Department of CSE
SSN College of Engineering

Vishakan Subramanian - 18 5001 196 - Semester VII

18 August 2021

UCS 1711 - Mobile Application Development Lab

Exercise 3: Application Using Graphical Primitives

Aim:

To develop an Android application that generates the following using graphical primitives.

1. Draw shapes such as Line, Circle, Rectangle and Arc.
2. Perform animations on an image.
3. Perform transformations like rotation and zoom.
4. Draw a car and animate the car.

Code: Main Activity:

```
package com.example.graphics;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
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);

        final Button shapesButton = findViewById(R.id.shapesButton);
        final Button animateButton = findViewById(R.id.animateButton);
        final Button transformButton = findViewById(R.id.transformButton);

        shapesButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent shapesIntent = new Intent(MainActivity.this,
   ShapesActivity.class);
                startActivity(shapesIntent);
            }
        });
        animateButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent animateIntent = new Intent(MainActivity.this,
   AnimateActivity.class);
                startActivity(animateIntent);
            }
        });
        transformButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent transformIntent = new Intent(MainActivity.this,
   TransformActivity.class);
                startActivity(transformIntent);
            }
        });
    }
}
```

Code: Shapes Activity:

```java
package com.example.graphics;

import androidx.appcompat.app.AppCompatActivity;

import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.RectF;
import android.os.Bundle;
import android.widget.ImageView;

public class ShapesActivity extends AppCompatActivity {

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

        final ImageView canvasImage = findViewById(R.id.canvasImage);

        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);

        paint.setColor(Color.MAGENTA);
        canvas.drawText("Rectangle",720,350, paint);
        canvas.drawRect(700,400,950,900, paint);

        paint.setColor(Color.BLUE);
        canvas.drawText("Circle",220,350, paint);
        canvas.drawCircle(300,550,150, paint);

        paint.setColor(Color.RED);
        canvas.drawText("Square",220,1100, paint);
        canvas.drawRect(150,1150,450,1450, paint);

        paint.setColor(Color.BLACK);
        canvas.drawText("Line",780,1100,paint);
        canvas.drawLine(820,1150,820,1450, paint);

        paint.setColor(Color.parseColor("#FFA500"));    //orange
        RectF oval = new RectF(200, 100, 800, 200);
        canvas.drawText("Arc", 475, 100, paint);
        canvas.drawArc(oval, 0, 180, false, paint);
    }
}
```

Code: Animate Activity:

```java
package com.example.graphics;

import androidx.appcompat.app.AppCompatActivity;

import android.animation.Animator;
import android.animation.AnimatorListenerAdapter;
import android.animation.AnimatorSet;
import android.animation.ObjectAnimator;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.RectF;
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);

        ImageView canvasAnimate = findViewById(R.id.canvasAnimate);

        Paint paint = new Paint();

        Bitmap carBitmap = Bitmap.createBitmap(1920, 1080, Bitmap.Config.ARGB_8888);
        canvasAnimate.setImageBitmap(carBitmap);

        Canvas carCanvas = new Canvas(carBitmap);

        //Car one
        paint.setColor(Color.GREEN);
        carCanvas.drawRect(50, 100, 450, 300, paint);

        paint.setColor(Color.YELLOW);
        RectF roof1 = new RectF(100, 0, 400, 200);
        carCanvas.drawArc(roof1, 180, 180, false, paint);

        paint.setColor(Color.DKGRAY);
        carCanvas.drawCircle(150, 300, 50, paint);
        carCanvas.drawCircle(350, 300, 50, paint);

        //Car two
        paint.setColor(Color.BLUE);
        carCanvas.drawRect(450, 600, 50, 800, paint);

        paint.setColor(Color.RED);
        RectF roof2 = new RectF(100, 500, 400, 700);
        carCanvas.drawArc(roof2, 180, 180, false, paint);

        paint.setColor(Color.BLACK);
        carCanvas.drawCircle(150, 800, 50, paint);
        carCanvas.drawCircle(350, 800, 50, paint);

        ObjectAnimator animateCar = ObjectAnimator.ofFloat(canvasAnimate, "x",  800);
        animateCar.setDuration(2000);
        AnimatorSet animation = new AnimatorSet();

        //Reverse the direction
        animation.play(animateCar).before(ObjectAnimator.ofFloat(canvasAnimate, "x", 0).setDuration(2000));

        //To loop the animation indefinitely
        animation.addListener(new AnimatorListenerAdapter() {
            @Override
            public void onAnimationEnd(Animator animation) {
                super.onAnimationEnd(animation);
                animation.start();
            }
        });

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

Code: Transform Activity:

```java
package com.example.graphics;

import androidx.appcompat.app.AppCompatActivity;

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

public class TransformActivity extends AppCompatActivity {

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

        final ImageView syncImage = findViewById(R.id.syncImage);
        final ImageView starImage = findViewById(R.id.starImage);
        final ImageView dollarImage = findViewById(R.id.dollarImage);

        Animation rotateAnimation = AnimationUtils.loadAnimation(getApplicationContext(), R.anim.rotate);
        syncImage.startAnimation(rotateAnimation);

        Animation zoomAnimation = AnimationUtils.loadAnimation(getApplicationContext(), R.anim.zoom);
        starImage.startAnimation(zoomAnimation);

        Animation fadeAnimation = AnimationUtils.loadAnimation(getApplicationContext(), R.anim.fade);
        dollarImage.startAnimation(fadeAnimation);
    }
}
```

Code: 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>
```

Code: 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>
```

Code: 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>
```

Code: Main Activity Layout

```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="173dp"
        android:layout_height="70dp"
        android:layout_marginTop="250dp"
        android:text="Shapes"
        android:textSize="18sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/animateButton"
        android:layout_width="173dp"
        android:layout_height="70dp"
        android:layout_marginTop="20dp"
        android:text="Animate"
        android:textSize="18sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/shapesButton" />

    <Button
        android:id="@+id/transformButton"
        android:layout_width="173dp"
        android:layout_height="70dp"
        android:layout_marginTop="20dp"
        android:text="Transform"
        android:textSize="18sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/animateButton" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Code: Shapes Activity Layout

```
<?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>
```

Code: Animate Activity Layout

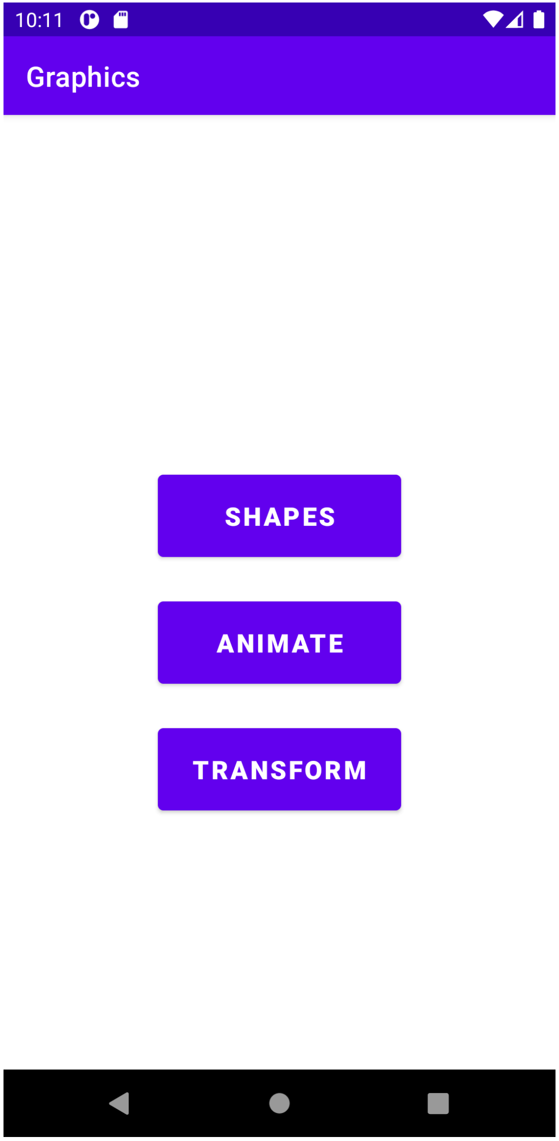
```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    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"
      app:srcCompat="@color/white" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Code: Transform Activity Layout

```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/syncImage"
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:layout_marginTop="50dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.497"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/ic_sync_black_24dp" />

    <ImageView
        android:id="@+id/starImage"
        android:layout_width="75dp"
        android:layout_height="75dp"
        android:layout_marginTop="100dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.493"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/dollarImage"
        app:srcCompat="@drawable/ic_auto_awesome_black_24dp" />

    <ImageView
        android:id="@+id/dollarImage"
        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_toBottomOf="@+id/syncImage"
        app:srcCompat="@drawable/ic_monetization_on_black_24dp" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

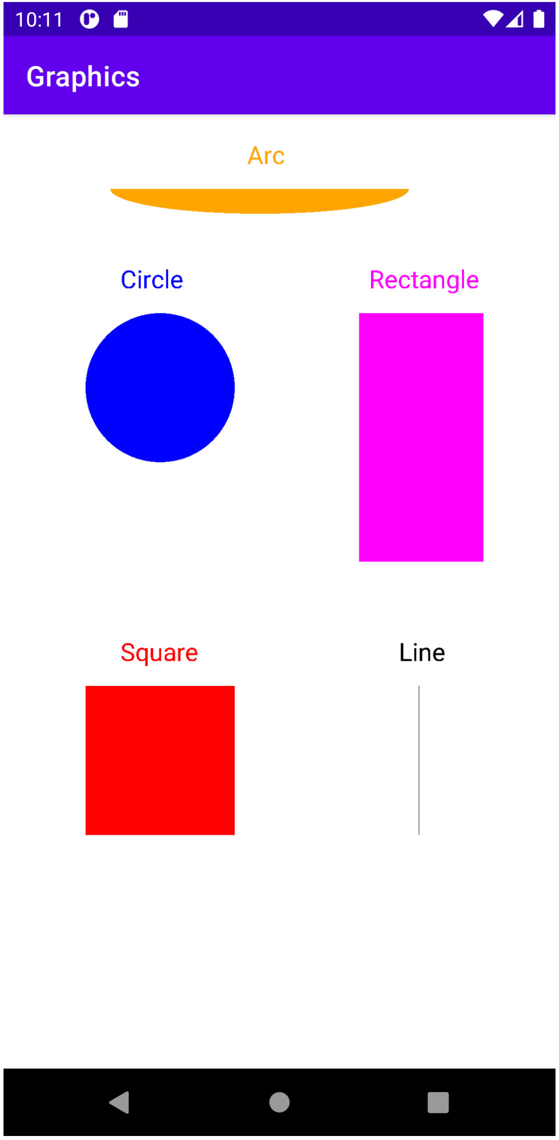
Output: Graphics App Main Screen:

Figure 1: Output: Graphics App Main Screen.



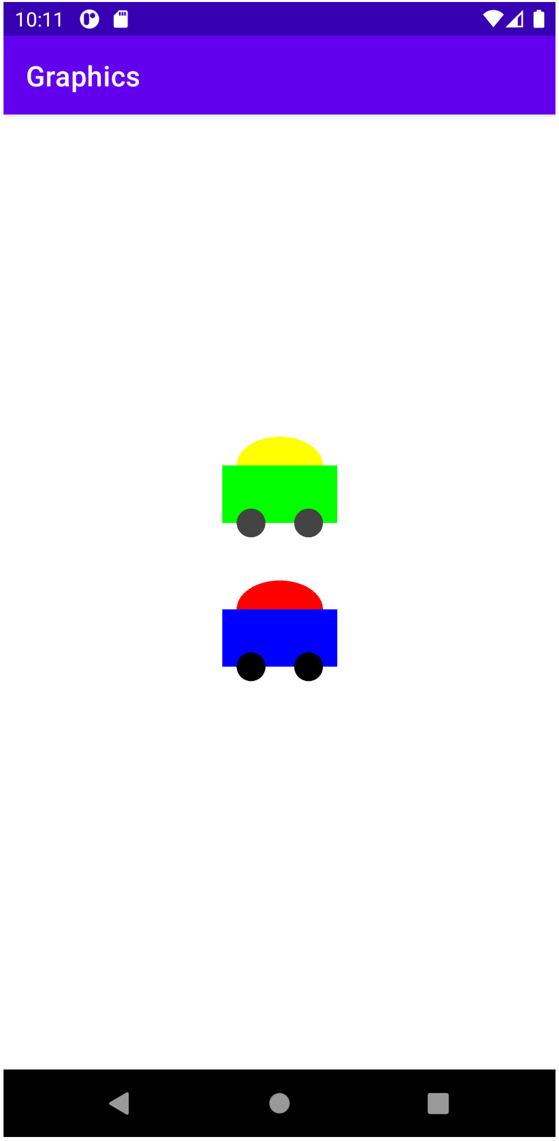
Output: Shapes Screen:

Figure 2: Output: Shapes Screen.



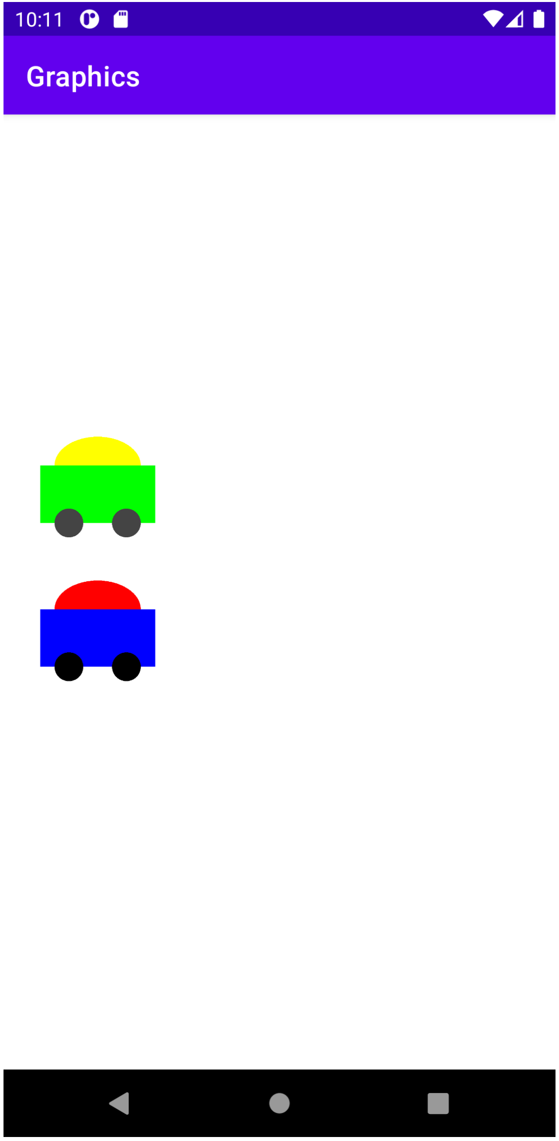
Output: Animations Screen - 1:

Figure 3: Output: Animations Screen - 1.



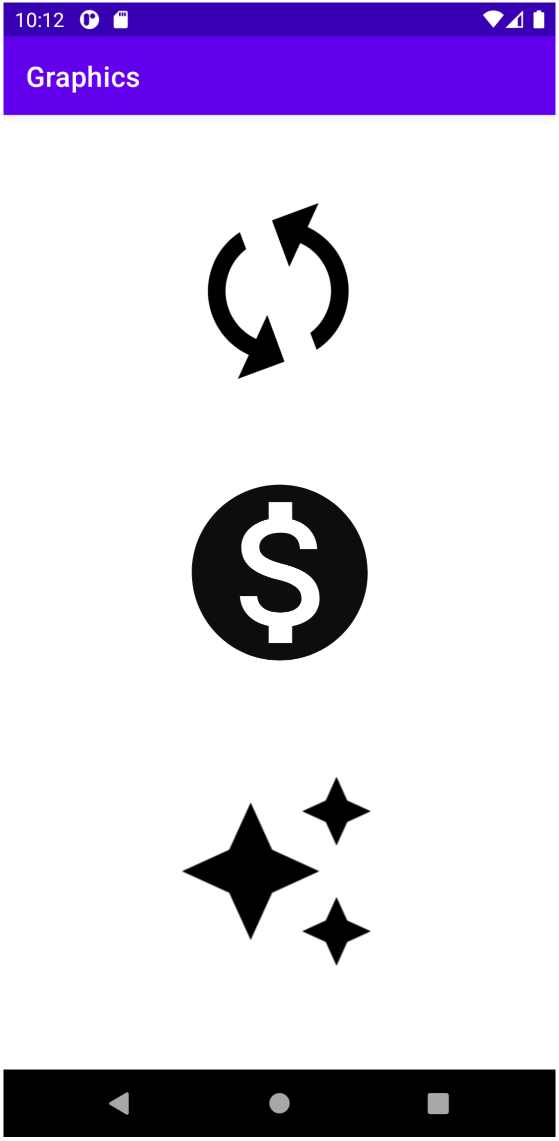
Output: Animations Screen - 2:

Figure 4: Output: Animations Screen - 2.



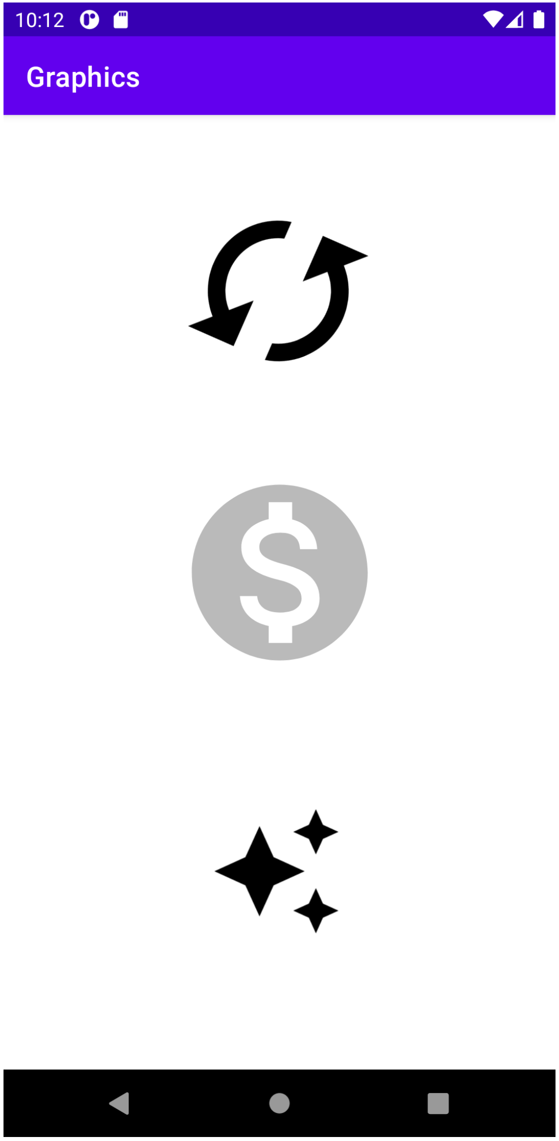
Output: Transformations Screen - 1:

Figure 5: Output: Transformations Screen - 1.



Output: Transformations Screen - 2:

Figure 6: Output: Transformations Screen - 2.



Learning Outcome:

- I understood how to draw graphical primitives like shapes using **Bitmap** and **Canvas** in Android.
- I was able to draw shapes like **Lines, Rectangles, Circles and Arcs** using the Canvas object.
- I was able to set **Colors** to the objects drawn using the Canvas object.
- I drew a simple **car** using the graphical primitives.
- I was able to animate the car to **move forwards and backwards** infinitely using **AnimatorSet** object.
- I understood how to add custom animation specifications with XML-based syntax for Android applications.
- I was able to add a **rotate** transformation to a static image using **AnimationUtils** object and **XML specification**.
- I was able to add a **fade** transformation to a static image using **AnimationUtils** object and **XML specification**.
- I was able to add a **zoom** transformation to a static image using **AnimationUtils** object and **XML specification**.
- I understood how to use several XML-based parameters like **duration, repeatCount, repeatMode, pivotX, pivotY, fromXScale, fromYScale etc.** to control the transformations as desired.