

```
public class Line
{
    private Point from;
    private Point to;
    private Color color;

    public Line(int pX1, int pY1, int pX2, int pY2, Color pColor)
    {
        from = new Point(pX1, pY1);
        to = new Point(pX2, pY2);
        color = pColor;
    }

    public Line(Point pFrom, Point pTo, Color pColor)
    {
        from = pFrom;
        to = pTo;
        color = pColor;
    }

    public Point getFrom()
    {
        return from;
    }

    public void setFrom(Point pFrom)
    {
        from = pFrom;
    }

    public Point getTo()
    {
        return to;
    }

    public void setTo(Point pTo)
    {
        to = pTo;
    }

    void setColor(Color pColor)
    {
        color = pColor;
    }

    public void draw(Graphics g)
    {
        // Dessiner la ligne dans la couleur de la ligne
        g.setColor(color);
        g.drawLine(from.x, from.y, to.x, to.y);
    }

    public String toString()
    {
        return "[" + from.x + "," + from.y + "] -> [" + to.x + "," + to.y + "];"
    }
}
```

```
public class Lines
{
    private ArrayList<Line> allLines = new ArrayList<>();

    public void add(Line l)
    {
        allLines.add(l);
    }

    public void clear()
    {
        // allLines = new ArrayList<Line>();
        allLines.clear();
    }

    public void draw(Graphics g)
    {
        // Chaque ligne se dessine elle-même
        for (int i = 0; i < allLines.size(); i++)
            allLines.get(i).draw(g);
    }

    public Object[] toArray()
    {
        return allLines.toArray();
    }
}
```

```
public class DrawPanel extends javax.swing.JPanel
{
    private Lines lines = null;

    public DrawPanel()
    {
        initComponents();
    }

    public void setLines(Lines l)
    {
        lines = l;
    }

    @Override
    public void paintComponent(Graphics g)
    {
        // Effacer le dessin
        int w = getWidth();
        int h = getHeight();

        g.setColor(Color.WHITE);
        g.fillRect(0, 0, w, h);

        // Dessiner les lignes si présentes
        if (lines != null)
            lines.draw(g);
    }
}
// Skipped: ... initComponents { ... }
// Variables declaration - do not modify//GEN-BEGIN:variables
// End of variables declaration//GEN-END:variables
}
```

```

public class MainFrame extends javax.swing.JFrame
{
    private Lines lines = new Lines();
    private Timer timer = null;

    public MainFrame()
    {
        initComponents();
        drawPanel.setLines(lines);
        timer = new Timer(10, stepButton.getActionListeners()[0]);
        timer.stop();
        stepButton.setVisible(false);
    }
    // Skipped: ... initComponents { ... }
    private void stepButtonActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_stepButtonActionPerformed
        int xMax = drawPanel.getWidth();
        int yMax = drawPanel.getHeight();
        int xFrom = (int) (Math.random()*xMax);
        int yFrom = (int) (Math.random()*yMax);
        int xTo = (int) (Math.random()*xMax);
        int yTo = (int) (Math.random()*yMax);

        Color c = new Color((int)(Math.random()*256),(int)(Math.random()*256),(int)(Math.random()*256),(int)(Math.random()*256));

        Line newLine = new Line(xFrom, yFrom, xTo, yTo, c);

        lines.add(newLine);
        repaint();
    } //GEN-LAST:event_stepButtonActionPerformed

    private void timerButtonActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_timerButtonActionPerformed
        if (timer.isRunning())
        {
            timer.stop();
            timerButton.setText("Start Timer");
        }
        else
        {
            timer.start();
            timerButton.setText("Stop Timer");
        }
    } //GEN-LAST:event_timerButtonActionPerformed
    // Skipped: ... Look & Feel
    // Variables declaration - do not modify //GEN-BEGIN:variables
    private DrawPanel drawPanel;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JButton stepButton;
    private javax.swing.JButton timerButton;
    // End of variables declaration //GEN-END:variables
}

```