

Practical No. 7

Title: Android program to work with graphics and animation

Aim: Create an application to demonstrate graphics and animation

Introduction

Create Drawing Objects

The android.graphics framework divides drawing into two areas:

1. *What* to draw, handled by Canvas
2. *How* to draw, handled by Paint.

For instance, Canvas provides a method to draw a line, while Paint provides methods to define that line's color. Canvas has a method to draw a rectangle, while Paint defines whether to fill that rectangle with a color or leave it empty. Simply put, Canvas defines shapes that you can draw on the screen, while Paint defines the color, style, font, and so forth of each shape you draw.

Draw!

Once you have your object creation and measuring code defined, you can implement onDraw(). Every view implements onDraw() differently, but there are some common operations that most views share:

- Draw text using drawText(). Specify the typeface by calling setTypeface(), and the text color by calling setColor().
- Draw primitive shapes using drawRect(), drawOval(), and drawArc(). Change whether the shapes are filled, outlined, or both by calling setStyle().
- Draw more complex shapes using the Path class. Define a shape by adding lines and curves to a Path object, then draw the shape using drawPath(). Just as with primitive shapes, paths can be outlined, filled, or both, depending on the setStyle().
- Define gradient fills by creating LinearGradient objects. Call setShader() to use your LinearGradient on filled shapes.
- Draw bitmaps using drawBitmap().

Tween Animation

Tween Animation takes some parameters such as start value, end value, size, time duration, rotation angle etc. and perform the required animation on that object. It can be applied to any type of object. So, in order to use this, android has provided us a class called Animation.

In order to perform animation in android, we are going to call a static function `loadAnimation()` of the class `AnimationUtils`. We are going to receive the result in an instance of Animation Object. Its syntax is as follows –

```
Animation animation = AnimationUtils.loadAnimation(getApplicationContext(),  
R.anim.blink);
```

Note the second parameter. It is the name of our animation xml file. You have to create a new folder called `anim` under `res` directory and make an xml file under `anim` folder.

This animation class has many useful functions which are listed below –

Sr.No	Method & description
1	start() This method starts the animation.
2	setDuration(long duration) This method sets the duration of an animation.
3	getDuration() This method gets the duration which is set by above method.
4	end() This method ends the animation.
5	cancel()

This method cancels the animation.

In order to apply this animation to an object, we will just call the `startAnimation()` method of the object. Its syntax is –

```
ImageView image1 = (ImageView)findViewById(R.id.imageView1);  
  
image.startAnimation(animation);
```

Exercise - Create an application using graphics and animation

Implementation:

Program:

activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout  
    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:id="@+id/main"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    tools:context=".MainActivity">  
  
    <ImageView  
        android:id="@+id/imageView"  
        android:layout_width="113dp"  
        android:layout_height="98dp"  
        android:contentDescription="@string/star"  
        app:srcCompat="@drawable/baseline_filter_vintage_24"  
        app:layout_constraintTop_toTopOf="parent"  
        app:layout_constraintStart_toStartOf="parent"  
        app:layout_constraintEnd_toEndOf="parent"  
        android:layout_marginTop="50dp"/>  
  
    <Button  
        android:id="@+id/button"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"
```

```
android:layout_marginStart="16dp"
android:layout_marginTop="36dp"
android:text="@string/blink"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/imageView" />
```

```
<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="36dp"
    android:layout_marginTop="36dp"
    android:text="@string/rotate"
    app:layout_constraintStart_toEndOf="@+id/button"
    app:layout_constraintTop_toBottomOf="@+id/imageView" />
```

```
<Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="40dp"
    android:layout_marginTop="36dp"
    android:text="@string/fade"
    app:layout_constraintStart_toEndOf="@+id/button2"
    app:layout_constraintTop_toBottomOf="@+id/imageView" />
```

```
<Button
    android:id="@+id/button4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="48dp"
    android:text="@string/move"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button" />
```

```
<Button
    android:id="@+id/button5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="36dp"
    android:layout_marginTop="48dp"
    android:text="@string/slide"
    app:layout_constraintStart_toEndOf="@+id/button4"
    app:layout_constraintTop_toBottomOf="@+id/button2" />
```

```
<Button
    android:id="@+id/button6"
```

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="40dp"
android:layout_marginTop="48dp"
android:text="@string/zoom"
app:layout_constraintStart_toEndOf="@+id/button5"
app:layout_constraintTop_toBottomOf="@+id/button3" />
```

```
<Button
    android:id="@+id/button7"
    android:layout_width="204dp"
    android:layout_height="49dp"
    android:text="@string/stop"
    app:layout_constraintTop_toBottomOf="@+id/button5"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginTop="60dp" />
```

```
<ImageView
    android:id="@+id/imageView2"
    android:layout_width="113dp"
    android:layout_height="98dp"
    android:layout_marginTop="64dp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button7"
    app:srcCompat="@drawable/baseline_animation_24"
    android:contentDescription="@string/plane"
    tools:layout_editor_absoluteX="149dp"
    tools:ignore="MissingConstraints" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

blink.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <alpha android:fromAlpha="0.0"
        android:toAlpha="1.0"
        android:interpolator="@android:anim/accelerate_interpolator"
        android:duration="500"
        android:repeatMode="reverse"
        android:repeatCount="infinite"
    />
</set>
```

fade.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/accelerate_interpolator">
    <alpha
        android:duration="1000"
        android:fromAlpha="0"
        android:toAlpha="1"
    />

    <alpha
        android:duration="1000"
        android:fromAlpha="1"
        android:startOffset="2000"
        android:toAlpha="0"
    />

</set>
```

move.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/accelerate_interpolator"
    android:fillAfter="true">
    <translate
        android:fromXDelta="0%p"
        android:toXDelta="75%p"
```

```
        android:duration="100"  
    />  
</set>
```

rotate.xml:

```
<?xml version="1.0" encoding="utf-8"?>  
<set xmlns:android="http://schemas.android.com/apk/res/android">  
    <rotate  
        android:duration="6000"  
        android:fromDegrees="0"  
        android:pivotX="50%"  
        android:pivotY="50%"  
        android:toDegrees="360"  
    />  
  
    <rotate  
        android:duration="6000"  
        android:fromDegrees="360"  
        android:pivotX="50%"  
        android:pivotY="50%"  
        android:startOffset="5000"  
        android:toDegrees="0"  
    />  
</set>
```

slide.xml:

```
<?xml version="1.0" encoding="utf-8"?>  
<set xmlns:android="http://schemas.android.com/apk/res/android"  
    android:fillAfter="true">  
    <scale
```

```
        android:duration="500"  
        android:fromXScale="1.0"  
        android:fromYScale="1.0"  
        android:interpolator="@android:anim/linear_interpolator"  
        android:toXScale="1.0"  
        android:toYScale="0.0"  
    />  
</set>
```

zoom.xml:

```
<?xml version="1.0" encoding="utf-8"?>  
<set xmlns:android="http://schemas.android.com/apk/res/android"  
    android:fillAfter="true">  
  
    <scale  
        android:duration="1000"  
        android:fromXScale="1"  
        android:fromYScale="1"  
        android:interpolator="@android:anim/linear_interpolator"  
        android:pivotX="50%"  
        android:pivotY="50%"  
        android:toXScale="2"  
        android:toYScale="2"  
  
    />  
</set>
```


MainActivity.java

```
package com.example.graphicsandanimation;

import android.os.Bundle;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.Button;
import android.widget.ImageView;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {

    ImageView imageView;

    Button blinkBtn, rotateBtn, fadeBtn, moveBtn, slideBtn, zoomBtn, stopBtn;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
```

```
v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);  
return insets;  
});
```

```
imageView = findViewById(R.id.imageView);  
blinkBtn = findViewById(R.id.button);  
rotateBtn = findViewById(R.id.button2);  
fadeBtn = findViewById(R.id.button3);  
moveBtn = findViewById(R.id.button4);  
slideBtn = findViewById(R.id.button5);  
zoomBtn = findViewById(R.id.button6);  
stopBtn = findViewById(R.id.button7);
```

```
blinkBtn.setOnClickListener(v->{  
    Animation blink = AnimationUtils.loadAnimation(getApplicationContext(), R.anim.blink);  
    imageView.startAnimation(blink);  
});
```

```
rotateBtn.setOnClickListener(v->{  
    Animation rotate = AnimationUtils.loadAnimation(getApplicationContext(), R.anim.rotate);  
    imageView.startAnimation(rotate);  
});
```

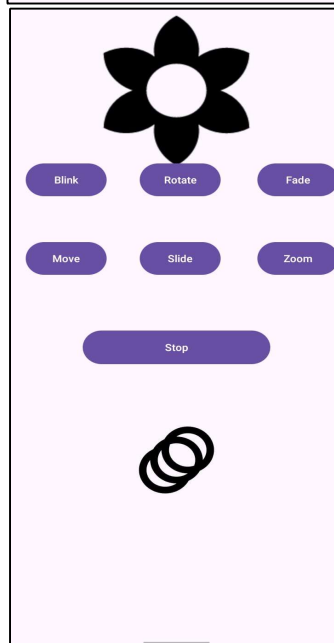
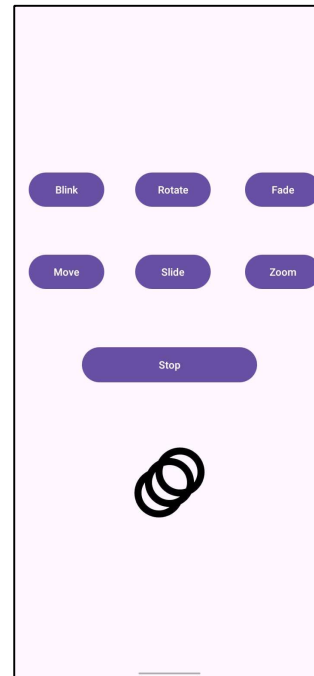
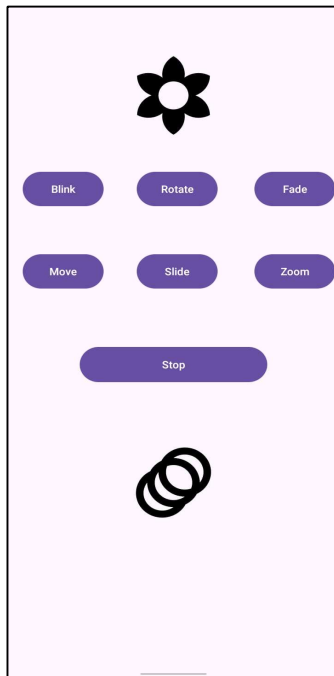
```
fadeBtn.setOnClickListener(v->{  
    Animation fade = AnimationUtils.loadAnimation(getApplicationContext(), R.anim.fade);  
    imageView.startAnimation(fade);  
});  
moveBtn.setOnClickListener(v->{
```

```
        Animation move = AnimationUtils.loadAnimation(getApplicationContext(), R.anim.move);
        imageView.startAnimation(move);
    });

    slideBtn.setOnClickListener(v->{
        Animation slide = AnimationUtils.loadAnimation(getApplicationContext(), R.anim.slide);
        imageView.startAnimation(slide);
    });

    zoomBtn.setOnClickListener(v->{
        Animation zoom = AnimationUtils.loadAnimation(getApplicationContext(), R.anim.zoom);
        imageView.startAnimation(zoom);
    });

    stopBtn.setOnClickListener(v->{
        imageView.clearAnimation();
    });
}
}
```

Output:

Conclusion - By mastering the concepts of Canvas, Paint, and tween animation, you can create compelling and dynamic graphics for your Android applications. This knowledge empowers you to breathe life into your user interfaces and enhance user experience.