```java
/**
 * Edited by Tony Shi
 * Honors Software Android
 * Drawing Project
 * Due Nov 28
 * points comment next to path commands are used to find x,y
values on graph paper.
 * DrawArc is unable to be used due to minimum API
requirement(current: API 15. required: API 21)
 */
package com.tonyxr.drawingproject;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.DashPathEffect;
import android.graphics.Paint;
import android.graphics.Path;
import android.graphics.Rect;
import android.view.View;



public class DrawingProject extends AppCompatActivity {
    DemoView demoview;

    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        demoview = new DemoView(this);
        setContentView(demoview);
    }

    private class DemoView extends View
    {
        public DemoView(Context context)
        {
            super(context);
        }

        @Override protected
        void onDraw(Canvas canvas)
        {
```

```
            super.onDraw(canvas);

            int x = 0;                  // horizontal placement
of graphic shapes
            int y = 0;                  // vertical placement of
graphic shapes

            Paint paint = new Paint();

            // make the entire canvas blue
            paint.setColor(Color.BLUE);
            canvas.drawPaint(paint);

            paint.setAntiAlias(true);
            Path pathCar = new Path();
            pathCar.moveTo(50,1400);//F
            pathCar.lineTo(50,1600);//A
            pathCar.lineTo(200,1600);//B
            pathCar.lineTo(200,1500);//C
            pathCar.lineTo(300,1500);//D
            pathCar.lineTo(300,1600);//E
            pathCar.lineTo(700,1600);//R
            pathCar.lineTo(700,1500);//O
            pathCar.lineTo(800,1500);//P
            pathCar.lineTo(800,1600);//Q
            pathCar.lineTo(950,1600);//N
            pathCar.lineTo(950,1400);//M
            pathCar.lineTo(845,1350);//L
            pathCar.lineTo(700,1300);//K
            pathCar.lineTo(600,1100);//J
            pathCar.lineTo(300,1100);//I
            pathCar.lineTo(200,1300);//H
            pathCar.lineTo(100,1300);//G
            pathCar.close();
            paint.setColor(Color.BLACK);
            canvas.drawPath(pathCar, paint);

            //door
            paint.setColor(Color.WHITE);
            canvas.drawRect(400,1430,430,1440,paint);
            paint.setColor(Color.WHITE);
            canvas.drawRect(660,1430,690,1430,paint);

            //left window
            paint.setAntiAlias(true);
            Path pathWindow = new Path();
```

```
pathWindow.moveTo(320,1220);
pathWindow.lineTo(430,1220);
pathWindow.lineTo(430,1380);
pathWindow.lineTo(220,1380);
pathWindow.close();
paint.setColor(Color.WHITE);
canvas.drawPath(pathWindow, paint);

//right window
paint.setAntiAlias(true);
Path pathWindow2 = new Path();
pathWindow2.moveTo(470,1220);
pathWindow2.lineTo(580,1220);
pathWindow2.lineTo(680,1380);
pathWindow2.lineTo(470,1380);
pathWindow2.close();
paint.setColor(Color.WHITE);
canvas.drawPath(pathWindow2, paint);

//door
paint.setColor(Color.WHITE);
paint.setStrokeWidth(3);
canvas.drawLine(450,1200,450,1650,paint);

paint.setColor(Color.WHITE);
paint.setStrokeWidth(3);
canvas.drawLine(300,1650,700,1650,paint);

paint.setColor(Color.WHITE);
paint.setStrokeWidth(3);
canvas.drawLine(200,1400,700,1400,paint);


//wheel right

paint.setColor(Color.GRAY);
canvas.drawCircle(750,1550,50,paint);
paint.setColor(Color.RED);
canvas.drawCircle(750,1550,40,paint);

//wheel left

paint.setColor(Color.GRAY);
canvas.drawCircle(250,1550,50,paint);
paint.setColor(Color.RED);
canvas.drawCircle(250,1550,40,paint);
```

```
            //canvas.drawArc(200,400,300,600,0,90,false,paint);
            //canvas.drawArc requires minimum API 21, we
currently have the minimum API set as API 15

            //wyo symbol
            paint.setAntiAlias(true);
            Path pathWyo = new Path();
            pathWyo.moveTo(200,400);//S
            pathWyo.lineTo(300,400);//W
            pathWyo.lineTo(400,600);//Z
            pathWyo.lineTo(443,400);//C1
            pathWyo.lineTo(547,400);//D1
            pathWyo.lineTo(600,600);//E1
            pathWyo.lineTo(700,400);//F1
            pathWyo.lineTo(800,400);//G1
            pathWyo.lineTo(700,600);//N1
            pathWyo.lineTo(700,800);//H1
            pathWyo.lineTo(600,800);//B1
            pathWyo.lineTo(500,600);//A1
            pathWyo.lineTo(400,800);//V
            pathWyo.lineTo(300,800);//U
            pathWyo.lineTo(300,600);//T
            pathWyo.close();
            paint.setColor(Color.WHITE);
            canvas.drawPath(pathWyo, paint);
            //paint.setColor((int)Color.pack(14,14,164)); API 26
required, suppose color is navy blue


            //ground
            paint.setStyle(Paint.Style.FILL);
            paint.setColor(Color.DKGRAY);
            canvas.drawRect(0,1800,1080,1920,paint);


            // draw some hollow text using STROKE style
            paint.setStyle(Paint.Style.FILL);
            paint.setColor(Color.WHITE);
            paint.setTextSize(72);
            canvas.drawText("TONY", 250, 75, paint);

            // draw some filled text using FILL style
            paint.setStyle(Paint.Style.FILL);
```

```
            paint.setColor(Color.WHITE);
            paint.setAntiAlias(true);        // turn
antialiasing on to smooth out the text
            paint.setTextSize(72);
            canvas.drawText("SHI", 250, 200, paint);


            /**
            // draw a solid blue circle
            paint.setStyle(Paint.Style.FILL);
            paint.setColor(Color.BLUE);
            canvas.drawCircle(20, 20, 15, paint); // originx,
originy, radius

            // draw a solid green rectangle
            // smooth edges
            paint.setColor(Color.GREEN);
            canvas.drawRect(100, 5, 130, 35, paint);   // left,
top, right, bottom

            paint.setStyle(Paint.Style.STROKE);        // next
shape will be hollow, not filled
            paint.setStrokeWidth(1);

            // using a Path object to store 3 line segments that
form a triangle
            Path path = new Path();
            path.moveTo(160, -30);
            path.lineTo(160, 0);
            path.lineTo(180, 0);
            path.close();

            paint.setColor(Color.RED);
            canvas.drawCircle(220, 20, 10, paint);

            // using offset to draw the same triangle in
multiple locations
            path.offset(10, 40);
            paint.setColor(Color.BLACK);
            canvas.drawPath(path, paint);    // first triangle
is black

            path.offset(40, 0);              // next triangle
placed 40 pixels to the right and 0 pixels up or down
            paint.setColor(Color.MAGENTA);
            canvas.drawPath(path, paint);    // reusing the same
```

```
path (i.e. triangle)

            path.offset(30, 40);            // offset is
cumulative
            paint.setColor(Color.GREEN);
            canvas.drawPath(path, paint);
            **/

            /**
            // draw some hollow text using STROKE style
            paint.setStyle(Paint.Style.STROKE);
            paint.setColor(Color.CYAN);
            paint.setTextSize(30);
            canvas.drawText("TONY", 25, 75, paint);

            // draw some filled text using FILL style
            paint.setStyle(Paint.Style.FILL);
            paint.setAntiAlias(true);       // turn
antialiasing on to smooth out the text
            paint.setTextSize(30);
            canvas.drawText("SHI", 25, 110, paint);

             **/

            // draw rotated text
            // get text width and height
            // set desired drawing location
            x = 75;
            y = 185;
            paint.setColor(Color.RED);
            paint.setTextSize(40);
            String word = "TNT";

            // draw bounding rect before rotating text
            Rect rect = new Rect();
            paint.getTextBounds(word, 0, word.length(), rect);
            canvas.translate(x, y);
            paint.setStyle(Paint.Style.FILL);

            // draw unrotated text
            x = 100;
            y = 185;
            paint.setColor(Color.RED);
            canvas.drawText("Not TNT", 0, 0, paint);
            paint.setStyle(Paint.Style.STROKE);
            canvas.drawRect(rect, paint);
```

```
            // undo the translate
            canvas.translate(-x, -y);

            // rotate the canvas on center of the text to draw
            canvas.rotate(-45, x + rect.exactCenterX(), y +
rect.exactCenterY());

            // draw the rotated text
            paint.setStyle(Paint.Style.FILL);
            canvas.drawText(word, x, y, paint);

            // this paragraph of code have issue
            // undo the rotate
            //canvas.restore();
            //canvas.drawText("After canvas.restore()", 50, 250,
paint);

            // draw a thick dashed line
            /**
            DashPathEffect dashPath = new DashPathEffect(new
float[]{20,5}, 1);
            paint.setPathEffect(dashPath);
            paint.setStrokeWidth(8);
            canvas.drawLine(0, 300 , 320, 300, paint);
            **/

            try {
                Thread.sleep(10000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
```