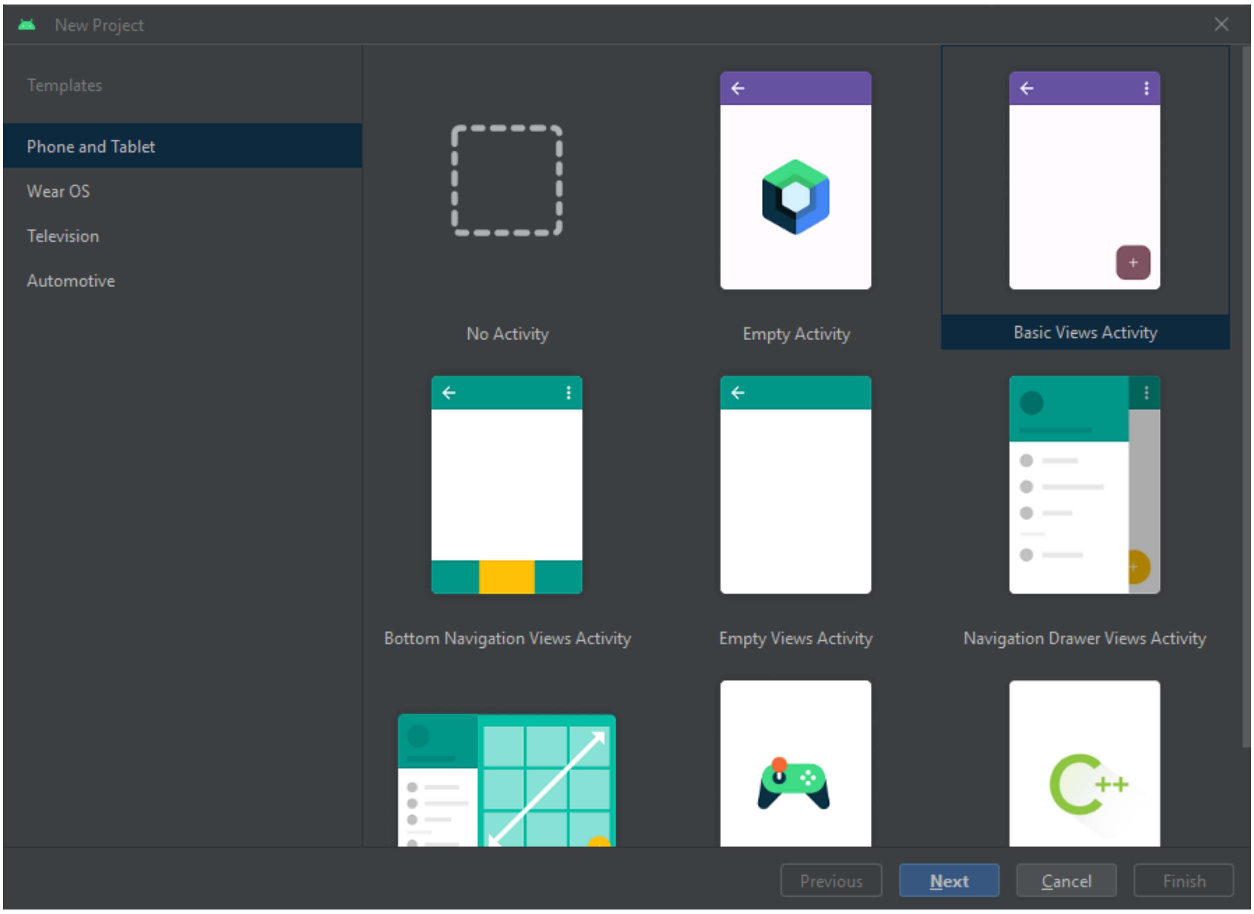
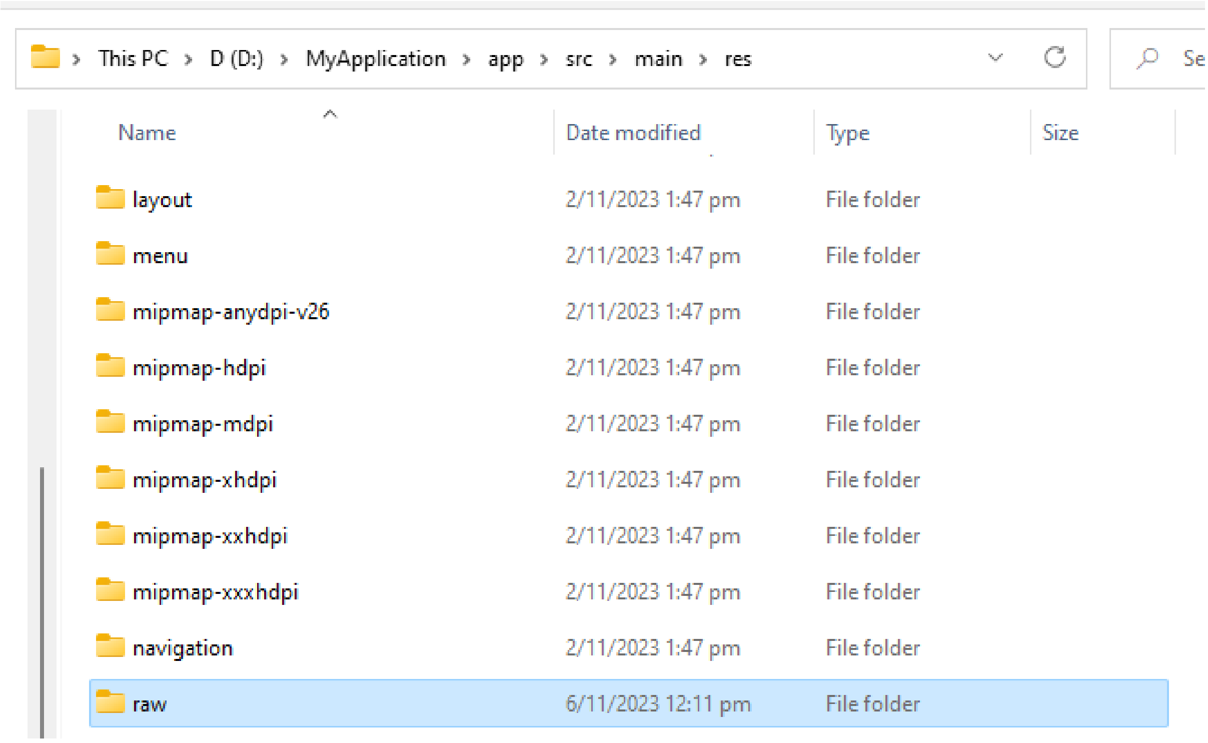
Using Audio and Images

Create a new project with Basic Views Activity



Use the same emulator you have used in the previous activities.

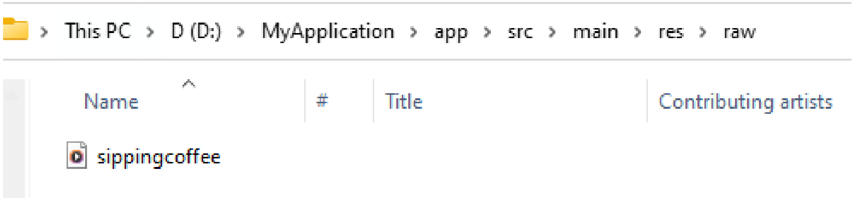
Create new folder named raw and place the given audio file inside the project folder.



This PC › D (D:) › MyApplication › app › src › main › res

Se

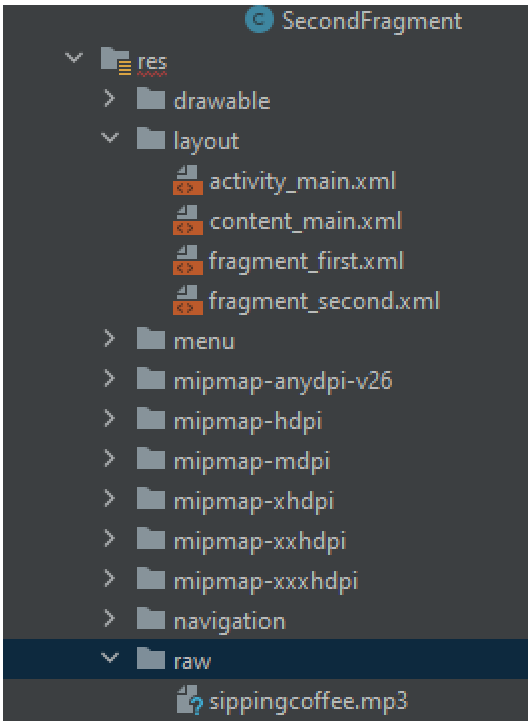
Name	Date modified	Type	Size
layout	2/11/2023 1:47 pm	File folder	
menu	2/11/2023 1:47 pm	File folder	
mipmap-anydpi-v26	2/11/2023 1:47 pm	File folder	
mipmap-hdpi	2/11/2023 1:47 pm	File folder	
mipmap-mdpi	2/11/2023 1:47 pm	File folder	
mipmap-xhdpi	2/11/2023 1:47 pm	File folder	
mipmap-xxhdpi	2/11/2023 1:47 pm	File folder	
mipmap-xxxhdpi	2/11/2023 1:47 pm	File folder	
navigation	2/11/2023 1:47 pm	File folder	
raw	6/11/2023 12:11 pm	File folder	



This PC › D (D:) › MyApplication › app › src › main › res › raw

Name	#	Title	Contributing artists
sippingcoffee			

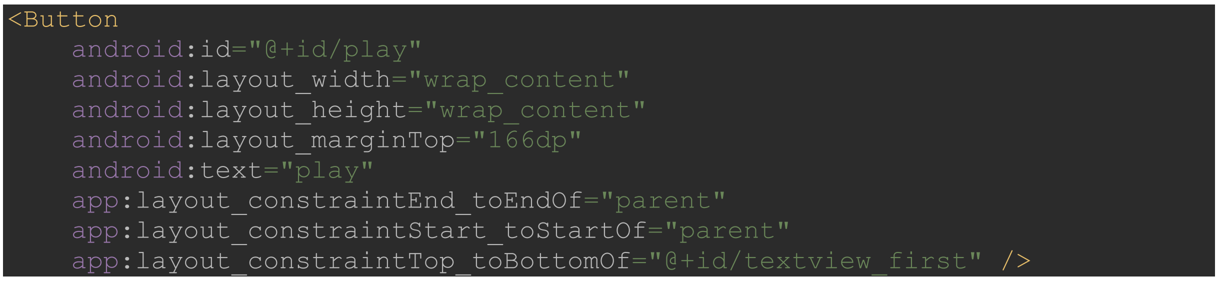
You can check the file inside Android Studio as well.



```
        SecondFragment
  ⌄ res
    › drawable
    ⌄ layout
        activity_main.xml
        content_main.xml
        fragment_first.xml
        fragment_second.xml
    › menu
    › mipmap-anydpi-v26
    › mipmap-hdpi
    › mipmap-mdpi
    › mipmap-xhdpi
    › mipmap-xxhdpi
    › mipmap-xxxhdpi
    › navigation
    ⌄ raw
        sippingcoffee.mp3
```

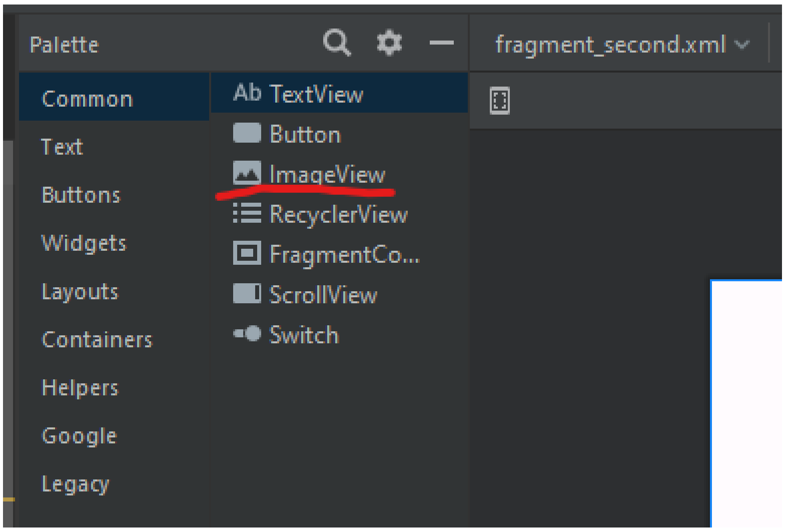
In fragment_first.xml inside res/layout folder,

Put in button. (refer to appendix to show the design of xml if unsure)

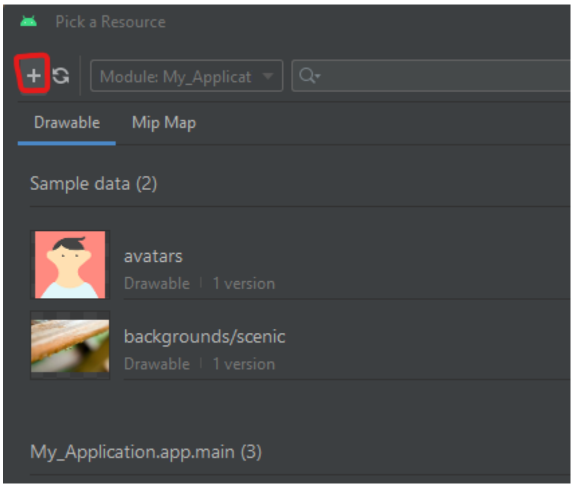


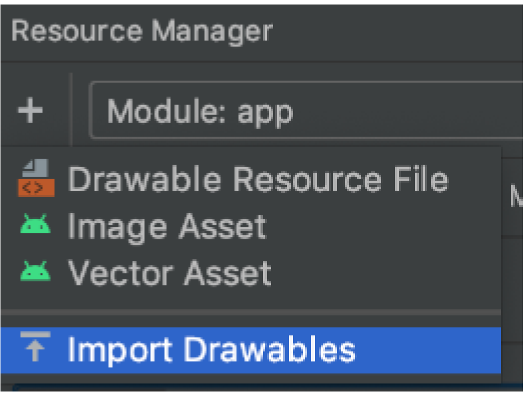
```
<Button
    android:id="@+id/play"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="166dp"
    android:text="play"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textview_first" />
```

In fragment_second.xml inside res/layout folder, show the design view and drag in an ImageView to the layout.

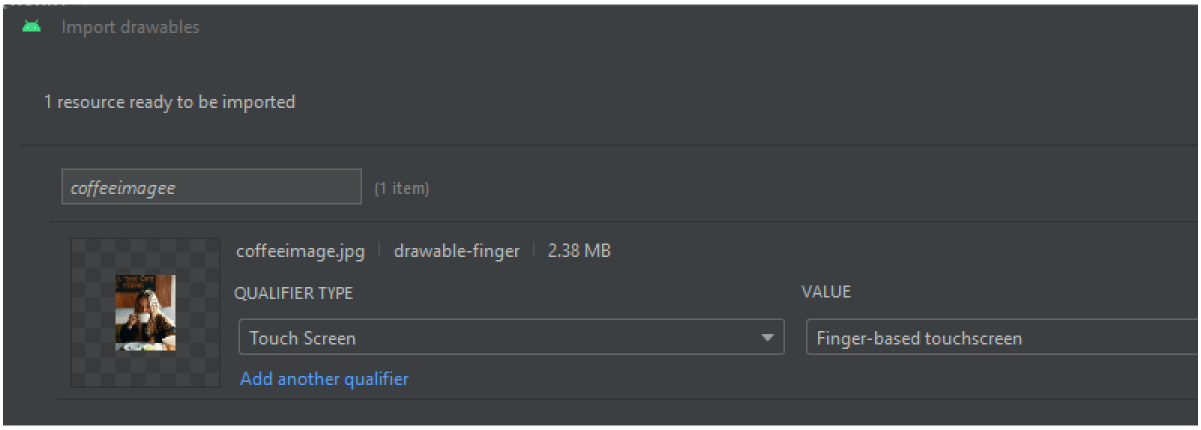


Click on the button and Import Drawables.





Select the jpeg provided and qualifier as so:



Import drawables

1 resource ready to be imported

coffeeimagee (1 item)

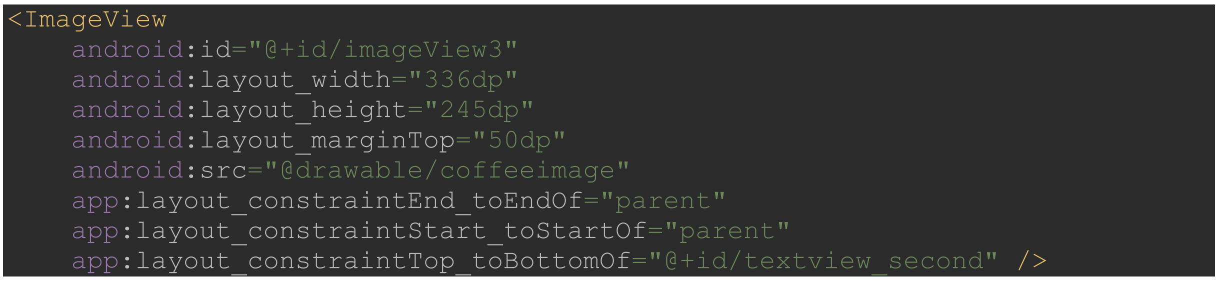
coffeeimage.jpg | drawable-finger | 2.38 MB

QUALIFIER TYPE: Touch Screen

VALUE: Finger-based touchscreen

Now the picture should be saved in a folder called drawable.

Edit the code inside fragment_second ImageView. You can also use the ConstraintLayout to manually edit the image location from the design screen.



```
<ImageView
    android:id="@+id/imageView3"
    android:layout_width="336dp"
    android:layout_height="245dp"
    android:layout_marginTop="50dp"
    android:src="@drawable/coffeeimage"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textview_second" />
```

Now all the front end has been completed, and now to the button functionalities.

In FirstFragment.java

```
import android.media.MediaPlayer;
import android.widget.Button;
```

inside FirstFragment public class, call MediaPlayer.

```
MediaPlayer mp;
```

and inside onCreateView() method, set up the MediaPlayer to call in the audio file and set up the button to start the audio file.

```
mp = MediaPlayer.create(getActivity(), R.raw.sippingcoffee);
Button play = (Button)binding.getRoot().findViewById(R.id.play);
```

Inside onViewCreated() method, put in the function of the play button.

```
binding.play.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        mp.start();
    }
});
```

The total code should look like this.

```
public class FirstFragment extends Fragment {
    MediaPlayer mp;
    private FragmentFirstBinding binding;
    @Override
    public View onCreateView(
            LayoutInflater inflater, ViewGroup container,
            Bundle savedInstanceState
    ) {
        binding = FragmentFirstBinding.inflate(inflater, container, false);
        mp = MediaPlayer.create(getActivity(), R.raw.sippingcoffee);
        Button play = (Button)binding.getRoot().findViewById(R.id.play);
        return binding.getRoot();
    }
    public void onViewCreated(@NonNull View view, Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        binding.buttonFirst.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                NavHostFragment.findNavController(FirstFragment.this)
                        .navigate(R.id.action_FirstFragment_to_SecondFragment);
            }
        });
        binding.play.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.start();
            }
        });

    }
    @Override
    public void onDestroyView() {
        super.onDestroyView();
        binding = null;
    }
}
```

In Second Fragment.java,

Similar to above, call the MediaPlayer in public class SecondFragment.

```
MediaPlayer mp;
```

Set up the mediaplayer inside onCreateView.

```
mp = MediaPlayer.create(getActivity(), R.raw.sippingcoffee);
```

set up ImageView and onClick function into the ImageView

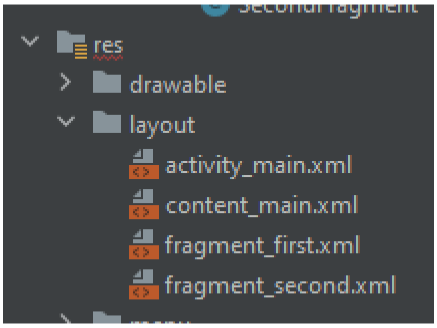
```
ImageView ci = (ImageView) binding.getRoot().findViewById(R.id.imageView3);
binding.imageView3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view)
    {
        mp.start();
    }
});
```

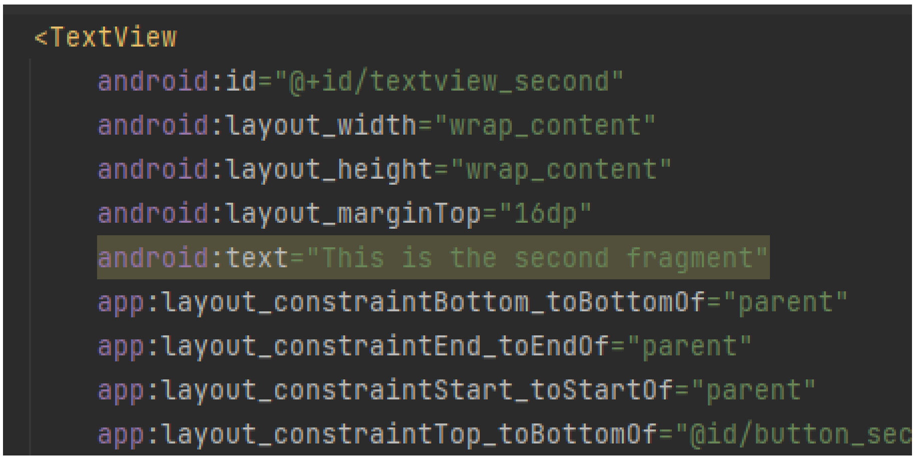
The total code should look like this

```
public class SecondFragment extends Fragment {
    MediaPlayer mp;
    private FragmentSecondBinding binding;

    @Override
    public View onCreateView(
            LayoutInflater inflater, ViewGroup container,
            Bundle savedInstanceState
    ) {
        binding = FragmentSecondBinding.inflate(inflater, container, false);
        mp = MediaPlayer.create(getActivity(), R.raw.sippingcoffee);
        return binding.getRoot();
    }

    public void onViewCreated(@NonNull View view, Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);
        binding.buttonSecond.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                NavHostFragment.findNavController(SecondFragment.this)
                        .navigate(R.id.action_SecondFragment_to_FirstFragment);
            }
        });
        ImageView ci = (ImageView) binding.getRoot().findViewById(R.id.imageView3);
        binding.imageView3.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view)
            {
                mp.start();
            }
        });
    }
    @Override
    public void onDestroyView() {
        super.onDestroyView();
        binding = null;
    }
}
```

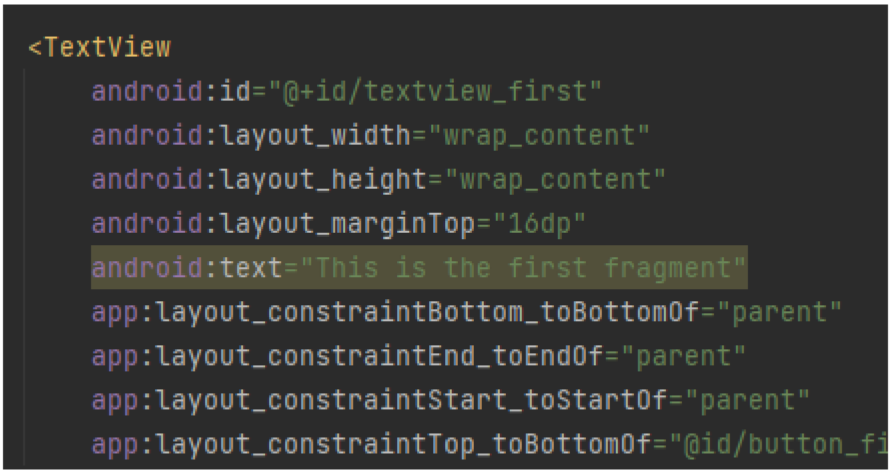
Now, run the app.

For easier viewing, I changed the TextView of fragment_second.xml and fragment_first.xml





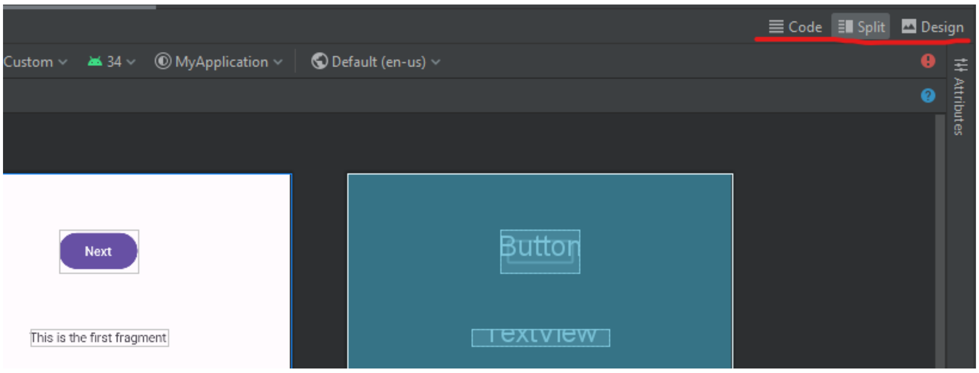
```
<TextView
    android:id="@+id/textview_second"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="This is the second fragment"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/button_sec
```



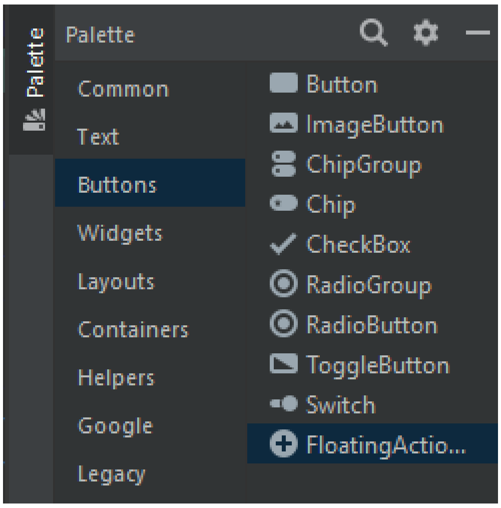
```
<TextView
    android:id="@+id/textview_first"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:text="This is the first fragment"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/button_fi
```

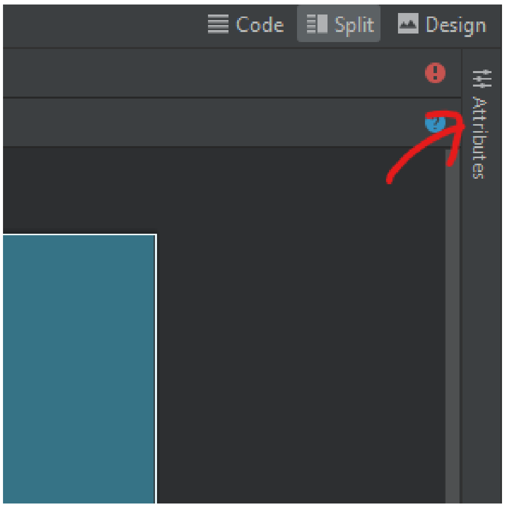
Appendix

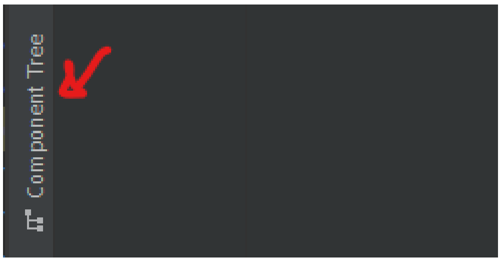
Open split view



From here, open the Component Tree and Attributes.







Sound Effect from <a href="https://pixabay.com/?utm_source=link-attribution&utm_medium=referral&utm_campaign=music&utm_content=95226">Pixabay</a>