CS 1331 Homework 12

Due Sunday, April 21st, 2013 8:00pm

Introduction

This assignment combines GUI drawing with LinkedLists.

You will be creating a GUI application where a user can click on the screen to place red, green, or blue dots. They will be able to remove a dot by clicking on it, and they will be able to remove all of the dots of a certain color by pressing a specific key. Storing and accessing the dots will be done through a linked list class that you write.

12.1 Dot.java

This class represents the Dot that will be drawn on the screen. It is relatively simple; it's essentially a wrapper for a Point and an associated color.

We have provided you with the file for this class which contains all of the instance variables and method headers for everything required. You are not allowed to add in any new instance variables or change the headers in any way.

12.2 DotLinkedList.java

This class is what we will use to store the dots in our program. You are NOT allowed to use the built-in java LinkedList class. The whole point of having you write your own is to understand how the linked list data structure works.

This linked list will be singly-linked with only a head reference.

We have provided you with the file for this class which contains all of the instance variables and method headers for everything required. You are not allowed to add in any new instance variables or change the headers in any way.

12.3 DotPanel.java

This class will be what actually displays the dots, and allows the user to interact with the program.

The required interactions are:

- Clicking the mouse on the screen creates a new dot
- Clicking an already created dot removes that dot
- Pressing "r", "g", or "b" changes the current color to red, green, or blue respectively; all dots created after changing the color will be that color until the color is changed again

- Pressing “1”, “2”, or “3” removes all of the dots of a specific color; 1 removes the red dots, 2 removes the green dots, and 3 removes the blue dots.
- Pressing “s” should display the String representation of the list on the left hand side of the page – if the list is already displayed, pressing “s” should remove that from view

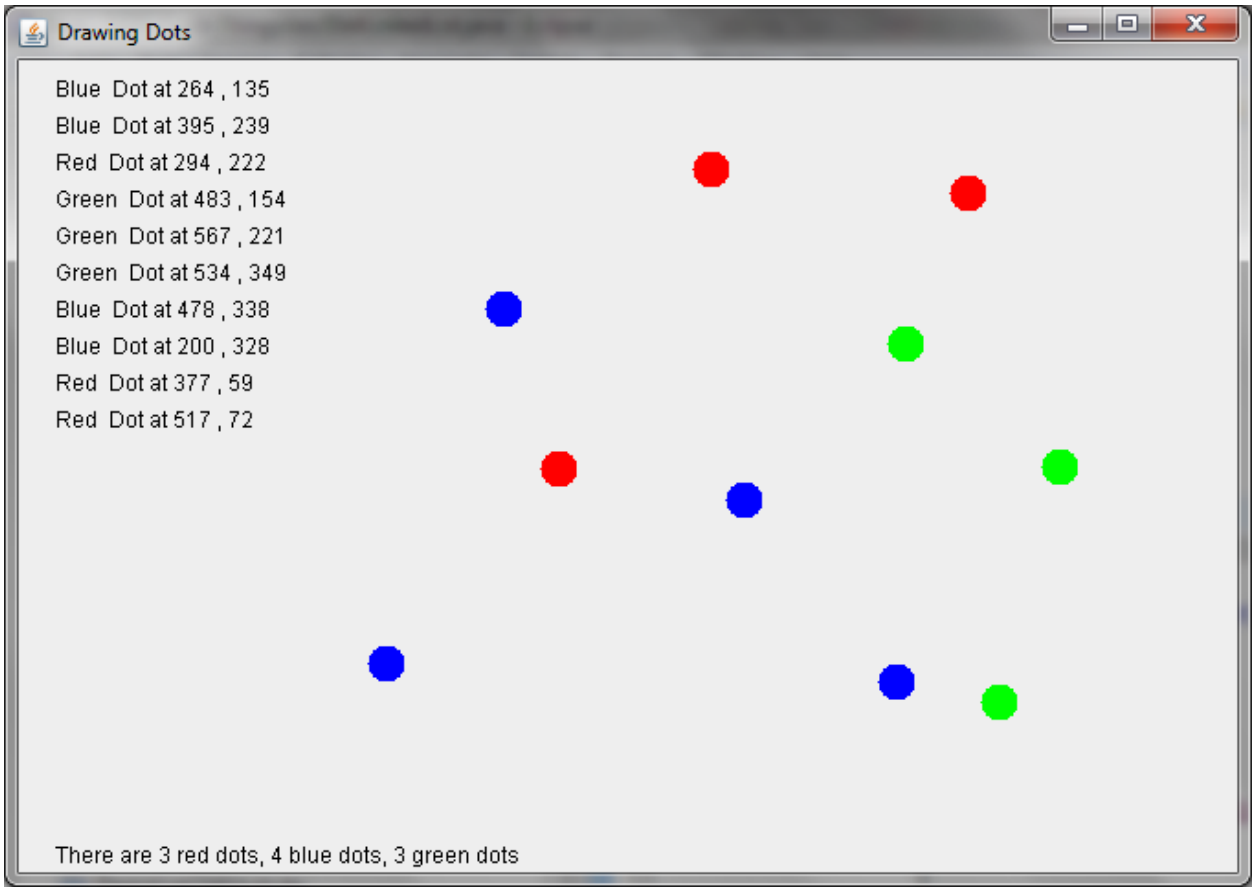
Additionally, you should display a count of each of the different colored dots on the bottom of the screen.

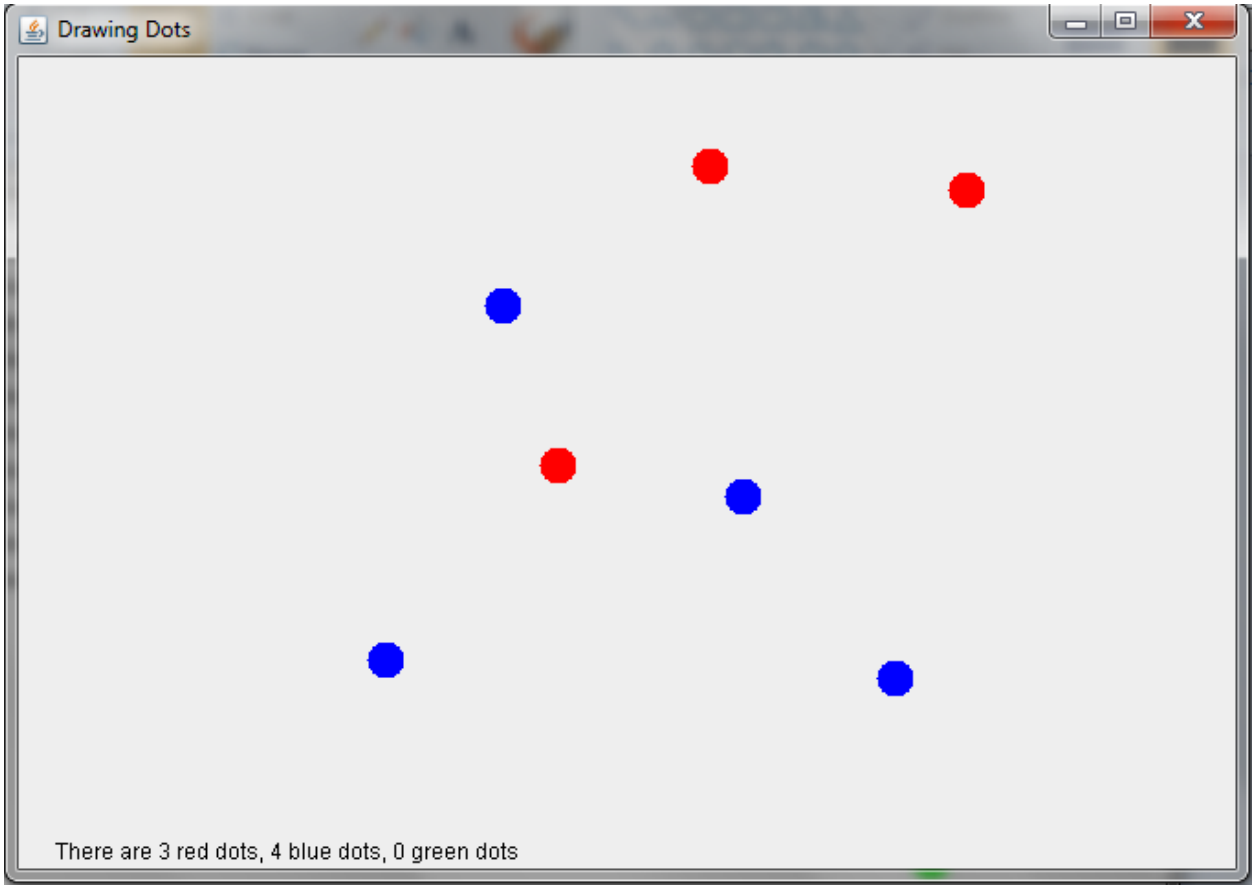
Hint: For an example of KeyListeners, take a look back at the Pacman Homework

12.4 DotDriver.java

This file will be very minimal. It should just contain a main method that creates an instance of our DotPanel and displays it to the user.

This is an example of what your program should look like:





Turn-in Procedure

Turn in the following files to T-Square. When you are ready, make sure that you have actually **submitted** your files, and not just saved them as a draft.

- Dot.java
- DotLinkedList.java
- DotPanel.java
- DotDriver.java

Note** Always submit .java files - never submit your .class files. Once you have submitted and received the email from T-Square, you should download your files into a fresh folder. Compile, run, and test those exact files that you turned in. It is pretty much foolproof if you use this technique. Anything less than this is a gamble. See "safe submission" info below. File issues, non-compiling files, non-source code files, etc are all problems and will cause a 0 for the HW. Also, make sure that your files are in on time; the real deadline is 8 pm. While you have until 2 am to get it turned in, we will not accept homework past 2 am for any reason. Don't wait until the last minute!

Verify the Success of Your HW Turn-in

Practice "safe submission"! Verify that your HW files were truly submitted correctly, the upload was successful, and that the files compile and run. It is solely your responsibility to turn in your homework and practice this safe submission safeguard.

1. After uploading the files to T-Square you should receive an email from T-Square listing the names of the files that were uploaded and received. If you do not get the confirmation email almost immediately, something is wrong with your HW submission and/or your email. Even receiving the email does not guarantee that you turned in exactly what you intended.
2. Read that email. Look at those filenames. We do not grade .class files. We require source code .java files.
3. After submitting the files to T-Square, return to the Assignment menu option and this homework. It should show the submitted files and the fact that you submitted.
4. Download copies of your submitted files from the T-Square Assignment page placing them in a new folder.
5. Recompile and test those exact files.
6. This helps guard against a few things.
    a. It helps insure that you turn in the correct files.
    b. It helps you realize if you omit a file or files. (If you do discover that you omitted a file, submit all of your files again, not just the missing one.)
    c. Reading the email and looking at the files you download helps prevent the turn-in of bytecode (.class) file for which you will receive no credit.
    d. Helps find last minute causes of files not compiling and/or running.