Experiment – 15

Aim: Design an application to perform various animations on a particular image

Solution:

Step 1: Create an anim resource directory

To create a animation, you need to create anim folder by selecting it and choosing "New", then "android resource directory" and entering "anim" as the name and type as anim.

Step 2: Add all animations files:

Create all animation transistions under anim directory

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">

   <ImageView
      android:id="@+id/logoImage"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      app:srcCompat="@drawable/download"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintTop_toTopOf="parent"
      app:layout_constraintBottom_toTopOf="@+id/rotBtn"/>

   <Button
      android:id="@+id/blinkBtn"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Blink"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintEnd_toStartOf="@+id/rotBtn"
      app:layout_constraintTop_toBottomOf="@+id/logoImage"
      app:layout_constraintBottom_toTopOf="@+id/moveBtn"/>

   <Button
      android:id="@+id/rotBtn"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Rotate"
      app:layout_constraintStart_toEndOf="@+id/blinkBtn"
      app:layout_constraintEnd_toStartOf="@+id/fadeBtn"
      app:layout_constraintTop_toBottomOf="@+id/logoImage"
```

```
    app:layout_constraintBottom_toTopOf="@+id/slideBtn"/>

  <Button
    android:id="@+id/fadeBtn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Fade"
    app:layout_constraintStart_toEndOf="@+id/rotBtn"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/logoImage"
    app:layout_constraintBottom_toTopOf="@+id/zoomBtn"/>

  <Button
    android:id="@+id/moveBtn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Move"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toStartOf="@+id/slideBtn"
    app:layout_constraintTop_toBottomOf="@+id/blinkBtn"
    app:layout_constraintBottom_toBottomOf="parent"/>

  <Button
    android:id="@+id/slideBtn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Slide"
    app:layout_constraintStart_toEndOf="@+id/moveBtn"
    app:layout_constraintEnd_toStartOf="@+id/zoomBtn"
    app:layout_constraintTop_toBottomOf="@+id/rotBtn"
    app:layout_constraintBottom_toBottomOf="parent"/>


  <Button
    android:id="@+id/zoomBtn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Zoom"
    app:layout_constraintStart_toEndOf="@+id/slideBtn"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/fadeBtn"
    app:layout_constraintBottom_toBottomOf="parent"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

blink.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"
```

```
      android:repeatMode = "reverse"
      android:repeatCount = "infinite"
      android:interpolator = "@android:anim/accelerate_interpolator" />

</set>
```

rotate.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
   <rotate android:fromDegrees="0"
      android:toDegrees="360"
      android:pivotX="50%"
      android:pivotY="50%"
      android:duration = "5000" />
   <rotate android:fromDegrees="360"
      android:toDegrees="0"
      android:pivotX="50%"
      android:pivotY="50%"
      android:startOffset = "6000"
      android:duration = "5000" />
</set>
```

fade.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
   <alpha android:fromAlpha="0"
      android:toAlpha="1"
      android:duration = "2000"
      android:interpolator = "@android:anim/accelerate_interpolator" />
   <alpha android:fromAlpha="1"
      android:toAlpha="0"
      android:duration = "2000"
      android:interpolator = "@android:anim/accelerate_interpolator" />
</set>
```

move.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
   <translate android:fromXDelta="0%"
      android:toXDelta="75%"
      android:duration = "2000"
      android:interpolator ="@android:anim/linear_interpolator" />
</set>
```

slide.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
   <scale android:fromYScale="1.0"
      android:toYScale="0.0"
      android:fromXScale="1.0"
```

```
      android:toXScale="1.0"
      android:duration = "2000"
      android:interpolator = "@android:anim/linear_interpolator"/>
</set>
```

zoom.xml:

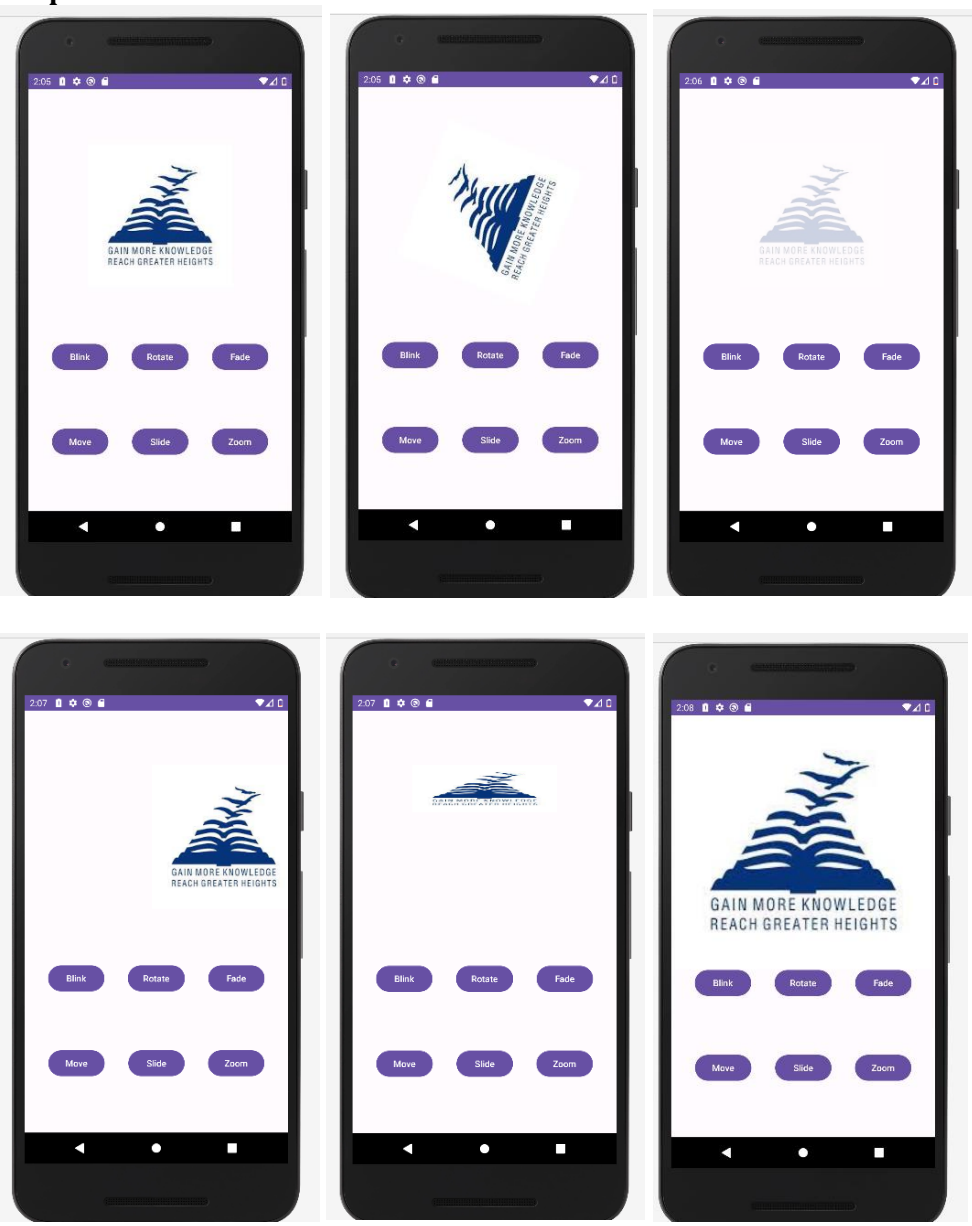
```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
   <scale android:fromXScale="1"
      android:toXScale="5"
      android:fromYScale="1"
      android:toYScale="5"
      android:pivotX="50%"
      android:pivotY="50%"
      android:duration = "5000" />
   <scale android:fromXScale="1"
      android:toXScale="0.2"
      android:fromYScale="1"
      android:toYScale="0.2"
      android:pivotX="50%"
      android:pivotY="50%"
      android:duration = "5000" />
</set>
```

MainActivity.java:

```
package com.example.animationsexample;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.Button;
import android.widget.ImageView;

public class MainActivity extends AppCompatActivity {
   ImageView im;
   Button bli,fad,mov,rot,sli,zoom;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      im = findViewById(R.id.logoImage);
      bli = findViewById(R.id.blinkBtn);
      rot = findViewById(R.id.rotBtn);
      fad = findViewById(R.id.fadeBtn);
      mov = findViewById(R.id.moveBtn);
      sli = findViewById(R.id.slideBtn);
      zoom = findViewById(R.id.zoomBtn);
```

```
        bli.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Animation animation =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.blink);
                im.startAnimation(animation);
            }
        });
        rot.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Animation animation =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.rotate);
                im.startAnimation(animation);
            }
        });
        fad.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Animation animation =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.fade);
                im.startAnimation(animation);
            }
        });
        mov.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Animation animation =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.move);
                im.startAnimation(animation);
            }
        });
        sli.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Animation animation =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.slide);
                im.startAnimation(animation);
            }
        });
        zoom.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Animation animation =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.zoom);
                im.startAnimation(animation);
            }
        });
    }
}
```

Output:


GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS
Blink
Rotate
Fade
Move
Slide
Zoom

Experiment – 16

Aim: Design an application to find the current location of user.

Solution:

Step 1: Add Google play location service dependency

You need to go to your app-level Gradle file and add the google play service dependency. Paste the below code under your dependencies section.

implementation("com.google.android.gms:play-services-location:21.0.1")

Step 2: Provide permissions in Manifest

Go to your manifest file and add the permission for internet, fine and coarse location access. You can find the permissions in the below code.

```
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.INTERNET"/>
```

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/addressTxt"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="User Address"
      android:textSize="24sp"
      android:textStyle="bold"
      app:layout_constraintBottom_toTopOf="@+id/getBtn"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent" />

   <Button
      android:id="@+id/getBtn"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Get Current Location"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintTop_toBottomOf="@+id/addressTxt"
      app:layout_constraintBottom_toBottomOf="parent"/>
```

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

MainActivity.java:

```
package com.example.findmylocation;

import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;

import android.Manifest;
import android.content.pm.PackageManager;
import android.location.Location;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;

public class MainActivity extends AppCompatActivity {
   TextView tv;
   Button locBtn;
   FusedLocationProviderClient locationProviderClient;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      tv = findViewById(R.id.addressTxt);
      locBtn = findViewById(R.id.getBtn);
      locationProviderClient = LocationServices.getFusedLocationProviderClient(this);
      locBtn.setOnClickListener(new View.OnClickListener() {
         @RequiresApi(api = Build.VERSION_CODES.M)
         @Override
         public void onClick(View view) {

if(ActivityCompat.checkSelfPermission(MainActivity.this,Manifest.permission.ACCESS_COARSE_LOC
ATION)!=PackageManager.PERMISSION_GRANTED){
               requestPermissions(new String[]{Manifest.permission.ACCESS_COARSE_LOCATION},1);
               return;
            }
            Task<Location> location = locationProviderClient.getLastLocation();
            location.addOnSuccessListener(new OnSuccessListener<Location>() {
               @Override
               public void onSuccess(Location location) {
                  double latitude = location.getLatitude();
                  double longitude = location.getLongitude();
```

```
                tv.setText("Latitude: "+latitude+"\n Longitude: "+longitude);
            }
        });
    }
});
  }
}
```

Output: