CO2220 Laboratory 6
Graphical Object-Oriented and Internet Programming in Java

Corresponds to Vol 1, Chapter 12 (Events and Graphics) of Study Guide

Learning Objectives:

To understand and apply the following concepts:
- how graphics classes can be used
- designing and developing event based GUI program

Task 1

You are supposed to draw a green color filled circle in the middle of a white background. You can name this class ***Drawing***.

You are required to follow the instructions stated:

- A **GPanel** object is instantiated (name your class "GPanel.java"). GPanel subclasses JPanel and overrides paintComponent.

- A **JFrame** object is created (within your class "Drawing.java"). A JFrame represents an actual window on the screen.

- The JVM will use the JFrame object that you have created to set up and draw the window.

- Implement the code to draw on the GPanel object, together with the shapes as shown in **Figure 1**. Note: you should write this code within the paintComponent() method.

- setDefaultCloseOperation ensures that the program terminates when the user closes the window.

- The panel is added to the frame. This is an unusual line of code, because it contains two method calls. The compiler works from left to right, so this line is equivalent to (frame.getContentPane()).add(myPanel);

  The call getContentPane() asks frame for a pane. The pane is a blank space inside the frame. The code asks the ContentPane object to add our panel to the frame's pane.

- The window's initial dimensions are set to 300 x 300 pixels.

- The window is displayed by sending a **setVisible** message to the frame and then main terminates. Normally the program would terminate at this point because main leaves the stack and nothing replaces it. However the Java graphics system is running a separate graphics thread. Your GPanel object is referenced by a variable in this thread and paintComponent is called on your object whenever the window needs refreshing or redrawing.

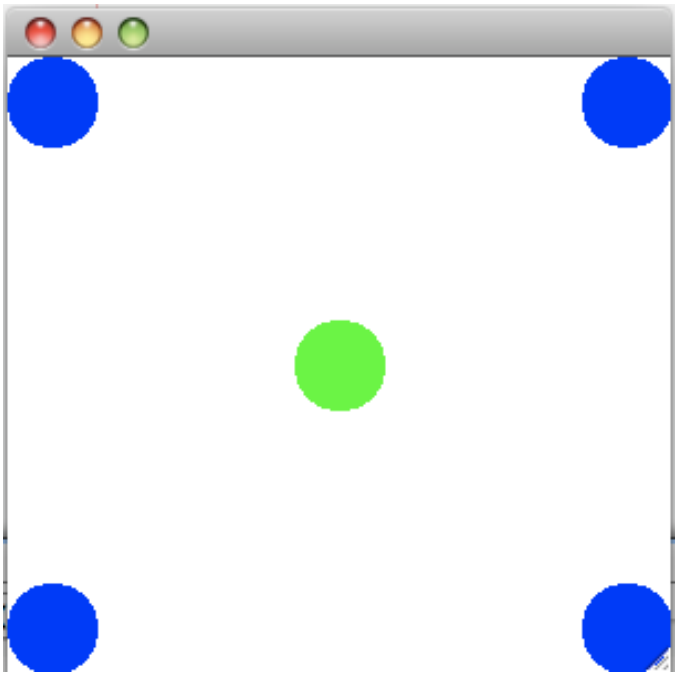

Figure 1 Fixed position of the circles

Task 2

Write an animation class. The four blue circles are supposed to move from where they are to the diagonally opposite direction. The diagram below shows the starting and ending position of the blue circles.

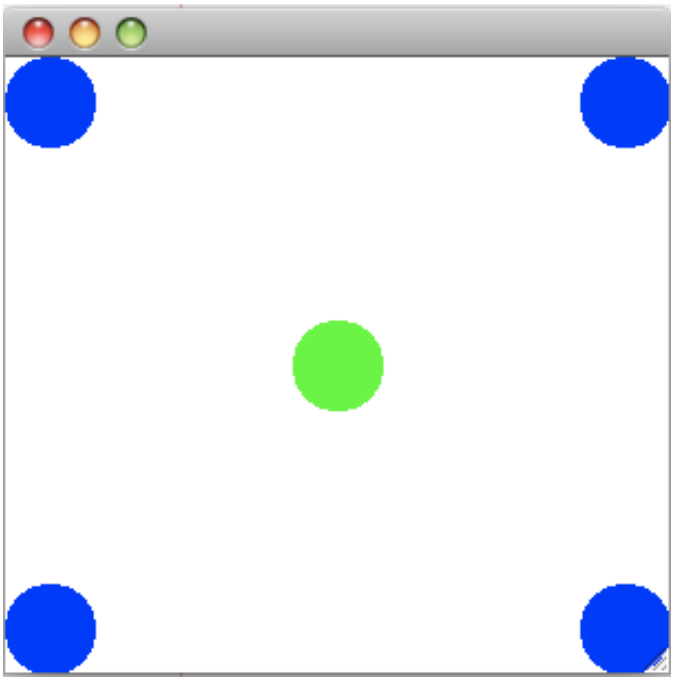

Figure 2 - Starting positions of the circles

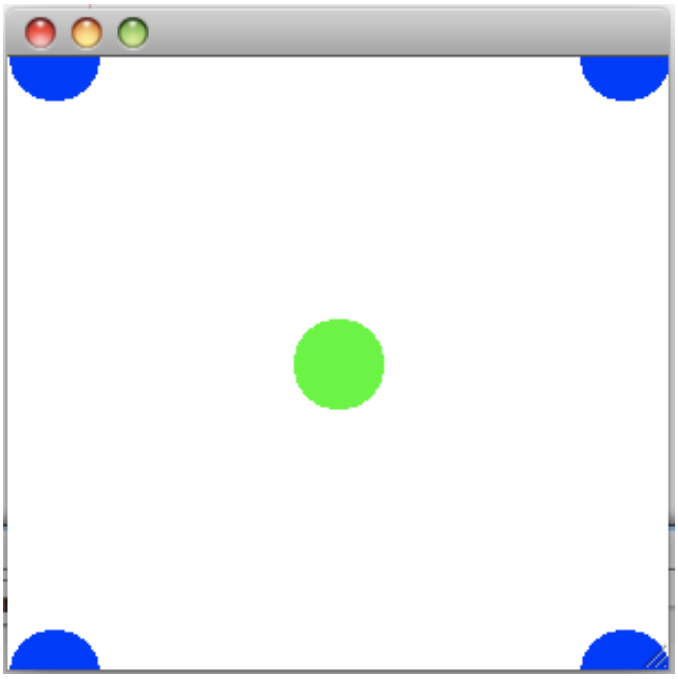

Figure 3 - End positions of the circles

Task 3

Write a **MoveMessageDemo** class to demonstrate that you can move the string to whichever position you want on the frame.

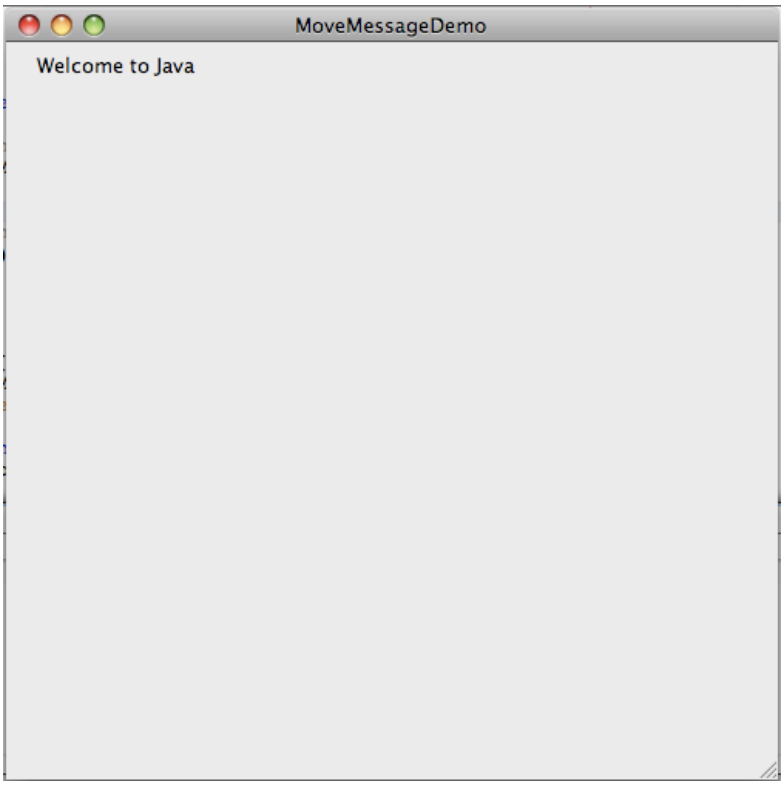

Figure 4 - MoveMessageDemo

Task 4

Write a **MouseMotionEventDemo** class to demonstrate that if you draw on the panel. A square will be registered on the points taken along the path. You are supposed to use a **MouseMotionListener** in your program.

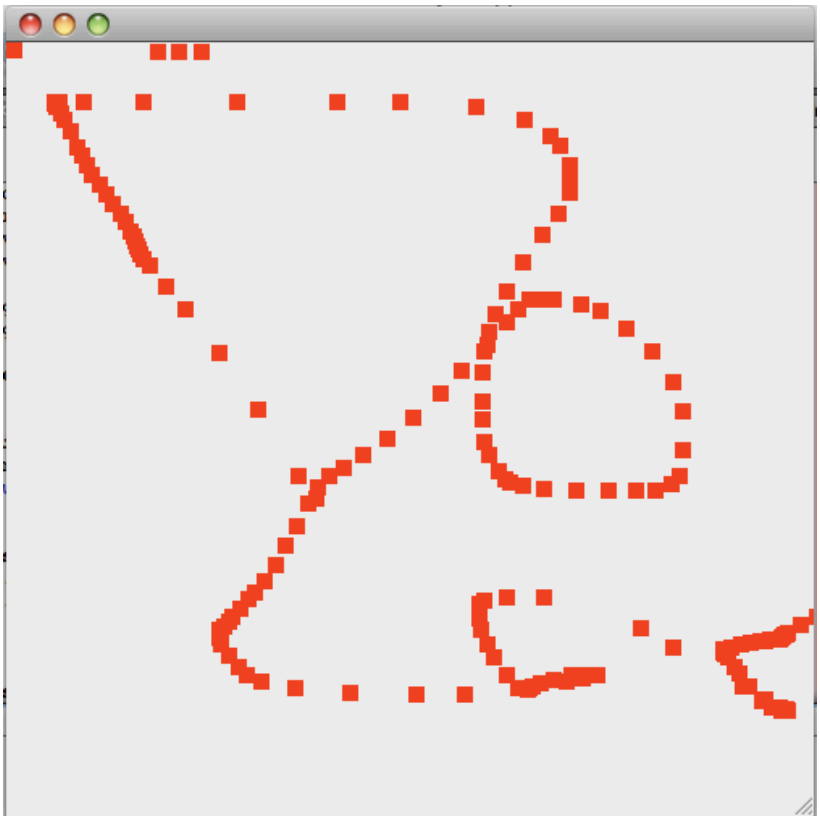

Figure 5 - MouseMotionEventDemo