CSE 142 Section Handout #3

Challenge Sheet

You are not expected or required to solve these problems. These problems are designed for students who want a fun, extra challenge to test their skills on harder programming problems. Have fun!

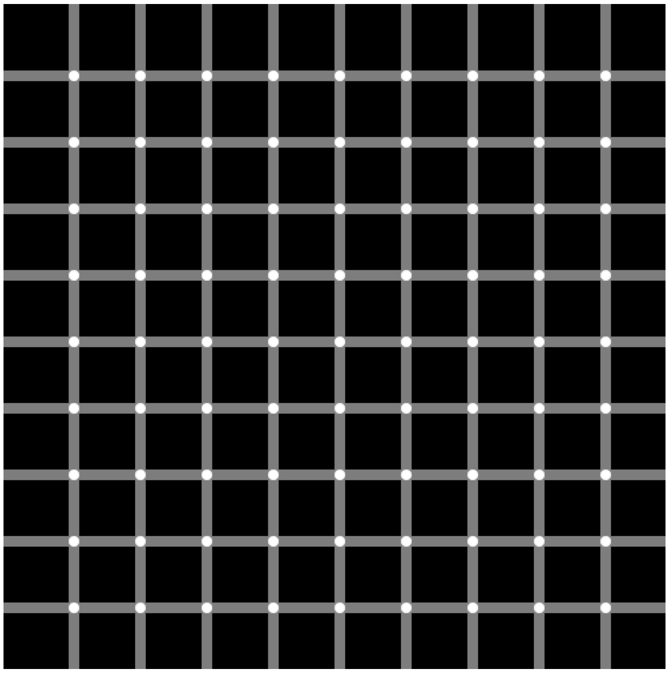
The Scintillating Grid Illusion is an optical illusion where white circles are placed on the intersections of perpendicular gray bars on a black background. With this construction, dark dots seem to appear and disappear randomly above the white circles as you look around at them. Note that the dots don’t do this if you only stare at a single dot but only if you look around at multiple. While this illusion works for non-square grids, we will only work with square grids in this problem.

Construct a method named drawIllusion that takes in three parameters: A graphics parameter to draw shapes with, an integer parameter denoting the size of the DrawingPanel, and an integer parameter denoting how many vertical and horizontal bars should be drawn on each side. Both the size of the white circles and the width of the light gray bars should be 1/60th of the length of the DrawingPanel. The gray bars should be evenly spaced between them (It is ok for the first bar to be a little misaligned in relation to the left side).

For example, the following piece of code should produce the graphic below

```
public static void main(String[] args) {
      int sideLength = 500;
      DrawingPanel p = new DrawingPanel(sideLength, sideLength);
      Graphics g = p.getGraphics();
      p.setBackground(Color.BLACK);

      drawIllusion(g, sideLength, 10);
}
```



CSE 142 Section Handout #3

Solutions (One Possible Solution Provided)

1.

```
import java.awt.*;

public class Dots {
    public static void main(String[] args) {
        int sideLength = 500;
        DrawingPanel p = new DrawingPanel(sideLength, sideLength);
        Graphics g = p.getGraphics();
        p.setBackground(Color.BLACK);

        drawIllusion(g, sideLength, 10);
    }

    public static void drawIllusion(Graphics g, int sideLength, int numberOfLines) {
        int size = sideLength / 60;

        g.setColor(Color.LIGHT_GRAY);
        for(int i = 1; i <= numberOfLines; i++) {
            g.fillRect(0, (i * sideLength / numberOfLines), sideLength, size);
        }

        for(int i = 1; i <= numberOfLines; i++) {
            g.fillRect((i * sideLength / numberOfLines), 0, size, sideLength);
        }

        g.setColor(Color.WHITE);
        for(int i = 1; i < numberOfLines; i++) {
            for (int j = 1; j < numberOfLines; j++) {
                g.fillOval((i * sideLength / numberOfLines), (j * sideLength /
                numberOfLines) ,size, size);
            }
        }
    }
}
```