**4. Android Zoom In / Out Animations with Examples**

In android, **Zoom In** and **Zoom Out** animations are used to change the appearance and behaviour of the objects over a particular interval of time. The Zoom In and Zoom Out animations will provide a better look and feel for our applications.

Generally, the animations are useful when we want to notify users about the change’s happening in our app, such as new content loaded or new actions available, etc.

To create an animation effect to the objects in our android application, we need to follow below steps.

**Create XML File to Define Animation**

We need to create an xml file that defines the type of animation to perform in a new folder **anim** under **res** directory (**res**  **anim**  **zoom\_in.xml**) with required properties. In case **anim** folder not exists in **res** directory, create a new one.

To use **Zoom In** or **Zoom Out** animations in our android applications, we need to define a new xml files with **<scale>** tag like as shown below.

For **Zoom In** animation, we need to set android:pivotX="50%" and android:pivotY="50%" to perform the zoom from centre of the element. Also we need to use fromXScale, fromYScale attributes to define the scaling of object and we need keep these value lesser than toXScale, toYScale like as shown below.

<?xml version="1.0" encoding="utf-8"?>  
<set xmlns:android="http://schemas.android.com/apk/res/android">  
    <scale  
        xmlns:android="http://schemas.android.com/apk/res/android"  
        android:duration="1000"  
        android:fromXScale="2"  
        android:fromYScale="2"  
        android:pivotX="50%"  
        android:pivotY="50%"  
        android:toXScale="4"  
        android:toYScale="4" >  
    </scale>  
</set>

**Zoom Out** animation is same as **Zoom In** animation but fromXScale, fromYScale attribute values must be greater than toXScale, toYScale like as shown below.

<?xml version="1.0" encoding="utf-8"?>  
<set xmlns:android="http://schemas.android.com/apk/res/android">  
    <scale  
        android:duration="2500"  
        android:fromXScale="1.0"  
        android:fromYScale="1.0"  
        android:pivotX="50%"  
        android:pivotY="50%"  
        android:toXScale=".2"  
        android:toYScale=".2" />  
</set>

Once we are done with creation of required animation XML files, we need to load those animation files using different properties.

**Android Load and Start the Animation**

In android, we can perform animations by using **AnimationUtils** component methods such as **loadAnimation()**. Following is the code snippet of loading and starting an animation using **loadAnimation()** and **startAnimation()**methods.

ImageView img = (ImageView)findViewById(R.id.imgvw);

Animation aniSlide = AnimationUtils.loadAnimation(getApplicationContext(),R.anim.zoom\_in);  
img.startAnimation(aniSlide);

If we observe above code snippet, we are adding an animation to the image using **loadAnimation()** method. The second parameter in **loadAnimation()** method is the name of our animation xml file.

Here we used another method **startAnimation()** to apply the defined animation to imageview object.

Now we will see how to implement zoom in and zoom out animations for imageview on button click in android applications with examples.

**Android Zoom In & Zoom Out Animations Example**

Following is the example of implementing a zoom in and zoom out animations to change the image size on button click in android applications.

Create a new android application using android studio and give names as **ZoomInOutExample**. In case if you are not aware of creating an app in android studio check this article [Android Hello World App](https://www.tutlane.com/tutorial/android/android-hello-world-app-example).

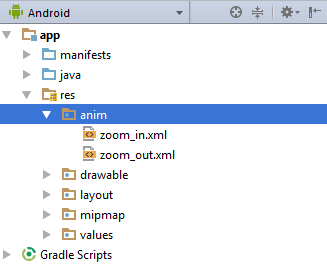
Once we create an application, open **activity\_main.xml** file from **\res\layout** folder path and write the code like as shown below.

**activity\_main.xml**

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout\_width="match\_parent"  
    android:layout\_height="match\_parent"  
    android:paddingLeft="10dp"  
    android:paddingRight="10dp">  
    <ImageView android:id="@+id/imgvw"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="250dp"  
        android:src="@drawable/bangkok"/>  
    <Button  
        android:id="@+id/btnZoomIn"  
        android:layout\_below="@+id/imgvw"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:text="Zoom In" android:layout\_marginLeft="100dp" />  
    <Button  
        android:id="@+id/btnZoomOut"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:layout\_alignBottom="@+id/btnZoomIn"  
        android:layout\_toRightOf="@+id/btnZoomIn"  
        android:text="Zoom Out" />  
</RelativeLayout>

As discussed, we need to create an xml files to define zoom in and zoom out animations in new folder **anim** under **res**directory (**res**  **anim**  **zoom\_in.xml**, **zoom\_out.xml**) with required properties. In case **anim** folder not exists in **res**directory, create a new one.

Following is the example of creating an XML files (**zoom\_in.xml**, **zoom\_out.zml**) under **anim** folder to define zoom in / out animation properties.



Now open **zoom\_in.xml** file and write the code to set zoom in animation properties like as shown below

**zoom\_in.xml**

<?xml version="1.0" encoding="utf-8"?>  
<set xmlns:android="http://schemas.android.com/apk/res/android">  
    <scale  
        xmlns:android="http://schemas.android.com/apk/res/android"  
        android:duration="1000"  
        android:fromXScale="2"  
        android:fromYScale="2"  
        android:pivotX="50%"  
        android:pivotY="50%"  
        android:toXScale="4"  
        android:toYScale="4" >  
    </scale>  
</set>

Now open **zoom\_out.xml** file and write the code to set zoom out animation properties like as shown below

**zoom\_out.xml**

<?xml version="1.0" encoding="utf-8"?>  
<set xmlns:android="http://schemas.android.com/apk/res/android">  
    <scale  
        android:duration="2500"  
        android:fromXScale="1.0"  
        android:fromYScale="1.0"  
        android:pivotX="50%"  
        android:pivotY="50%"  
        android:toXScale=".2"  
        android:toYScale=".2" />  
</set>

Now open your main activity file **MainActivity.java** from **\java\com.tutlane.zoominoutexample**

 path and write the code like as shown below.

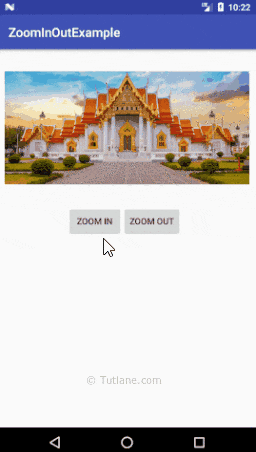
**MainActivity.java**

package com.sarker.zoominoutexample;  
import android.media.Image;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.view.animation.Animation;  
import android.view.animation.AnimationUtils;  
import android.widget.Button;  
import android.widget.ImageView;  
  
public class MainActivity extends AppCompatActivity {  
    private Button btnzIn;  
    private Button btnzOut;  
    private ImageView img;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity\_main);  
        btnzIn = (Button)findViewById(R.id.btnZoomIn);  
        btnzOut = (Button)findViewById(R.id.btnZoomOut);  
        img = (ImageView)findViewById(R.id.imgvw);  
        btnzIn.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                Animation animZoomIn = AnimationUtils.loadAnimation(getApplicationContext(),R.anim.zoom\_in);  
                img.startAnimation(animZoomIn);  
            }  
        });  
        btnzOut.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                Animation animZoomOut = AnimationUtils.loadAnimation(getApplicationContext(),R.anim.zoom\_out);  
                img.startAnimation(animZoomOut);  
            }  
        });  
    }  
}

If we observe above code, we are adding an animation to the image using **loadAnimation()** method and used **startAnimation()** method to apply the defined animation to imageview object.

**Output of Android Zoom In / Out Animations Example**

When we run above program in android studio we will get the result like as shown below.



If we observe above result, whenever we are clicking on **Zoom In** or **Zoom Out** buttons, the image size varies based on our functionality.

This is how we can implement zoom in and zoom out animations for imageview in android applications based on our requirements.