# UNIT III: MULTIMEDIA, ANIMATION AND GRAPHICS

## 3.1 Playing Audio, Video.

We can play and control the audio files in android by the help of **MediaPlayer class**.
**MediaPlayer** class we can easily fetch, decode and play both audio and video files with minimal setup.

The android media framework provides built-in support for playing a variety of common media types, such as audio or video. We have multiple ways to play audio or video but the most important component of media framework is **MediaPlayer** class.
**MediaPlayer** class we can access audio or video files from application (raw) resources, standalone files in file system or from a data stream arriving over a network connection and play audio or video files with the multiple playback options such as play, pause, forward, backward, etc.

```
MediaPlayer mPlayer = MediaPlayer.create(this, R.raw.baitikochi_chuste);
mPlayer.start();
```

The second parameter in **create()** method is the name of the song that we want to play from our application resource directory (**res/raw**).
create a new **raw** folder under **res** directory and add a properly encoded and formatted media files in it.

There are list of methods of MediaPlayer class.

1. **public void setDataSource(String path)-** sets the data source (file path or http url) to use.
2. **public void prepare()-**prepares the player for playback synchronously.
3. **public void start()-**it starts or resumes the playback.
4. **public void stop()-**it stops the playback.
5. **public void pause()-**it pauses the playback.
6. **public void seekTo(int millis)-** seeks to specified time in miliseconds.
7. **public int getDuration()-**returns duration of the file.
8. **public void setVolume(float leftVolume,float rightVolume)-** sets the volume on this player.

## 3.2 Rotate Animation

**Rotate** animation is used to change the appearance and behavior of the objects over a particular interval of time. The Rotate animation will provide a better look and feel for our applications.
the animations are useful when we want to notify users about the changes happening in our app, such as new content loaded or new actions available, etc.

**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** à **anim** à **rotate.xml**) with required properties. In case, if **anim** folder does not exist in **res** directory, create a new one.
To use **Rotate** animation in our android applications, we need to define a new xml file with **<rotate>** tag
To **Rotate** animation in **Clockwise**, we need to set android:fromDegrees and android:toDegrees property values and these will defines a rotation angles

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
android:interpolator="@android:anim/cycle_interpolator">
<rotate
android:fromDegrees="0"
android:toDegrees="360"
android:pivotX="50%"
android:pivotY="50%"
android:duration="5000" /> </set>
```

To **Rotate** animation in **Anti Clockwise**, we need to set android:fromDegrees and android:toDegrees property values and these will defines a rotation angles

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
android:interpolator="@android:anim/cycle_interpolator">
<rotate
android:fromDegrees="360"
android:toDegrees="0"
android:pivotX="50%"
android:pivotY="50%"
android:duration="5000" /> </set>
```

## 3.3 FadeIn / FadeOut Animation.

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

To use **Fade In** or **Fade Out** animations in our android applications, we need to define a new XML file with **<alpha>** tag

**Fade In** animation

```
<?xml version="1.0" encoding="utf-8"?> <set
xmlns:android="http://schemas.android.com/apk/res/android"
android:interpolator="@android:anim/linear_interpolator">
<alpha
android:duration="2000"
android:fromAlpha="0.1"
android:toAlpha="1.0">
</alpha>
</set>
```

**Fade Out** animation

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
android:interpolator="@android:anim/linear_interpolator">
<alpha android:duration="2000"
android:fromAlpha="1.0"
android:toAlpha="0.1" >
</alpha>
 </set>
```

```
ImageView img = (ImageView)findViewById(R.id.imgvw);
Animation aniFade =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.fade_in);
img.startAnimation(aniFade);
```

## 3.3 Zoom Animation.

### Zoom Animation

Zoom In and Zoom Out animations are used to enlarge and reduce the size of a view in Android applications respectively. These types of animations are often used by developers to provide a dynamic nature to the applications.
To use **Zoom In** or **Zoom Out** animations in our android applications, we need to define new XML files with **<scale>** tag

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

```
<?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

```
<?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>
```

**The End**