[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: Jesik Pargi
Enrollment number: 21012022030

Department of Computer Engineering/Information Technology

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.

2. 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.kt1:

```
package com.example.madpractical8_21012022030

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_21012022030

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){
```

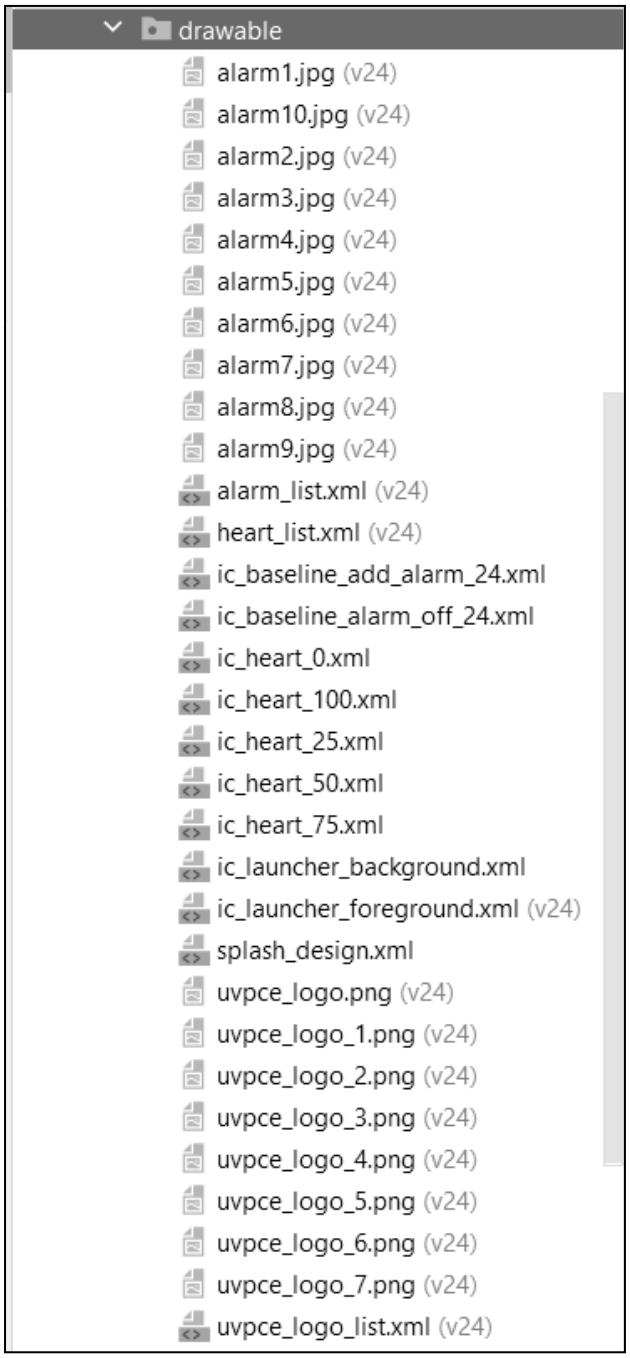
```
                alarm_animation.start()
                heart_animation.start()
            }
            else{
                alarm_animation.stop()
                heart_animation.stop()
            }
        }
}
```

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_20012021068">

    <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_20012021068"

        tools:targetApi="31">

        <activity

            android:name=".SplashActivity"

            android:exported="true">

            <intent-filter>

                <action android:name="android.intent.action.MAIN"
/>
```

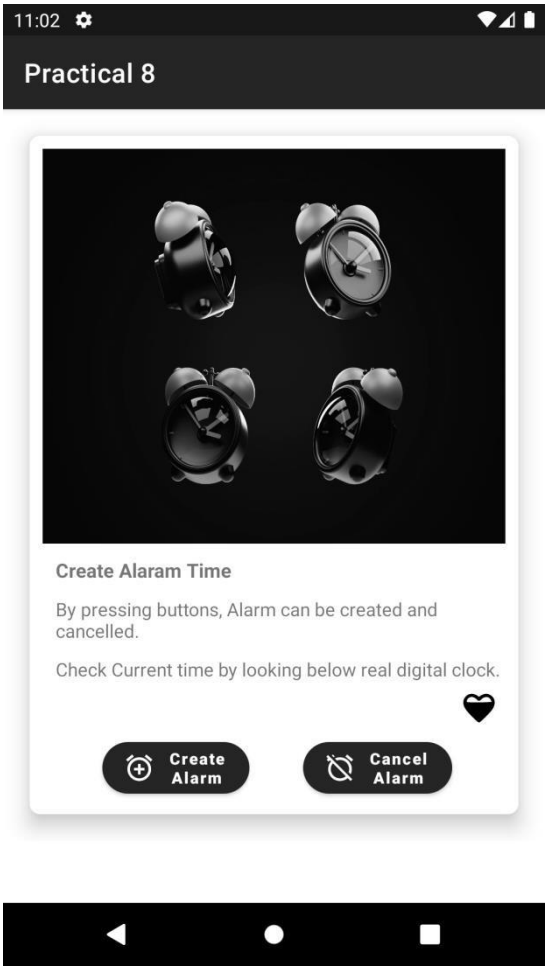
```
                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".MainActivity"
            android:exported="false"/>
    </application>
</manifest>
```

Resources:


drawable
alarm1.jpg (v24)
alarm10.jpg (v24)
alarm2.jpg (v24)
alarm3.jpg (v24)
alarm4.jpg (v24)
alarm5.jpg (v24)
alarm6.jpg (v24)
alarm7.jpg (v24)
alarm8.jpg (v24)
alarm9.jpg (v24)
alarm_list.xml (v24)
heart_list.xml (v24)
ic_baseline_add_alarm_24.xml
ic_baseline_alarm_off_24.xml
ic_heart_0.xml
ic_heart_100.xml
ic_heart_25.xml
ic_heart_50.xml
ic_heart_75.xml
ic_launcher_background.xml
ic_launcher_foreground.xml (v24)
splash_design.xml
uvpce_logo.png (v24)
uvpce_logo_1.png (v24)
uvpce_logo_2.png (v24)
uvpce_logo_3.png (v24)
uvpce_logo_4.png (v24)
uvpce_logo_5.png (v24)
uvpce_logo_6.png (v24)
uvpce_logo_7.png (v24)
uvpce_logo_list.xml (v24)

Output:


11:01
Ganpat University
U.V. Patel College of Engineering
॥ विद्यया समाजोत्कर्षः ॥


11:02
Practical 8
Create Alaram Time
By pressing buttons, Alarm can be created and cancelled.
Check Current time by looking below real digital clock.
Create Alarm
Cancel Alarm