Assignment 3

Graphical Primitives and Animation

Shashanka Venkatesh - 185001145

26th August, 2021

Learning Objective

Generate the following using graphical primitives.

Specifications

1. Draw shapes such as Line, Circle, Rectangle and Arc
2. Perform animation using any Image (Gif, Jpeg)
3. Perform transformation – Rotation, Zooming

Home (Main Activity)

XML (activity_main.xml)

```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">

    <Button
        android:id="@+id/shapesButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp"
        android:text="@string/primitive_shapes"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/animateButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/animations"
        app:layout_constraintBottom_toTopOf="@+id/transformButton"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/shapesButton" />

    <Button
        android:id="@+id/transformButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/transformations"
        android:layout_marginBottom="50dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Java (MainActivity.java)

```java
package com.madlab.graphicalprimitivesandanimation;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.graphics.drawable.shapes.Shape;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Get references to the 3 buttons
        final Button shapesButton = findViewById(R.id.shapesButton);
        final Button animateButton = findViewById(R.id.animateButton);
        final Button transformButton = findViewById(R.id.transformButton);

        // Set On-Click Listeners to all of them
        shapesButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent shapesIntent = new Intent(MainActivity.this, ShapesActivity.class);
                startActivity(shapesIntent);
            }
        });
        animateButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent animateIntent = new Intent(MainActivity.this, AnimateActivity.class);
                startActivity(animateIntent);
            }
        });
        transformButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent transformIntent = new Intent(MainActivity.this, TransformActivity.class);
                startActivity(transformIntent);
            }
        });
    }
}
```

Primitive Shapes

XML (activity_shapes.xml)

```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=".ShapesActivity">

    <ImageView
        android:id="@+id/canvasImage"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@color/white"/>

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

Java (ShapesActivity.java)

```java
package com.madlab.graphicalprimitivesandanimation;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.graphics.*;
import android.widget.ImageView;

public class ShapesActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_shapes);

        // Getting a reference to the canvas (ImageView)
        final ImageView canvasImage = findViewById(R.id.canvasImage);

        // Setting up a Bit map over which graphics can be drawn
        Bitmap bitmap = Bitmap.createBitmap(1080, 1920, Bitmap.Config.
    ARGB_8888);
        canvasImage.setImageBitmap(bitmap);
        Canvas canvas = new Canvas(bitmap);
        Paint paint = new Paint();
        paint.setTextSize(50);

        // Painting the shapes with their names as captions
        paint.setColor(Color.rgb(41, 22, 72));
        canvas.drawText("Rectangle", 150, 1050, paint);
        canvas.drawRect(50, 100, 450, 1000, paint);

        paint.setColor(Color.rgb(80, 0, 40));
        canvas.drawText("Square", 710, 700, paint);
        canvas.drawRect(550 ,150 ,1050 ,650, paint);

        paint.setColor(Color.rgb(0, 77, 81));
        canvas.drawText("Circle", 750, 1200, paint);
        canvas.drawCircle(800, 1000, 150, paint);

        paint.setColor(Color.rgb(165, 56, 0));
        canvas.drawText("Arc", 250, 1450, paint);
        RectF oval = new RectF(50, 1150, 550, 1400);
        canvas.drawArc(oval, 0, 180, false, paint);

        paint.setColor(Color.BLACK);
        canvas.drawText("Line", 900, 1500, paint);
        canvas.drawLine(820, 1300, 820, 1700, paint);
    }
}
```

Animation using an Image

XML (activity_animate.xml)

```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=".AnimateActivity">

    <ImageView
        android:id="@+id/canvasAnimate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:src="@drawable/car"/>

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

Java (AnimateActivity.java)

```java
package com.madlab.graphicalprimitivesandanimation;

import androidx.appcompat.app.AppCompatActivity;
import android.animation.*;
import android.os.Bundle;
import android.widget.ImageView;

public class AnimateActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_animate);

        // Getting a reference to the canvas (ImageView)
        ImageView canvasAnimate = findViewById(R.id.canvasAnimate);

        ObjectAnimator moveForwards = ObjectAnimator.ofFloat(canvasAnimate, "x", 150).setDuration(1000);
        ObjectAnimator moveBackwards = ObjectAnimator.ofFloat(canvasAnimate, "x", -150).setDuration(2000);
        ObjectAnimator moveToOriginal = ObjectAnimator.ofFloat(canvasAnimate, "x", 0).setDuration(2000);
        AnimatorSet animation = new AnimatorSet();

        // Reverse the direction
        animation.play(moveBackwards).before(moveForwards);
        animation.play(moveForwards).before(moveToOriginal);

        // Loop
        animation.addListener(new AnimatorListenerAdapter() {
            @Override
            public void onAnimationEnd(Animator animation) {
                super.onAnimationEnd(animation);
                animation.start();
            }
        });

        // Start the animation
        animation.start();
    }
}
```

Transformations

XML (activity_transform.xml)

```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=".TransformActivity">

    <ImageView
        android:id="@+id/rotateImage"
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:layout_marginTop="50dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/ic_rotate"/>

    <ImageView
        android:id="@+id/profileImage"
        android:layout_width="150dp"
        android:layout_height="150dp"
        app:layout_constraintBottom_toTopOf="@+id/zoomImage"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/rotateImage"
        app:srcCompat="@drawable/ic_profile" />

    <ImageView
        android:id="@+id/zoomImage"
        android:layout_width="75dp"
        android:layout_height="75dp"
        android:layout_marginBottom="50dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:srcCompat="@drawable/ic_zoom"/>

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

Animation XML (`rotate.xml`)

```xml
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
   android:interpolator="@android:anim/cycle_interpolator">
    <rotate android:fromDegrees="0"
        android:toDegrees="360"
        android:pivotX="50%"
        android:pivotY="50%"
        android:duration="4000"
        android:repeatCount="infinite"
        android:repeatMode="restart"/>
</set>
```

Animation XML (`fade.xml`)

```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="2000"
        android:repeatCount="infinite"
        android:repeatMode="reverse"/>
</set>
```

Animation XML (`zoom.xml`)

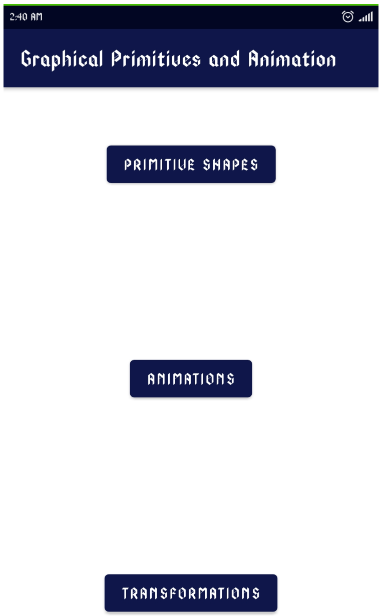
```xml
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <scale
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:duration="2000"
        android:fromXScale="1"
        android:fromYScale="1"
        android:pivotX="50%"
        android:pivotY="50%"
        android:toXScale="2"
        android:toYScale="2"
        android:repeatCount="infinite"
        android:repeatMode="reverse">
    </scale>
</set>
```

Java (TransformActivity.java)

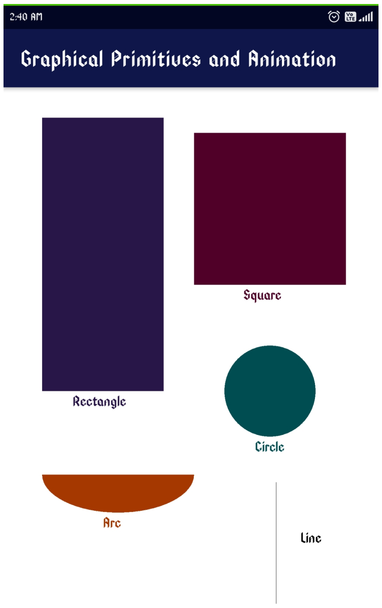
```java
package com.madlab.graphicalprimitivesandanimation;

import androidx.appcompat.app.AppCompatActivity;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.os.Bundle;
import android.widget.ImageView;

public class TransformActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_transform);

        // Get the 3 ImageViews
        final ImageView rotateImage = findViewById(R.id.rotateImage);
        final ImageView profileImage = findViewById(R.id.profileImage);
        final ImageView zoomImage = findViewById(R.id.zoomImage);

        // Define the animations using the XML resources
        Animation rotateAnimation = AnimationUtils.loadAnimation(getApplicationContext(), R.anim.rotate);
        Animation fadeAnimation = AnimationUtils.loadAnimation(getApplicationContext(), R.anim.fade);
        Animation zoomAnimation = AnimationUtils.loadAnimation(getApplicationContext(), R.anim.zoom);

        // Start the animations
        rotateImage.startAnimation(rotateAnimation);
        profileImage.startAnimation(fadeAnimation);
        zoomImage.startAnimation(zoomAnimation);
    }
}
```

App Screenshots

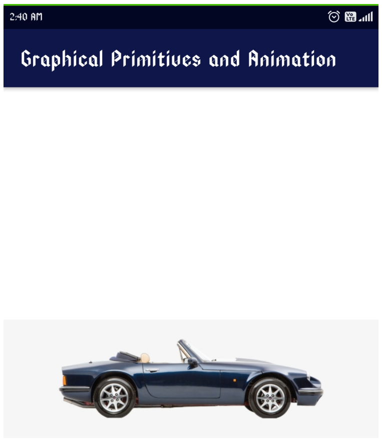
Home

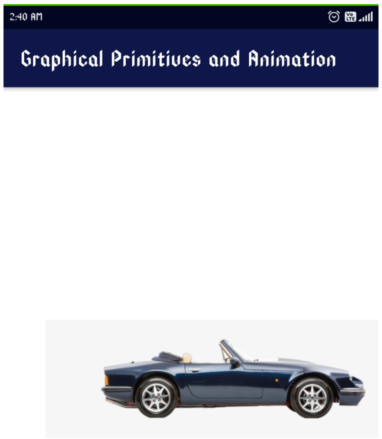


Graphical Primitives

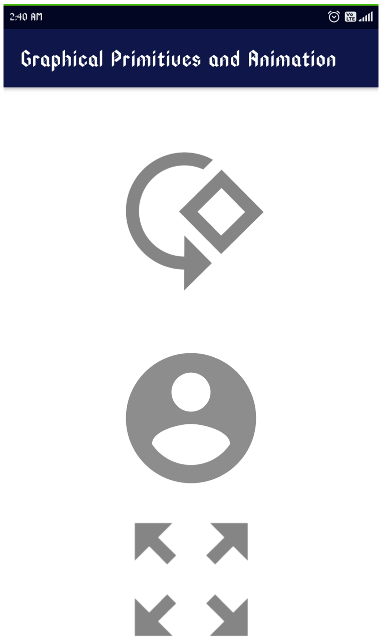


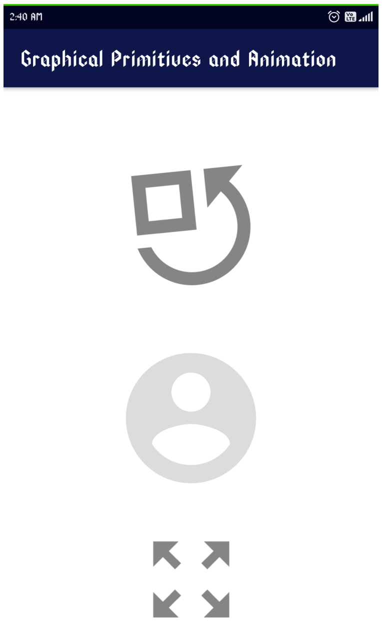
Animation





Transformation


2:40 AM
Graphical Primitives and Animation


2:40 AM
Graphical Primitives and Animation