**[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: Kanzariya Dhavanik

Enrollment number: 21012022022



**Department of Computer Engineering/Information Technolo**

**What is Frame by Frame Animation?**

Frame-by-frame Animation can be defined as a method to produce the illusion of movement. It makes incremental changes between every keyframe to create this illusion. As the name implies, Frame-by-frame Animation is shot one frame at a time to form an Animation.

**What is Twin Animation?**

Tween Animation usually works upon the view elements of the layout. It can rotate, scale, shift, translate, or zoom the UI element.

* **Implementing edge-to-edge in your app requires the following steps:**

1. **Layout your app full-screen.**

**override fun onCreate(savedInstanceState: Bundle?) {  
  super.onCreate(savedInstanceState)  
  WindowCompat.setDecorFitsSystemWindows(window, false)  
}**

This method tells the window that the content of the app will expand to full screen. Once this is called the app content view will be drawn from the edges.

1. **Change the system bar colors and transparency**

**<!-- values-v29/themes.xml -->  
<style name="Theme.MyApp">  
  <item name="android:navigationBarColor">  
     @android:color/transparent  
  </item>  
  
  <!-- Optional: set to transparent if your app is drawing behind the status bar. -->  
  <item name="android:statusBarColor">  
     @android:color/transparent  
  </item>  
  
  <!-- Optional: set the status bar light and content dark. -->  
  <item name="android:windowLightStatusBar">  
    true  
  </item>  
</style>**

Now that content is drawn behind system bars we need to update the bars colors so that the app content is visible. For this we will set the colours of status bar and navigation bar to transparent. Note this is only required for API 29 and above as the system bars have a translucent scrim behind system bars. For API 28 level and below the system bars is by default transparent. Check the below screenshot after making system bars transparent.

### Handle any visual overlaps.

The system bars are made transparent. Even if they are transparent they are part of the hierarchy and can consume the touch. So in the scenario where we have an app widget that intersects with system bars it will not allow the widget to handle the user touch. Hence blocking the usage of that widget. Check the screen added in above section where bottom buttons are overlapped by navigation bar.  
  
To prevent this from happening the android system has a concept of insets which provides the required space taken by the system bars. We can use this space value to position the view such that it can handle the user touch.

The types of insets that apply to displaying your app edge-to-edge are:

* **System bars insets**: These insets are best used for views that are tappable and that shouldn’t be visually obscured by the system bars.
* **System gesture insets**: These insets apply to gesture-navigational areas used by the system that take priority over your app.

### System bars insets

System bars insets are the most commonly-used type of insets, and represent where the system UI is displayed in the z-axis above your app. They are best used to move or pad views in your app that are both tappable and shouldn't be visually obscured by the system bars.

**ViewCompat.setOnApplyWindowInsetsListener(view) { view, windowInsets ->  
  val insets = windowInsets.getInsets(WindowInsetsCompat.Type.systemBars())  
  // Apply the insets as a margin to the view. Here the system is setting  
  // only the bottom, left, and right dimensions, but apply whichever insets are  
  // appropriate to your layout. You can also update the view padding  
  // if that's more appropriate.  
  view.updateLayoutParams<MarginLayoutParams>(  
      leftMargin = insets.left,  
      bottomMargin = insets.bottom,  
      rightMargin = insets.right,  
  )  
  
  // Return CONSUMED if you don't want want the window insets to keep being  
  // passed down to descendant views.  
  WindowInsetsCompat.CONSUMED  
}**

**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"  
 android:fitsSystemWindows="true"  
 android:background="@drawable/splash\_design">  
 <ImageView  
 android:id="@+id/img"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="150dp"  
 android:layout\_gravity="center"  
 />  
</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">  
 <com.google.android.material.appbar.AppBarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
 <androidx.appcompat.widget.Toolbar  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:title="Practical 8"  
 app:titleTextColor="@color/white"/>  
 </com.google.android.material.appbar.AppBarLayout>  
 <androidx.core.widget.NestedScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 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="20dp"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 app:cardCornerRadius="8dp"  
 app:cardElevation="12dp"  
 android:layout\_marginBottom="20dp"  
 >  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:layout\_margin="10dp">  
 <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>  
  
 <androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="5dp"  
 android:layout\_marginEnd="5dp"  
 android:orientation="horizontal">  
  
 <com.google.android.material.button.MaterialButton  
 android:id="@+id/add\_alaram"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="5dp"  
 android:text="Create\nAlarm"  
 android:textSize="12dp"  
 android:textAllCaps="false"  
 android:textStyle="bold"  
 app:cornerRadius="20dp"  
 app:icon="@drawable/ic\_baseline\_add\_alarm\_24"  
 app:layout\_constraintEnd\_toStartOf="@id/Cancel\_alaram"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <com.google.android.material.button.MaterialButton  
 android:id="@+id/Cancel\_alaram"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="5dp"  
 android:text="Cancel\nAlarm"  
 android:textSize="12dp"  
 android:textAllCaps="false"  
 android:textStyle="bold"  
 app:cornerRadius="20dp"  
 app:icon="@drawable/ic\_baseline\_alarm\_off\_24"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toEndOf="@id/add\_alaram"  
 app:layout\_constraintTop\_toTopOf="@id/add\_alaram" />  
 </androidx.constraintlayout.widget.ConstraintLayout>  
 </LinearLayout>  
 </com.google.android.material.card.MaterialCardView>  
 </LinearLayout>  
 </androidx.core.widget.NestedScrollView>  
</androidx.coordinatorlayout.widget.CoordinatorLayout>**

**SplashActivity.kt**

**package com.example.madpractical8\_21012022022  
  
import android.content.Intent  
import android.graphics.drawable.AnimationDrawable  
import android.os.Bundle  
import android.view.animation.Animation  
import android.view.animation.AnimationUtils  
import android.widget.ImageView  
import androidx.appcompat.app.AppCompatActivity  
  
class SplashActivity : AppCompatActivity(){  
 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.img)  
 logo\_img.setBackgroundResource(R.drawable.uvpce\_logo\_list)  
 logoframbyframanimation=logo\_img.background as AnimationDrawable  
 twinanimation= AnimationUtils.loadAnimation(this,R.anim.twin\_animation)  
  
 }  
 override fun onWindowFocusChanged(hasFocus: Boolean) {  
 super.onWindowFocusChanged(hasFocus)  
 if(hasFocus){  
 logoframbyframanimation.start()  
 logo\_img.startAnimation(twinanimation)  
 twinanimation.setAnimationListener(object :Animation.AnimationListener{  
 override fun onAnimationEnd(p0: Animation?) {  
 var intent = Intent(applicationContext,MainActivity::class.java)  
  
 startActivity(intent)  
 overridePendingTransition(R.anim.scale\_in,R.anim.scale\_out)  
 finish()  
 }  
 override fun onAnimationRepeat(p0: Animation?) {}  
 override fun onAnimationStart(p0: Animation?) {}  
 })  
 }  
 else{  
 logoframbyframanimation.stop()  
 }  
 }  
}**

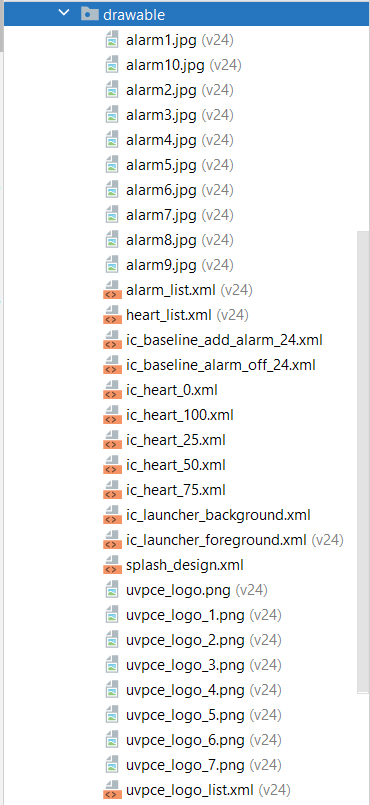
**MainActivity.kt**

**package com.example.madpractical8\_21012022022  
  
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.alarm\_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()  
 }  
 }  
}**

**AndroidManifest.xml**

**<?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\_21012022022">  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="Practical 8"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.MADPractical8\_21012022022"  
 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="false"/>  
 </application>  
</manifest>**

Resources:



**Output:**