University of Puerto Rico
Department of Electrical and Computer Engineering

ICOM 4015 Laboratory: Advanced Programming

Laboratory 4: Random Numbers, Methods and Animation

Completed by:
ID:
Date:

1 Introduction

In this laboratory we will practice the use and design of methods working with graphics animation.

2 Random numbers generator

Java has the class Random that provides several alternatives to generate random numbers. In this lab we will only work with a few of them.

Random is defined in the "java.util" library package. So in order to use this class in our programs we can use

```
import java.util.Random;
```

One way to create a random number generator of class Random is

```
Random generator = new Random();
```

Java provides method `nextInt()` to generate random integer numbers and `nextDouble()` to generate random double numbers. The method `nextInt()` returns any integer number, but sometimes we need to bound the range. To do this, we can send an integer argument to the method. If for example we use

```
value = generator.nextInt(n);
```

`value` will receive an integer between 0 and n – 1.

Lab Practice

Design a program to generate random integer values within a range. The program will ask the user for the minimum and maximum possible values. Then the program will display a list of 20 random integers within the established range. *When done, paste your code below.*

//

//

3 Using our own designed methods to control graphics animation

Previously at the classroom we studied an example of a program where a square moved inside a panel. In this section we will work on the design of a program that will control the movement of a rectangle so that it bounces within the limits of the panel. We provide part of the code and you,

using your knowledge on methods, complete the program. You can modify the provided code to correct any misbehavior you find. When done, *replace the code below with yours.*

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
import javax.swing.*;
import javax.swing.JFrame;
import java.awt.*;
import java.util.Random;

public class BouncingSquare2
{
  Random generator = new Random();
  final int PANEL_WIDTH = 400,
            PANEL_HEIGHT = 400,
            MAX_RECTANGLE_WIDTH = 50,
            MIN_RECTANGLE_WIDTH = 10,
            MAX_RECTANGLE_HEIGHT = 50,
            MIN_RECTANGLE_HEIGHT = 10;

  /**
   * rectangleWidth will receive a value within [MIN_RECTANGLE_WIDTH, MAX_RECTANGLE_WIDTH]
   * rectangleHeight will receive a value within [MIN_RECTANGLE_HEIGHT, MAX_RECTANGLE_HEIGHT]
   */
  int rectangleWidth = 20;   //How to change this???????????
  int rectangleHeight = 20;   //How to change this???????????

  /**
   * x and y will receive random values so the initial position of the object is within
   * the container dimensions. The object's height and width are taken into account.
   */
  int x = 5;   //How to change this???????????
  int y = 15;   //How to change this???????????

  class DrawPanel extends JPanel
  {
    public void paintComponent(Graphics g)
    {
      //Fill all DrawPanel with Color.White
      g.setColor(Color.WHITE);
      g.fillRect(0,0,this.getWidth(),this.getHeight());
      //Draw a Blue Rectangle
      g.setColor(Color.BLUE);
      g.fillRect(x,y,rectangleHeight,rectangleWidth);
    }
  }

  /**
   * To detect if the object hits the leftBorder of the container
   *
   * @return  true if the object hits the left border of the container
   *            and false otherwise.
   */

  public boolean hitLeftBorder()
  {
       return false;   //How to change this???????????
  }


  /**
   * To detect if the object hits the topBorder of the container
   *
   * @return  true if the object hits the top border of the container
   *            and false otherwise.
   */
```

```
public boolean hitTopBorder()
{
     return false;   //How to change this???????????
}


/**
 * To detect if the object hits the right border of the container
 *
 * @param   border is the width of the container
 * @return  true if the object hits the right border and false otherwise
 */

public boolean hitRightBorder(int border)
{
     return false;   //How to change this???????????
}


/**
 * To detect if the object hits the right border of the container
 *
 * @param border  is the height of the container
 * @return  True if the object hits the bottom border and false otherwise
 */

public boolean hitBottomBorder(int border)
{
     return false;   //How to change this???????????
}

public void animate()
{
  JFrame frame = new JFrame();
  frame.setSize(PANEL_WIDTH,PANEL_HEIGHT);
  frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  DrawPanel draw = new DrawPanel();
  frame.getContentPane().add(draw);
  frame.setVisible(true);

  /**
   * xIncrement will have a random number within [-3, 3].
   * yIncrement will have a random number within [-3, 3].
   */
  int xIncrement = 2;   //How to change this???????????
  int yIncrement = 3;   //How to change this???????????

  while (true)
  {
    System.out.println(x+"__"+y);

    y = y + yIncrement;
    x = x + xIncrement;

    if (hitRightBorder(PANEL_WIDTH) || hitLeftBorder())
    {
      xIncrement = xIncrement * -1;
    }
    if (hitBottomBorder(PANEL_HEIGHT) || hitTopBorder())
    {
      yIncrement = yIncrement * -1;
    }

    //tells the panel to redraw itself
    draw.repaint();
    try
    {
      //20 milliseconds delay
```

```
        Thread.sleep(20);
      }
      catch(Exception e)
      {
      }
    }
  }
  public static void main(String[] args)
  {

    BouncingSquare2 animation = new BouncingSquare2();
    animation.animate();
  }

}
```
//