Exercise 3 - Application using graphical primitives

Objective

Design a CAR using Shape drawables with the help of relevant shapes such as Line, Circle, Rectangle and Arc.
a. Move the car forward by pressing forward button so that car moves from a predefined starting point to the predefined endpoint.
b. On pressing backward button, rotate the car to 180 degrees from the current point to the starting point.
c. Implement a Tap-to-zoom animation on any image
d. Implement the Card flipping animation.

Algorithm

1. Initialize the Project:
   - Open Android Studio and create a new Android project.
   - Set up the project with a suitable name and package.

2. Design the Car Layout:
   - Utilize shape drawables to create a car using lines, circles, rectangles, and arcs.
   - Arrange these shapes to form the outline of a car.

3. Implement Forward Movement:
   - Add a "Forward" button to the layout.
   - On pressing the button, move the car from a predefined starting point to the predefined endpoint, creating a forward animation.

4. Implement Backward Movement:
   - Add a "Backward" button to the layout.

- On pressing the button, rotate the car 180 degrees from the current point to the starting point, creating a backward animation.

5. Tap-to-Zoom Animation:
- Implement a tap-to-zoom animation on any image (possibly an image of the car or another component).
- Detect tap gestures, and smoothly zoom in/out on the image.

6. Card Flipping Animation:
- Implement a card flipping animation.
- Utilize rotation animations to make the car or any image appear as if it is flipping like a card.

Features used

Main Features:
- Car Design: Utilization of shape drawables to create a car.
- Forward Movement: Animation of the car moving forward on button press.
- Backward Movement: Animation of the car rotating backward on button press.
- Tap-to-Zoom Animation: Interactive zoom animation on tapping an image.
- Card Flipping Animation: Animation simulating a card flip effect.
- User Interaction: Buttons for user interaction to trigger animations.

Source Code

activity_main.xml

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

    - Car Group (Body + Wheels) >
    <RelativeLayout
        android:id="@+id/carGroup"
        android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content">

<RelativeLayout
    android:id="@+id/carGroupF"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="12dp">

    - Car Body >
    <ImageView
        android:id="@+id/carBodyImageViewF"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:src="@drawable/car_body" >

    - Left Wheel >
    <ImageView
        android:id="@+id/leftWheelImageViewF"
        android:layout_width="24dp"
        android:layout_height="24dp"
        android:layout_below="@id/carBodyImageViewF"
        android:layout_marginEnd="-24dp"
        android:layout_toStartOf="@id/carBodyImageViewF"
        android:src="@drawable/wheel" >

    - Right Wheel >
    <ImageView
        android:id="@+id/rightWheelImageViewF"
        android:layout_width="24dp"
        android:layout_height="24dp"
        android:layout_below="@id/carBodyImageViewF"
        android:layout_marginStart="-24dp"
        android:layout_toEndOf="@id/carBodyImageViewF"
        android:src="@drawable/wheel" >

/RelativeLayout>

<RelativeLayout
    android:id="@+id/carGroupB"
    android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:layout_margin="12dp">

        - Car Body >
        <ImageView
            android:id="@+id/carBodyImageViewB"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerInParent="true"
            android:src="@drawable/car_body" >

        - Left Wheel >
        <ImageView
            android:id="@+id/leftWheelImageViewB"
            android:layout_width="24dp"
            android:layout_height="24dp"
            android:layout_below="@id/carBodyImageViewB"
            android:layout_marginEnd="-24dp"
            android:layout_toStartOf="@id/carBodyImageViewB"
            android:src="@drawable/wheel" >

        - Right Wheel >
        <ImageView
            android:id="@+id/rightWheelImageViewB"
            android:layout_width="24dp"
            android:layout_height="24dp"
            android:layout_below="@id/carBodyImageViewB"
            android:layout_marginStart="-24dp"
            android:layout_toEndOf="@id/carBodyImageViewB"
            android:src="@drawable/wheel" >

    /RelativeLayout>

/RelativeLayout>

<RelativeLayout
    android:id="@+id/ButtonGrp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true">
```

```
        <Button
            android:id="@+id/changeDirectionButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="186dp"
            android:text="Change Direction" >
        <Button
            android:id="@+id/accelerateButton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="140dp"
            android:text="Move" >
    /RelativeLayout>
/RelativeLayout>
```

MainActivity.java

```
package com.example.a3;

import android.animation.AnimatorInflater;
import android.animation.AnimatorSet;
import android.animation.ObjectAnimator;
import android.animation.PropertyValuesHolder;
import android.animation.ValueAnimator;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.DisplayMetrics;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {
    private int direction;
    private boolean isFront;
    private boolean isZoomed;

    public MainActivity() {
        this.direction = 1;
        this.isFront = true;
        this.isZoomed = false;
    }
```

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

        View carGroup = findViewById(R.id.carGroup);
        Button accelerateBtn = findViewById(R.id.accelerateButton);
        Button changeDirBtn = findViewById(R.id.changeDirectionButton);

        carGroup.setOnClickListener((e) > {
            if (isZoomed) zoomOutCar(carGroup);
            else zoomInCar(carGroup);

            isZoomed = !isZoomed;
        });

        DisplayMetrics displayMetrics = getResources().getDisplayMetrics(); float
        scale = displayMetrics.density;

        View front = findViewById(R.id.carGroupF);
        View back = findViewById(R.id.carGroupB);

        / Set camera distance for front and back
        front.setCameraDistance(8000 * scale);
        back.setCameraDistance(8000 * scale);

        / Load front and back animations
        AnimatorSet front_animation = (AnimatorSet)
   AnimatorInflater.loadAnimator(this, R.animator.flip_front);
            AnimatorSet back_animation = (AnimatorSet)
AnimatorInflater.loadAnimator(this, R.animator.flip_back);

        changeDirBtn.setOnClickListener(e > {
            this.direction *= -1;

            if (isFront) {
                front_animation.setTarget(front);
                back_animation.setTarget(back);
                front_animation.start();
                back_animation.start();
```

```
                isFront = false;
            } else {
                front_animation.setTarget(back);
                back_animation.setTarget(front);
                back_animation.start();
                front_animation.start();
                isFront = true;
            }
        });
        accelerateBtn.setOnClickListener(e > {
            float currentTranslationX = carGroup.getTranslationX();
            float targetTranslationX = currentTranslationX + this.direction * 50f;
                  int screenWidth = getResources().getDisplayMetrics().widthPixels;

            if (targetTranslationX > screenWidth) {
                targetTranslationX = 0;
            }

            ValueAnimator animator = ValueAnimator.ofFloat(currentTranslationX,
targetTranslationX);
            animator.setDuration(500); / Set the animation duration in
milliseconds
            animator.addUpdateListener(animation > {
                float animatedValue = (float) animation.getAnimatedValue();
                carGroup.setTranslationX(animatedValue);
            });
            animator.start();
        });
    }

    private void zoomInCar(View view) {
        PropertyValuesHolder scaleX =
PropertyValuesHolder.ofFloat(View.SCALE_X, 1.5f);
        PropertyValuesHolder scaleY =
PropertyValuesHolder.ofFloat(View.SCALE_Y, 1.5f);
        ObjectAnimator animator = ObjectAnimator.ofPropertyValuesHolder(view,
scaleX, scaleY);
        animator.setDuration(500); / Adjust the duration as needed
        animator.start();
    }
```

```
    private void zoomOutCar(View view) {
        PropertyValuesHolder scaleX =
PropertyValuesHolder.ofFloat(View.SCALE_X, 1f);
        PropertyValuesHolder scaleY =
PropertyValuesHolder.ofFloat(View.SCALE_Y, 1f);
        ObjectAnimator animator = ObjectAnimator.ofPropertyValuesHolder(view,
scaleX, scaleY);
        animator.setDuration(500); / Adjust the duration as needed
        animator.start();
    }
}
```

res/drawable/car_body.xml

```
- car_body.xml >
<shape xmlns:android="http: /schemas.android.com/apk/res/android">
    <solid android:color="#FF1100" >
    <size android:width="128dp" android:height="60dp" >
    <corners android:radius="10dp" >
/shape>
```

res/drawable/wheel.xml

```
- wheel.xml >
<shape xmlns:android="http: /schemas.android.com/apk/res/android">
    <solid android:color="#000000" >
    <size android:width="40dp" android:height="40dp" >
    <corners android:radius="20dp" >
/shape>
```

res/animatior/flip_front.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http: /schemas.android.com/apk/res/android">
    <objectAnimator
        android:valueFrom="0"
        android:valueTo="180"
        android:propertyName="rotationY"
        android:duration="1000"
        >
    <objectAnimator
```

```
        android:valueFrom="1.0"
        android:valueTo="0.0"
        android:propertyName="alpha"
        android:startOffset="500"
        android:duration="1"
        >
/set>
```

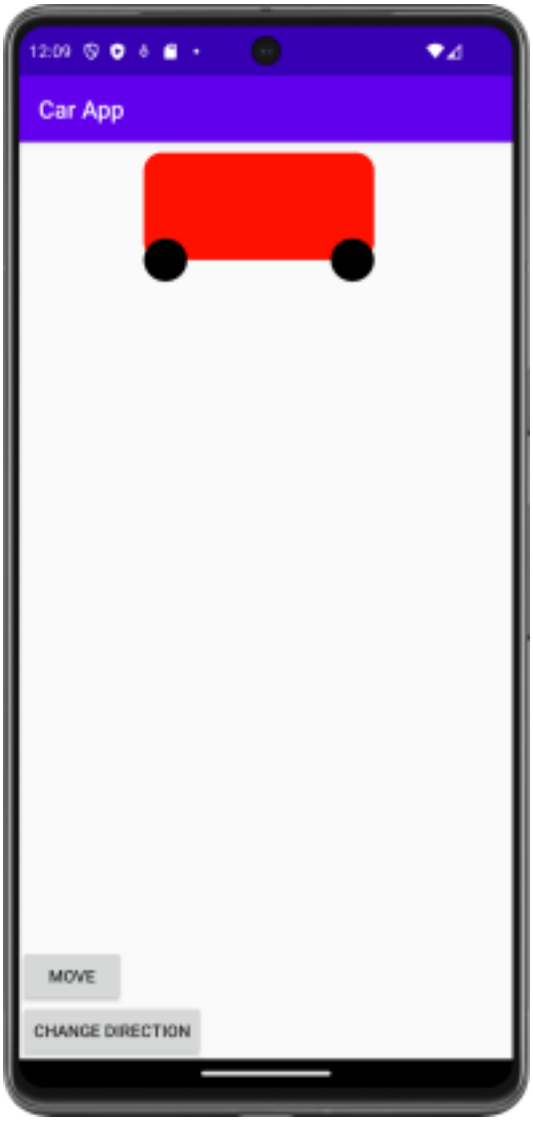
res/animatior/flip_back.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http: /schemas.android.com/apk/res/android">
   <objectAnimator
        android:valueFrom="1.0"
        android:valueTo="0.0"
        android:propertyName="alpha"
        android:duration="0"
        >
   <objectAnimator
        android:valueFrom="180"
        android:valueTo="0"
        android:propertyName="rotationY"
        android:repeatMode="reverse"
        android:duration="1000"
        >

   <objectAnimator
        android:valueFrom="0.0"
        android:valueTo="1.0"
        android:propertyName="alpha"
        android:startOffset="500"
        android:duration="0"
        >

/set>
```

Output Screenshots



Result

Thus a car animation was implemented using drawables

Best Practices

1. User friendly design
2. Readable layouts
3. modularity

Learning outcomes

- Implement Drawable
- Implement animation