

# Practical: 8

**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.

Submitted By: Pavan Modi  
Enrollment number: 21012012016



**Ganpat  
University**

॥ विद्यया समाजोत्कर्षः ॥

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

Department of Computer  
Engineering/Information Technology

### Practical-8

**AIM:** Create Android Application to demonstrate Frame by frame animation and splash screen to demonstrate twin animation according to below instructions.

#### **What is Frame by Frame Animation?**

Frame-by-frame animation changes the contents of the Stage in every frame. It is best suited to complex animation in which an image changes in every frame instead of simply moving across the Stage. Frame-by-frame animation increases file size more rapidly than twined animation. In frame-by-frame animation, Animate stores the values for each complete frame.

To create a frame-by-frame animation, define each frame as a keyframe and create a different image for each frame. Each new keyframe initially contains the same contents as the keyframe preceding it, so you can modify the frames in the animation incrementally.

#### **What is Twin Animation?**

Animation has always been about the illusion of movement. Twin is a major part of making that illusion look real. Twin in animation is a short for *inbetweening*, and it's the process of generating images that go between keyframes.

Keyframes are the images at the beginning and end of a smooth transition. For example, an animated character could appear to jump from one point to another. That character would be clearly defined in each keyframe, but between those two frames, the figure could look distorted or stretched to fit with the motion and direction.

#### **How can you achieve edge-to-edge content display in your app?**

You can configure your app to draw its content behind the system bars. Together, the status bar and the navigation bar are called the *system bars*.

Your app achieves an edge-to-edge layout by drawing behind these system bars. When implementing edge-to-edge, your app should do the following:

Draw behind the navigation bar to achieve a more compelling and modern user experience.

Draw behind the status bar if it makes sense for your content and layout, such as in the case of full-width imagery. To do this, use APIs such as [AppBarLayout](#), which defines an app bar pinned to the top of the screen.

#### **Uvpce\_logo\_list.xml:**

```
<?xml version="1.0" encoding="utf-8"?>
<animation-list
    android:oneshot="true"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@drawable/uvpce_logo_1" android:duration="100"/>
    <item android:drawable="@drawable/uvpce_logo_2" android:duration="100"/>
</animation-list>
```

## Practical: 8

---

```
<item android:drawable="@drawable/uvpce_logo_3" android:duration="100"/>
<item android:drawable="@drawable/uvpce_logo_4" android:duration="100"/>
<item android:drawable="@drawable/uvpce_logo_5" android:duration="100"/>
<item android:drawable="@drawable/uvpce_logo_6" android:duration="100"/>
<item android:drawable="@drawable/uvpce_logo_7" android:duration="100"/>
<item android:drawable="@drawable/uvpce_logo" android:duration="100"/>
</animation-list>
```

### Heart\_list.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<animation-list xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@drawable/ic_heart_0" android:duration="100"/>
    <item android:drawable="@drawable/ic_heart_25" android:duration="100"/>
    <item android:drawable="@drawable/ic_heart_50" android:duration="100"/>
    <item android:drawable="@drawable/ic_heart_75" android:duration="100"/>
    <item android:drawable="@drawable/ic_heart_100" android:duration="100"/>
</animation-list>
```

### Alaram\_list.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<animation-list xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@drawable/alarm1" android:duration="100"/>
    <item android:drawable="@drawable/alarm2" android:duration="100"/>
    <item android:drawable="@drawable/alarm3" android:duration="100"/>
    <item android:drawable="@drawable/alarm4" android:duration="100"/>
    <item android:drawable="@drawable/alarm5" android:duration="100"/>
    <item android:drawable="@drawable/alarm6" android:duration="100"/>
    <item android:drawable="@drawable/alarm7" android:duration="100"/>
    <item android:drawable="@drawable/alarm8" android:duration="100"/>
    <item android:drawable="@drawable/alarm9" android:duration="100"/>
    <item android:drawable="@drawable/alarm10" android:duration="100"/>
</animation-list>
```

### Gradiant\_rectangle.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<shape android:shape="rectangle"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <gradient
        android:type="radial"
        android:centerX="0.9"
        android:centerY="0.9"
        android:gradientRadius="1500"
        android:centerColor="#EA80F1"
        android:endColor="#2079BF"
        android:startColor="#FB73A2"/>
```

</shape>

### **Scale\_in.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:duration="1000"/>
</set>
```

### **Scale\_out.xml:**

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
  <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="500"/>
  <alpha
    android:fromAlpha="1.0"
    android:toAlpha="0.0"
    android:duration="500"/>
</set>
```

### **Twin\_animation.xml:**

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
  android:startOffset="800">
  <translate
    android:fromXDelta="1.0"
    android:fromYDelta="1.0"
    android:toXDelta="10.0"
    android:toYDelta="100.0"
    android:duration="750"/>
  <rotate
    android:fromDegrees="0"
    android:toDegrees="360"
    android:pivotX="50%"
    android:pivotY="50%"
    android:duration="1500"/>
```

```
<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"/>
<scale
  android:fromXScale="1.0"
  android:fromYScale="1.0"
  android:toXScale="0.5"
  android:toYScale="0.5"
  android:pivotX="50%"
  android:pivotY="50%"
  android:duration="750"
  android:startOffset="750"/>
<translate
  android:fromXDelta="1.0"
  android:fromYDelta="1.0"
  android:toXDelta="-10.0"
  android:toYDelta="-100.0"
  android:duration="750"
  android:startOffset="750"/>
</set>
```

### Activity\_splash.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
  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">
  <androidx.constraintlayout.widget.ConstraintLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/gradiat_rectangle">
    <ImageView
      android:id="@+id/image"
      android:layout_width="wrap_content"
      android:layout_height="120dp"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintTop_toTopOf="parent"
```

```
        app:layout_constraintBottom_toBottomOf="parent"/>
    </androidx.constraintlayout.widget.ConstraintLayout>
</FrameLayout>
```

### Activity\_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
    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">
    <androidx.core.widget.NestedScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:layout_marginTop="30dp"
        app:layout_behavior="@string/appbar_scrolling_view_behavior">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">
            <com.google.android.material.card.MaterialCardView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginTop="5dp"
                android:layout_marginStart="20dp"
                android:layout_marginEnd="20dp"
                app:cardCornerRadius="30dp"
                app:cardElevation="20dp">
                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:orientation="vertical">
                    <ImageView
                        android:id="@+id/alaram_list"
                        android:layout_width="wrap_content"
                        android:layout_height="300dp" />
                    <LinearLayout
                        android:layout_width="wrap_content"
                        android:layout_height="match_parent"
                        android:orientation="vertical">
                        <TextView
                            android:layout_width="wrap_content"
```

## Practical: 8

---

```
        android:layout_height="wrap_content"
        android:text="Create Alaram Time "
        android:layout_marginTop="10dp"
        android:layout_marginStart="10dp"
        android:textSize="15sp"
        android:textStyle="bold"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:layout_marginStart="10dp"
        android:text="By pressing buttons, Alarm can be created and cancelled."/>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Check Current time by looking below real digital clock."
        android:layout_marginTop="10dp"
        android:layout_marginStart="10dp"/>
    <ImageView
        android:id="@+id/heart_list"
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:layout_gravity="right" />
</LinearLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <com.google.android.material.button.MaterialButton
        android:id="@+id/add_alaran"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:cornerRadius="20dp"
        android:text="Create Alarm"
        android:textAllCaps="false"
        android:layout_marginStart="5dp"
        app:icon="@drawable/ic_baseline_add_alarm_24" />
    <com.google.android.material.button.MaterialButton
        android:id="@+id/Cancel_alaran"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:cornerRadius="20dp"
        android:text="Cancel Alarm"
        android:textAllCaps="false"
        android:layout_marginStart="5dp"
        app:icon="@drawable/ic_baseline_alarm_off_24" />
</LinearLayout>
```

```
        </LinearLayout>
    </com.google.android.material.card.MaterialCardView>
</LinearLayout>
</androidx.core.widget.NestedScrollView>
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

### Manifests:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.madpractical8_21012012016">
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.MADPractical8_21012012016"
        tools:targetApi="31">
        <activity
            android:name=".SplashActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".MainActivity"
            android:exported="true">
        </activity>
    </application>
</manifest>
```

### SplashActivity.kt:

```
package com.example.madpractical8_21012012016

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
```



## Practical: 8

---

```
import android.widget.ImageView

class SplashActivity : AppCompatActivity() ,Animation.AnimationListener{
    lateinit var logo_img:ImageView
    lateinit var logoframbyframanimation:AnimationDrawable
    lateinit var twinanimation:Animation
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_splash)
        logo_img=findViewById(R.id.image)
        logo_img.setBackgroundResource(R.drawable.uvpce_logo_list)
        logoframbyframanimation=logo_img.background as AnimationDrawable
        twinanimation=AnimationUtils.loadAnimation(this,R.anim.twin_animation)
        twinanimation.setAnimationListener(this)
    }
    override fun onWindowFocusChanged(hasFocus: Boolean) {
        super.onWindowFocusChanged(hasFocus)
        if(hasFocus){
            logoframbyframanimation.start()
            logo_img.startAnimation(twinanimation)
        }
        else{
            logoframbyframanimation.stop()
        }
    }
    override fun onAnimationStart(p0: Animation?) {
    }
    override fun onAnimationEnd(p0: Animation?) {
        intent=Intent(this,MainActivity::class.java).apply {
            overridePendingTransition(R.anim.sacle_in,R.anim.scale_out)
            startActivity(this) }
    }
    override fun onAnimationRepeat(p0: Animation?) {
    }
}
```

### MainActivity.kt:

```
package com.example.madpractical8_21012012016

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

class MainActivity : AppCompatActivity() {
    lateinit var alaram_img:ImageView
```

## Practical: 8

---

```
lateinit var heart_img:ImageView
lateinit var alarm_animation: AnimationDrawable
lateinit var heart_animation:AnimationDrawable

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)
    alarm_img=findViewById(R.id.alarm_list)
    alarm_img.setBackgroundResource(R.drawable.alarm_list)
    alarm_animation=alarm_img.background as AnimationDrawable
    heart_img=findViewById(R.id.heart_list)
    heart_img.setBackgroundResource(R.drawable.heart_list)
    heart_animation=heart_img.background as AnimationDrawable
}
override fun onFocusChanged(hasFocus: Boolean) {
    super.onFocusChanged(hasFocus)
    if(hasFocus){
        alarm_animation.start()
        heart_animation.start()
    }
    else{
        alarm_animation.stop()
        heart_animation.stop()
    }
}
}
```

## Practical: 8

Output:

