CS 1063 Lab 7: Random Walks

The `RandomWalk` Class

Objectives

- Call methods with parameters and return values.
- Use the Graphics, Random, Scanner, and String classes.
- Use while statements

Hand-in Requirements

All projects and laboratories will be submitted electronically through Blackboard. Zip up your entire lab directory to submit as the source. (Right click on the lab folder and follow **Send To > Compressed (zipped) Folder**.) The lab folder should include the following:

- `RandomWalk.java`
- `DrawingPanel.java`
- `RandomWalkOutput.txt`

Task

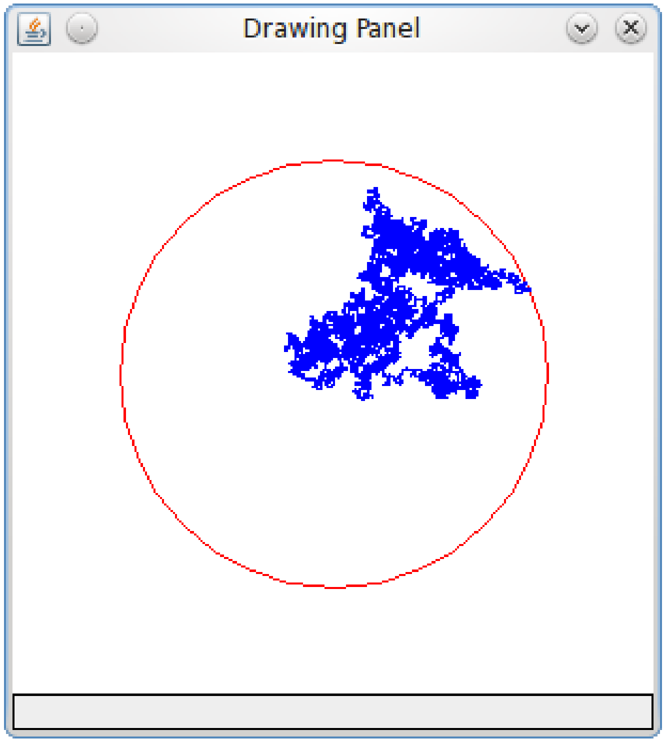
Simulate a random walk using a Graphics object. Ask the user for information.

Similar to previous programs, your program should start by printing

```
Lab 7 written by YOURNAME
```

Details

A random walk begins at a point and repeatedly takes a step in a randomly chosen direction. In our version, the random walk will start at the center of a circle and continue until it goes outside the circle. Each step will be randomly chosen from one pixel up, one pixel down, one pixel left, and one pixel right (this kind of random walk is called the "drunkard's walk"). Call the `nextInt` method on a Random object to generate an integer between 0 and 3 and map each value to one of the steps. The random walk will be drawn on a Graphics object. Here is an example for a circle with a radius of 100.



The user should be asked for the radius of the circle and the color of the circle (from at least two choices for colors). The program should print the values that the user enters. The program should draw the circle on a DrawingPanel that is big enough to hold the circle, leaving some space (but not a lot of space) around the circle. Call the `drawOval` on your Graphics object to draw the circle. Each step of the random walk should drawn by calling the `drawLine` method on your Graphics object. To animate the walk, call the `sleep` method on your DrawingPanel to pause a millisecond between each step. After the random walk is finished, the number of steps of the random walk should be printed.

Loops

This lab **must** have a while loop to ensure that the radius of the circle is between 50 and 400.

This lab **must** have a while loop to ensure that the user selects one of the color choices.

This lab **must** have a while loop to continue drawing the random walk until the walk leaves the circle.

Methods

Write and use the following methods to support your random walk.

Write a `boolean` method to determine whether the current position of the random walk is outside the circle. What values do you need to make this decision? These values need to be parameters of the method. How will you calculate the distance from the center of the circle? You did something similar in Lab 5.

Write a `boolean` method to determine whether a `String` answer entered by the user matches a `String` choice for the question.

```
public static boolean matchesChoice(String answer, String choice)
```

For example, if "blue" is one of the choices for a color, then the user should be able to enter "blue", "Blue", "BLUE", "b", or "B" to answer with this color. However, "bluish", "navyBlue", or "blueGreen" should not match. That is,

`matchesChoice("blue", "blue")` should return `true`.
`matchesChoice("Blue", "blue")` should return `true`.
`matchesChoice("B", "blue")` should return `true`.
`matchesChoice("bluish", "blue")` should return `false`.
`matchesChoice("navyBlue", "blue")` should return `false`.
`matchesChoice("blueGreen", "blue")` should return `false`.

Rubric

Your program should compile without any errors. A program with more than one or two compile errors will likely get a zero for the whole assignment.

The following criteria will also be used to determine the grade for this assignment:

- [2 points] If your submission includes the following:
  - The main method prints "Lab 7 written by [...]".
  - Your submission was a Zip file named `lab7.zip` containing a folder named `lab7`, which contains the other files.
  - If your Java program was in a file named `RandomWalk.java`.
  - If the output of your Java program was in a file named `RandomWalkOutput.java`.
  - If your program contains a comment that describes what the program does and contains a comment describing each method.
  - If your program is indented properly.
- [3 Points] For a while loop ensuring that the user enters a number between 50 and 400 for the radius of the circle.
- [3 Points] For a correct implementation of the method that determines if the random walk is outside the circle.
- [3 Points] For a correct implementation of the `matchesChoice` method.
- [2 Points] For selecting the right circle/line color based on the user's input.
- [2 Points] For printing what the user entered and printing the number of steps of the random walk.
- [5 Points] For a correct random walk simulation.

Revision Date Tue Apr 14 2015 05:08:53 GMT-0500 (Central Daylight Time)