LECTURE 21: ANIMATIONS

BY LINA HAMMAD & AHMAD BARGHASH

In this lecture, we will learn how to add animations to the application elements.

Source: Geeks for Geeks

ANIMATIONS

- Animation is a method in which a collection of images are combined in a specific way and processed then they appear as moving images. Building animations make on-screen objects seems to be alive.

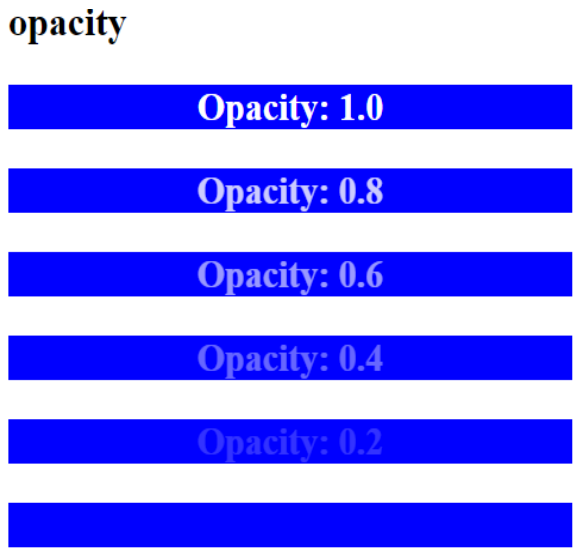
- Android has quite a few tools to help you create animations with relative ease. so in this lecture we will learn to create animations using Kotlin.

XML ATTRIBUTES I

- Below are some attributes which we are using while writing the code in xml.

XML ATTRIBUTES	DESCRIPTION
android:duration	It is used to specify the duration of animation to run
android:fromAlpha	It is the starting alpha value for the animation, where 1.0 means fully opaque and 0.0 means fully transparent
android:toAlpha	It is the ending alpha value
android:id	Sets unique id of the view
android:fromYDelta	It is the change in Y coordinate to be applied at the start of the animation
android:toYDelta	It is the change in Y coordinate to be applied at the end of the animation
android:startOffset	Delay occur when an animation runs (in miliseconds), once start time is reached.

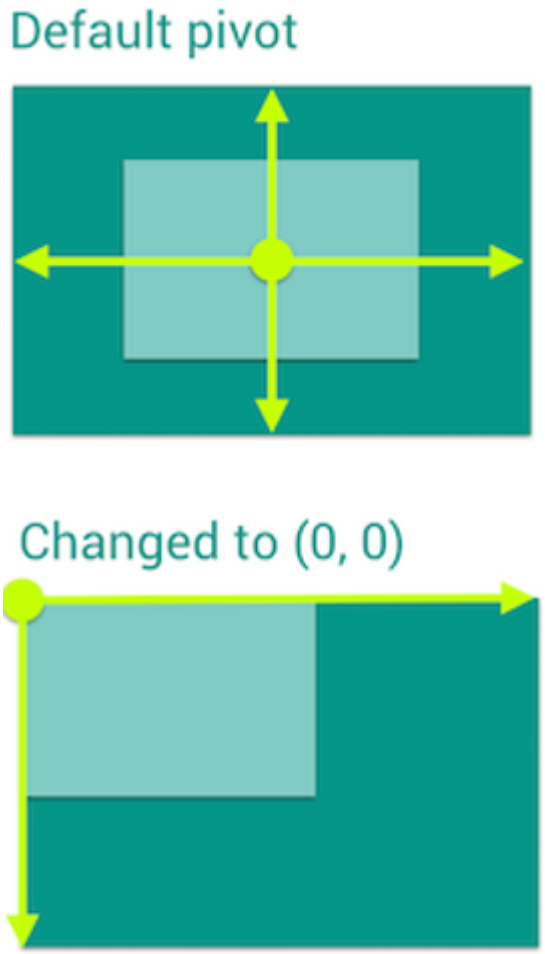
- Opacity is the characteristic of being difficult to understand or unclear.



XML ATTRIBUTES 2

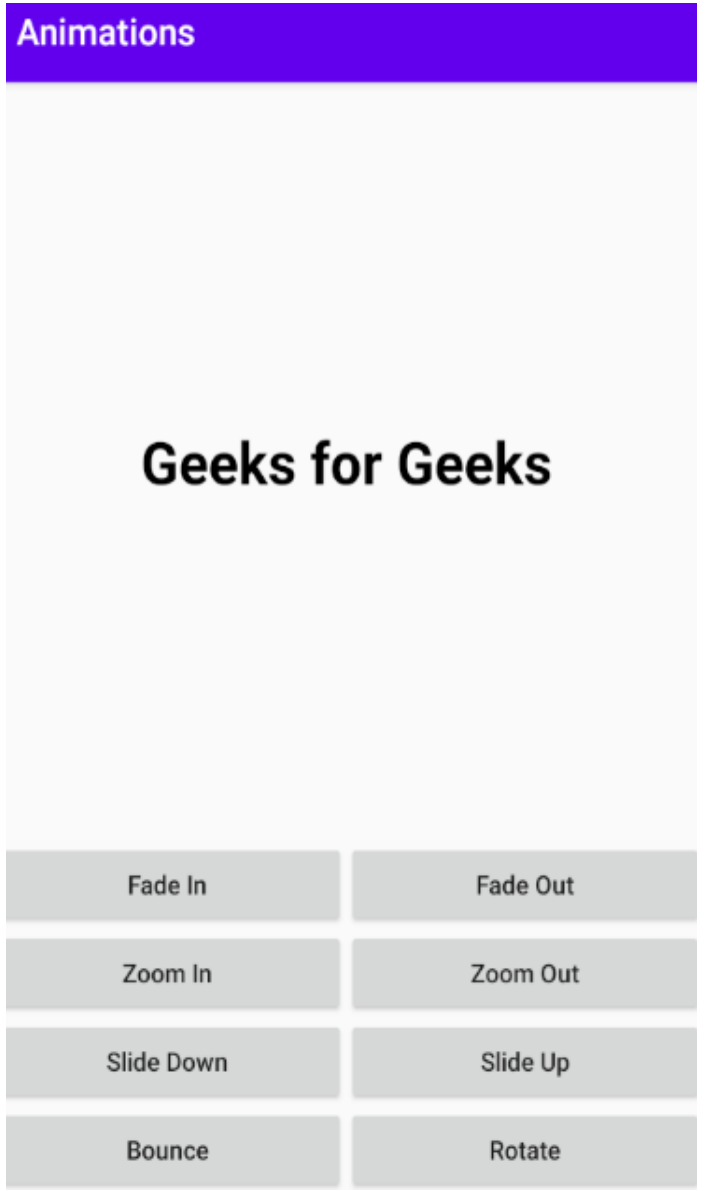
- Below are some attributes which we are using while writing the code in xml.

XML ATTRIBUTES	DESCRIPTION
android:pivotX	It represents the X-axis coordinates to zoom from starting point.
android:pivotY	It represents the Y-axis coordinates to zoom from starting point.
android:fromXScale	Starting X size offset,
android:fromYScale	Starting Y size offset,
android:toXScale	Ending of X size offset
android:toYScale	Ending of Y size offset
android:fromDegrees	Starting angular position, in degrees.



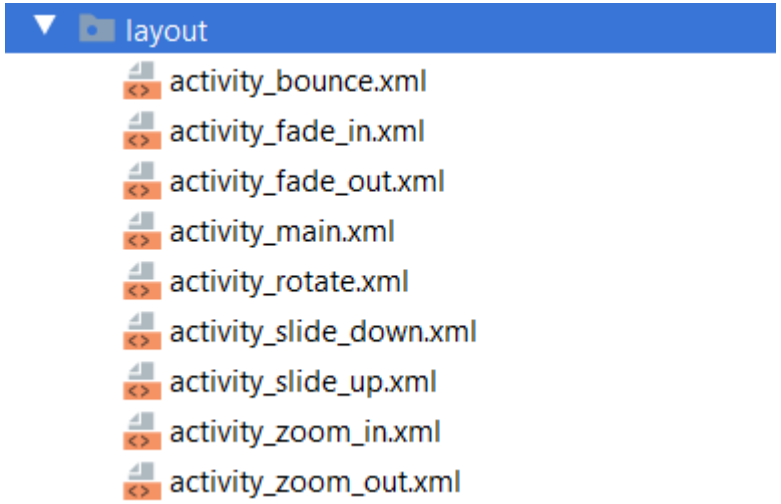
Step 1 : activity_main.xml

- After creating project we will modify xml files. In xml file we will create one *TextView* where all the animations are performed and *Eight Buttons* for Eight different animations.



Step2: Create The XML Files

- After modifying the activity_main.xml, we will create xml files for *animations*. so we will first create a folder name *anim*.
- In this folder, we will be adding the XML files which will be used to produce the animations. For this to happen, go to app/res right click and then select, *Android Resource Directory* and name it as anim.
- The XML files are:
  - bounce.xml
  - Fade_in.xml
  - fade_out.xml
  - rotate.xml
  - slide_down.xml
  - slide_up.xml
  - zoom_in.xml
  - zoom_out.xml



bounce.xml

- In this animation the text is bounce like a ball.

```
<?xml version="1.0" encoding="utf-8"?>
<set
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/linear_interpolator"
    android:fillAfter="true">
    <translate
        android:fromYDelta="100%"
        android:toYDelta="-20%"
        android:duration="300" />
    <translate
        android:startOffset="500"
        android:fromYDelta="-20%"
        android:toYDelta="10%"
        android:duration="150" />
    <translate
        android:startOffset="1000"
        android:fromYDelta="10%"
        android:toYDelta="0"
        android:duration="100" />
</set>
```

fade_in.xml

- In Fade In animation the text will appear from background.

```
<?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="1000"
        android:fromAlpha="0.1"
        android:toAlpha="1.0" />
</set>
```

fade_out.xml

- In Fade Out animation the color of text is faded for a particular amount of time.

```
<?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="1000"
        android:fromAlpha="1.0"
        android:toAlpha="0.1" />
</set>
```

rotate.xml

- In rotate animation the text is rotated for a particular amount of time.

```
<?xml version="1.0" encoding="utf-8"?>
<rotate xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="1000"
    android:fromDegrees="0"
    android:interpolator="@android:anim/linear_interpolator"
    android:pivotX="50%"
    android:pivotY="50%"
    android:startOffset="0"
    android:toDegrees="360" />
```

slide_down.xml

- In this animation the text will come from top to bottom.

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <translate
        android:duration="1000"
        android:fromYDelta="-100%"
        android:toYDelta="0" />
</set>
```

slide_up.xml

- In this animation the text will go from bottom to top.

```
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <translate
        android:duration="1000"
        android:fromYDelta="0"
        android:toYDelta="-100%" />
</set>
```

zoom_in.xml

- In this animation the text will appear bigger for a particular amount of time.

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:fillAfter="true">
    <scale xmlns:android="http://schemas.android.com/apk/res/android"
        android:duration="1000"
        android:fromXScale="1"
        android:fromYScale="1"
        android:pivotX="50%"
        android:pivotY="50%"
        android:toXScale="1.5"
        android:toYScale="1.5">
    </scale>
</set>
```

zoom_out.xml

- In this animation the text will appear smaller for a particular amount of time.

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:fillAfter="true" >
    <scale
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:duration="1000"
        android:fromXScale="1.0"
        android:fromYScale="1.0"
        android:pivotX="50%"
        android:pivotY="50%"
        android:toXScale="0.5"
        android:toYScale="0.5" >
    </scale>
</set>
```

MainActivity.kt file

- The magic relays in the AnimationUtils.loadAnimation

  AnimationUtils is a public class that Defines common utilities for working with animations.

  It has lots of public methods such as **loadAnimation** which Loads an Animation object from a resource

  One example:

```
val animationZoomIn = AnimationUtils.loadAnimation(this, R.anim.zoom_in)
textView.startAnimation(animationZoomIn)
```

MainActivity.kt file

- After creating all animations in xml. we will create MainActivity.kt

```
// bounce button
bounce.setOnClickListener {
    val animationBounce = AnimationUtils.loadAnimation(this, R.anim.activity_bounce)
    textView.startAnimation(animationBounce)
}

// fade in button
fade_in.setOnClickListener {
    textView.visibility = View.VISIBLE
    val animationFadeIn = AnimationUtils.loadAnimation(this, R.anim.activity_fade_in)
    textView.startAnimation(animationFadeIn)
}
```

MainActivity.kt file

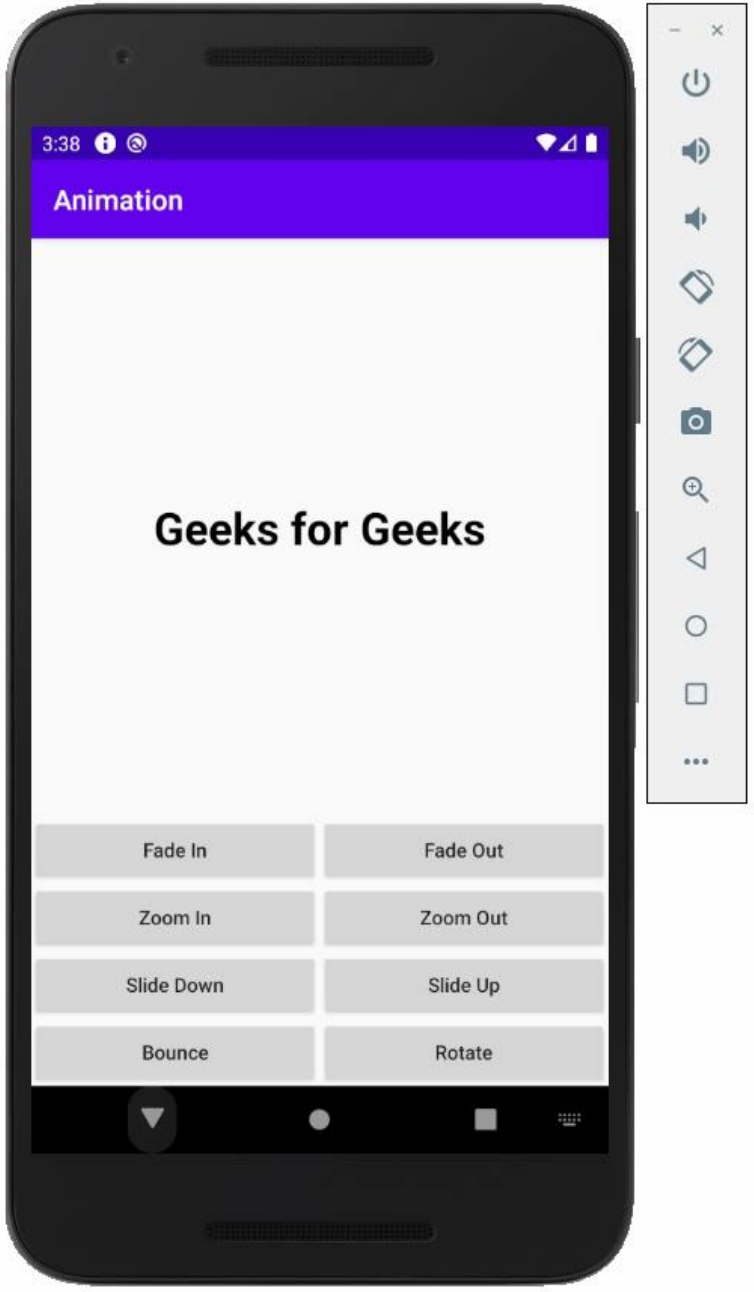
```
// fade out button
fade_out.setOnClickListener {
    val animationFadeOut = AnimationUtils.loadAnimation(this, R.anim.activity_fade_out)
    textView.startAnimation(animationFadeOut)
    Handler().postDelayed({
        textView.visibility = View.GONE
    }, 1000)
}

// rotate button
rotate.setOnClickListener {
    val animationRotate = AnimationUtils.loadAnimation(this, R.anim.activity_rotate)
    textView.startAnimation(animationRotate)
}

// rotate button
slide_down.setOnClickListener {
    val animationSlideDown = AnimationUtils.loadAnimation(this, R.anim.activity_slide_down)
    textView.startAnimation(animationSlideDown)
}
```

MainActivity.kt file

```
// slide down button
slide_down.setOnClickListener {
    val animationSlideDown = AnimationUtils.loadAnimation(this, R.anim.activity_slide_down)
    textView.startAnimation(animationSlideDown)
}

// slide up button
slide_up.setOnClickListener {
    val animationSlideUp = AnimationUtils.loadAnimation(this, R.anim.activity_slide_up)
    textView.startAnimation(animationSlideUp)
}

// zoom in button
zoom_in.setOnClickListener {
    val animationZoomIn = AnimationUtils.loadAnimation(this, R.anim.activity_zoom_in)
    textView.startAnimation(animationZoomIn)
}

// zoom out button
zoom_out.setOnClickListener {
    val animationZoomOut = AnimationUtils.loadAnimation(this, R.anim.activity_zoom_out)
    textView.startAnimation(animationZoomOut)
}
```

MainActivity.kt file

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    // bounce button
    bounce.setOnClickListener {...}

    // fade in button
    fade_in.setOnClickListener {...}

    // fade out button
    fade_out.setOnClickListener {...}

    // rotate button
    rotate.setOnClickListener {...}

    // rotate button
    slide_down.setOnClickListener {...}

    // slide down button
    slide_down.setOnClickListener {...}

    // slide up button
    slide_up.setOnClickListener {...}

    // zoom in button
    zoom_in.setOnClickListener {...}

    // zoom out button
    zoom_out.setOnClickListener {...}
}
```



YOUR APPLICATION SHOULD LOOK LIKE