Assignment 3

Title:- Write an application that draws basic graphical primitives on the screen.

Code:-

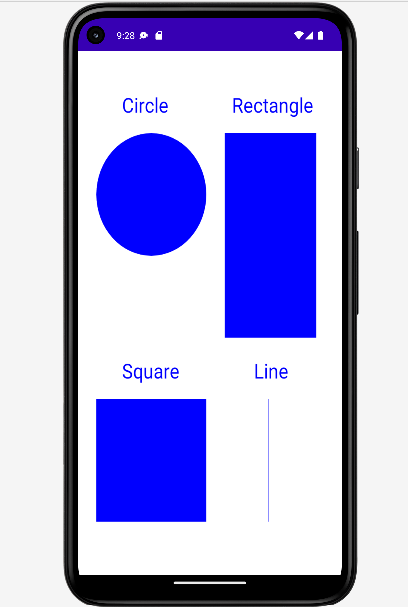
activity\_main.xml:

*<?*xml version="1.0" encoding="utf-8"*?>*<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <ImageView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:id="@+id/imageView" />  
</RelativeLayout>

MainActivity.java:

package com.example.mclab3;  
  
import android.app.Activity;  
import android.graphics.Bitmap;  
import android.graphics.Canvas;  
import android.graphics.Color;  
import android.graphics.Paint;  
import android.graphics.drawable.BitmapDrawable;  
import android.os.Bundle;  
import android.widget.ImageView;  
  
public class MainActivity extends Activity  
{  
 @Override  
 public void onCreate(Bundle savedInstanceState)  
 **{** super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 *//Creating a Bitmap* Bitmap bg = Bitmap.*createBitmap*(720, 1280, Bitmap.Config.*ARGB\_8888*);  
  
 *//Setting the Bitmap as background for the ImageView* ImageView i = (ImageView) findViewById(R.id.*imageView*);  
 i.setBackgroundDrawable(new BitmapDrawable(bg));  
  
 *//Creating the Canvas Object* Canvas canvas = new Canvas(bg);  
  
 *//Creating the Paint Object and set its color & TextSize* Paint paint = new Paint();  
 paint.setColor(Color.*BLUE*);  
 paint.setTextSize(50);  
  
 *//To draw a Rectangle* canvas.drawText("Rectangle", 420, 150, paint);  
 canvas.drawRect(400, 200, 650, 700, paint);  
  
 *//To draw a Circle* canvas.drawText("Circle", 120, 150, paint);  
 canvas.drawCircle(200, 350, 150, paint);  
  
 *//To draw a Square* canvas.drawText("Square", 120, 800, paint);  
 canvas.drawRect(50, 850, 350, 1150, paint);  
  
 *//To draw a Line* canvas.drawText("Line", 480, 800, paint);  
 canvas.drawLine(520, 850, 520, 1150, paint);  
 **}**}

Output:-



Assignment 4

Title:- Develop a native application that uses GPS location information.

Code:-

activity\_main.xml:

<?xml version = "1.0" encoding = "utf-8"?>  
<LinearLayout xmlns:android = "http://schemas.android.com/apk/res/android"  
 android:layout\_width = "fill\_parent"  
 android:layout\_height = "fill\_parent"  
 android:orientation = "vertical" >  
  
  
 <Button  
 android:id = "@+id/button"  
 android:layout\_width = "fill\_parent"  
 android:layout\_height = "wrap\_content"  
 android:text = "getlocation"/>  
  
</LinearLayout>

MainActivity.java:

package com.example.mclab4;  
  
import android.Manifest;  
import android.app.Activity;  
import android.content.pm.PackageManager;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.Toast;  
import androidx.core.app.ActivityCompat;  
  
public class MainActivity extends Activity {  
  
 Button btnShowLocation;  
 private static final int *REQUEST\_CODE\_PERMISSION* = 2;  
 String mPermission = Manifest.permission.*ACCESS\_FINE\_LOCATION*;  
  
 *// GPSTracker class* GPSTracker gps;  
  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 try {  
 if (ActivityCompat.*checkSelfPermission*(this, mPermission)  
 != PackageManager.*PERMISSION\_GRANTED*) {  
  
 ActivityCompat.*requestPermissions*(this, new String[]{mPermission},  
 *REQUEST\_CODE\_PERMISSION*);  
  
  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 btnShowLocation = (Button) findViewById(R.id.*button*);  
  
 *// show location button click event* btnShowLocation.setOnClickListener(new View.OnClickListener() {  
  
 @Override  
 public void onClick(View arg0) {  
 *// create class object* gps = new GPSTracker(MainActivity.this);  
  
 *// check if GPS enabled* if(gps.canGetLocation()){  
  
 double latitude = gps.getLatitude();  
 double longitude = gps.getLongitude();  
  
 *// \n is for new line* Toast.*makeText*(getApplicationContext(), "Your Location is - \nLat: "  
 + latitude + "\nLong: " + longitude, Toast.*LENGTH\_LONG*).show();  
 }else{  
 *// can't get location  
 // GPS or Network is not enabled  
 // Ask user to enable GPS/network in settings* gps.showSettingsAlert();  
 }  
  
 }  
 });  
 }  
}

GPSTracker.java:

package com.example.mclab4;  
  
import android.Manifest;  
import android.app.AlertDialog;  
import android.app.Service;  
import android.content.Context;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.location.Location;  
import android.location.LocationListener;  
import android.location.LocationManager;  
import android.os.Bundle;  
import android.os.IBinder;  
import android.provider.Settings;  
import android.util.Log;  
  
import androidx.core.app.ActivityCompat;  
  
public class GPSTracker extends Service implements LocationListener {  
  
 private final Context mContext;  
  
 *// flag for GPS status* boolean isGPSEnabled = false;  
  
 *// flag for network status* boolean isNetworkEnabled = false;  
  
 *// flag for GPS status* boolean canGetLocation = false;  
  
 Location location; *// location* double latitude; *// latitude* double longitude; *// longitude  
  
 // The minimum distance to change Updates in meters* private static final long *MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES* = 10; *// 10 meters  
  
 // The minimum time between updates in milliseconds* private static final long *MIN\_TIME\_BW\_UPDATES* = 1000 \* 60 \* 1; *// 1 minute  
  
 // Declaring a Location Manager* protected LocationManager locationManager;  
  
 public GPSTracker(Context context) {  
 this.mContext = context;  
 getLocation();  
 }  
  
 public Location getLocation() {  
 try {  
 locationManager = (LocationManager) mContext.getSystemService(*LOCATION\_SERVICE*);  
  
 *// getting GPS status* isGPSEnabled = locationManager.isProviderEnabled(LocationManager.*GPS\_PROVIDER*);  
  
 *// getting network status* isNetworkEnabled = locationManager  
 .isProviderEnabled(LocationManager.*NETWORK\_PROVIDER*);  
  
 if (!isGPSEnabled && !isNetworkEnabled) {  
 *// no network provider is enabled* } else {  
 this.canGetLocation = true;  
 *// First get location from Network Provider* if (isNetworkEnabled) {  
 if (ActivityCompat.*checkSelfPermission*(this, Manifest.permission.*ACCESS\_FINE\_LOCATION*) != PackageManager.*PERMISSION\_GRANTED* && ActivityCompat.*checkSelfPermission*(this, Manifest.permission.*ACCESS\_COARSE\_LOCATION*) != PackageManager.*PERMISSION\_GRANTED*) {  
 *// TODO: Consider calling  
 // ActivityCompat#requestPermissions  
 // here to request the missing permissions, and then overriding  
 // public void onRequestPermissionsResult(int requestCode, String[] permissions,  
 // int[] grantResults)  
 // to handle the case where the user grants the permission. See the documentation  
 // for ActivityCompat#requestPermissions for more details.* }  
 locationManager.requestLocationUpdates(  
 LocationManager.*NETWORK\_PROVIDER*,  
 *MIN\_TIME\_BW\_UPDATES*,  
 *MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES*, this);  
  
 Log.*d*("Network", "Network");  
 if (locationManager != null) {  
 location = locationManager  
 .getLastKnownLocation(LocationManager.*NETWORK\_PROVIDER*);  
  
 if (location != null) {  
 latitude = location.getLatitude();  
 longitude = location.getLongitude();  
 }  
 }  
 }  
  
 *// if GPS Enabled get lat/long using GPS Services* if (isGPSEnabled) {  
 if (location == null) {  
 locationManager.requestLocationUpdates(  
 LocationManager.*GPS\_PROVIDER*,  
 *MIN\_TIME\_BW\_UPDATES*,  
 *MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES*, this);  
  
 Log.*d*("GPS Enabled", "GPS Enabled");  
 if (locationManager != null) {  
 location = locationManager  
 .getLastKnownLocation(LocationManager.*GPS\_PROVIDER*);  
  
 if (location != null) {  
 latitude = location.getLatitude();  
 longitude = location.getLongitude();  
 }  
 }  
 }  
 }  
 }  
  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 return location;  
 }  
  
 */\*\*  
 \* Stop using GPS listener  
 \* Calling this function will stop using GPS in your app  
 \* \*/* public void stopUsingGPS(){  
 if(locationManager != null){  
 locationManager.removeUpdates(GPSTracker.this);  
 }  
 }  
  
 */\*\*  
 \* Function to get latitude  
 \* \*/* public double getLatitude(){  
 if(location != null){  
 latitude = location.getLatitude();  
 }  
  
 *// return latitude* return latitude;  
 }  
  
 */\*\*  
 \* Function to get longitude  
 \* \*/* public double getLongitude(){  
 if(location != null){  
 longitude = location.getLongitude();  
 }  
  
 *// return longitude* return longitude;  
 }  
  
 */\*\*  
 \* Function to check GPS/wifi enabled  
 \* @return boolean  
 \* \*/* public boolean canGetLocation() {  
 return this.canGetLocation;  
 }  
  
 */\*\*  
 \* Function to show settings alert dialog  
 \* On pressing Settings button will lauch Settings Options  
 \* \*/* public void showSettingsAlert(){  
 AlertDialog.Builder alertDialog = new AlertDialog.Builder(mContext);  
  
 *// Setting Dialog Title* alertDialog.setTitle("GPS is settings");  
  
 *// Setting Dialog Message* alertDialog.setMessage("GPS is not enabled. Do you want to go to settings menu?");  
  
 *// On pressing Settings button* alertDialog.setPositiveButton("Settings", new DialogInterface.OnClickListener() {  
 public void onClick(DialogInterface dialog,int which) {  
 Intent intent = new Intent(Settings.*ACTION\_LOCATION\_SOURCE\_SETTINGS*);  
 mContext.startActivity(intent);  
 }  
 });  
  
 *// on pressing cancel button* alertDialog.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {  
 public void onClick(DialogInterface dialog, int which) {  
 dialog.cancel();  
 }  
 });  
  
 *// Showing Alert Message* alertDialog.show();  
 }  
  
 @Override  
 public void onLocationChanged(Location location) {  
 }  
  
 @Override  
 public void onProviderDisabled(String provider) {  
 }  
  
 @Override  
 public void onProviderEnabled(String provider) {  
 }  
  
 @Override  
 public void onStatusChanged(String provider, int status, Bundle extras) {  
 }  
  
 @Override  
 public IBinder onBind(Intent arg0) {  
 return null;  
 }  
}

Output:-



Assignment 5

Title:- Design an android Application for Frame Animation.

Code:-

activity\_main.xml:

<RelativeLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:paddingBottom="@dimen/activity\_vertical\_margin" tools:context=".MainActivity">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Alert Dialog"

android:id="@+id/textView"

android:textSize="35dp"

android:layout\_alignParentTop="true"

android:layout\_centerHorizontal="true" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Tutorialspoint"

android:id="@+id/textView2"

android:textColor="#ff3eff0f"

android:textSize="35dp"

android:layout\_below="@+id/textView"

android:layout\_centerHorizontal="true" />

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/imageView"

android:src="@drawable/abc"

android:layout\_below="@+id/textView2"

android:layout\_alignRight="@+id/textView2"

android:layout\_alignEnd="@+id/textView2"

android:layout\_alignLeft="@+id/textView"

android:layout\_alignStart="@+id/textView"/>

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="zoom"

android:id="@+id/button"

android:layout\_below="@+id/imageView"

android:layout\_alignParentLeft="true"

android:layout\_alignParentStart="true"

android:layout\_marginTop="40dp"

android:onClick="clockwise"/>

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="clockwise"

android:id="@+id/button2"

android:layout\_alignTop="@+id/button"

android:layout\_centerHorizontal="true"

android:onClick="zoom"/>

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="fade"

android:id="@+id/button3"

android:layout\_alignTop="@+id/button2"

android:layout\_alignParentRight="true"

android:layout\_alignParentEnd="true"

android:onClick="fade"/>

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="blink"

android:onClick="blink"

android:id="@+id/button4"

android:layout\_below="@+id/button"

android:layout\_alignParentLeft="true"

android:layout\_alignParentStart="true" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="move"

android:onClick="move"

android:id="@+id/button5"

android:layout\_below="@+id/button2"

android:layout\_alignRight="@+id/button2"

android:layout\_alignEnd="@+id/button2"

android:layout\_alignLeft="@+id/button2"

android:layout\_alignStart="@+id/button2" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="slide"

android:onClick="slide"

android:id="@+id/button6"

android:layout\_below="@+id/button3"

android:layout\_toRightOf="@+id/textView"

android:layout\_toEndOf="@+id/textView" />

</RelativeLayout>

MainActivity.java:

import android.app.Activity;

import android.os.Bundle;

import android.view.View;

import android.view.animation.Animation;

import android.view.animation.AnimationUtils;

import android.widget.ImageView;

import android.widget.Toast;

public class MainActivity extends Activity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

public void clockwise(View view){

ImageView image = (ImageView)findViewById(R.id.imageView);

Animation animation = AnimationUtils.loadAnimation(getApplicationContext(),

R.anim.myanimation);

image.startAnimation(animation);

}

public void zoom(View view){

ImageView image = (ImageView)findViewById(R.id.imageView);

Animation animation1 = AnimationUtils.loadAnimation(getApplicationContext(),

R.anim.clockwise);

image.startAnimation(animation1);

}

public void fade(View view){

ImageView image = (ImageView)findViewById(R.id.imageView);

Animation animation1 =

AnimationUtils.loadAnimation(getApplicationContext(),

R.anim.fade);

image.startAnimation(animation1);

}

public void blink(View view){

ImageView image = (ImageView)findViewById(R.id.imageView);

Animation animation1 =

AnimationUtils.loadAnimation(getApplicationContext(),

R.anim.blink);

image.startAnimation(animation1);

}

public void move(View view){

ImageView image = (ImageView)findViewById(R.id.imageView);

Animation animation1 =

AnimationUtils.loadAnimation(getApplicationContext(), R.anim.move);

image.startAnimation(animation1);

}

public void slide(View view){

ImageView image = (ImageView)findViewById(R.id.imageView);

Animation animation1 =

AnimationUtils.loadAnimation(getApplicationContext(), R.anim.slide);

image.startAnimation(animation1);

}

}

res/anim/myanimation.xml:

<?xml version="1.0" encoding="utf-8"?>

<set xmlns:android="http://schemas.android.com/apk/res/android">

<scale xmlns:android="http://schemas.android.com/apk/res/android"

android:fromXScale="0.5"

android:toXScale="3.0"

android:fromYScale="0.5"

android:toYScale="3.0"

android:duration="5000"

android:pivotX="50%"

android:pivotY="50%" >

</scale>

<scale xmlns:android="http://schemas.android.com/apk/res/android"

android:startOffset="5000"

android:fromXScale="3.0"

android:toXScale="0.5"

android:fromYScale="3.0"

android:toYScale="0.5"

android:duration="5000"

android:pivotX="50%"

android:pivotY="50%" >

</scale>

</set>

res/anim/clockwise.xml:

<?xml version="1.0" encoding="utf-8"?>

<set xmlns:android="http://schemas.android.com/apk/res/android">

<rotate xmlns:android="http://schemas.android.com/apk/res/android"

android:fromDegrees="0"

android:toDegrees="360"

android:pivotX="50%"

android:pivotY="50%"

android:duration="5000" >

</rotate>

<rotate xmlns:android="http://schemas.android.com/apk/res/android"

android:startOffset="5000"

android:fromDegrees="360"

android:toDegrees="0"

android:pivotX="50%"

android:pivotY="50%"

android:duration="5000" >

</rotate>

</set>

res/anim/fade.xml:

<?xml version="1.0" encoding="utf-8"?>

<set xmlns:android="http://schemas.android.com/apk/res/android"

android:interpolator="@android:anim/accelerate\_interpolator" >

<alpha

android:fromAlpha="0"

android:toAlpha="1"

android:duration="2000" >

</alpha>

<alpha

android:startOffset="2000"

android:fromAlpha="1"

android:toAlpha="0"

android:duration="2000" >

</alpha>

</set>

res/anim/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:interpolator="@android:anim/accelerate\_interpolator"

android:duration="600"

android:repeatMode="reverse"

android:repeatCount="infinite"/>

</set>

res/anim/move.xml:

<?xml version="1.0" encoding="utf-8"?>

<set

xmlns:android="http://schemas.android.com/apk/res/android"

android:interpolator="@android:anim/linear\_interpolator"

android:fillAfter="true">

<translate

android:fromXDelta="0%p"

android:toXDelta="75%p"

android:duration="800" />

</set>

res/anim/slide.xml:

<?xml version="1.0" encoding="utf-8"?>

<set xmlns:android="http://schemas.android.com/apk/res/android"

android:fillAfter="true" >

<scale

android:duration="500"

android:fromXScale="1.0"

android:fromYScale="1.0"

android:interpolator="@android:anim/linear\_interpolator"

android:toXScale="1.0"

android:toYScale="0.0" />

</set>

Output:-

