My Solution on CS106A - Section 20 - Problem 4 - Drawing Lines

This is my Solution on CS106A - Section 20 - Problem 4 - Drawing Lines from the Computer Science Course CS106A of Prof. Mehran Sahami at the STANFORD University.

My Code:

DrawLines.java

```
/*
 * File: DrawLines.java
 * --------------------
 * Name: Axel Werner [mail@awerner.myhome-server.de]
 * Section Leader:
 *
 * CS106A - Section 20 - Problem 4 - Draw Lines
 *
 * Write a GraphicsProgram that allows the user to
 * draw lines on the canvas. Pressing the mouse
 * button sets the starting point for the line.
 * Dragging the mouse moves the other endpoint
 * around as the drag proceeds. Releasing the
 * mouse fixes the line in its current position
 * and gets ready to start a new line.
 *
 * For example, suppose that you press the mouse
 * button somewhere on the screen and then drag it
 * rightward an inch, holding the button down.
 * What you'd like to see is the following picture:
 *
 * {picture1}
 *
 * If you then move the mouse downward without
 * releasing the button, the displayed line will
 * track the mouse, so that you might see the
 * following picture:
 *
 * {picture2}
 *
 * Because the original point and the mouse position
 * appear to be joined by some elastic string, this
 * technique is called rubber-banding.
 * Although this program may seem quite powerful, it
 * is also simple to implement. The entire program
 * requires fewer than 20 lines of code.
```

```
 *
 * PERSONAL FEATURE BONUS:
 *
 * The temporary Line while in "rubber band mode" is
 * colored in RED to indicate the mode. When mouse
 * button is released the Line is drawn in standard
 * Color BLACK as supposed to.
 *
 */

import acm.graphics.*;
import acm.program.*;
import acm.util.*;

import java.applet.*;
import java.awt.*;
import java.awt.event.*;

public class DrawLines extends GraphicsProgram {


    public void run() {
        addMouseListeners();
    }

    public void mousePressed (MouseEvent e) {
        tmpLine.setStartPoint(    e.getX(), e.getY()    );
        tmpLine.setEndPoint(    e.getX(), e.getY()    );
        tmpLine.setColor( Color.RED );
        add(tmpLine);
    }

    public void mouseDragged(MouseEvent e) {
        tmpLine.setEndPoint( e.getX(), e.getY()    );
    }

    public void mouseReleased(MouseEvent e) {
        GLine newLine = new GLine(    tmpLine.getX(), tmpLine.getY(),
                                      e.getX(), e.getY() );
        add(newLine);
        remove(tmpLine);
    }


    /*
     * Define some Instance Variables
     */

    /**
     * Define a temporary Line to work with
```

```
     */
    GLine tmpLine = new GLine(0,0,0,0);

}
```

— *Axel Werner* *2012-04-04 00:10*

java, karel, stanford, university, cs106, computer, science, learning, programming

From:
https://awerner.myhome-server.de/ - **Axel Werner's OPEN SOURCE Knowledge Base**

Permanent link:
https://awerner.myhome-server.de/doku.php?id=it-artikel:java:my-solution-on-cs106a-section-20-problem-4-drawing-lines

Last update: **2022-08-31 12:30**