA");
    transaction.commit();
}

```

Now to implement addFragmentB, all you have to do is to copy and paste this code in the body of addFragmentB() method and change A to B as shown:



```
FragmentTransactions - [C:\Android Demos\Fragments\FragmentTransactions] - [app] - ...\\app\\src\\main\\java\\com\\example\\reza\\fragmenttransactions\\MainActivity.java - Android Studio 3.0.1

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
FragmentTransactions app src main java com.example.reza.fragmenttransactions MainActivity
Android manifests java com.example.reza.fragmenttransactions FragmentA FragmentB MainActivity
APP com.example.reza.fragmenttransactions (androidTest) com.example.reza.fragmenttransactions (test)
res drawable layout activity_main.xml fragment_a.xml fragment_b.xml mipmap values
Gradle Scripts

MainActivity addFragmentA()
import android.app.FragmentManager;
import android.app.FragmentTransaction;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class MainActivity extends AppCompatActivity {

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

    public void addFragmentA(View view) {
        FragmentA fragmentA = new FragmentA();
        FragmentManager manager = getFragmentManager();
        FragmentTransaction transaction = manager.beginTransaction();
        transaction.add(R.id.container, fragmentA, "fragA");
        transaction.commit();
    }

    public void removeFragmentA(View view) {
    }

    public void addFragmentB(View view) {
    }

    public void removeFragmentB(View view) {
    }

    public void replaceByFragmentA(View view) {
    }

    public void replaceByFragmentB(View view) {
    }

    public void attachFragmentA(View view) {
    }

    public void detachFragmentA(View view) {
    }
}

25 chars, 51 Create Gist... 257 chars, 51 Create Gist...
Copy to clipboard
```

The screenshot shows the Android Studio interface with the code editor open to MainActivity.java. A context menu is displayed over the code, with the 'Copy' option highlighted. The code itself contains methods for adding, removing, and replacing fragments A and B.

FragmentTransactions - [C:\Android Demos\Fragments\FragmentTransactions] - [app] - ...\\app\\src\\main\\java\\com\\example\\reza\\fragmenttransactions\\MainActivity.java - Android Studio 3.0.1

```

    import android.app.FragmentManager;
    import android.app.FragmentTransaction;
    import android.support.v7.app.AppCompatActivity;
    import android.os.Bundle;
    import android.view.View;

    public class MainActivity extends AppCompatActivity {

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

        public void addFragmentA(View view) {
            FragmentA fragmentA = new FragmentA();

            FragmentManager manager = getFragmentManager();
            FragmentTransaction transaction = manager.beginTransaction();
            transaction.add(R.id.container, fragmentA, "fragA");
            transaction.commit();
        }

        public void removeFragmentA(View view) {
        }

        public void addFragmentB(View view) {
        }

        public void removeFragmentB(View view) {
        }

        public void replaceByFragmentA(View view) {
        }

        public void replaceByFragmentB(View view) {
        }
    }

```

Paste from clipboard

Event Log Gradle Console

FragmentTransactions - [C:\Android Demos\Fragments\FragmentTransactions] - [app] - ...\\app\\src\\main\\java\\com\\example\\reza\\fragmenttransactions\\MainActivity.java - Android Studio 3.0.1

```

    import android.app.FragmentManager;
    import android.app.FragmentTransaction;
    import android.support.v7.app.AppCompatActivity;
    import android.os.Bundle;
    import android.view.View;

    public class MainActivity extends AppCompatActivity {

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

        public void addFragmentA(View view) {
            FragmentA fragmentA = new FragmentA();

            FragmentManager manager = getFragmentManager();
            FragmentTransaction transaction = manager.beginTransaction();
            transaction.add(R.id.container, fragmentA, "fragA");
            transaction.commit();
        }

        public void removeFragmentA(View view) {
        }

        public void addFragmentB(View view) {
            FragmentB fragmentB = new FragmentB();

            FragmentManager manager = getFragmentManager();
            FragmentTransaction transaction = manager.beginTransaction();
            transaction.add(R.id.container, fragmentB, "fragB");
            transaction.commit();
        }

        public void removeFragmentB(View view) {
        }

        public void replaceByFragmentA(View view) {
        }

        public void replaceByFragmentB(View view) {
        }
    }

```

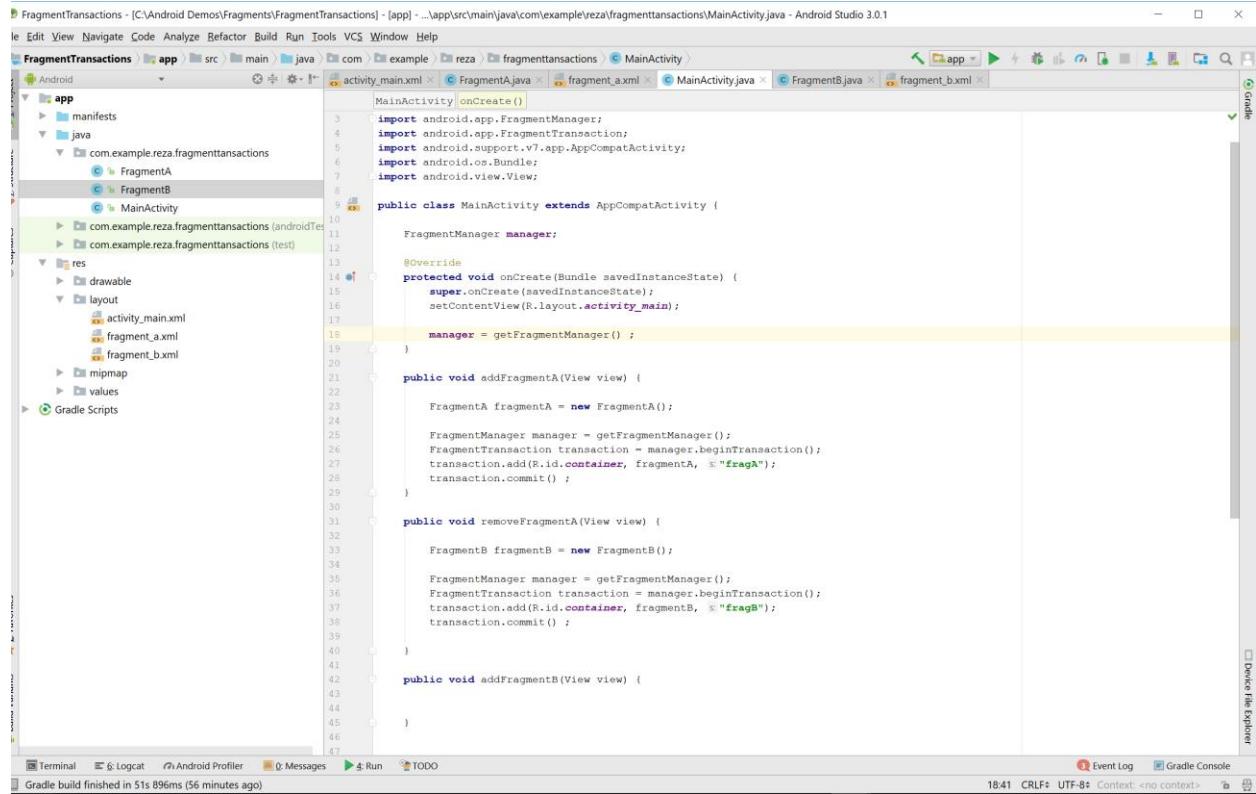
Grade build finished in 51s 896ms (today 12:59 PM)

Event Log Gradle Console

Here is the code for this two methods with a larger font:

```
public void addFragmentA(View view) {  
  
    FragmentA fragmentA = new FragmentA();  
  
    FragmentManager manager = getFragmentManager();  
    FragmentTransaction transaction = manager.beginTransaction();  
    transaction.add(R.id.container, fragmentA, s: "fragA");  
    transaction.commit();  
}  
  
public void removeFragmentA(View view) {  
  
}  
  
public void addFragmentB(View view) {  
    FragmentB fragmentB = new FragmentB();  
  
    FragmentManager manager = getFragmentManager();  
    FragmentTransaction transaction = manager.beginTransaction();  
    transaction.add(R.id.container, fragmentB, s: "fragB");  
    transaction.commit();  
}
```

Since the line that creates a FragmentManager object is going to be needed again and again, we can remove it and make it a class field as shown:



The screenshot shows the Android Studio interface with the project 'FragmentTransactions' open. The code editor displays the MainActivity.java file. The code has been modified to move the creation of the FragmentManager object into a class field named 'manager'. The code is as follows:

```
import android.app.FragmentManager;
import android.app.FragmentManager;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class MainActivity extends AppCompatActivity {

    FragmentManager manager;

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

        manager = getFragmentManager();
    }

    public void addFragmentA(View view) {
        FragmentA fragmentA = new FragmentA();

        FragmentManager manager = getFragmentManager();
        FragmentTransaction transaction = manager.beginTransaction();
        transaction.add(R.id.container, fragmentA, "fragA");
        transaction.commit();
    }

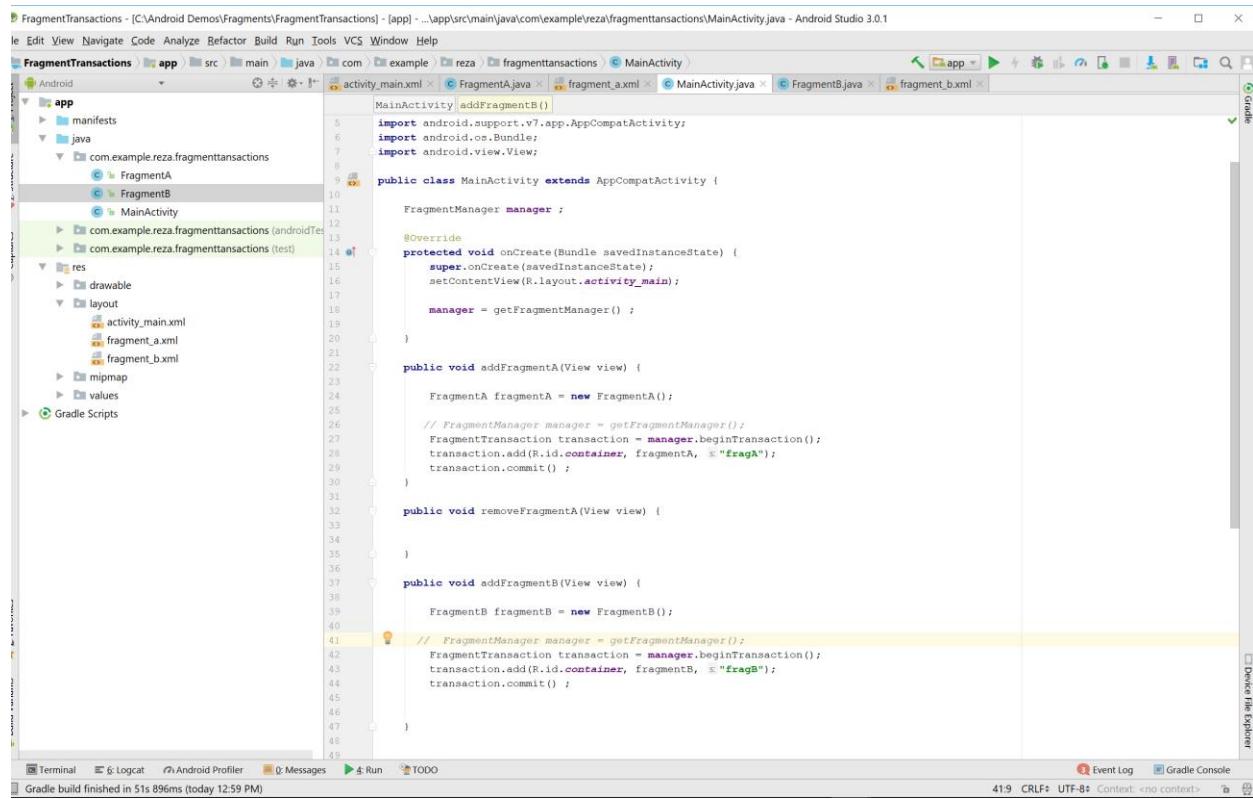
    public void removeFragmentA(View view) {
        FragmentB fragmentB = new FragmentB();

        FragmentManager manager = getFragmentManager();
        FragmentTransaction transaction = manager.beginTransaction();
        transaction.add(R.id.container, fragmentB, "fragB");
        transaction.commit();
    }

    public void addFragmentB(View view) {
    }
}
```

The code editor shows several tabs at the top: activity_main.xml, FragmentA.java, fragment_a.xml, MainActivity.java, FragmentB.java, and fragment_b.xml. The Java code is the active tab. The bottom of the screen shows the Android Studio navigation bar with icons for Terminal, Logcat, Android Profiler, Messages, Run, and TODO.

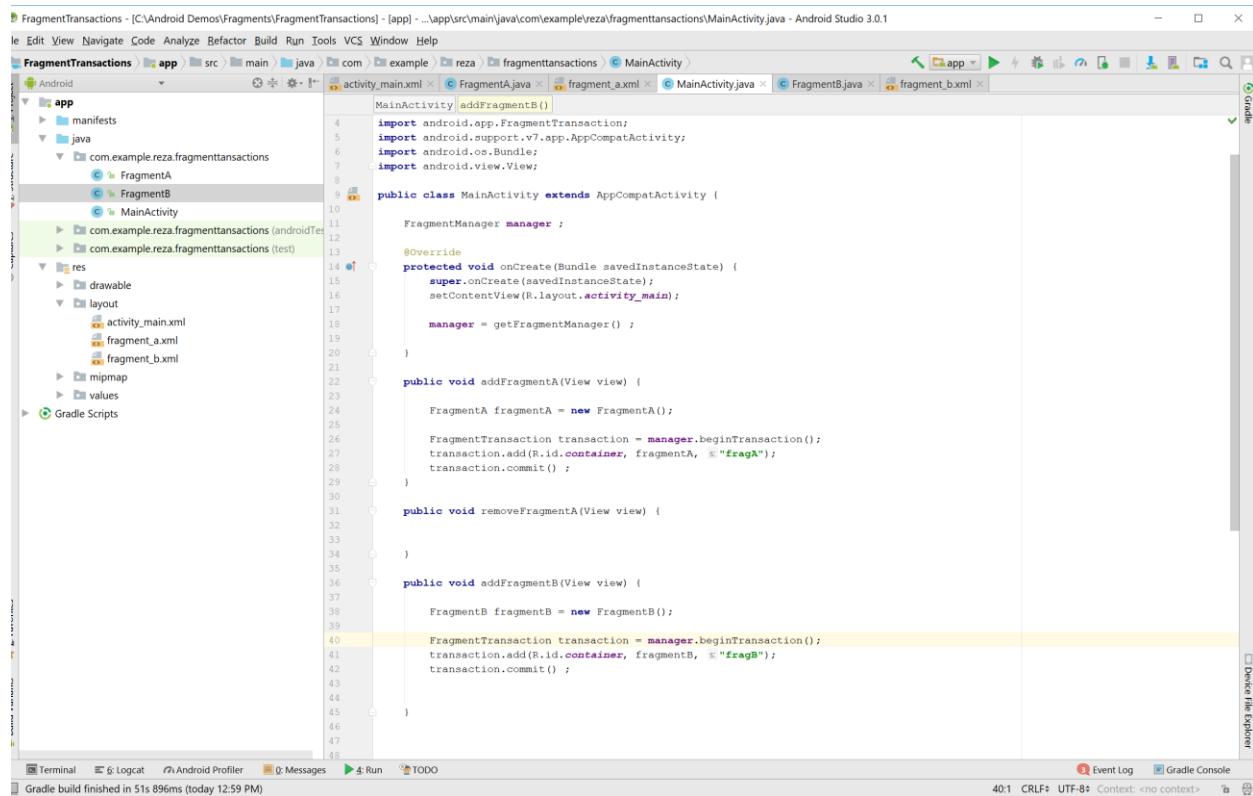
Lines 11 and 18 were added and we don't need the commented lines anymore:



The screenshot shows the Android Studio interface with the project 'FragmentTransactions' open. The code editor displays the MainActivity.java file. The code implements methods for adding and removing two fragments, A and B, from a container view. The original code included commented-out lines for fragment B, which have now been removed.

```
public class MainActivity extends AppCompatActivity {
    FragmentManager manager;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        manager = getSupportFragmentManager();
    }
    public void addFragmentA(View view) {
        FragmentA fragmentA = new FragmentA();
        FragmentTransaction transaction = manager.beginTransaction();
        transaction.add(R.id.container, fragmentA, "fragA");
        transaction.commit();
    }
    public void removeFragmentA(View view) {
    }
    public void addFragmentB(View view) {
        FragmentB fragmentB = new FragmentB();
        // FragmentManager manager = getSupportFragmentManager();
        FragmentTransaction transaction = manager.beginTransaction();
        transaction.add(R.id.container, fragmentB, "fragB");
        transaction.commit();
    }
}
```

Now we can remove these lines from our methods so the modified code for the methods will be as shown:



The screenshot shows the Android Studio interface with the project 'FragmentTransactions' open. The code editor displays the MainActivity.java file. The code is as follows:

```
import android.app.FragmentTransaction;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class MainActivity extends AppCompatActivity {

    FragmentManager manager;

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

        manager = getSupportFragmentManager();
    }

    public void addFragmentA(View view) {
        FragmentA fragmentA = new FragmentA();

        FragmentTransaction transaction = manager.beginTransaction();
        transaction.add(R.id.container, fragmentA, "fragA");
        transaction.commit();
    }

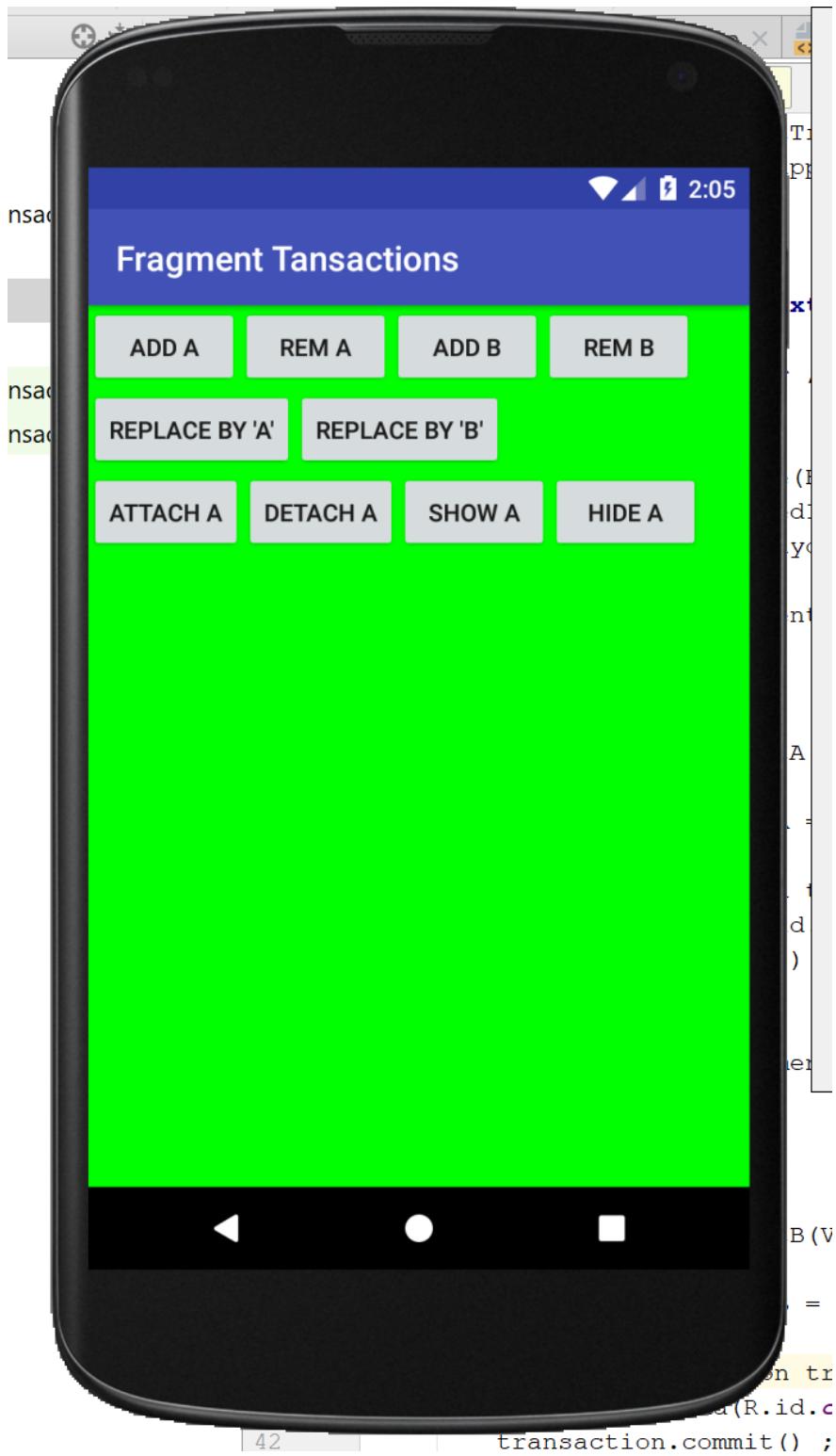
    public void removeFragmentA(View view) {
    }

    public void addFragmentB(View view) {
        FragmentB fragmentB = new FragmentB();

        FragmentTransaction transaction = manager.beginTransaction();
        transaction.add(R.id.container, fragmentB, "fragB");
        transaction.commit();
    }
}
```

The code editor highlights the lines that were removed in yellow. The bottom status bar shows 'Gradle build finished in 51s 896ms (today 12:59 PM)'.

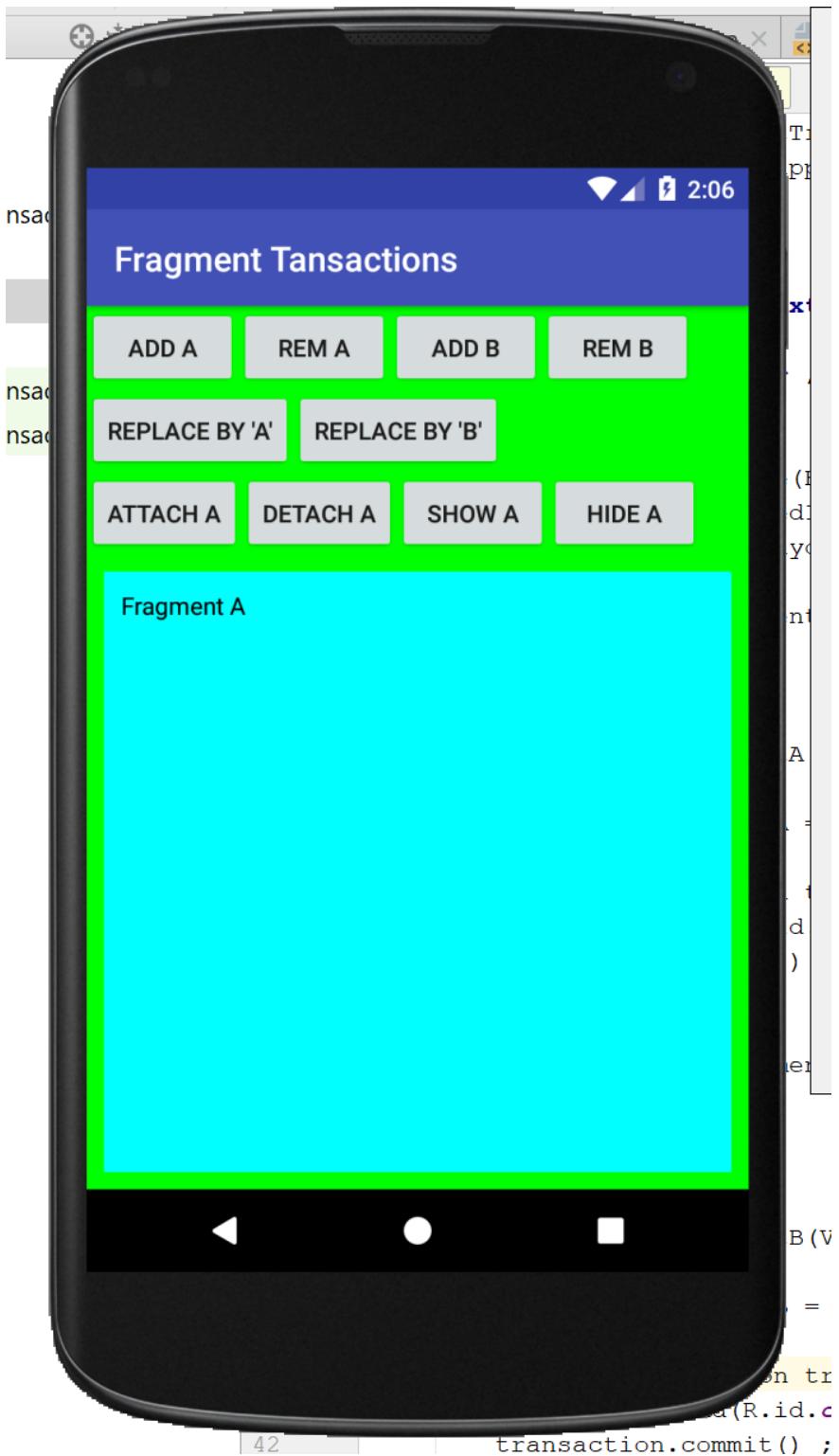
At this point we can run our app to test it. Here is our application up and running:



```
        transaction.commit() ;
```

42

Click the “ADD A” button and you’ll see the fragment is added to the container (frame layout here) as shown:



```
| 42 |           transaction.commit();
```

Now examine the life cycle methods that were executed inside our Logcat window (use the word “Fragment” as the filter as shown):

The screenshot shows the Android Studio interface. The top part displays the code for `MainActivity.java`, which contains logic for adding and removing `FragmentA`. The bottom part shows the `Logcat` window with the following log entries:

```

03-09 14:05:29.444 8048-8048/com.example.reza.fragmenttransactions I/FragmentA: onAttach()
03-09 14:05:29.445 8048-8048/com.example.reza.fragmenttransactions I/FragmentA: onCreate()
03-09 14:05:29.451 8048-8048/com.example.reza.fragmenttransactions I/FragmentA: onCreateView()
03-09 14:05:29.451 8048-8048/com.example.reza.fragmenttransactions I/FragmentA: onActivityCreated()
03-09 14:05:29.451 8048-8048/com.example.reza.fragmenttransactions I/FragmentA: onStart()
03-09 14:05:29.451 8048-8048/com.example.reza.fragmenttransactions I/FragmentA: onResume()

```

Now click on “ADD B” method. You’ll see that Fragment B is added to our container as shown:



And here are the life cycle methods called for Fragment B (they appear in red because of using Log.e instead of Log.i)

The screenshot shows the Android Studio interface. The top navigation bar includes File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, and Help. Below the toolbar, the project structure is visible with packages app, com.example.reza.fragmenttransactions, and com.example.reza.fragmenttransactions (androidTest). The FragmentB.java file is open, showing its code:

```
48     public void onStart() {
49         super.onStart();
50         Log.e(TAG, msg: "onStart()");
51     }
52
53     @Override
54     public void onResume() {
55         super.onResume();
56         Log.e(TAG, msg: "onResume()");
57     }
58
59     @Override
60     public void onPause() {
61         super.onPause();
62         Log.e(TAG, msg: "onPause()");
63     }
64
65     @Override
```

The Logcat tab at the bottom displays log messages from the Emulator Nexus_4 API_25. The messages show the execution of FragmentB's lifecycle methods (onStart, onResume, onPause) and the corresponding Log.e statements. The Logcat interface includes filters for Verbose, Fragment, Regex, and Show only selected application, along with a Device File Explorer sidebar.

Notice that `onDestory()` method was not called for fragment A, another word fragment A is still there but it has been covered by fragment B and it is really behind fragment B.

So inside our main activity we added Fragment A and then we added Fragment B which covered Fragment A.

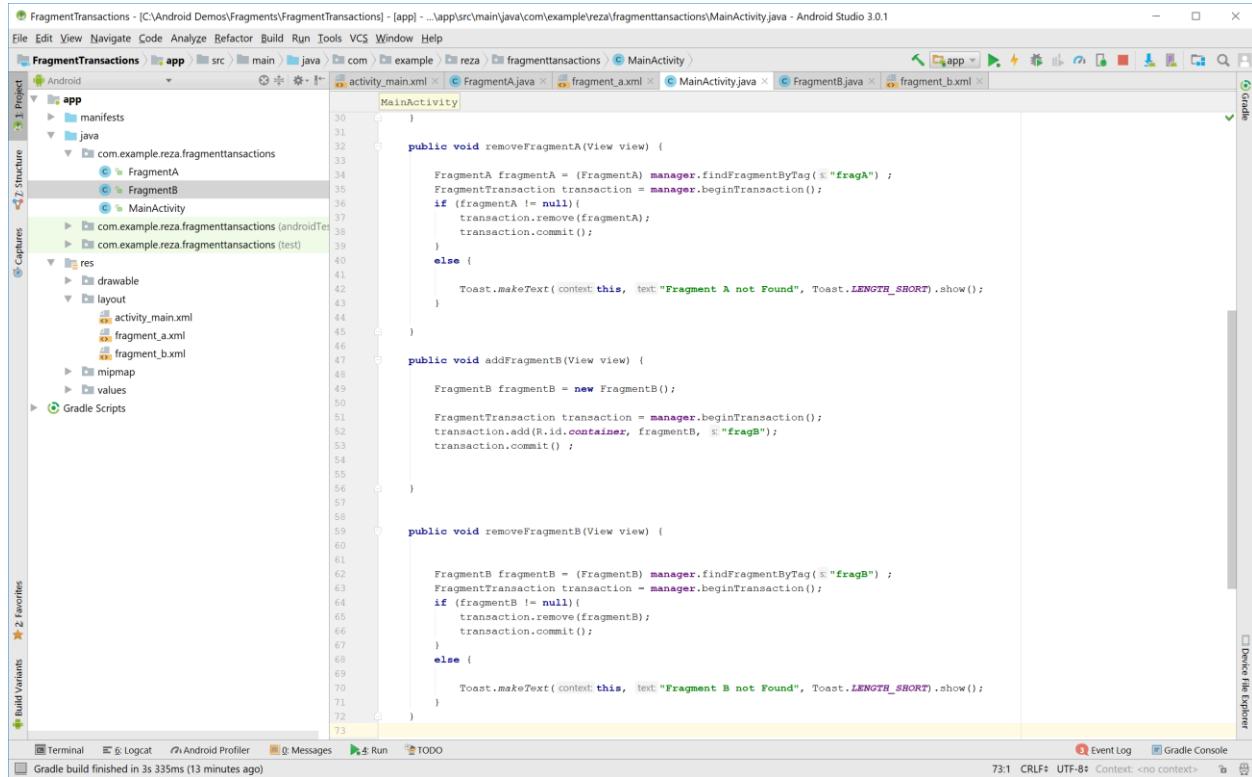
We can write the code for `remove()` methods so we can remove fragment B and see Fragment A again.

Now the steps to remove a fragment which has been already added to an activity we need to do the following:

Get the reference to the fragment which is going to be removed

Then use `remove()` transaction method to remove the fragment which is going to call `onDestory()` method which destroys the fragment. So once you remove a fragment, the fragment is destroyed and is detached from the host activity.

Now add the following code to the removeFragmentA() and removeFragmentB() methods to perform the remove transaction as shown:



The screenshot shows the Android Studio interface with the project 'FragmentTransactions' open. The code editor displays the MainActivity.java file, which contains the following Java code:

```
public void removeFragmentA(View view) {
    FragmentA fragmentA = (FragmentA) manager.findFragmentByTag("fragA");
    FragmentTransaction transaction = manager.beginTransaction();
    if (fragmentA != null) {
        transaction.remove(fragmentA);
        transaction.commit();
    } else {
        Toast.makeText(context, "Fragment A not Found", Toast.LENGTH_SHORT).show();
    }
}

public void addFragmentB(View view) {
    FragmentB fragmentB = new FragmentB();

    FragmentTransaction transaction = manager.beginTransaction();
    transaction.add(R.id.container, fragmentB, "fragB");
    transaction.commit();
}

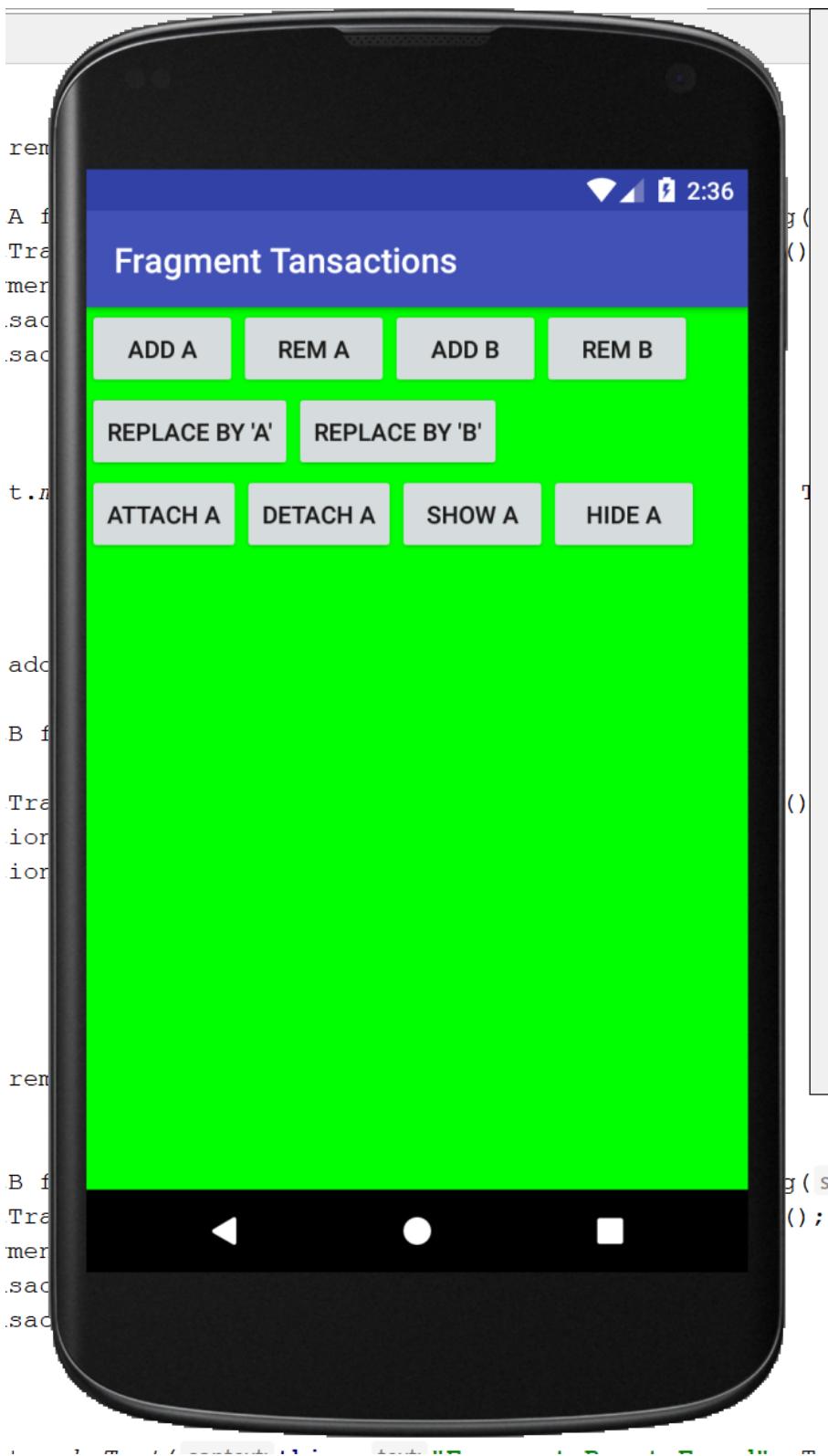
public void removeFragmentB(View view) {
    FragmentB fragmentB = (FragmentB) manager.findFragmentByTag("fragB");
    FragmentTransaction transaction = manager.beginTransaction();
    if (fragmentB != null) {
        transaction.remove(fragmentB);
        transaction.commit();
    } else {
        Toast.makeText(context, "Fragment B not Found", Toast.LENGTH_SHORT).show();
    }
}
```

The code implements three methods: removeFragmentA(), addFragmentB(), and removeFragmentB(). The remove methods use a FragmentTransaction to remove a fragment by its tag ('fragA' or 'fragB'). The add method adds a new FragmentB to the container with tag 'fragB'. Toaster messages are used to inform the user if a fragment is not found.

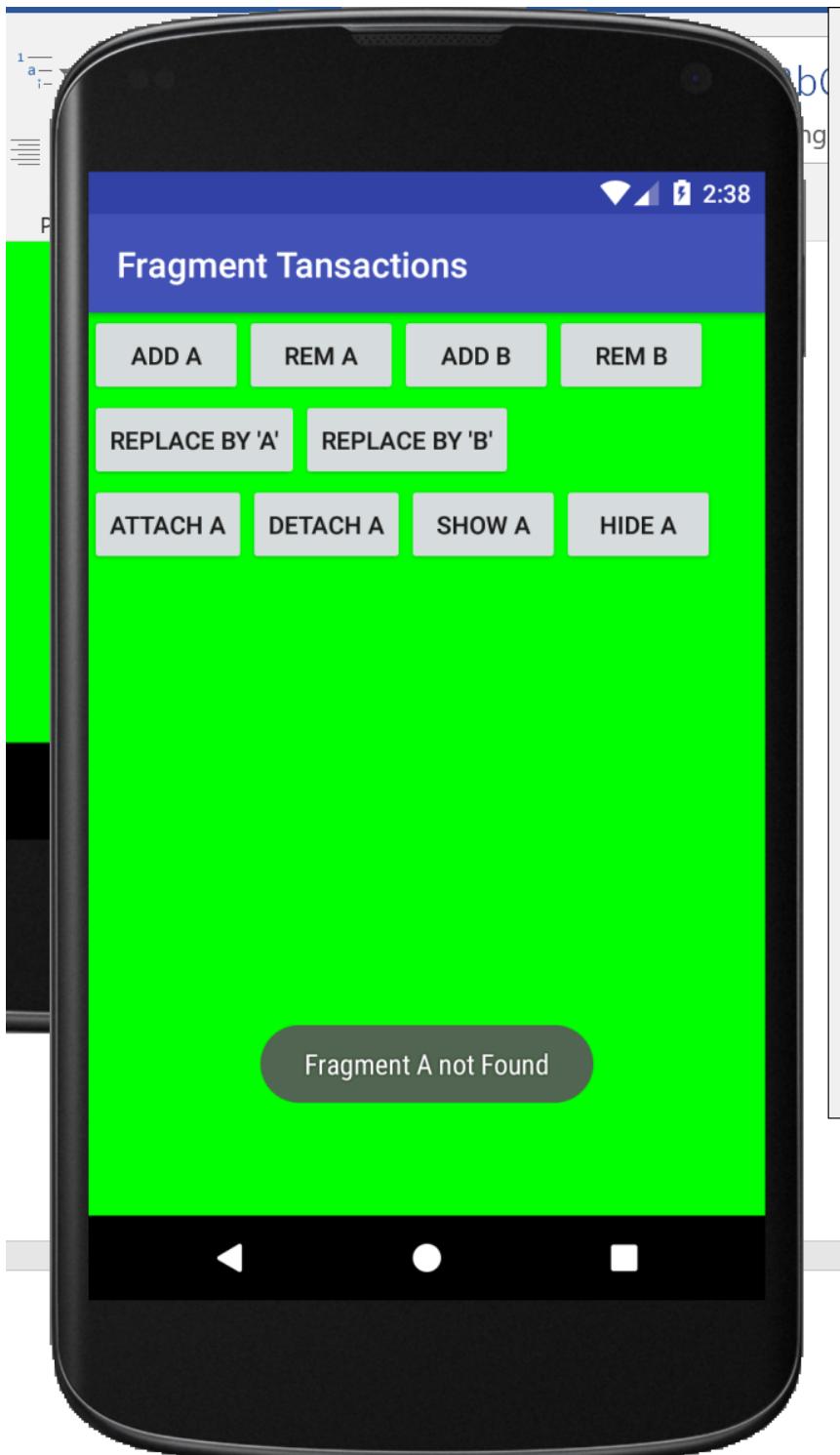
Here is the code with a larger font. Note how we use the tag we assigned to Fragment A (“fragA”) to find it using “findFragmentByTag() method” and note we check to make sure if Fragment A has been added (it is not null) otherwise we will get a null exception. The code for RemoveFragmentB() is identical except for changing A to B:

```
public void removeFragmentA(View view) {  
  
    FragmentA fragmentA = (FragmentA) manager.findFragmentByTag("fragA") ;  
    FragmentTransaction transaction = manager.beginTransaction();  
    if (fragmentA != null){  
        transaction.remove(fragmentA);  
        transaction.commit();  
    }  
    else {  
  
        Toast.makeText(context: this, text: "Fragment A not Found", Toast.LENGTH_SHORT).show();  
    }  
}
```

Now run the app again:



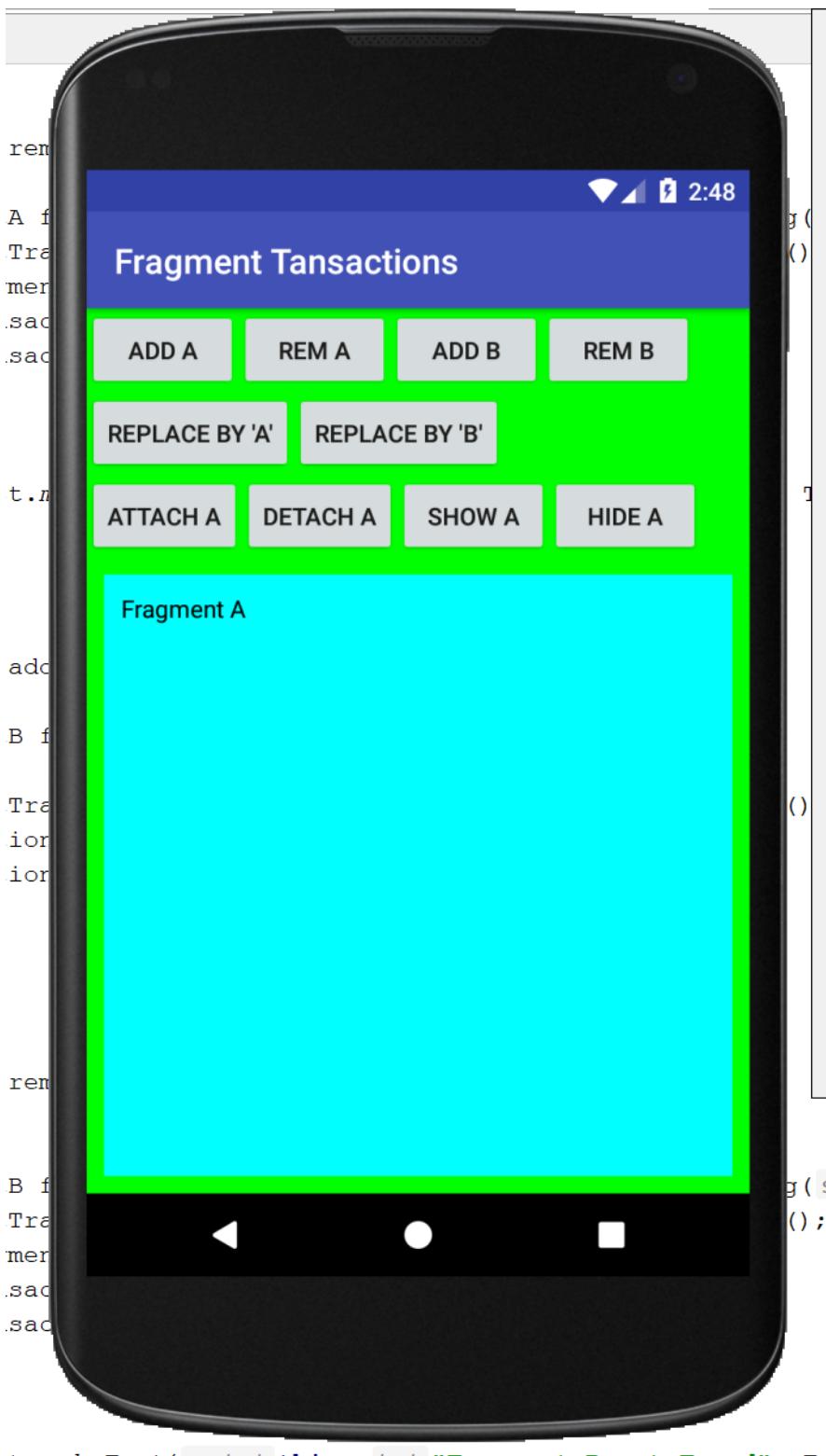
At this point, since we haven't added Fragment A or Fragment B, if we click the remove buttons for fragment A and B we get the toast message as shown:

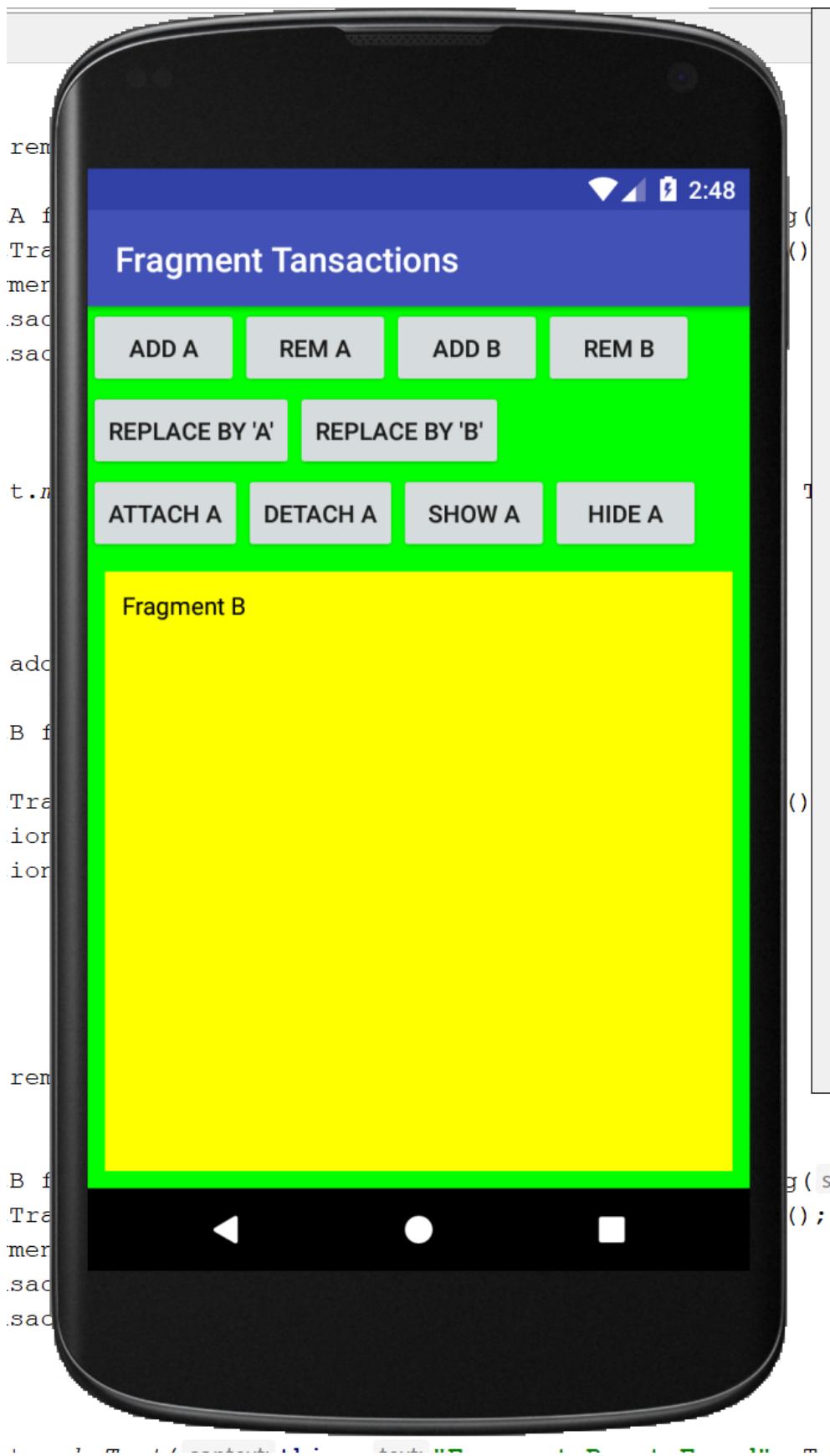


Added Fragment A or Fragment B. if we click the remove buttons for



Now click the “ADD A” followed by the “ADD B” buttons:





And here are the life cycle methods which were called:

The screenshot shows the Android Studio interface with the FragmentTransactions project open. The code editor displays MainActivity.java with the following code:

```
public void removeFragmentA(View view) {
    FragmentA fragmentA = (FragmentA) manager.findFragmentByTag("fragA");
    FragmentTransaction transaction = manager.beginTransaction();
    if (fragmentA != null) {
        transaction.remove(fragmentA);
        transaction.commit();
    } else {
        Toast.makeText(context, "Fragment A not Found", Toast.LENGTH_SHORT).show();
    }
}
```

The Logcat tab shows log entries for the FragmentA lifecycle:

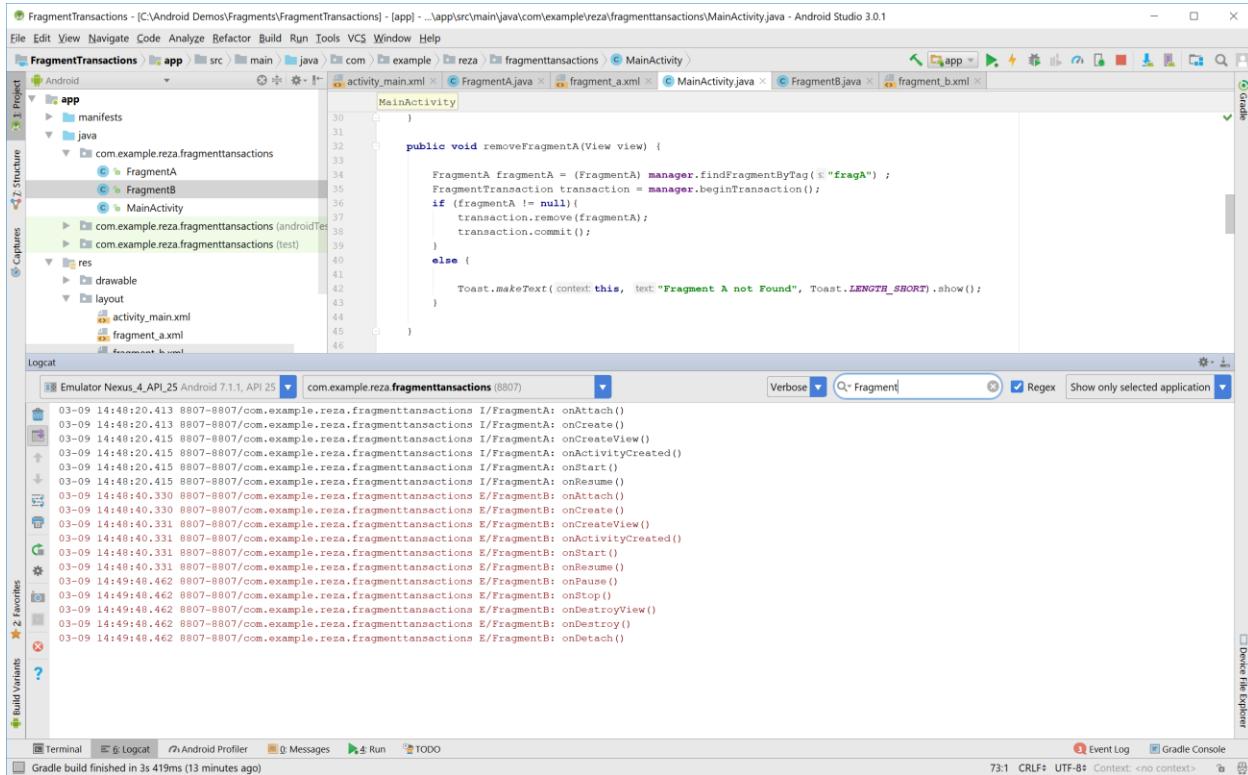
```
03-09 14:48:20.413 8807-8807/com.example.reza.fragmenttransactions I/FragmentA: onAttach()
03-09 14:48:20.413 8807-8807/com.example.reza.fragmenttransactions I/FragmentA: onCreate()
03-09 14:48:20.415 8807-8807/com.example.reza.fragmenttransactions I/FragmentA: onCreateView()
03-09 14:48:20.415 8807-8807/com.example.reza.fragmenttransactions I/FragmentA: onActivityCreated()
03-09 14:48:20.415 8807-8807/com.example.reza.fragmenttransactions I/FragmentA: onStart()
03-09 14:48:20.415 8807-8807/com.example.reza.fragmenttransactions I/FragmentA: onResume()
03-09 14:48:40.331 8807-8807/com.example.reza.fragmenttransactions E/FragmentB: onAttach()
03-09 14:48:40.331 8807-8807/com.example.reza.fragmenttransactions E/FragmentB: onCreate()
03-09 14:48:40.331 8807-8807/com.example.reza.fragmenttransactions E/FragmentB: onCreateView()
03-09 14:48:40.331 8807-8807/com.example.reza.fragmenttransactions E/FragmentB: onActivityCreated()
03-09 14:48:40.331 8807-8807/com.example.reza.fragmenttransactions E/FragmentB: onStart()
```

The bottom status bar indicates "Build finished in 3s 419ms (12 minutes ago)".

Now click Remove B ("REM B" button):



And here are the life cycle methods:



Notice the last 3 calls which shows fragment was destroyed and detached from the host activity.

Now click Remove A (“REM A” button) to remove fragment A as shown:



And here are the methods which were called:

```

public void removeFragmentA(View view) {
    FragmentA fragmentA = (FragmentA) manager.findFragmentByTag("fragA");
    FragmentTransaction transaction = manager.beginTransaction();
    if (fragmentA != null) {
        transaction.remove(fragmentA);
        transaction.commit();
    } else {
        Toast.makeText(context, "Fragment A not Found", Toast.LENGTH_SHORT).show();
    }
}

```

```

public void removeFragmentA(View view) {
    FragmentA fragmentA = (FragmentA) manager.findFragmentByTag("fragA");
    FragmentTransaction transaction = manager.beginTransaction();
    if (fragmentA != null) {
        transaction.remove(fragmentA);
        transaction.commit();
    } else {
        Toast.makeText(context, "Fragment A not Found", Toast.LENGTH_SHORT).show();
    }
}

```

Again notice the last 3 calls that shows fragment A has been destroyed and detached from the host activity.

At this point our container is blank and we don't have any fragments attached to the activity.



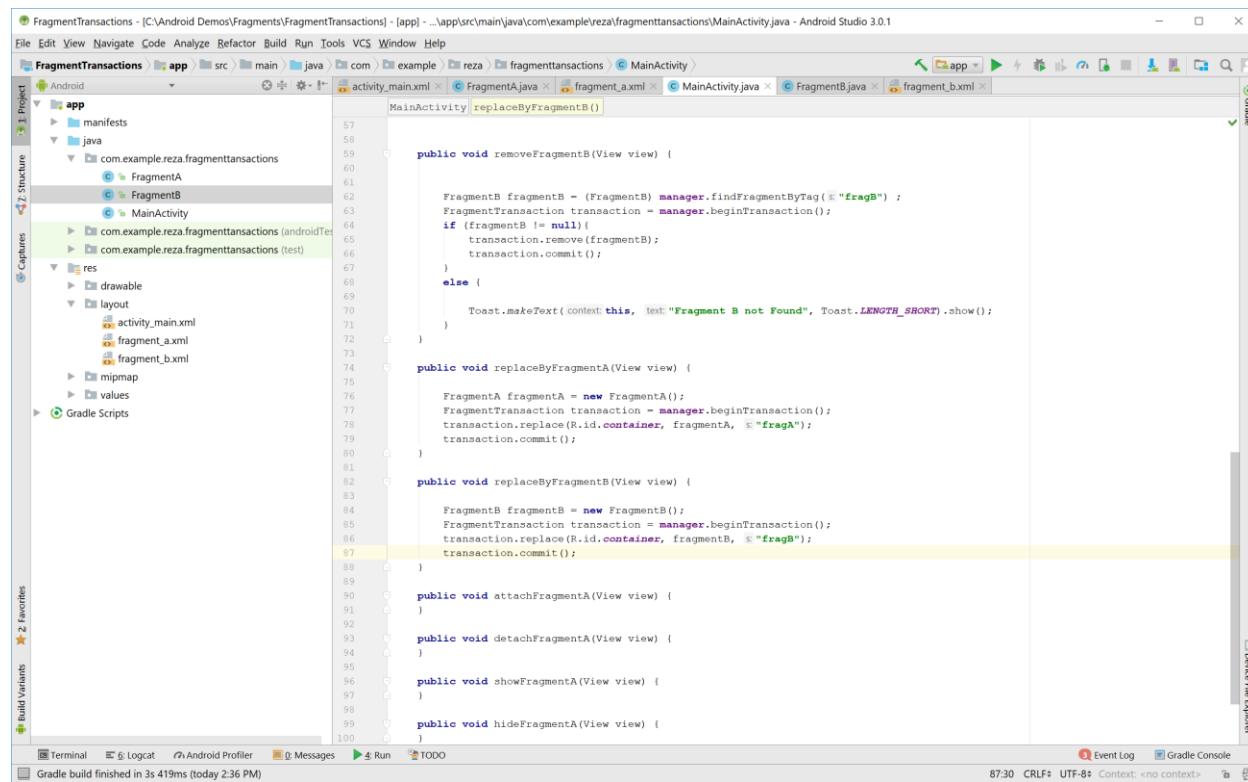
Another one of the transaction methods is the replace method that allows you to replace a fragment. What follows are the steps that you perform to replace a fragment:

First you get the id of the container layout inside which you need to replace a fragment, the next step is to get a reference to the new fragment that you want to insert in the place of the old fragment and finally you use the replace() method to replace the old fragment with the new one.

Note that when you replace an old fragment, the old fragment gets destroyed.

Modify your MainActivity.java file and provide implementation for replaceByFragmentA() method as shown:

Note that the implementation for replaceByFragmentB() is identical, you just change A for B.



The screenshot shows the Android Studio interface with the project 'FragmentTransactions' open. The code editor displays the MainActivity.java file. The code implements three methods: removeFragmentB, replaceByFragmentA, and replaceByFragmentB. The replaceByFragmentA and replaceByFragmentB methods are identical, showing how to find a fragment by tag, begin a transaction, and replace it with a new fragment. The removeFragmentB method shows how to remove a fragment if it exists. The code uses FragmentTransaction.replace(R.id.container, fragment, "fragB"); to replace the fragment.

```
public void removeFragmentB(View view) {
    FragmentB fragmentB = (FragmentB) manager.findFragmentByTag("fragB");
    FragmentTransaction transaction = manager.beginTransaction();
    if (fragmentB != null) {
        transaction.remove(fragmentB);
        transaction.commit();
    } else {
        Toast.makeText(context, "Fragment B not Found", Toast.LENGTH_SHORT).show();
    }
}

public void replaceByFragmentA(View view) {
    FragmentA fragmentA = new FragmentA();
    FragmentTransaction transaction = manager.beginTransaction();
    transaction.replace(R.id.container, fragmentA, "fragA");
    transaction.commit();
}

public void replaceByFragmentB(View view) {
    FragmentB fragmentB = new FragmentB();
    FragmentTransaction transaction = manager.beginTransaction();
    transaction.replace(R.id.container, fragmentB, "fragB");
    transaction.commit();
}

public void attachFragmentA(View view) {
}

public void detachFragmentA(View view) {
}

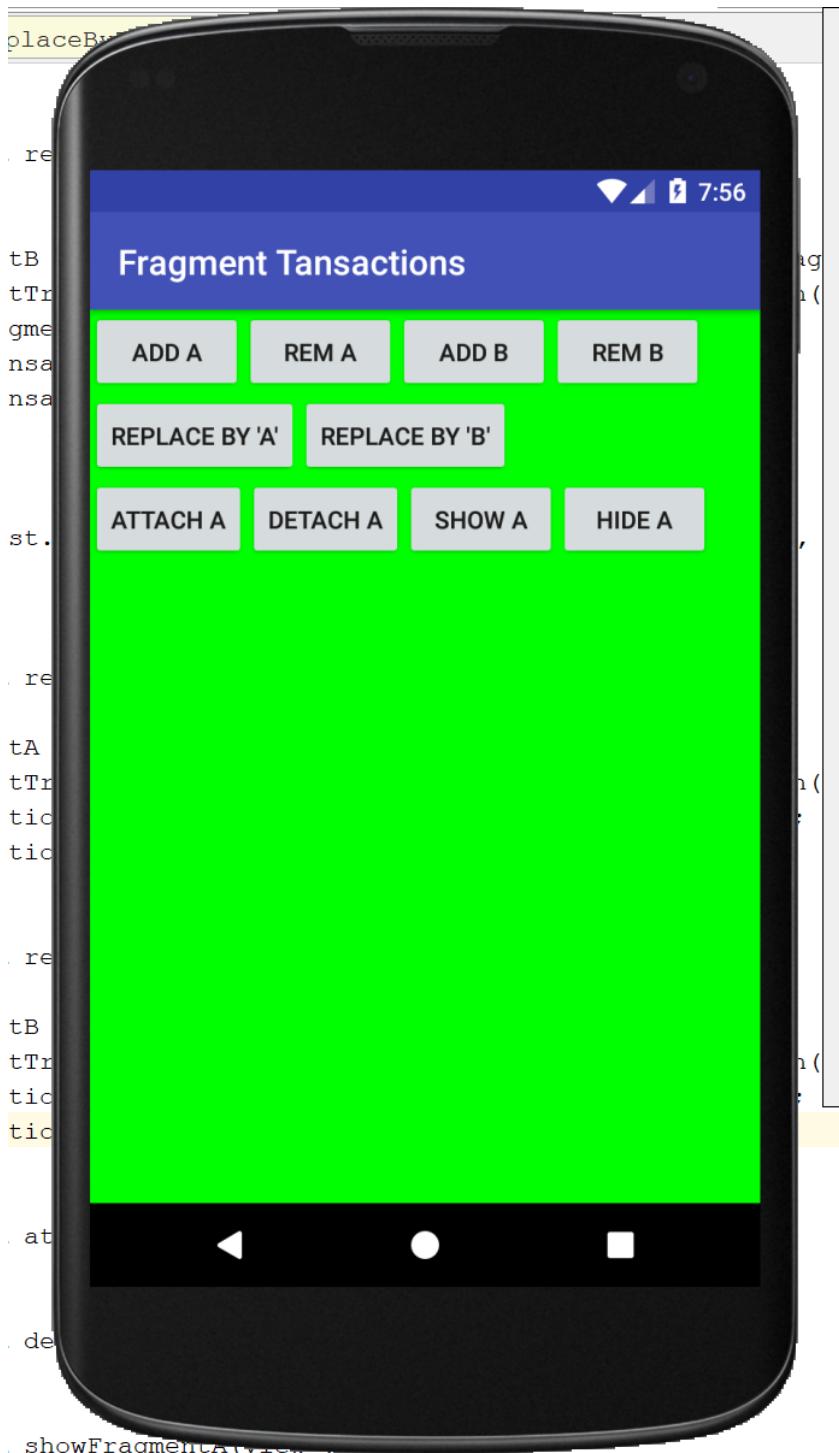
public void showFragmentA(View view) {
}

public void hideFragmentA(View view) {
}
```

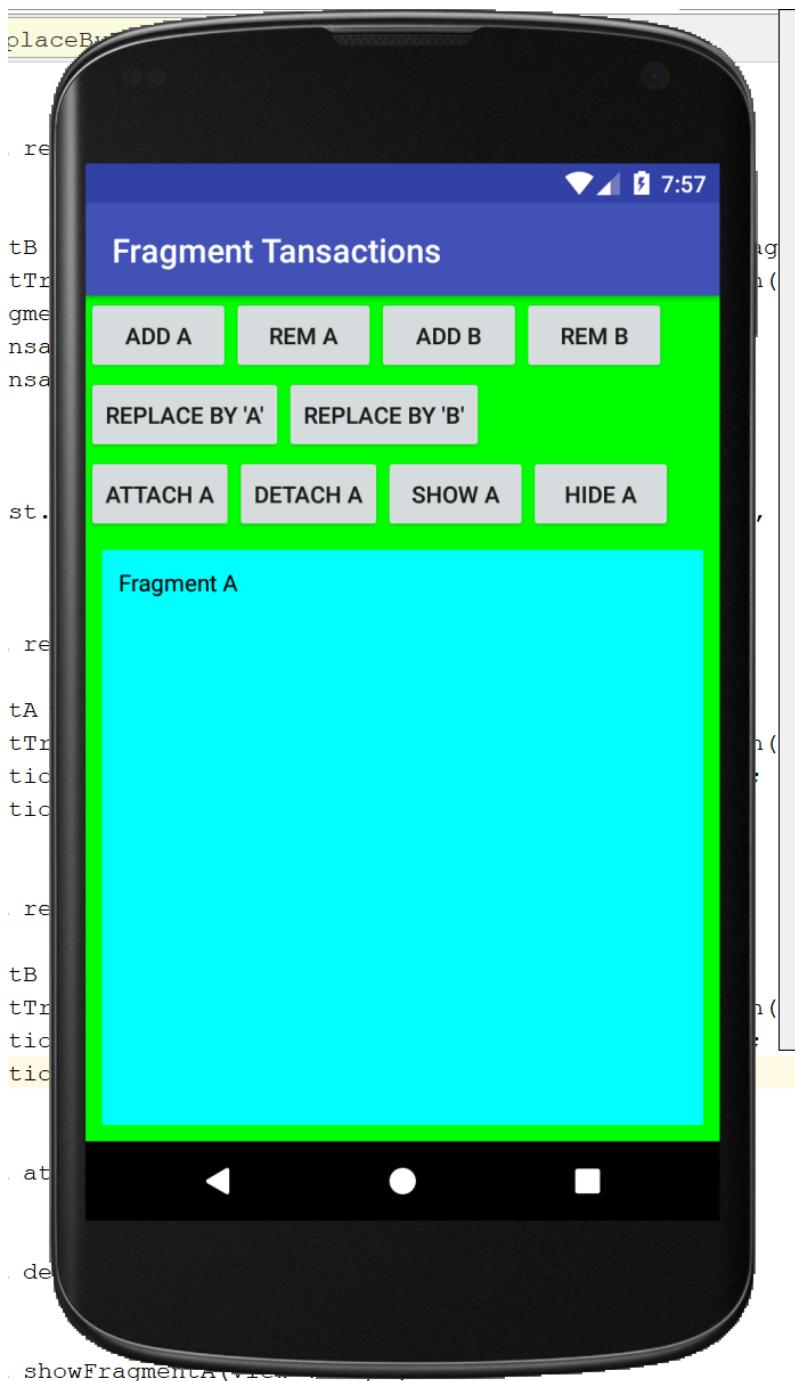
And here is the code with a larger font:

```
public void replaceByFragmentA(View view) {  
  
    FragmentA fragmentA = new FragmentA();  
    FragmentTransaction transaction = manager.beginTransaction();  
    transaction.replace(R.id.container, fragmentA, s: "fragA");  
    transaction.commit();  
}  
  
public void replaceByFragmentB(View view) {  
  
    FragmentB fragmentB = new FragmentB();  
    FragmentTransaction transaction = manager.beginTransaction();  
    transaction.replace(R.id.container, fragmentB, s: "fragB");  
    transaction.commit();  
}
```

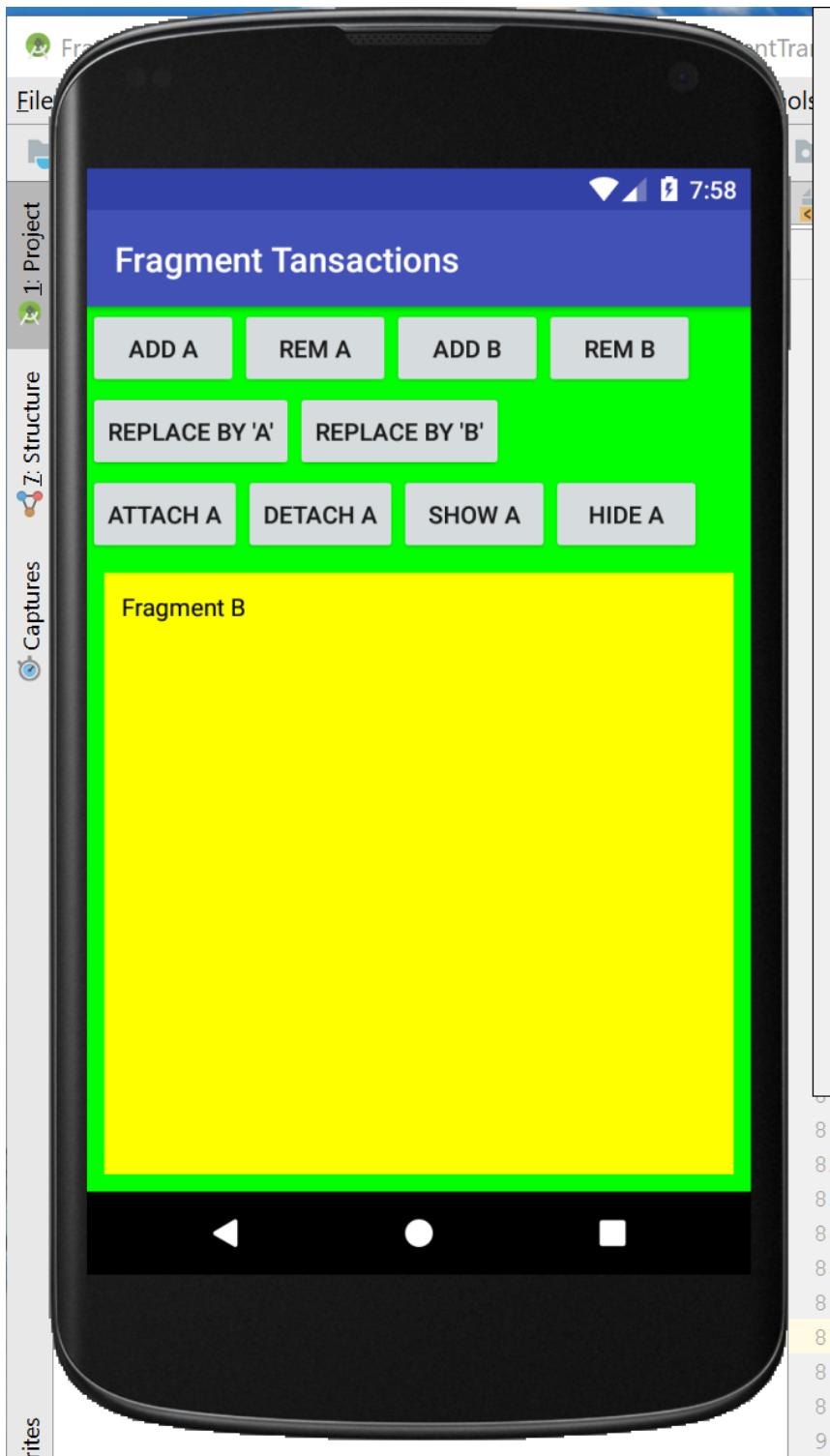
Now run the app:



Click “ADD A” to add fragment A:



Now let's try to replace fragment A with fragment B with clicking "ReplaceByB" button and here is the result:



And here are the life cycle methods which were called:

The screenshot shows the Android Studio interface with the FragmentTransactions project open. The code editor displays MainActivity.java with the following code:

```
public void removeFragmentB(View view) {
    FragmentB fragmentB = (FragmentB) manager.findFragmentByTag("fragB");
    FragmentTransaction transaction = manager.beginTransaction();
    if (fragmentB != null) {
        transaction.remove(fragmentB);
        transaction.commit();
    } else {
        Toast.makeText(context, "Fragment B not Found", Toast.LENGTH_SHORT).show();
    }
}
```

The Logcat tab shows the following log entries:

```
03-09 20:00:37.257 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onAttach()
03-09 20:00:37.257 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onCreate()
03-09 20:00:37.257 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onCreateView()
03-09 20:00:37.257 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onActivityCreated()
03-09 20:00:37.257 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onStart()
03-09 20:00:37.257 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onResume()
03-09 20:00:40.352 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: onAttach()
03-09 20:00:40.352 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: onCreate()
03-09 20:00:40.352 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: onPause()
03-09 20:00:40.352 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: onStop()
03-09 20:00:40.352 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: onDestroyView()
03-09 20:00:40.353 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: onDestroy()
03-09 20:00:40.353 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: onDetach()
03-09 20:00:40.353 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: oncreateView()
03-09 20:00:40.353 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: onActivityCreated()
03-09 20:00:40.353 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: onStart()
03-09 20:00:40.353 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: onResume()
```

The bottom status bar indicates "Gradle build finished in 1s 354ms (a minute ago)".

Now click “ReplaceByA” button and here is the result:



And here are the life cycle methods in the order they were called:

The screenshot shows the Android Studio interface with the FragmentTransactions project open. The code editor displays MainActivity.java with the following code:

```
public void showFragmentA() {
    FragmentA fragmentA = (FragmentA) getSupportFragmentManager().findFragmentByTag("tagA");
    FragmentTransaction transaction = manager.beginTransaction();
    transaction.replace(R.id.fragmentContainer, fragmentA, "tagA");
    transaction.commit();
}
```

The Logcat tab shows log messages from the Emulator Nexus_4 API_25. The messages detail the creation and destruction of FragmentA and FragmentB. Key log entries include:

- 03-09 20:00:37.257 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onAttach()
- 03-09 20:00:37.257 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onCreate()
- 03-09 20:00:37.257 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onCreateView()
- 03-09 20:00:37.258 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onActivityCreated()
- 03-09 20:00:37.258 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onStart()
- 03-09 20:00:37.258 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onResume()
- 03-09 20:00:40.352 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: onAttach()
- 03-09 20:00:40.352 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: onCreate()
- 03-09 20:00:40.352 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onPause()
- 03-09 20:00:40.352 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onStop()
- 03-09 20:00:40.353 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onDestroyView()
- 03-09 20:00:40.353 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onDestroy()
- 03-09 20:00:40.353 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onDetach()
- 03-09 20:00:40.353 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: onCreateView()
- 03-09 20:00:40.353 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: onActivityCreated()
- 03-09 20:00:40.353 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: onStart()
- 03-09 20:00:40.353 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: onResume()
- 03-09 20:03:20.495 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onAttach()
- 03-09 20:03:20.495 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onCreate()
- 03-09 20:03:20.495 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: onPause()
- 03-09 20:03:20.495 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: onStop()
- 03-09 20:03:20.495 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: onDestroyView()
- 03-09 20:03:20.495 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: onDestroy()
- 03-09 20:03:20.495 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: onDetach()
- 03-09 20:03:20.495 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onCreateView()
- 03-09 20:03:20.495 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onActivityCreated()
- 03-09 20:03:20.495 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onStart()
- 03-09 20:03:20.495 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onResume()

The bottom status bar shows "Grade build finished in 1s 354ms (4 minutes ago)".

The highlighted section in the next screen capture shows the methods that were called after clicking the “ReplaceByA” button:

```

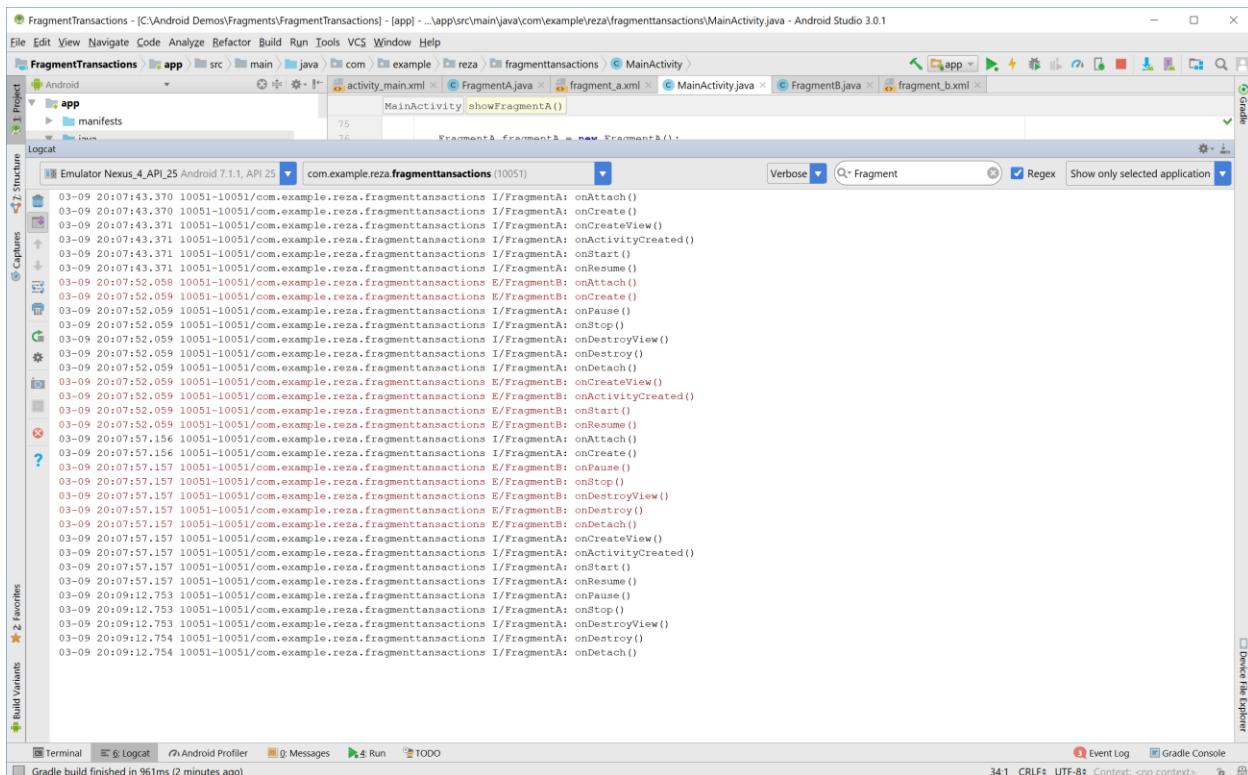
FragmentTransactions - [C:\Android Demos\Fragments\FragmentTransactions] - [app] - ...\\app\\src\\main\\java\\com\\example\\reza\\fragmenttransactions\\MainActivity.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
FragmentTransactions > app > src > main > java > com > example > reza > fragmenttransactions > MainActivity >
activity_main.xml FragmentA.java fragment_a.xml MainActivity.java FragmentB.java fragment_b.xml
Gradle
Project
app
manifests
java
com.example.reza.fragmenttransactions
FragmentA
FragmentB
MainActivity
Logcat
Emulator Nexus_4 API 25 Android 7.1.1, API 25 com.example.reza.fragmenttransactions (9832)
Verbose Fragment Regex Show only selected application
03-09 20:00:37.257 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onAttach()
03-09 20:00:37.257 9832-9832/com.example.reza.fragmenttransactions I/FragmentB: onCreate()
03-09 20:00:37.257 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onCreateView()
03-09 20:00:37.259 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onActivityCreated()
03-09 20:00:37.259 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onStart()
03-09 20:00:37.259 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onResume()
03-09 20:00:40.352 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: onAttach()
03-09 20:00:40.352 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: onCreate()
03-09 20:00:40.352 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onPause()
03-09 20:00:40.352 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onStop()
03-09 20:00:40.352 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onDestoryView()
03-09 20:00:40.353 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onDestory()
03-09 20:00:40.353 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onDetach()
03-09 20:00:40.353 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: onCreateView()
03-09 20:00:40.353 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: onActivityCreated()
03-09 20:00:40.353 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: onStart()
03-09 20:00:40.353 9832-9832/com.example.reza.fragmenttransactions E/FragmentB: onResume()
03-09 20:03:20.494 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onAttach()
03-09 20:03:20.494 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onCreate()
03-09 20:03:20.494 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onCreateView()
03-09 20:03:20.494 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onActivityCreated()
03-09 20:03:20.495 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onStart()
03-09 20:03:20.495 9832-9832/com.example.reza.fragmenttransactions I/FragmentA: onResume()

```

Now you can click "REM A" button to remove the fragment A as shown:



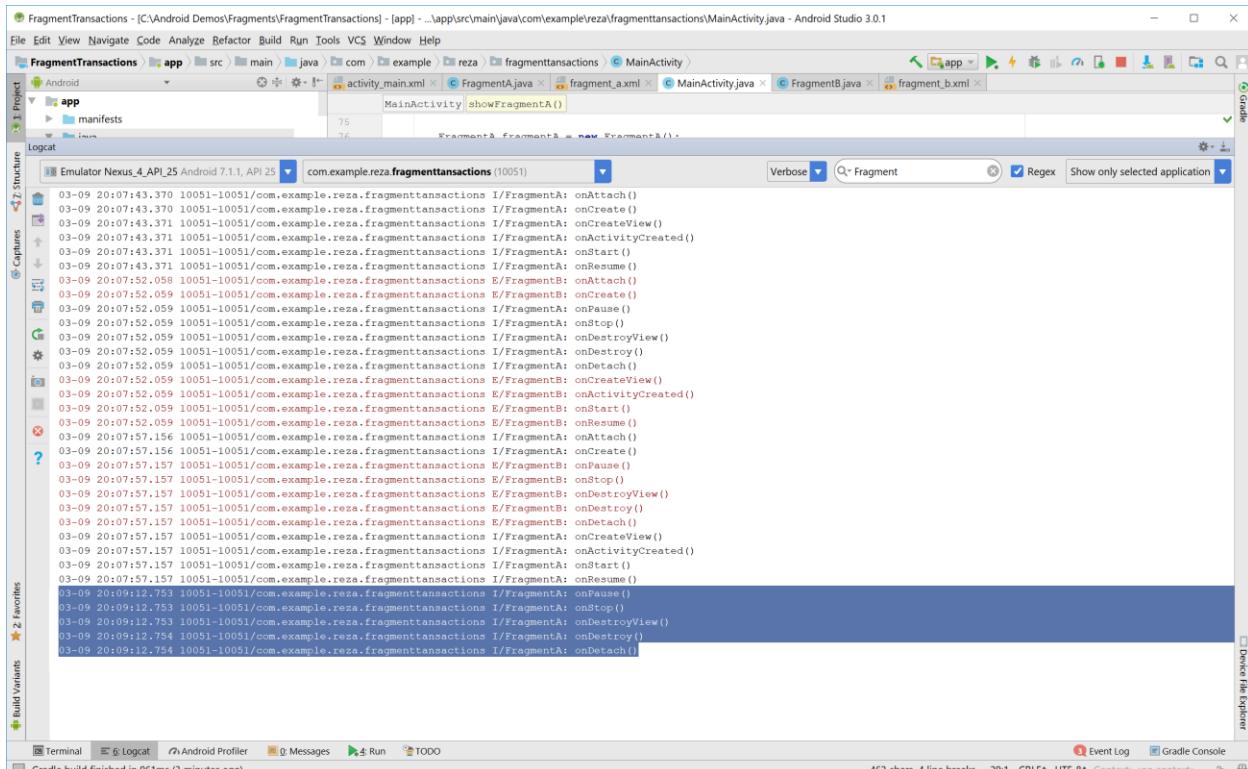
And here is the Logcat window:



The screenshot shows the Android Studio Logcat window. The title bar reads "FragmentTransactions - [C:\Android Demos\Fragments\FragmentTransactions] - [app] - ...\\app\\src\\main\\java\\com\\example\\fragmenttransactions\\MainActivity.java - Android Studio 3.0.1". The Logcat tab is selected, showing log entries for the Emulator Nexus_4_API_25. The log entries are as follows:

```
03-09 20:07:43.370 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onAttach()
03-09 20:07:43.370 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onCreate()
03-09 20:07:43.371 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onCreateView()
03-09 20:07:43.371 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onActivityCreated()
03-09 20:07:43.371 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onStart()
03-09 20:07:43.371 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onResume()
03-09 20:07:52.059 10051-10051/com.example.reza.fragmenttransactions E/FragmentB: onAttach()
03-09 20:07:52.059 10051-10051/com.example.reza.fragmenttransactions E/FragmentB: onCreate()
03-09 20:07:52.059 10051-10051/com.example.reza.fragmenttransactions I/FragmentB: onPause()
03-09 20:07:52.059 10051-10051/com.example.reza.fragmenttransactions I/FragmentB: onStop()
03-09 20:07:52.059 10051-10051/com.example.reza.fragmenttransactions I/FragmentB: onDestroyView()
03-09 20:07:52.059 10051-10051/com.example.reza.fragmenttransactions I/FragmentB: onDestroy()
03-09 20:07:52.059 10051-10051/com.example.reza.fragmenttransactions I/FragmentB: onDetach()
03-09 20:07:52.059 10051-10051/com.example.reza.fragmenttransactions E/FragmentB: onCreateView()
03-09 20:07:52.059 10051-10051/com.example.reza.fragmenttransactions E/FragmentB: onActivityCreated()
03-09 20:07:52.059 10051-10051/com.example.reza.fragmenttransactions E/FragmentB: onStart()
03-09 20:07:52.059 10051-10051/com.example.reza.fragmenttransactions E/FragmentB: onResume()
03-09 20:07:57.159 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onAttach()
03-09 20:07:57.159 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onCreate()
03-09 20:07:57.159 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onPause()
03-09 20:07:57.159 10051-10051/com.example.reza.fragmenttransactions E/FragmentB: onStop()
03-09 20:07:57.159 10051-10051/com.example.reza.fragmenttransactions E/FragmentB: onDestroyView()
03-09 20:07:57.159 10051-10051/com.example.reza.fragmenttransactions E/FragmentB: onDestroy()
03-09 20:07:57.159 10051-10051/com.example.reza.fragmenttransactions I/FragmentB: onDetach()
03-09 20:07:57.159 10051-10051/com.example.reza.fragmenttransactions I/FragmentB: onCreateView()
03-09 20:07:57.159 10051-10051/com.example.reza.fragmenttransactions I/FragmentB: onActivityCreated()
03-09 20:07:57.159 10051-10051/com.example.reza.fragmenttransactions I/FragmentB: onStart()
03-09 20:07:57.159 10051-10051/com.example.reza.fragmenttransactions I/FragmentB: onResume()
03-09 20:09:12.750 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onPause()
03-09 20:09:12.750 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onStop()
03-09 20:09:12.750 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onDestroyView()
03-09 20:09:12.750 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onDestroy()
03-09 20:09:12.754 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onDetach()
```

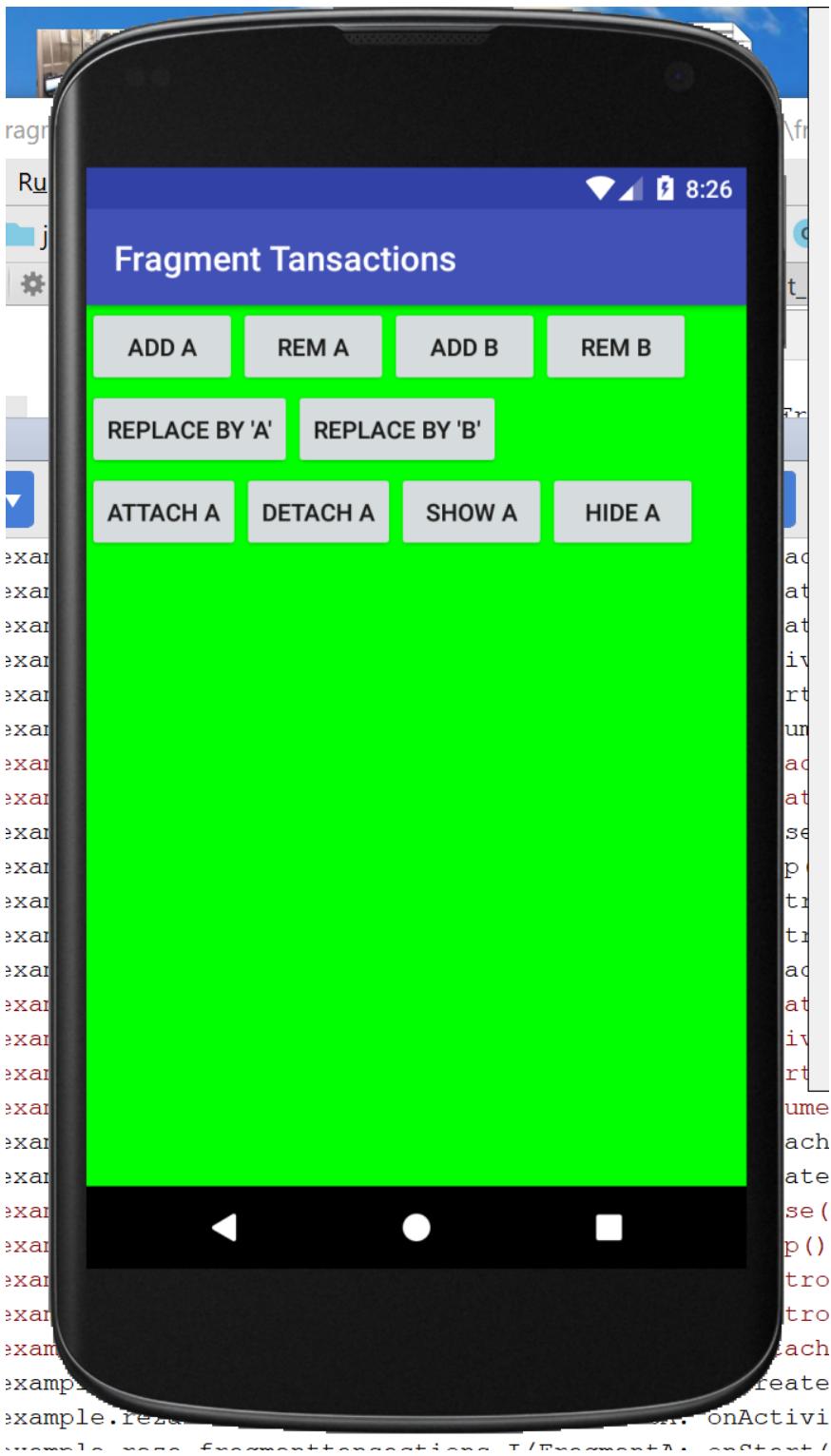
And here are the methods that were called when removing the fragment A:



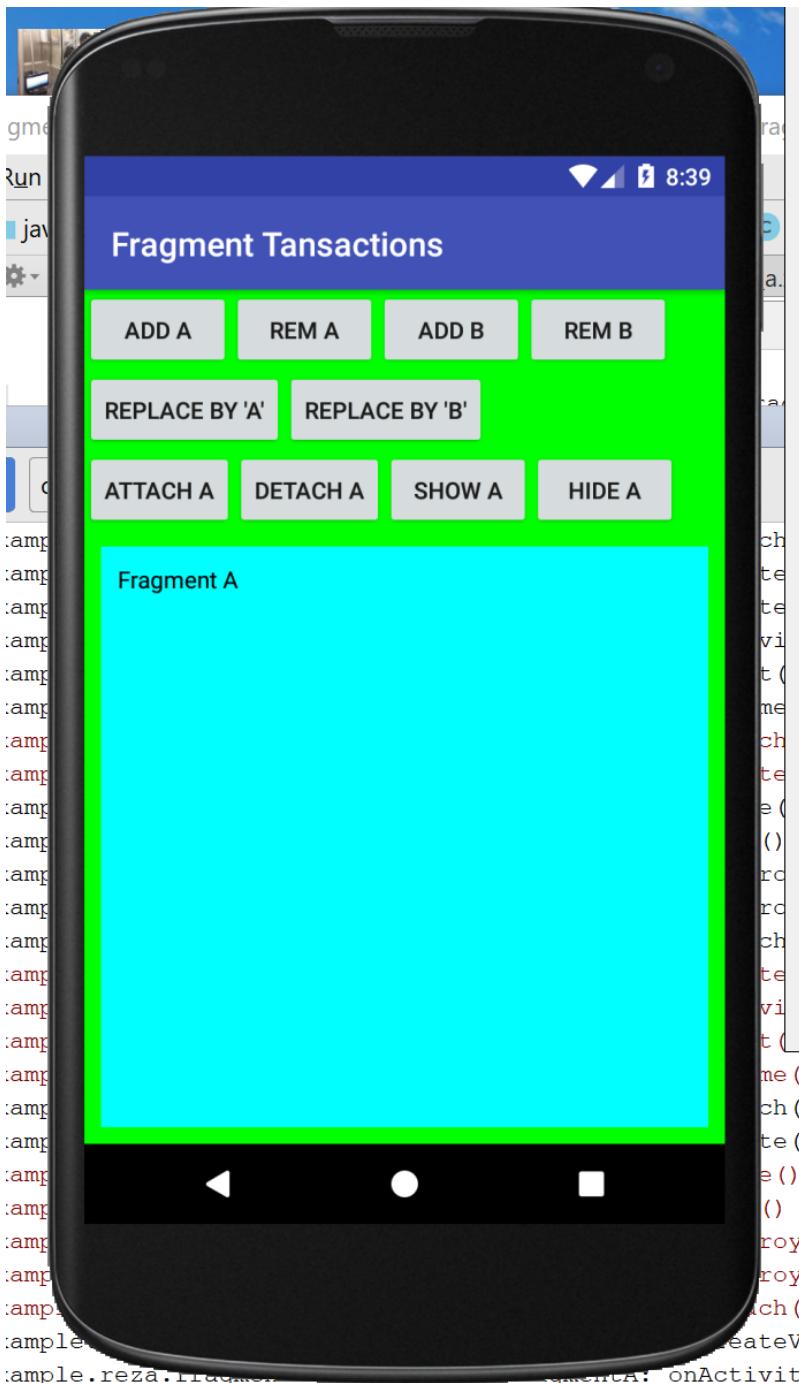
The screenshot shows the Android Studio interface with the Logcat tab selected. The log output is as follows:

```
03-09 20:07:43.370 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onAttach()
03-09 20:07:43.370 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onCreate()
03-09 20:07:43.371 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onCreateView()
03-09 20:07:43.371 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onActivityCreated()
03-09 20:07:43.371 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onStart()
03-09 20:07:43.371 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onResume()
03-09 20:07:52.059 10051-10051/com.example.reza.fragmenttransactions E/FragmentB: onAttach()
03-09 20:07:52.059 10051-10051/com.example.reza.fragmenttransactions E/FragmentB: onCreate()
03-09 20:07:52.059 10051-10051/com.example.reza.fragmenttransactions I/FragmentB: onPause()
03-09 20:07:52.059 10051-10051/com.example.reza.fragmenttransactions I/FragmentB: onStop()
03-09 20:07:52.059 10051-10051/com.example.reza.fragmenttransactions I/FragmentB: onDestroyView()
03-09 20:07:52.059 10051-10051/com.example.reza.fragmenttransactions I/FragmentB: onDestroy()
03-09 20:07:52.059 10051-10051/com.example.reza.fragmenttransactions I/FragmentB: onDetach()
03-09 20:07:52.059 10051-10051/com.example.reza.fragmenttransactions E/FragmentB: onCreateView()
03-09 20:07:52.059 10051-10051/com.example.reza.fragmenttransactions E/FragmentB: onActivityCreated()
03-09 20:07:52.059 10051-10051/com.example.reza.fragmenttransactions E/FragmentB: onStart()
03-09 20:07:52.059 10051-10051/com.example.reza.fragmenttransactions E/FragmentB: onResume()
03-09 20:07:57.159 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onAttach()
03-09 20:07:57.159 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onCreate()
03-09 20:07:57.159 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onPause()
03-09 20:07:57.159 10051-10051/com.example.reza.fragmenttransactions E/FragmentB: onStop()
03-09 20:07:57.159 10051-10051/com.example.reza.fragmenttransactions E/FragmentB: onDestroyView()
03-09 20:07:57.159 10051-10051/com.example.reza.fragmenttransactions E/FragmentB: onDestroy()
03-09 20:07:57.159 10051-10051/com.example.reza.fragmenttransactions I/FragmentB: onDetach()
03-09 20:07:57.159 10051-10051/com.example.reza.fragmenttransactions I/FragmentB: onCreateView()
03-09 20:07:57.159 10051-10051/com.example.reza.fragmenttransactions I/FragmentB: onActivityCreated()
03-09 20:07:57.159 10051-10051/com.example.reza.fragmenttransactions I/FragmentB: onStart()
03-09 20:07:57.159 10051-10051/com.example.reza.fragmenttransactions I/FragmentB: onResume()
03-09 20:09:12.753 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onPause()
03-09 20:09:12.753 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onStop()
03-09 20:09:12.753 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onDestroyView()
03-09 20:09:12.753 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onDestroy()
03-09 20:09:12.754 10051-10051/com.example.reza.fragmenttransactions I/FragmentA: onDetach()
```

Now we don't have any fragment displayed in the container as shown:



If you click the ReplaceByA button at this point, the container gets filled with the fragment A as shown even if there was no fragment in the container to replace:



And here is the Logcat window (I ran the app again to get rid of the old output to make it easier to focus on the new methods being called)

The screenshot shows the Android Studio interface with the FragmentTransactions project open. The Project tool window on the left shows the file structure, including Java files like MainActivity.java and FragmentA.java, and XML files like activity_main.xml and fragment_a.xml. The Logcat window at the bottom displays log messages from the Emulator Nexus_4 API_25. The log messages are as follows:

```
03-09 20:42:41.910 10190-10190/com.example.reza.fragmenttransactions I/FragmentA: onAttach()
03-09 20:42:41.910 10190-10190/com.example.reza.fragmenttransactions I/Fragment: onCreate()
03-09 20:42:41.911 10190-10190/com.example.reza.fragmenttransactions I/FragmentA: onCreateView()
03-09 20:42:41.911 10190-10190/com.example.reza.fragmenttransactions I/Fragment: onActivityCreated()
03-09 20:42:41.911 10190-10190/com.example.reza.fragmenttransactions I/FragmentA: onStart()
03-09 20:42:41.912 10190-10190/com.example.reza.fragmenttransactions I/FragmentA: onResume()
```

That means even if there is not fragment to replace, the fragment A will still be displayed.

Now if you click the ReplaceByA button again, then first fragment A will be destroyed and a new fragment A will be created and that new fragment replaces the old fragment A.

So what you see in the emulator below, is the new fragment A which replaced the old fragment A:



And here is the Logcat and life cycle method which were called:

The screenshot shows the Android Studio interface with the FragmentTransactions project open. The code editor displays MainActivity.java with the following code:

```
75     FragmentA fragmentA = new FragmentA();
76     FragmentTransaction transaction = manager.beginTransaction();
77     transaction.replace(R.id.container, fragmentA, "fragA");
78     transaction.commit();
79 }
80
81 public void replaceByFragmentB(View view) {
82
83     FragmentB fragmentB = new FragmentB();
84     FragmentTransaction transaction = manager.beginTransaction();
85     transaction.replace(R.id.container, fragmentB, "fragB");
86     transaction.commit();
87 }
88
89 public void attachFragmentA(View view) {
90 }
```

The Logcat tab shows log entries for FragmentA and FragmentB lifecycle events, such as onCreate, onCreateView, onStart, onResume, onPause, onStop, onDestroyView, and onDestroy. The bottom status bar indicates a successful build: "Gradle build finished in 927ms (3 minutes ago)".

In the screen capture below I have highlighted all the methods which were called after we called ReplaceByA() button for a second time. They clearly shows that the old fragment A was destroyed and a new fragment A was created to replace the old one.

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** Shows the project tree with modules: app, manifests, java, com.example.reza.fragmenttransactions (containing FragmentA, FragmentB, and MainActivity).
- MainActivity.java:** The code is shown with several methods highlighted in green:


```

    FragmentA fragmentA = new FragmentA();
    FragmentTransaction transaction = manager.beginTransaction();
    transaction.replace(R.id.container, fragmentA, "fragA");
    transaction.commit();

    public void replaceByFragmentB(View view) {
        FragmentB fragmentB = new FragmentB();
        FragmentTransaction transaction = manager.beginTransaction();
        transaction.replace(R.id.container, fragmentB, "fragB");
        transaction.commit();
    }

    public void attachFragmentA(View view) {
    }
      
```
- Logcat:** Displays log entries from the Emulator Nexus_4 API 25 device. The log shows the following sequence of events:


```

03-09 20:42:41.910 10190-10190/com.example.reza.fragmenttransactions I/Fragment: onAttach()
03-09 20:42:41.910 10190-10190/com.example.reza.fragmenttransactions I/Fragment: onCreate()
03-09 20:42:41.911 10190-10190/com.example.reza.fragmenttransactions I/Fragment: onCreateView()
03-09 20:42:41.911 10190-10190/com.example.reza.fragmenttransactions I/Fragment: onActivityCreated()
03-09 20:42:41.911 10190-10190/com.example.reza.fragmenttransactions I/Fragment: onStart()
03-09 20:42:41.912 10190-10190/com.example.reza.fragmenttransactions I/Fragment: onResume()
03-09 20:44:39.670 10190-10190/com.example.reza.fragmenttransactions I/Fragment: onAttach()
03-09 20:44:39.670 10190-10190/com.example.reza.fragmenttransactions I/Fragment: onCreate()
03-09 20:44:39.670 10190-10190/com.example.reza.fragmenttransactions I/Fragment: onPause()
03-09 20:44:39.670 10190-10190/com.example.reza.fragmenttransactions I/Fragment: onStop()
03-09 20:44:39.670 10190-10190/com.example.reza.fragmenttransactions I/Fragment: onDestroyView()
03-09 20:44:39.670 10190-10190/com.example.reza.fragmenttransactions I/Fragment: onDestroy()
03-09 20:44:39.670 10190-10190/com.example.reza.fragmenttransactions I/Fragment: onDetach()
03-09 20:44:39.670 10190-10190/com.example.reza.fragmenttransactions I/Fragment: onCreateView()
03-09 20:44:39.671 10190-10190/com.example.reza.fragmenttransactions I/Fragment: onActivityCreated()
03-09 20:44:39.671 10190-10190/com.example.reza.fragmenttransactions I/Fragment: onStart()
03-09 20:44:39.671 10190-10190/com.example.reza.fragmenttransactions I/Fragment: onResume()
      
```
- Bottom Bar:** Includes tabs for Terminal, Logcat, Android Profiler, Messages, Run, TODO, Event Log, and Gradle Console.

Note that the highlighted code in the above screen capture also contains a few calls related to the old fragment. The screen capture below shows everything related to the old fragment A which was called after the second time we clicked the “replaceByA” button:

The screenshot shows the Android Studio interface with the following details:

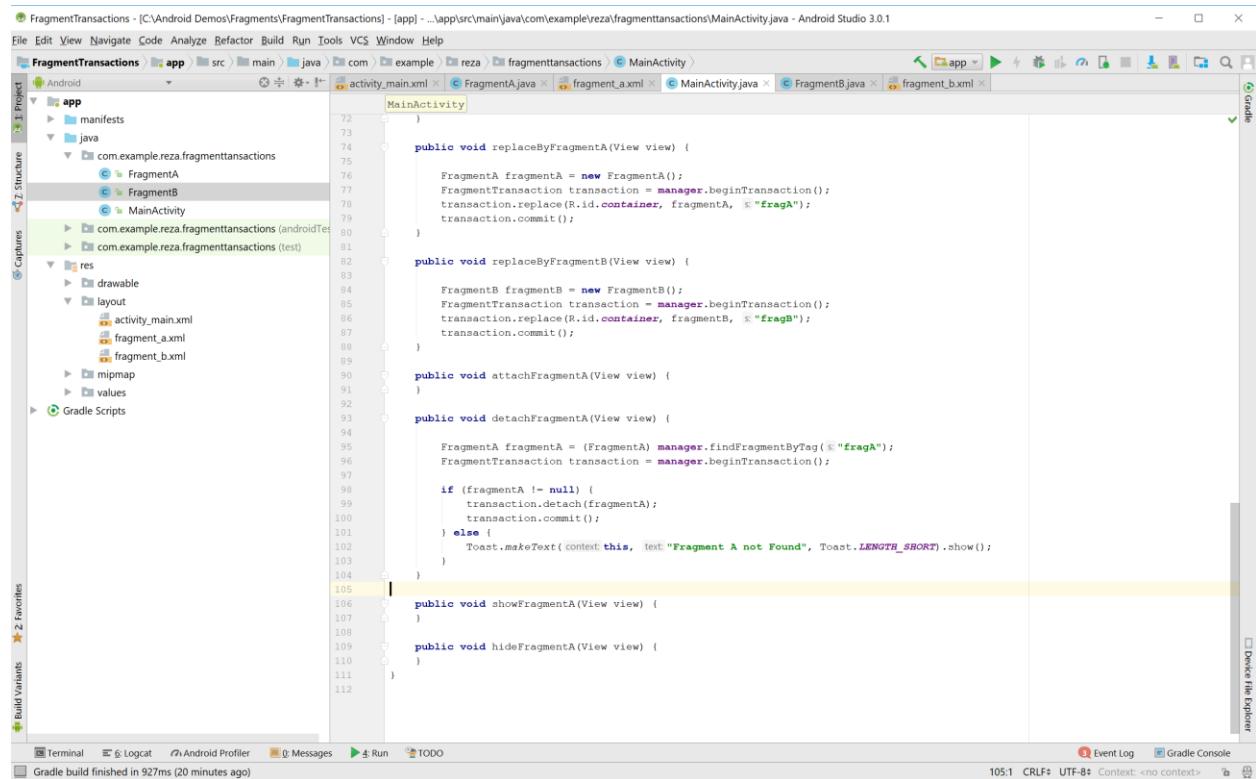
- Project Structure:** The project is named "FragmentTransactions". The "app" module contains Java files like FragmentA.java, FragmentB.java, and MainActivity.java, along with XML files for activity_main.xml, fragment_a.xml, and fragment_b.xml.
- Code Editor:** The FragmentA.java file is open, showing methods like onActivityCreated(), onDestroy(), and onDetach().
- Logcat:** The logcat window displays numerous log entries from the "com.example.reza.fragmenttransactions" package, detailing the lifecycle events of FragmentA (onAttach, onCreate, onCreateView, onActivityCreated, onStart, onResume, onAttach, onCreate) and FragmentB (onDestroyView, onStop, onDestroy, onDetach).

The next method from the fragment transaction methods is the detach method. You use this method to detach the fragment from the Activity UI and to do so you need to perform the following steps:

Get a reference to the fragment that you want to detach from the UI and then use the `detach()` method to detach the fragment from Activity UI.

Note that when you detach a fragment, the fragment's View is destroyed which simply means the fragment won't be visible to the user but the fragment is still present inside the activity, that is the fragment is not destroyed. Only the view of the fragment is destroyed which simply means the fragment and its UI is detached from the host activity but the instance of the fragment is still living and not destroyed.

What follows shows the code for detachFragmentA() method:



The screenshot shows the Android Studio interface with the project 'FragmentTransactions' open. The code editor displays the 'MainActivity.java' file. The 'detachFragmentA' method is highlighted with a yellow background. The code uses FragmentManager to find, detach, and commit transactions for FragmentA.

```
public void detachFragmentA(View view) {
    FragmentA fragmentA = (FragmentA) manager.findFragmentByTag("fragA");
    FragmentTransaction transaction = manager.beginTransaction();
    transaction.detach(fragmentA);
    transaction.commit();
}
}
}
```

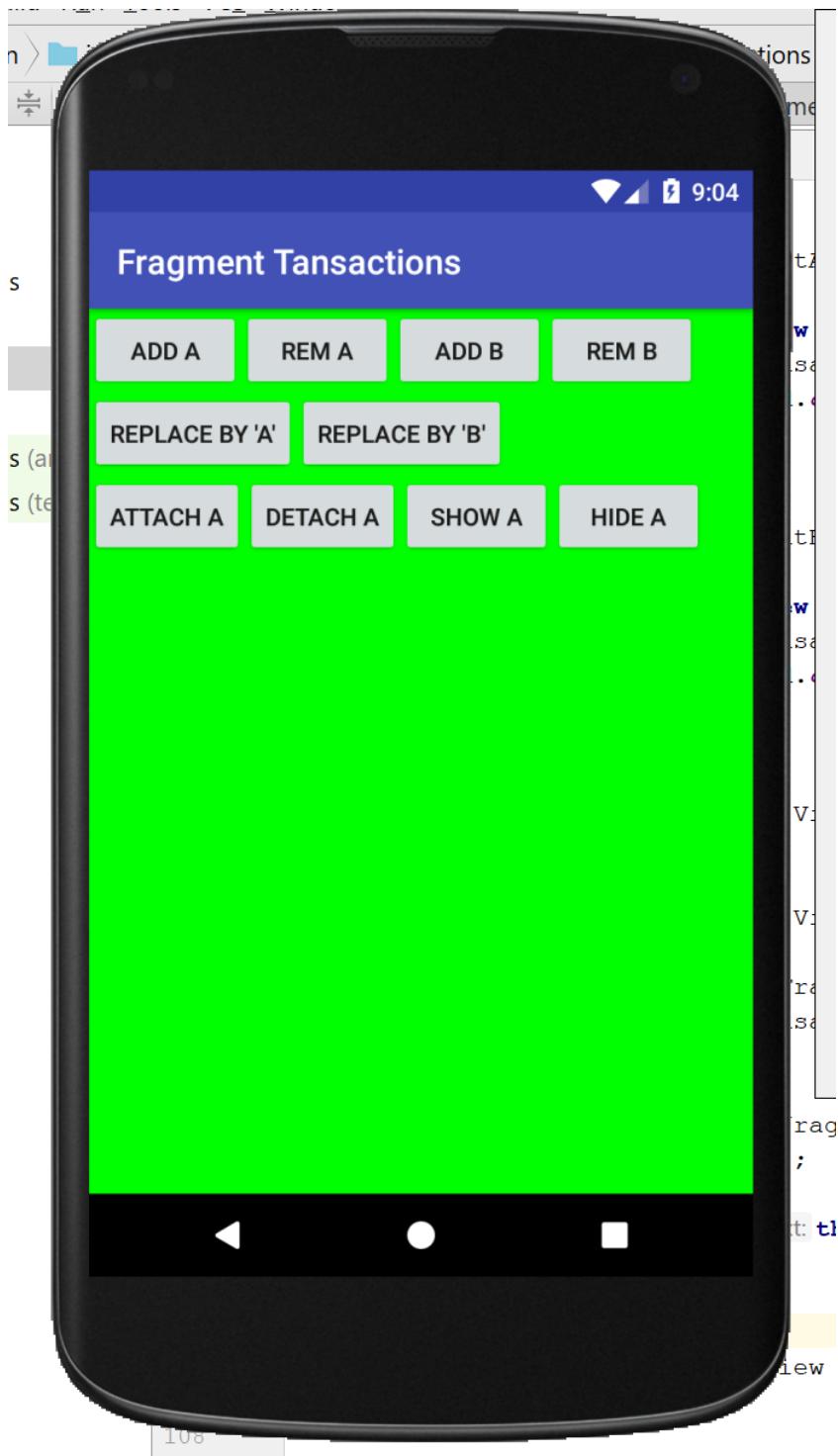
And here is the code with larger font:

```
public void detachFragmentA(View view) {

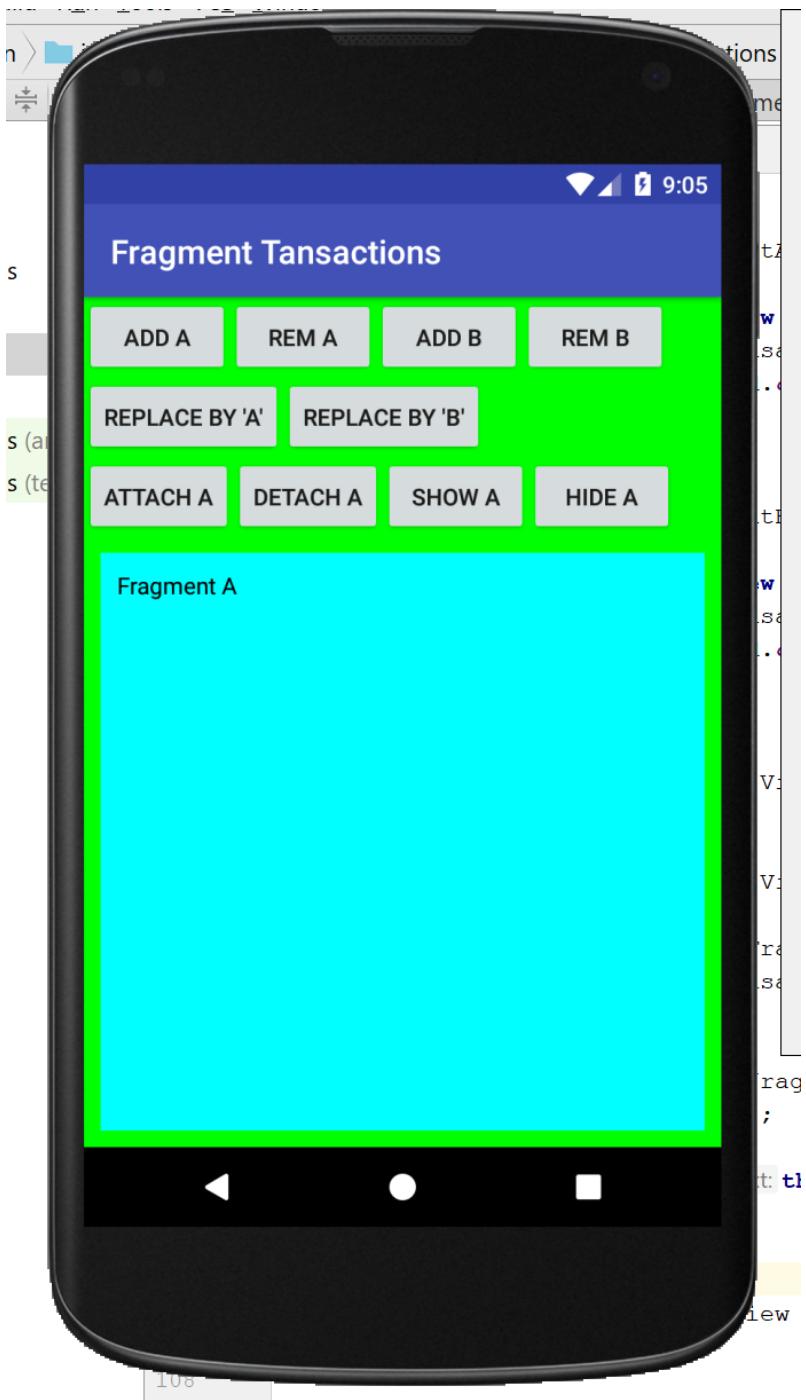
    FragmentA fragmentA = (FragmentA) manager.findFragmentByTag("fragA");
    FragmentTransaction transaction = manager.beginTransaction();

    if (fragmentA != null) {
        transaction.detach(fragmentA);
        transaction.commit();
    } else {
        Toast.makeText(context, "Fragment A not Found", Toast.LENGTH_SHORT).show();
    }
}
```

Now run the app:



Then click “ADD A” button:



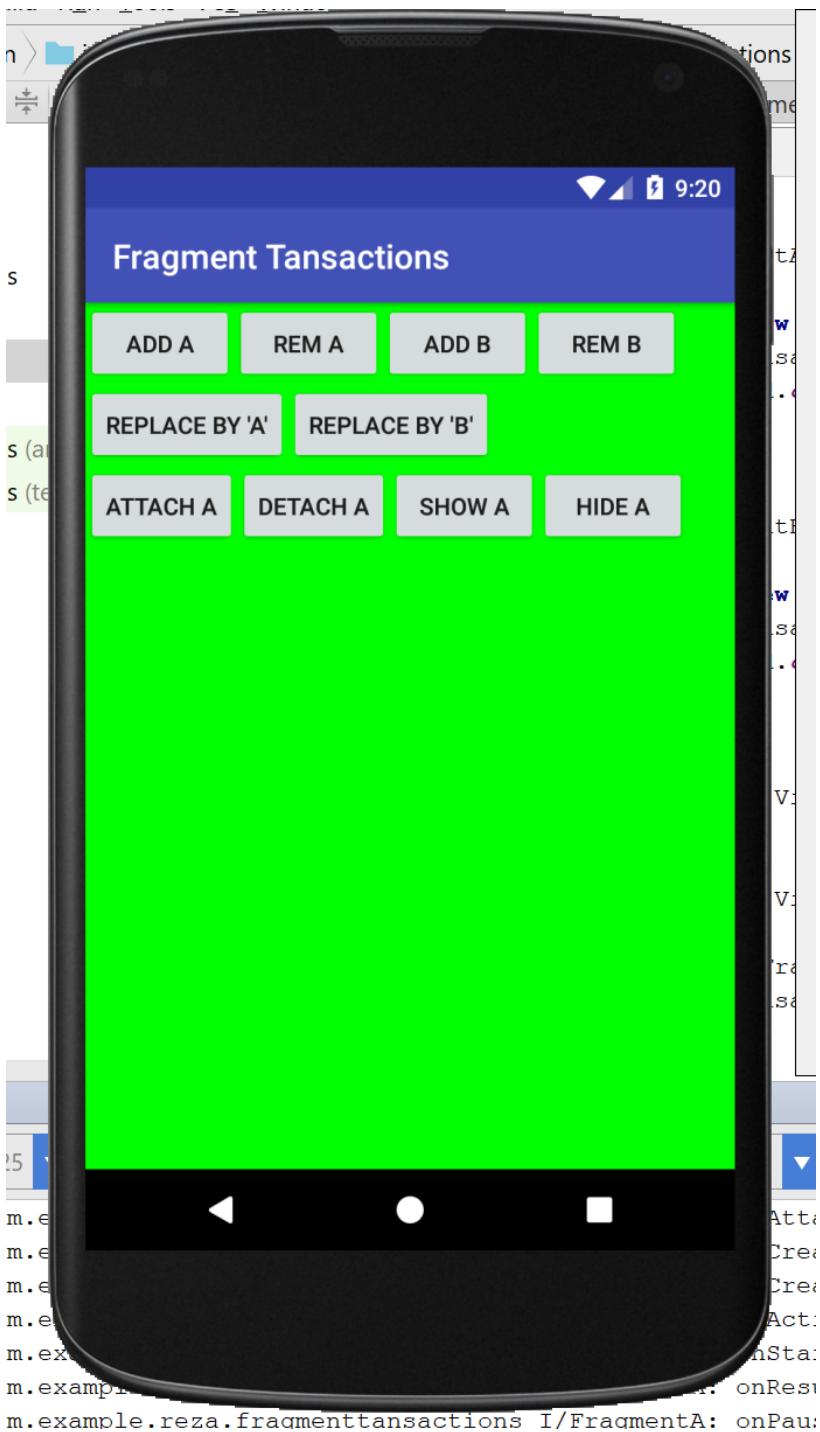
Notice the life cycle methods that were executed:

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The project is named "FragmentTransactions". The "app" module contains Java files for "MainActivity", "FragmentA", and "FragmentB", along with XML files for "activity_main.xml", "fragment_a.xml", and "fragment_b.xml".
- Code Editor:** The "MainActivity.java" file is open, showing code for replacing fragments. It includes methods like `replaceByFragmentA`, `replaceByFragmentB`, `attachFragmentA`, and `detachFragmentA`.
- Logcat:** The Logcat tab shows logs from the "Emulator Nexus_4_API_25" device. The logs list several fragment lifecycle events:

```
03-09 21:05:02.034 10513/com.example.reza.fragmenttransactions I/FragmentA: onAttach()
03-09 21:05:02.034 10513/com.example.reza.fragmenttransactions I/FragmentA: onCreate()
03-09 21:05:02.035 10513/com.example.reza.fragmenttransactions I/FragmentA: onCreateView()
03-09 21:05:02.035 10513/com.example.reza.fragmenttransactions I/FragmentA: onActivityCreated()
03-09 21:05:02.035 10513/com.example.reza.fragmenttransactions I/FragmentA: onStart()
03-09 21:05:02.035 10513/com.example.reza.fragmenttransactions I/FragmentA: onResume()
```
- Bottom Bar:** The bottom navigation bar includes tabs for Terminal, Logcat, Android Profiler, Messages, Run, TODO, Event Log, and Grade Console.

Now click “DETACH A” button:



```

FragmentTransactions - [C:\Android Demos\Fragments\FragmentTransactions] - [app] - ...\\app\\src\\main\\java\\com\\example\\reza\\fragmenttransactions\\MainActivity.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
FragmentTransactions app main java com example reza fragmenttransactions MainActivity
Android Project Structure Captures Gradle
1 Project app manifests java com.example.reza.fragmenttransactions FragmentA FragmentB MainActivity
Structure com.example.reza.fragmenttransactions (androidTest) com.example.reza.fragmenttransactions (test)
res drawable layout activity_main.xml fragment_a.xml fragment_b.xml mipmap values
Gradle Scripts
MainActivity.java
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98

public void replaceByFragmentA(View view) {
    FragmentA fragmentA = new FragmentA();
    FragmentTransaction transaction = manager.beginTransaction();
    transaction.replace(R.id.container, fragmentA, "fragA");
    transaction.commit();
}

public void replaceByFragmentB(View view) {
    FragmentB fragmentB = new FragmentB();
    FragmentTransaction transaction = manager.beginTransaction();
    transaction.replace(R.id.container, fragmentB, "fragB");
    transaction.commit();
}

public void attachFragmentA(View view) {
}

public void detachFragmentA(View view) {
    FragmentA fragmentA = (FragmentA) manager.findFragmentByTag("fragA");
    FragmentTransaction transaction = manager.beginTransaction();
    if (fragmentA != null) {
        transaction.detach(fragmentA);
        transaction.commit();
    }
}

```

Logcat

Emulator Nexus_4 API_25 Android 7.1.1, API 25 com.example.reza.fragmenttransactions (10513)

Verbose Fragment Regex Show only selected application

03-09 21:05:02.034 10513-10513/com.example.reza.fragmenttransactions I/Fragment: onAttach()
03-09 21:05:02.034 10513-10513/com.example.reza.fragmenttransactions I/Fragment: onCreate()
03-09 21:05:02.035 10513-10513/com.example.reza.fragmenttransactions I/Fragment: onCreateView()
03-09 21:05:02.035 10513-10513/com.example.reza.fragmenttransactions I/Fragment: onActivityCreated()
03-09 21:05:02.035 10513-10513/com.example.reza.fragmenttransactions I/Fragment: onStart()
03-09 21:05:02.035 10513-10513/com.example.reza.fragmenttransactions I/Fragment: onResume()
03-09 21:20:02.393 10513-10513/com.example.reza.fragmenttransactions I/Fragment: onPause()
03-09 21:20:02.393 10513-10513/com.example.reza.fragmenttransactions I/Fragment: onStop()
03-09 21:20:02.393 10513-10513/com.example.reza.fragmenttransactions I/Fragment: onDestroyView()

Terminal Logcat Android Profiler Messages Run TODO Event Log Gradle Console

Gradle build finished in 3s 747ms (16 minutes ago)

10:1 CRLF: UTF-8 Context: <no context>

The only 3 life cycle method which were called were, onPause(), onStop() and onDestroyView(). It is important to notice that the onDestroy() and onDetach() methods for fragment A were not called so even if the fragment A is not visible to the user, it is still alive and its instance remains in existence.

The next transaction method we are going to look at is the Attach() method. Now to use this method you have to perform the following steps:

First you need to get a reference to the detached fragment, then you use the attach() method to re-attach the fragment which was previously detached with calling the detach() method.

When you call the attach() method, the fragment's view is recreated and the fragment is reattached to the activity and becomes visible to the user.

What follows shows the code for attachFragmentA() method:

The screenshot shows the Android Studio interface with the project 'FragmentTransactions' open. The code editor displays the 'MainActivity.java' file, specifically the implementation of the 'attachFragmentA' method. The code uses a FragmentManager to find a fragment by tag ('fragA') and attach it to the current transaction if it exists. If the fragment is not found, a toast message is displayed. The code editor has syntax highlighting and line numbers. The bottom status bar shows 'Gradle build finished in 3s 747ms (24 minutes ago)'.

```
public void attachFragmentA(View view) {
    FragmentA fragmentA = (FragmentA) manager.findFragmentByTag("fragA");
    FragmentTransaction transaction = manager.beginTransaction();

    if (fragmentA != null) {
        transaction.attach(fragmentA);
        transaction.commit();
    } else {
        Toast.makeText(context, "Fragment A not Found", Toast.LENGTH_SHORT).show();
    }
}
```

And here is the code with a larger font:

```
public void attachFragmentA(View view) {

    FragmentA fragmentA = (FragmentA) manager.findFragmentByTag("fragA");
    FragmentTransaction transaction = manager.beginTransaction();

    if (fragmentA != null) {
        transaction.attach(fragmentA);
        transaction.commit();
    } else {
        Toast.makeText(context, "Fragment A not Found", Toast.LENGTH_SHORT).show();
    }
}
```

Now try the code with running the app:



Click “ADD A” button to add fragment A:



Then click “DETACH A” button :



The click “ATTACH A” button:



And here are the life cycle methods which were called:

The screenshot shows the Android Studio interface. The top navigation bar includes File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, Help. Below the toolbar, the Project tab is selected, showing the app module structure with Java files like FragmentTransactions.java, FragmentA.java, FragmentB.java, and MainActivity.java. The FragmentTransactions.java file has code for replacing and attaching fragments. The MainActivity.java file contains logic for attaching FragmentA. The Logcat window at the bottom displays log entries from the Emulator Nexus_4 API 25. The log entries show fragment lifecycle events such as onAttach(), onCreate(), onCreateView(), onStart(), onResume(), onPause(), onStop(), and onDestroyView().

Notice the call to the `onDestroyView()` method which was called when you clicked “DETACH A” button and `onCreateView()` method which was called when you clicked the “ATTACH A” button.

The last transaction methods we need to discuss are the `hide()` and `show()` transaction methods.

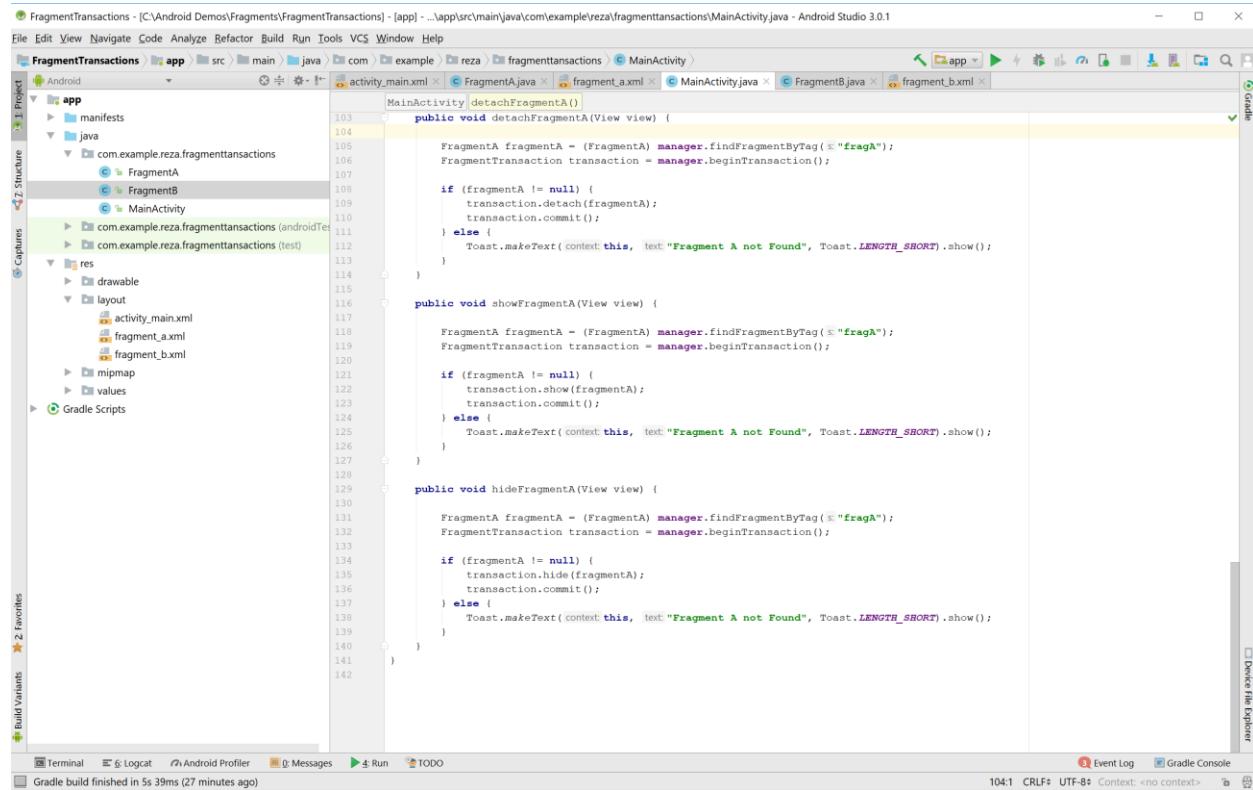
To perform the `hide()` transaction method on the fragment, we need to do the following:

First we need to get a reference to the fragment that we want to hide, then use `hide()` method to hide the fragment. When you call the `hide()` method, the fragment’s View becomes invisible but the fragment continues to be in the Resumed state and remains attached to the activity.

To perform the `show()` transaction method, we need to get a reference to a fragment that is currently hidden and we want to show it and then we use the `show()` method to make it visible to the user again. The fragment remains in the resumed state as it was earlier.

So there is no change in the state of a fragment when you use hide() and show() methods, All that happens is the fact that the fragment becomes visible or hidden otherwise it is still in the resumed state and attached to the activity.

What follows shows the code for showFragmentA() and hideFragmentA() methods:



The screenshot shows the Android Studio interface with the project 'FragmentTransactions' open. The code editor displays the MainActivity.java file, which contains three methods: detachFragmentA(), showFragmentA(), and hideFragmentA().

```
public void detachFragmentA(View view) {
    FragmentA fragmentA = (FragmentA) manager.findFragmentByTag("fragA");
    FragmentTransaction transaction = manager.beginTransaction();

    if (fragmentA != null) {
        transaction.detach(fragmentA);
        transaction.commit();
    } else {
        Toast.makeText(context, "Fragment A not Found", Toast.LENGTH_SHORT).show();
    }
}

public void showFragmentA(View view) {
    FragmentA fragmentA = (FragmentA) manager.findFragmentByTag("fragA");
    FragmentTransaction transaction = manager.beginTransaction();

    if (fragmentA != null) {
        transaction.show(fragmentA);
        transaction.commit();
    } else {
        Toast.makeText(context, "Fragment A not Found", Toast.LENGTH_SHORT).show();
    }
}

public void hideFragmentA(View view) {
    FragmentA fragmentA = (FragmentA) manager.findFragmentByTag("fragA");
    FragmentTransaction transaction = manager.beginTransaction();

    if (fragmentA != null) {
        transaction.hide(fragmentA);
        transaction.commit();
    } else {
        Toast.makeText(context, "Fragment A not Found", Toast.LENGTH_SHORT).show();
    }
}
```

The code uses a FragmentManager to find a fragment by tag ('fragA'). It then performs a beginTransaction() and either detach, show, or hide the fragment based on the method call. If the fragment is not found, a toast message is displayed indicating that 'Fragment A not Found'.

And here is the code with a larger font:

```
public void showFragmentA(View view) {

    FragmentA fragmentA = (FragmentA) manager.findFragmentByTag( s: "fragA");
    FragmentTransaction transaction = manager.beginTransaction();

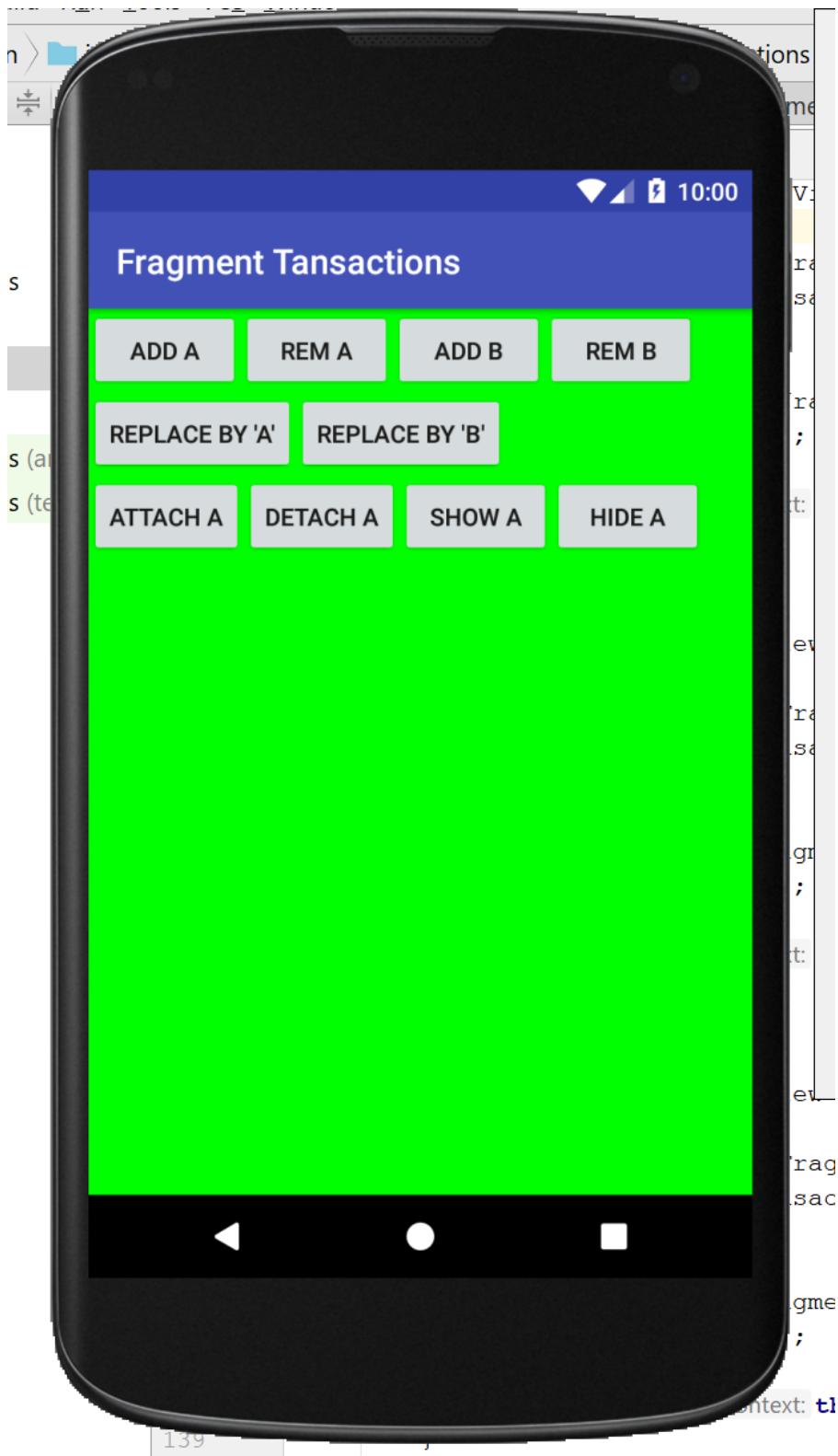
    if (fragmentA != null) {
        transaction.show(fragmentA);
        transaction.commit();
    } else {
        Toast.makeText( context: this, text: "Fragment A not Found", Toast.LENGTH_SHORT).show();
    }
}

public void hideFragmentA(View view) {

    FragmentA fragmentA = (FragmentA) manager.findFragmentByTag( s: "fragA");
    FragmentTransaction transaction = manager.beginTransaction();

    if (fragmentA != null) {
        transaction.hide(fragmentA);
        transaction.commit();
    } else {
        Toast.makeText( context: this, text: "Fragment A not Found", Toast.LENGTH_SHORT).show();
    }
}
```

Now let's run the app again:



Click “ADD A” button to add the fragment A:



Here the life cycle methods which were called:

The screenshot shows the Android Studio interface. The top navigation bar includes File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, and Help. The main window has tabs for FragmentTransactions, app, src, main, java, com, example, reza, fragmenttransactions, and MainActivity. The Project tool window on the left shows the app module with its manifest, Java files (MainActivity, FragmentA, FragmentB), and XML files (activity_main.xml, fragment_a.xml, fragment_b.xml). The Structure tool window is also visible. The bottom Logcat window displays log entries from the Emulator Nexus_4 API_25 Android 7.1.1, API 25. The log entries are as follows:

```
03-09 22:01:10.195 11317-11317/com.example.reza.fragmenttransactions I/FragmentA: onAttach()
03-09 22:01:10.196 11317-11317/com.example.reza.fragmenttransactions I/FragmentA: onCreate()
03-09 22:01:10.196 11317-11317/com.example.reza.fragmenttransactions I/FragmentA: onCreateView()
03-09 22:01:10.197 11317-11317/com.example.reza.fragmenttransactions I/FragmentA: onActivityCreated()
03-09 22:01:10.197 11317-11317/com.example.reza.fragmenttransactions I/FragmentA: onStart()
03-09 22:01:10.197 11317-11317/com.example.reza.fragmenttransactions I/FragmentA: onResume()
```

The bottom status bar indicates "Gradle build finished in 3s 536ms (a minute ago)".

Now click on “HIDE A” button:



And here is the Logcat window:

The screenshot shows the Android Studio interface with the FragmentTransactions project open. The Project tool window on the left shows the app module structure. The code editor displays MainActivity.java with the following code:

```
public void detachFragmentA(View view) {
    FragmentA fragmentA = (FragmentA) manager.findFragmentByTag("fragA");
    FragmentTransaction transaction = manager.beginTransaction();

    if (fragmentA != null) {
        transaction.detach(fragmentA);
        transaction.commit();
    } else {
        Toast.makeText(context, "Fragment A not Found", Toast.LENGTH_SHORT).show();
    }
}

public void showFragmentA(View view) {
    FragmentA fragmentA = (FragmentA) manager.findFragmentByTag("fragA");
    FragmentTransaction transaction = manager.beginTransaction();

    if (fragmentA != null) {
        transaction.attach(fragmentA);
        transaction.commit();
    } else {
        Toast.makeText(context, "Fragment A not Found", Toast.LENGTH_SHORT).show();
    }
}
```

The Logcat window at the bottom shows log entries from the Emulator Nexus_4 API_25 Android 7.1.1, API 25. The log entries are:

```
03-09 22:01:10,195 11317-11317/com.example.reza.fragmenttransactions I/FragmentA: onAttach()
03-09 22:01:10,196 11317-11317/com.example.reza.fragmenttransactions I/FragmentA: onCreate()
03-09 22:01:10,196 11317-11317/com.example.reza.fragmenttransactions I/FragmentA: onCreateView()
03-09 22:01:10,197 11317-11317/com.example.reza.fragmenttransactions I/FragmentA: onActivityCreated()
03-09 22:01:10,197 11317-11317/com.example.reza.fragmenttransactions I/FragmentA: onStart()
03-09 22:01:10,197 11317-11317/com.example.reza.fragmenttransactions I/FragmentA: onResume()
```

There is no change in the state of the fragment whatsoever another word no new methods were called when we clicked the “HIDE A” button. The fragment is not visible to the user but it remains in the resumed state.

Now click on “SHOW A” button:



And here is the Logcat window. There is no new calls, only the fragment became visible again but no life cycle methods were called another word the state of the fragment didn't change.

```

FragmentTransactions - [C:\Android Demos\Fragments\FragmentTransactions] - [app] - ...app\src\main\java\com\example\reza\fragmenttransactions\MainActivity.java - Android Studio 3.0.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
FragmentTransactions app src main java com example reza fragmenttransactions MainActivity
activity_main.xml FragmentA.java fragment_a.xml MainActionBar.java FragmentB.java fragment_b.xml
MainActivity.java
1 Project 2 Structure 3 Captures 4 Device File Explorer
app manifests java com.example.reza.fragmenttransactions FragmentA FragmentB MainActivity
res drawable layout activity_main.xml fragment_a.xml fragment_b.xml
mipmap
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
public void detachFragmentA(View view) {
    FragmentA fragmentA = (FragmentA) manager.findFragmentByTag("fragA");
    FragmentTransaction transaction = manager.beginTransaction();
    if (fragmentA != null) {
        transaction.detach(fragmentA);
        transaction.commit();
    } else {
        Toast.makeText(context, "Fragment A not Found", Toast.LENGTH_SHORT).show();
    }
}

public void showFragmentA(View view) {
    FragmentA fragmentA = (FragmentA) manager.findFragmentByTag("fragA");
    FragmentTransaction transaction = manager.beginTransaction();
    if (fragmentA != null) {
        transaction.attach(fragmentA);
        transaction.commit();
    }
}

```

Logcat

Emulator Nexus_4 API_25 Android 7.1.1, API 25 com.example.reza.fragmenttransactions (11317)

Verbose Fragment Regex Show only selected application

```

03-09 22:01:10.195 11317-11317/com.example.reza.fragmenttransactions I/FragmentA: onAttach()
03-09 22:01:10.196 11317-11317/com.example.reza.fragmenttransactions I/FragmentA: onCreate()
03-09 22:01:10.196 11317-11317/com.example.reza.fragmenttransactions I/FragmentA: onCreateView()
03-09 22:01:10.197 11317-11317/com.example.reza.fragmenttransactions I/FragmentA: onActivityCreated()
03-09 22:01:10.197 11317-11317/com.example.reza.fragmenttransactions I/FragmentA: onStart()
03-09 22:01:10.197 11317-11317/com.example.reza.fragmenttransactions I/FragmentA: onResume()

```

Build Variants Terminal & Logcat Android Profiler Messages Run TODO Event Log Gradle Console

Gradle build finished in 3s 536ms (5 minutes ago) 7:1 CRLF: UTF-8: Context: <no context>

And this concludes our demo app for the fragment transaction methods.