

Practical: 9

AIM- What is Frame by Frame Animation? What is Twin Animation?

How can you achieve edge-to-edge content display in your app? Create Android Application to demonstrate Frame by frame animation and splash screen to demonstrate twin animation according to below instructions.

1. Create MainActivity according to below UI design.
2. Create SplashActivity according to Video
3. Create gradient Rectangle by using `<gradient>` tag in `<shape>` tag for background of SplashActivity. Use radial rectangle with $x = 0.9$, $y = 0.9$, radius = 1500. Start Color pink and End Color blue. shape should be rectangle
4. Add these all in project: `<animation-list>`, `oneShot` attribute, `<set>` tag, `android:startOffset = 100`, `android:duration=1000`, `<scale>` tag, `<translate>` tag, `<rotate>` tag, `<alpha>` tag

Submitted By :- Jaydip Changani
Enrollment number :- 21012021012



**Ganpat
University**

॥ विद्यया समाजोत्कर्षः ॥

**U.V. Patel
College of
Engineering**

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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        android:text="@string/app_name"
        android:textAppearance="?attr/textAppearanceHeadline6"
        android:textSize="20sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <com.google.android.material.card.MaterialCardView
        android:id="@+id/card"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        app:cardElevation="20dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <ImageView
                android:id="@+id/alarm"
                android:layout_width="match_parent"
                android:layout_height="250dp"
                android:scaleType="center" />

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
```

Practical: 9

```
android:orientation="vertical"
android:padding="16dp">

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Create Alarm Time"
    android:textAppearance="?attr/textAppearanceHeadline6" />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:text="By pressing buttons, Alarm can be created and cancelled.\n\nCheck current
time by looking below real digital clock. "
    android:textAppearance="?attr/textAppearanceBody2" />
<ImageView
    android:id="@+id/heart"
    android:layout_width="40dp"
    android:layout_height="40dp"
    android:layout_marginTop="2dp"
    android:layout_marginEnd="25dp"
    android:layout_gravity="right"
    android:src="@drawable/heart_animation_list"/>

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="10dp">
    <com.google.android.material.button.MaterialButton
        android:id="@+id/create_alarm"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Create Alarm"
        android:backgroundTint="@color/blue"
        app:icon="@drawable/baseline_add_alarm_24" />
    <com.google.android.material.button.MaterialButton
        android:id="@+id/cancel_alarm"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Cancel Alarm"
        android:textColor="@color/blue"
        android:background="#00FFFFFF"
        app:icon="@drawable/baseline_alarm_off_24"
        app:iconTint="@color/blue"/>
    </LinearLayout>
</LinearLayout>
```

Practical: 9

```
</LinearLayout>
</com.google.android.material.card.MaterialCardView>
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.kt :-

```
package com.example.mad_practical_9_21012021012

import android.graphics.drawable.AnimationDrawable
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.ImageView

class MainActivity : AppCompatActivity() {

    lateinit var alarmanimation : AnimationDrawable
    lateinit var heartanimation : AnimationDrawable

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

        val alarm : ImageView = findViewById(R.id.alarm)
        alarm.setBackgroundResource(R.drawable.alarm_animation_list)
        alarmanimation = alarm.background as AnimationDrawable

        val heart : ImageView = findViewById(R.id.heart)
        heart.setBackgroundResource(R.drawable.heart_animation_list)
        heartanimation = heart.background as AnimationDrawable
    }

    override fun onWindowFocusChanged(hasFocus: Boolean) {
        super.onWindowFocusChanged(hasFocus)
        if (hasFocus) {
            alarmanimation.start()
            heartanimation.start()
        }
        else {
            alarmanimation.stop()
            heartanimation.stop()
        }
    }
}
```

tween_animation.xml :-

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:startOffset="1000">

    <scale
        android:fromXScale="1.0"
        android:fromYScale="1.0"
        android:toXScale="2.0"
        android:toYScale="2.0"
        android:pivotX="50%"
        android:pivotY="50%"
        android:duration = "750" />

    <rotate
        android:fromDegrees="0"
        android:toDegrees="360"
        android:pivotX="50%"
        android:pivotY="50%"
        android:duration = "1500" />

    <translate
        android:fromXDelta="1.0"
        android:fromYDelta="1.0"
        android:toXDelta="10.0"
        android:toYDelta="100.0"
        android:duration = "750" />

    <scale
        android:fromXScale="1.0"
        android:fromYScale="1.0"
        android:toXScale="0.5"
        android:toYScale="0.5"
        android:duration="750"
        android:pivotX="50%"
        android:pivotY="50%"
        android:startOffset="750"/>

    <translate
        android:fromXDelta="1.0"
        android:fromYDelta="1.0"
        android:toXDelta="0.5"
        android:toYDelta="0.5"
        android:duration="750"
        android:startOffset="1500"/>
```

```
<scale
  android:fromXScale="1.0"
  android:fromYScale="1.0"
  android:toXScale="1.5"
  android:toYScale="1.5"
  android:duration="750"
  android:pivotX="50%"
  android:pivotY="50%"
  android:startOffset="1500"/>

<alpha
  android:duration="750"
  android:startOffset="1500"
  android:fromAlpha="1.0"
  android:toAlpha="0" />

</set>
```

alarm animation list.xml:-

```
<?xml version="1.0" encoding="utf-8"?>
<animation-list xmlns:android="http://schemas.android.com/apk/res/android"
  android:oneshot="false">

  <item
    android:drawable="@drawable/alarm1"
    android:duration="200" />
  <item
    android:drawable="@drawable/alarm2"
    android:duration="200" />
  <item
    android:drawable="@drawable/alarm3"
    android:duration="200" />
  <item
    android:drawable="@drawable/alarm4"
    android:duration="200" />
  <item
    android:drawable="@drawable/alarm5"
    android:duration="200" />
  <item
    android:drawable="@drawable/alarm6"
    android:duration="200" />
  <item
    android:drawable="@drawable/alarm7"
```

Practical: 9

```
        android:duration="200" />
    <item
        android:drawable="@drawable/alarm8"
        android:duration="200" />
    <item
        android:drawable="@drawable/alarm9"
        android:duration="200" />
    <item
        android:drawable="@drawable/alarm10"
        android:duration="200" />
</animation-list>
```

uypce animation list :-

```
<?xml version="1.0" encoding="utf-8"?>
<animation-list xmlns:android="http://schemas.android.com/apk/res/android"
    android:oneshot="false">

    <item
        android:drawable="@drawable/uypce_logo"
        android:duration="100" />
    <item
        android:drawable="@drawable/uypce_logo_1"
        android:duration="100" />
    <item
        android:drawable="@drawable/uypce_logo_2"
        android:duration="100" />
    <item
        android:drawable="@drawable/uypce_logo_3"
        android:duration="200" />
    <item
        android:drawable="@drawable/uypce_logo_4"
        android:duration="200" />
    <item
        android:drawable="@drawable/uypce_logo_5"
        android:duration="200" />
    <item
        android:drawable="@drawable/uypce_logo_6"
        android:duration="50" />
    <item
        android:drawable="@drawable/uypce_logo_7"
        android:duration="50" />
```

```
</animation-list>
```

heart animation list.xml :-

```
<?xml version="1.0" encoding="utf-8"?>
<animation-list xmlns:android="http://schemas.android.com/apk/res/android"
    android:oneshot="false">

    <item
        android:drawable="@drawable/ic_heart_0"
        android:duration="200" />
    <item
        android:drawable="@drawable/ic_heart_25"
        android:duration="200" />
    <item
        android:drawable="@drawable/ic_heart_50"
        android:duration="200" />
    <item
        android:drawable="@drawable/ic_heart_75"
        android:duration="200" />
    <item
        android:drawable="@drawable/ic_heart_100"
        android:duration="200" />

</animation-list>
```

gradient rectangle.xml :-

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

    <gradient
        android:startColor="@color/pink"
        android:endColor="@color/blue"
        android:type="radial"
        android:gradientRadius="1500"
        android:centerX="0.9"
        android:centerY="0.9" />

</shape>
```


activity_splash.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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".SplashActivity"
    android:background="@drawable/gradient_rectangle">

    <ImageView
        android:id="@+id/uvpce"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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

SplashActivity.kt :-

```
package com.example.mad_practical_9_21012021012

import android.content.Intent
import android.graphics.drawable.AnimationDrawable
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.animation.Animation
import android.view.animation.AnimationUtils
import android.widget.ImageView

class SplashActivity : AppCompatActivity() {
    lateinit var logoanimation: AnimationDrawable

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

        val logo: ImageView = findViewById(R.id.uvpce)
```

Practical: 9

```
logo.setBackgroundResource(R.drawable.uvpce_animation_list)
logoanimation = logo.background as AnimationDrawable;

val myAnimation = AnimationUtils.loadAnimation(this, R.anim.twin_animation)
logo.startAnimation(myAnimation)

// Add an animation listener to your animation
myAnimation.setAnimationListener(object : Animation.AnimationListener {
    override fun onAnimationStart(animation: Animation) {
        // Animation started
    }

    override fun onAnimationEnd(animation: Animation) {
        // Animation ended, start a new activity here
        val intent = Intent(this@SplashActivity, MainActivity::class.java)
        startActivity(intent)
    }

    override fun onAnimationRepeat(animation: Animation) {
        // Animation repeated
    }
})
}

override fun onWindowFocusChanged(hasFocus: Boolean) {
    super.onWindowFocusChanged(hasFocus)
    if (hasFocus) {
        logoanimation.start()
    } else {
        logoanimation.stop()
    }
}
}
```

Practical: 9

Output :-

