

Practical - Android

Develop an application for working with graphics and animation.

XML Code:

Here, along with the layout xml file 4 other xml files are to be created which define the Shapes and Animation Transition Effect Data

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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="ty.practical7.MainActivity" android:id="@+id/mainLayout">

<ImageView
    android:id="@+id/imgShape1" android:layout_width="150dp"
    android:layout_height="150dp" app:srcCompat="@drawable/shape1"
    android:layout_alignParentTop="true" android:layout_alignParentEnd="true" />

<ImageView
    android:id="@+id/imgShape2" android:layout_width="200dp"
    android:layout_height="200dp" android:layout_alignEnd="@+id/imgShape1"
    android:layout_alignParentBottom="true" />

<Button
    android:id="@+id/btnAnimation" android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:layout_marginBottom="75dp"
    android:layout_marginEnd="32dp" android:onClick="Animation"
    android:text="Animation" android:layout_alignParentBottom="true"
    android:layout_toStartOf="@+id/imgShape2" />

</RelativeLayout>
```

shape1.xml

shape2.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android" android:shape="oval"
>

<solid android:color="#3F51B5" />

<size android:width="300dp" android:height="300dp"></size> </shape>
```

```

<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
android:shape="rectangle" >

<solid android:color="#303F9F" />

<size android:width="300dp" android:height="300dp"></size> </shape>
shape3.xml

```

shape_transition.xml

Image: Add a drawable object an Image in the res/drawable folder, here for example an image file name demo.png is used.

Project Structure:

Source Code:

```

<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android" android:shape="oval"
>

<solid android:color="#FF4081" />

<size android:width="300dp" android:height="300dp"></size> </shape>

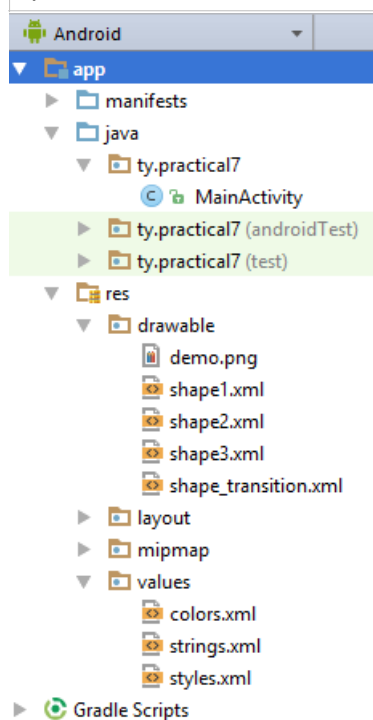
```

```

<?xml version="1.0" encoding="utf-8"?>
<transition xmlns:android="http://schemas.android.com/apk/res/android" >
<item android:drawable="@drawable/shape1" android:right="100dp"/> <item
android:drawable="@drawable/shape3" android:left="100dp"/>

</transition>

```



```

public class MainActivity extends AppCompatActivity {

    @Override

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

        DrawableGraphic();
        ShapeDrawableGraphic();
    }
    public void DrawableGraphic(){

        //Get Parent Layout
        RelativeLayout rl = (RelativeLayout)findViewById(R.id.mainLayout); //Create a dynamic
        ImageView object to place it within the Relative Layout

        ImageView demoimg = new ImageView(MainActivity.this); //Get the Image Source from
        @drawable, here demo.png

        demoimg.setImageResource(R.drawable.demo);
        //Specify the placement of ImageView within the RelativeLayout

        RelativeLayout.LayoutParams params = new RelativeLayout.LayoutParams(300,

        //Add rule to align the image to the left of the parent.

        params.addRule(RelativeLayout.ALIGN_PARENT_LEFT); demoimg.setLayoutParams(params);

        //Add ImageView within the Relative Layout
        rl.addView(demoimg); }

    public void ShapeDrawableGraphic() {

        //Assign Shape Properties
        int alpha = 127; int width = 300; int height = 300; int padding = 10;

        //Get Parent Layout
        RelativeLayout rl = (RelativeLayout)findViewById(R.id.mainLayout);

        // Create Shape 2
        ShapeDrawable shape2 = new ShapeDrawable(new RectShape());
        shape2.getPaint().setColor(Color.CYAN); shape2.setIntrinsicHeight(height);
        shape2.setIntrinsicWidth(width);

        shape2.setAlpha(alpha);

        // Put Shape 2 into an ImageView
        ImageView shape2View = new ImageView(getApplicationContext());
        shape2View.setImageDrawable(shape2); shape2View.setPadding(padding, padding, padding,
        padding);

        //Specify the placement of ImageView within the RelativeLayout

```

```
RelativeLayout.LayoutParams s2params = new RelativeLayout.LayoutParams(height, width);
```

```
//Add rule to align the image to the left of the parent.
```

```
s2params.addRule(RelativeLayout.CENTER_IN_PARENT);  
shape2View.setLayoutParams(s2params );
```

```
//Add ImageView within the Relative Layout  
rl.addView(shape2View); }
```

```
public void Animation(View v){
```

```
//Get the Shape Transition Drawable Objects TransitionDrawable transition =  
(TransitionDrawable)
```

```
ResourcesCompat.getDrawable(getResources(), R.drawable.shape_transition,  
null); //Apply Effect
```

```
transition.setCrossFadeEnabled(true); //Assign the effect to an ImageView object
```

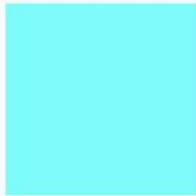
```
((ImageView) findViewById  
(R.id.imgShape2)).setImageDrawable(transition); //Set the Transition Effect  
Time
```

```
transition.startTransition(5000); }
```

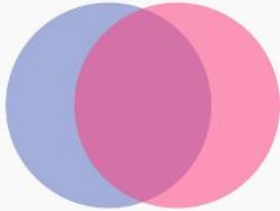
```
}
```

Output:

Practical7



ANIMATION



Practical7



ANIMATION