My Solution on CS106A - Section 20 - Problem 3 - Random Circles

This is my Solution on CS106A - Section 20 - Problem 3 - Random Circles from the Computer Science Course CS106A of Prof. Mehran Sahami at the STANFORD University.

Question: On some runs of this program you might not see ten circles. Why?

My Answer: Since there is no special controlling on where to place new circles on the canvas it is possible that it puts a new larger circle on top of an older smaller circle, covering the older circle completly. So its possible to have LESS THEN 10 circles visible on the canvas.

My Code:

RandomCircles.java

```
/*
 * File: RandomCircles.java
 * ------------------
 * Name: Axel Werner [mail@awerner.myhome-server.de]
 * Section Leader:
 *
 * CS106A - Section 20 - Problem 3 - Random Circles
 *
 * Write a GraphicsProgram that draws a set of ten circles
 * with different sizes, positions, and colors.
 * Each circle should have a randomly chosen color,
 * a randomly chosen radius between 5 and 50 pixels, and a
 * randomly chosen position on the canvas, subject to the
 * condition that the entire circle must fit inside the
 * canvas without extending past the edge.
 *
 */

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

import java.awt.*;

/**
 * Program to display random colored and random size circles
 * on the canvas. Number of circles and valid radiuses are
 * configurable by static finals (class variables).
 *
 * @author Axel Werner [mail@awerner.myhome-server.de]
```

```
 *
 */
public class RandomCircles extends GraphicsProgram {

    private static final int MAX_CIRCLES = 10;

    private static final int MIN_RADIUS = 5;
    private static final int MAX_RADIUS  = 50;

/**
 * Main Program
 */
    public void run(){
        final int CANVAS_WIDTH = getWidth();
        final int CANVAS_HEIGHT = getHeight();

        for(int circ=MAX_CIRCLES; circ > 0 ; circ--){

            int r = getRand(MIN_RADIUS,MAX_RADIUS);

            /* Note:
             * Formular "CANVAS_WIDTH-2*r" makes sure that the
             * Object stays completly within the Canvas surface
             * and does not exceed edges.
             */
            int xPos = getRand(0,CANVAS_WIDTH-2*r);
            int yPos = getRand(0,CANVAS_HEIGHT-2*r);

            addCircle(r, xPos, yPos);
        }
    }

    /**
     * Method to put Circle GObjects of random Color on the Canvas.
     *
     * @param r a integer Radius value
     * @param xPos a integer X position on the Canvas
     * @param yPos a integer Y position on the Canvas
     */
    private void addCircle(int r, int xPos, int yPos) {
        GOval newCircle = new GOval(xPos, yPos, r*2, r*2);
        newCircle.setFilled(true);
        Color rndColor = getRandColor();
        newCircle.setColor( rndColor );
        newCircle.setFillColor( rndColor );
        add(newCircle);
    }

    /**
     * Method to return a random Color
```

```
     *
     * @return a random Color.
     */
    private Color getRandColor() {
        RandomGenerator rnd = RandomGenerator.getInstance();
        return rnd.nextColor();
    }

    /**
     * Method to return a random Number between
     * the given min + max Parameters.
     *
     * @param min minimal integer value to pick a random number from.
     * @param max maximum integer value to pick a random number from.
     * @return a random integer number
     */
    private int getRand(int min, int max) {
        RandomGenerator rnd = RandomGenerator.getInstance();
        return rnd.nextInt(min, max);
    }
}
```

— *Axel Werner* *2012-04-03 22:49*

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-3-random-circles

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