**[ 2CEIT5PE5: MOBILE APPLICATION DEVELOPMENT]**

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



**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](https://developer.android.com/reference/com/google/android/material/appbar/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"/>

<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"

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\_alaram"

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\_alaram"

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:supportsRtl="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

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

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)

alaram\_img=findViewById(R.id.alaram\_list)

alaram\_img.setBackgroundResource(R.drawable.alaram\_list)

alarm\_animation=alaram\_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 onWindowFocusChanged(hasFocus: Boolean) {

super.onWindowFocusChanged(hasFocus)

if(hasFocus){

alarm\_animation.start()

heart\_animation.start()

}

else{

alarm\_animation.stop()

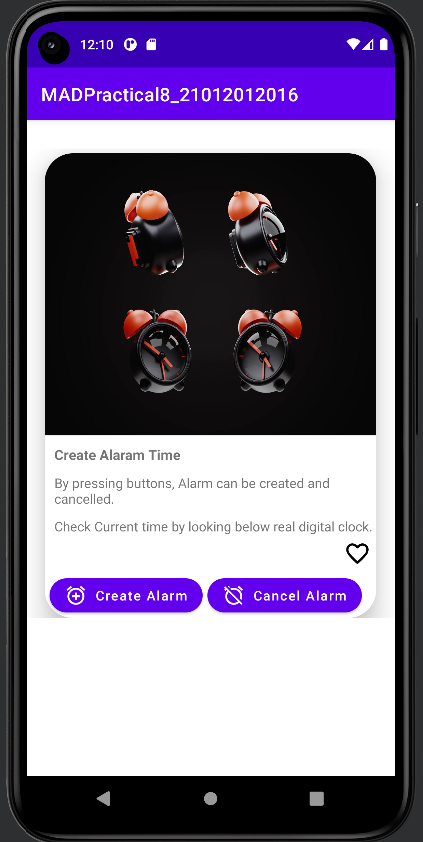
heart\_animation.stop()

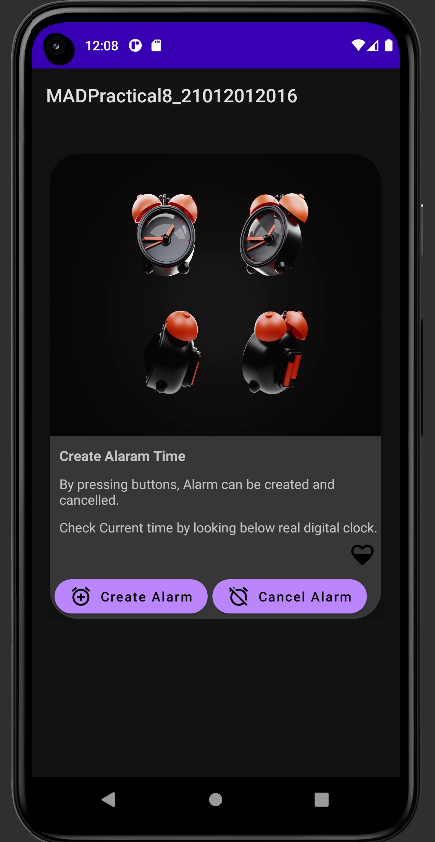
}

}

}

**Output:**

** **

** **