Exercise 3 - Application Development using basic graphical primitives

Sabarivasan V
205001085
CSE-B

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 the forward button so that the car moves from a predefined starting point to the predefined endpoint. b. On pressing the 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.

Android widgets used :

1. TextView - to display the text typed in the keyboard
2. Buttons - to create keys in the keyboard
3. GridLayout - to achieve layout of the keyboard

car.xml

```
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
    <!-- Car Body (Rectangle) -->
    <item android:drawable="@drawable/car_body" />

<!--    &lt;!&ndash; Car Roof (Rectangle) &ndash;&gt;-->
    <item
        android:drawable="@drawable/car_
        roof" android:gravity="top"
        />

    <!-- Car Windows (Rectangles) -->
    <item
        android:drawable="@drawable/car_wi
        ndow" android:top="10dp"
        android:left="20
        dp"
        android:right="4
        5dp"
        android:bottom="
        20dp"
        />
    <item
        android:drawable="@drawable/car_wi
        ndow" android:top="10dp"
        android:left="75
        dp"
        android:right="5
        dp"
        android:bottom="
        20dp"
        />

    <!-- Car Wheels (Circles) -->
    <item
        android:drawable="@drawable/car_w
        heel"
        android:gravity="left|bottom"
        android:left="10dp"
```

```
        android:bottom="0dp"
        />
    <item
        android:drawable="@drawable/car_wheel"
        android:gravity="right|bottom"
        android:right="10dp"
        android:bottom="0dp"
        />
</layer-list>
```

car_roof.xml

```
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="#000000" /> <!-- White color -->
    <corners android:radius="16dp" />
    <size android:width="20dp" android:height="15dp" />
</shape>
```

road.xml

```
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
    <!-- Road Background (Gray) -->
    <item>
        <shape android:shape="rectangle">
            <solid android:color="#808080" /> <!-- Gray color -->
        </shape>
    </item>

     Lane Markings (White Lines)
    <item android:drawable="@drawable/lane_marking"
        android:top="10dp"
        android:right="80dp"
        android:left="10dp"
        android:bottom="10dp"/>
    <item android:drawable="@drawable/lane_marking"
        android:top="10dp"
        android:right="45dp"
        android:left="45dp"
        android:bottom="10dp"/>
    <item android:drawable="@drawable/lane_marking"
    android:top="10dp"
    android:right="10dp"
    android:left="80dp"
    android:bottom="10dp"/>
</layer-list>
```

Lane_marking.xml

```
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="#FFFFFF" /> <!-- White color -->
    <size android:width="10dp" android:height="2dp" />
</shape>
```

Car_window.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="#FFFFFF" /> <!-- White color -->
    <corners android:radius="4dp" />
    <size android:width="20dp" android:height="20dp" />
</shape>
```

Car_body.xml

```
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="#FFA500" /> <!-- Orange color -->
    <corners android:radius="16dp" />
    <size android:width="100dp" android:height="40dp"/>
</shape>
```

Car_wheel.xml

```
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="#000000" /> <!-- Black color -->
    <size android:width="20dp" android:height="20dp" />
    <corners android:radius="10dp" />
</shape>
```

Main_activity.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/carImageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="206dp"
        android:src="@drawable/car"
        app:layout_constraintTop_toTopOf="parent"
        tools:layout_editor_absoluteX="0dp" />

    <ImageView
        android:id="@+id/roadImageView"
        android:layout_width="410dp"
        android:layout_height="111dp"
        android:layout_marginTop="246dp"
        android:layout_marginEnd="1dp"
        android:src="@drawable/road"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/forwardButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="57dp"
        android:layout_marginTop="91dp"
        android:text="Forward"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/roadImageView" />

    <Button
        android:id="@+id/sunFlipId"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="140dp"
        android:layout_marginEnd="170dp"
        android:layout_marginBottom="45dp"
        android:text="Flip Image"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="parent" />

    <Button
        android:id="@+id/backwardButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="91dp"
        android:layout_marginEnd="54dp"
        android:text="Backward"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/roadImageView" />

    <ImageView
        android:id="@+id/sunId"
        android:layout_width="79dp"
        android:layout_height="55dp"
        android:layout_marginStart="278dp"
        android:layout_marginTop="43dp"
        android:layout_marginEnd="54dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="?attr/actionModeCloseDrawable" />

    <!-- Add other UI elements here if needed -->
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

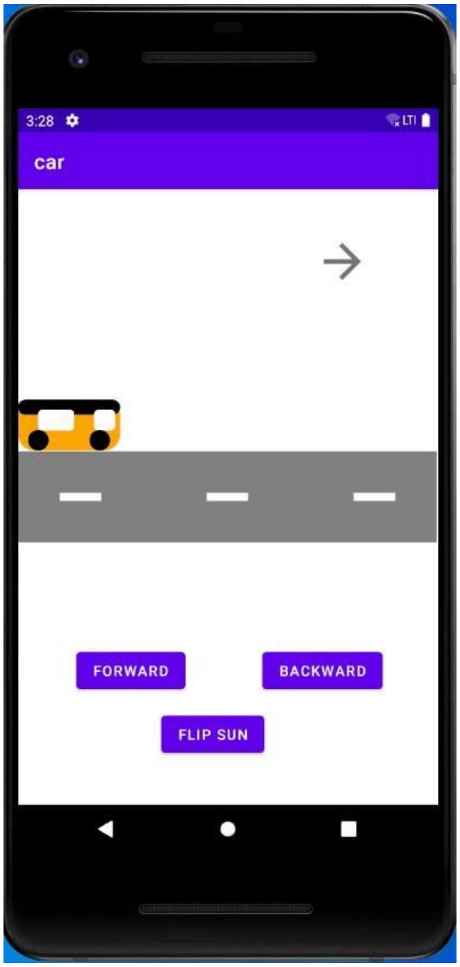
```
package com.example.car;
import androidx.appcompat.app.AppCompatActivity;
import android.animation.ObjectAnimator;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.view.animation.AccelerateDecelerateInterpolator;
import android.widget.Button;
import android.widget.ImageView;

public class MainActivity extends AppCompatActivity {


    private ImageView carImageView;
    private ImageView sunImageView;
    private Button forwardButton;
    private Button backwardButton;
    private Button sunFlipButton;

    private int carXPosition = 0; // Initial X position of the car
    private final int endpoint = 800;// Predefined endpoint
    private final int startpoint = 0;
    boolean isClicked=false;

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

        carImageView = findViewById(R.id.carImageView);
        sunImageView = findViewById(R.id.sunId);
        forwardButton = findViewById(R.id.forwardButton);
        backwardButton =
        findViewById(R.id.backwardButton);
        sunFlipButton = findViewById(R.id.sunFlipId);

        forwardButton.setOnClickListener(new View.OnClickListener() {
            boolean isOperationInProgress = false;
            @Override
            public void onClick(View v) {
                if (!isOperationInProgress) {
                    isOperationInProgress = true;
                    backwardButton.setEnabled(false);
                    forwardButton.setEnabled(false);
                    final Handler handler = new Handler();
                    final Runnable carMovement = new Runnable() {
                        @Override
                        public void run() {
                            carXPosition +=
                            10;
                            if (carXPosition <= endpoint) {
                                carImageView.setX(carXPosition);
                                handler.postDelayed(this, 100); // 100
milliseconds delay                           } else {
                                                 isOperationInPr
                                                 ogress = false;
                                                 backwardButton.
                                                 setEnabled(true
                                                 );
                                                 forwardButton.s
                                                 etEnabled(true)
                                                 ;
                            }
                        }
                    };
                                              handler.post(carMovement);
                                          }
            }
        });


        backwardButton.setOnClickListener(new View.OnClickListener() {
            boolean isOperationInProgress = false;
            private boolean isCarFlipped = false;
            @Override
            public void onClick(View v) {
                if (!isOperationInProgress) {
                    isOperationInProgress = true;
                    backwardButton.setEnabled(false);
                    forwardButton.setEnabled(false);
                    flipCar();
                    final Handler handler = new Handler();
                    final Runnable carMovement = new Runnable() {
                        @Override
                        public void run() {
```

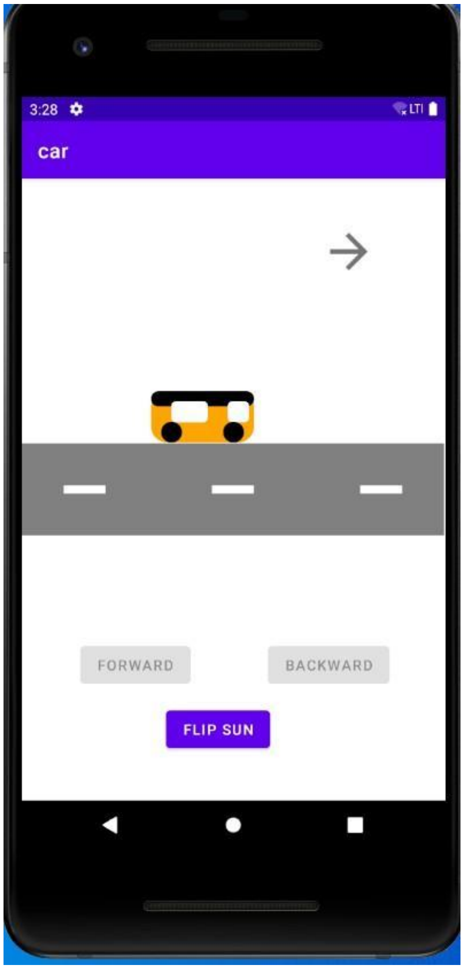
```
carXPosition -=
10;
if (carXPosition >= startpoint) {
```

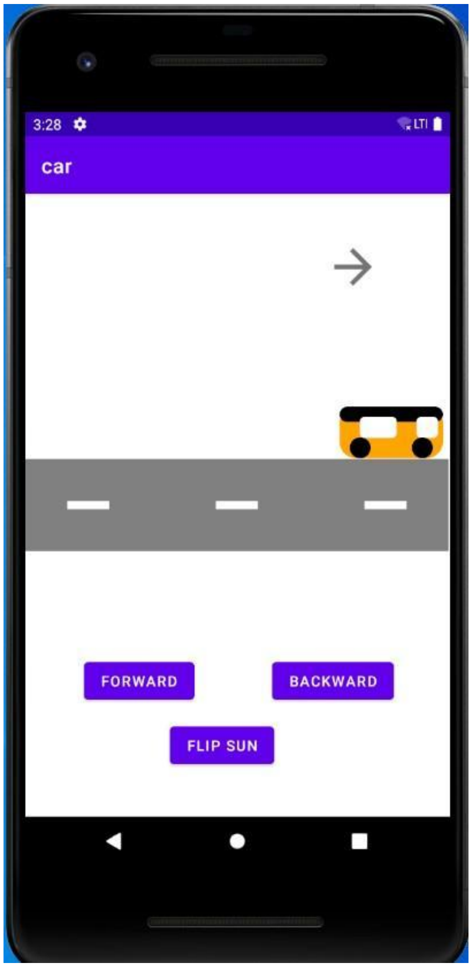
```
                                                carImageView.setX(car
                                                XPosition);
milliseconds delay                              handler.postDelayed(t
                                                his, 100); // 100

                                            } else {
                                                isOperationInPr
                                                ogress = false;
                                                backwardButton.
                                                setEnabled(true
                                                );
                                                forwardButton.s
                                                etEnabled(true)
                                                ; flipCar();
                            }
                        }
                    };
                    handler.post(carMovement);
                }
            }
            private void flipCar() {
                ObjectAnimator flipAnimator;

                if (isCarFlipped) {
                    flipAnimator = ObjectAnimator.ofFloat(carImageView,
"rotationY", 180f, 0f);
                    isCarFlipped = false;
                } else {
                    flipAnimator = ObjectAnimator.ofFloat(carImageView,
"rotationY", 0f, 180f);
                    isCarFlipped = true;
                }

                flipAnimator.setDuration(500);
                flipAnimator.setInterpolator(new
AccelerateDecelerateInterpolator());
                flipAnimator.start();
            }
        });
        sunImageView.setOnClickListener(new
            View.OnClickListener(){ @Override
            public void onClick(View v) {
                if(!isClicked){
                    sunImageView.setScaleX(2)
                    ;
                    sunImageView.setScaleY(2)
                    ; isClicked=true;
                }
                else{
                    sunImageView.setScaleX(1)
                    ;
                    sunImageView.setScaleY(1)
                    ; isClicked=false;
                }
            }
```

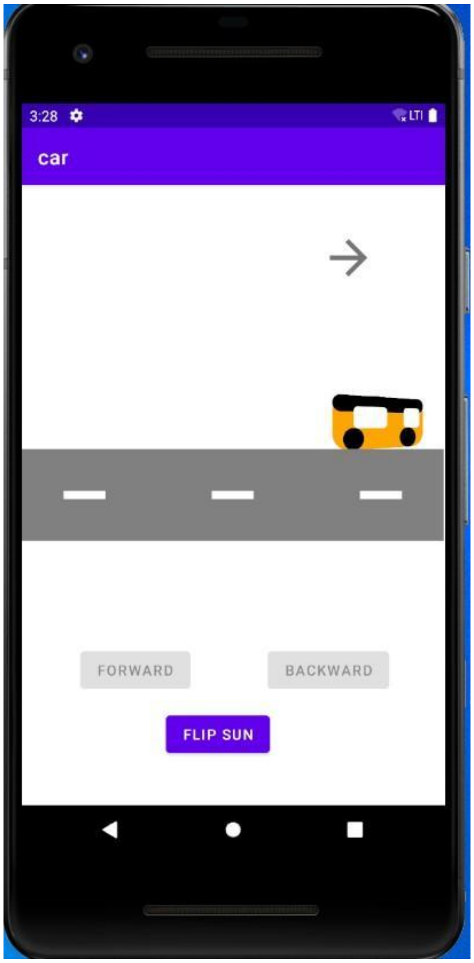
```
        });
        sunFlipButton.setOnClickListener(new View.OnClickListener() {
            private boolean isArrowFlipped = false;
            @Override
            public void onClick(View v) {
                ObjectAnimator flipAnimator;

                if (isArrowFlipped) {
                    flipAnimator = ObjectAnimator.ofFloat(sunImageView,
"rotationY", 180f, 0f);
                    isArrowFlipped = false;
                } else {
                    flipAnimator = ObjectAnimator.ofFloat(sunImageView,
"rotationY", 0f, 180f);
                    isArrowFlipped = true;
                }

                flipAnimator.setDuration(500);
                flipAnimator.setInterpolator(new
AccelerateDecelerateInterpolator());
                flipAnimator.start();
            }
        });
    }
}
```

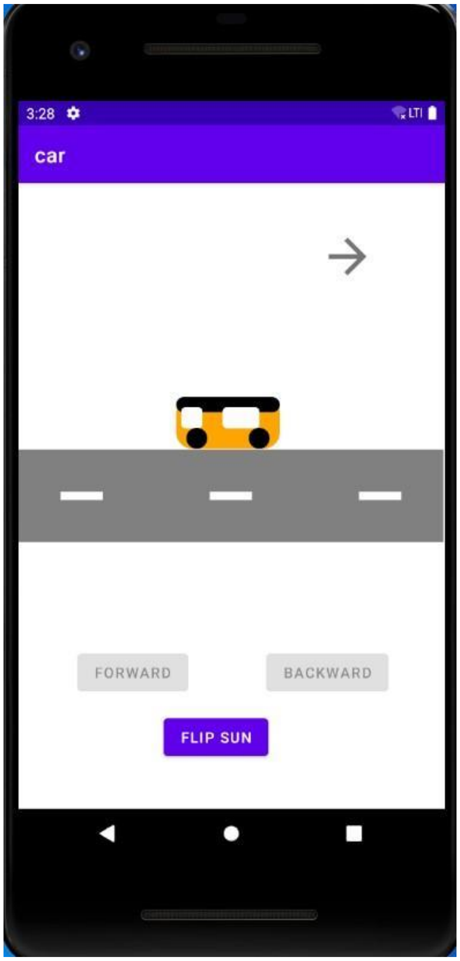
Output

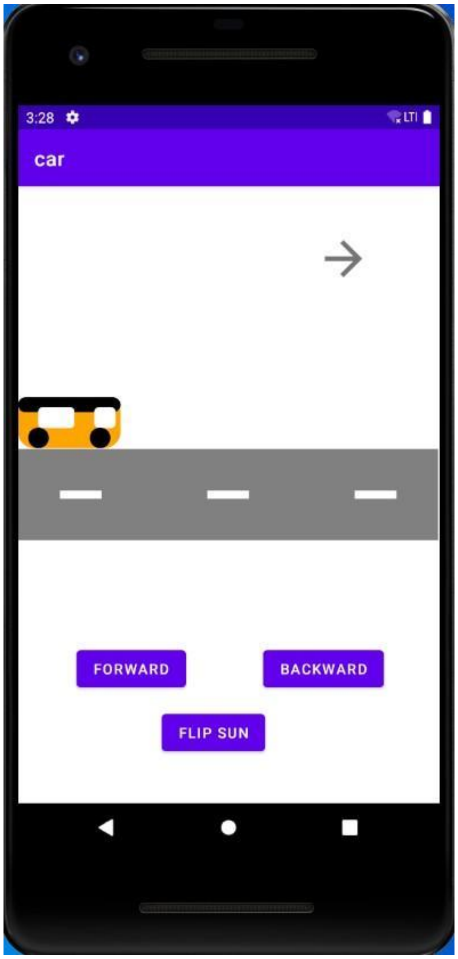

3:28
car
FORWARD
BACKWARD
FLIP SUN

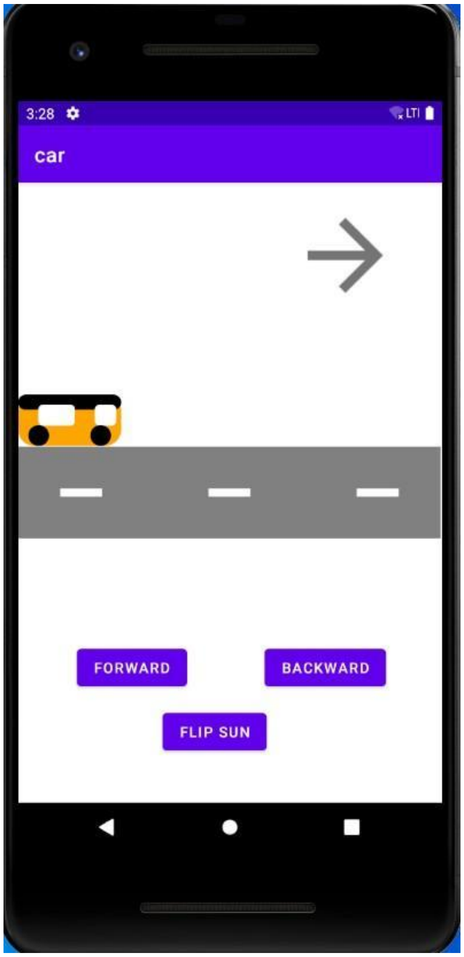

3:28
car
FORWARD
BACKWARD
FLIP SUN

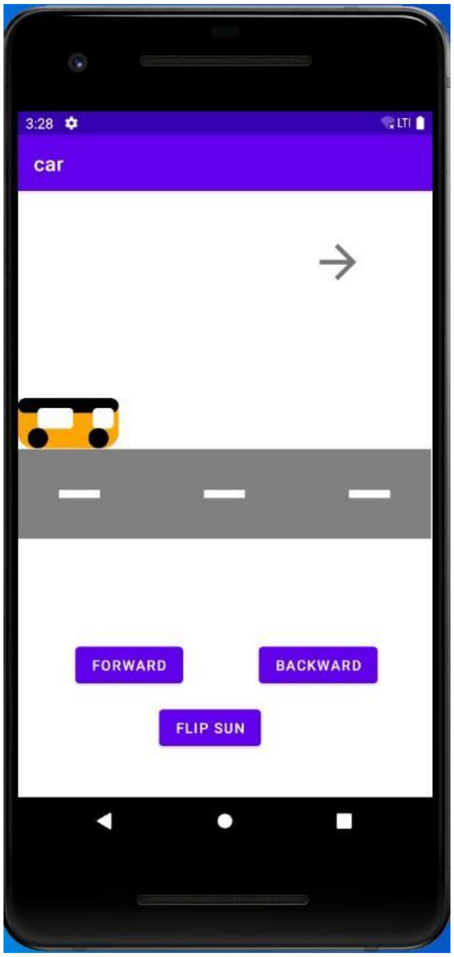

3:28
car
FORWARD
BACKWARD
FLIP SUN

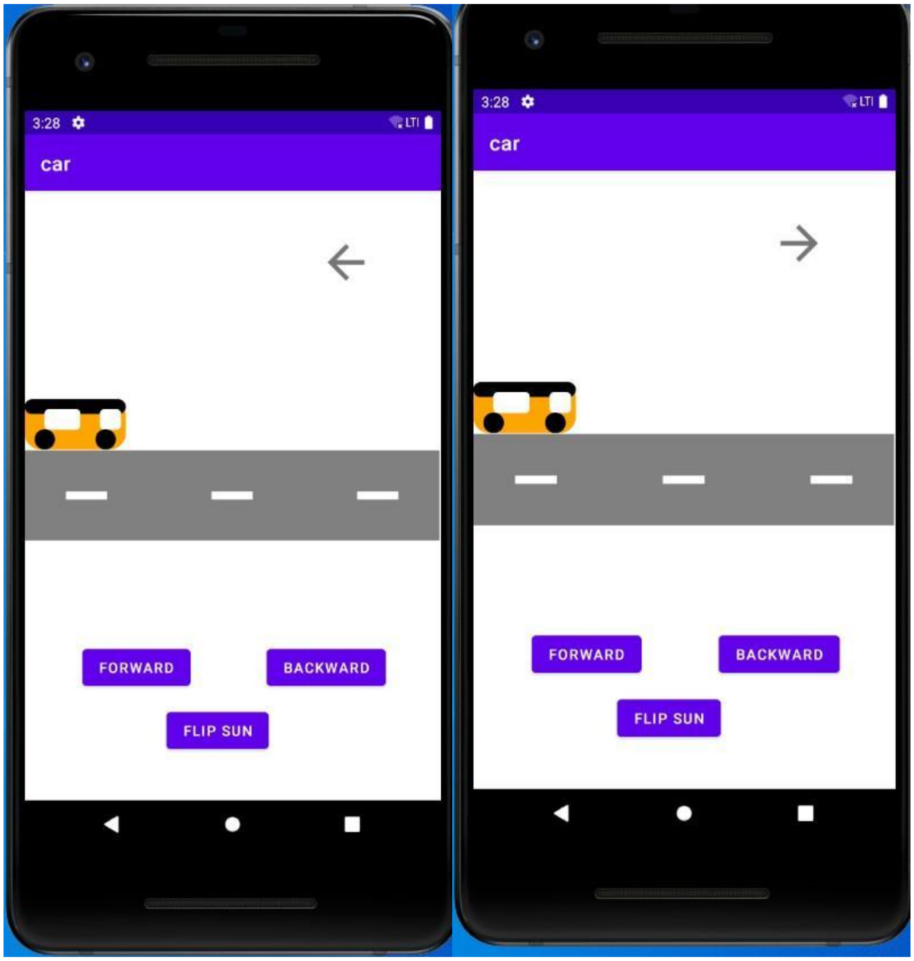

3:28
car
FORWARD
BACKWARD
FLIP SUN


3:28
car
FORWARD
BACKWARD
FLIP SUN


3:28
car
FORWARD
BACKWARD
FLIP SUN


3:28
car
FORWARD
BACKWARD
FLIP SUN


3:28
car
FORWARD
BACKWARD
FLIP SUN



Best Practices followed :

1. The naming convention has been followed to name all the variables used in the code
2. Appropriate versions of the java builds are used to run the mobile application
3. Comments explaining the code written have been included.

Learning Outcomes :

1. Develop mobile applications using GUI, Layouts and Event Listener
2. Develop a mobile app for simple needs
3. Learnt to build custom images by defining .xml file
4. Learnt to rotate,scale images.
5. Learnt to combine images and functions to create a moving effect