Metropolitan State University
ICS 372 Object-Oriented Design and Implementation
Spring 2021

Assignment 2 Graphical User Interfaces

Due: 11:59 PM on February 5, 2021
Total points: 50

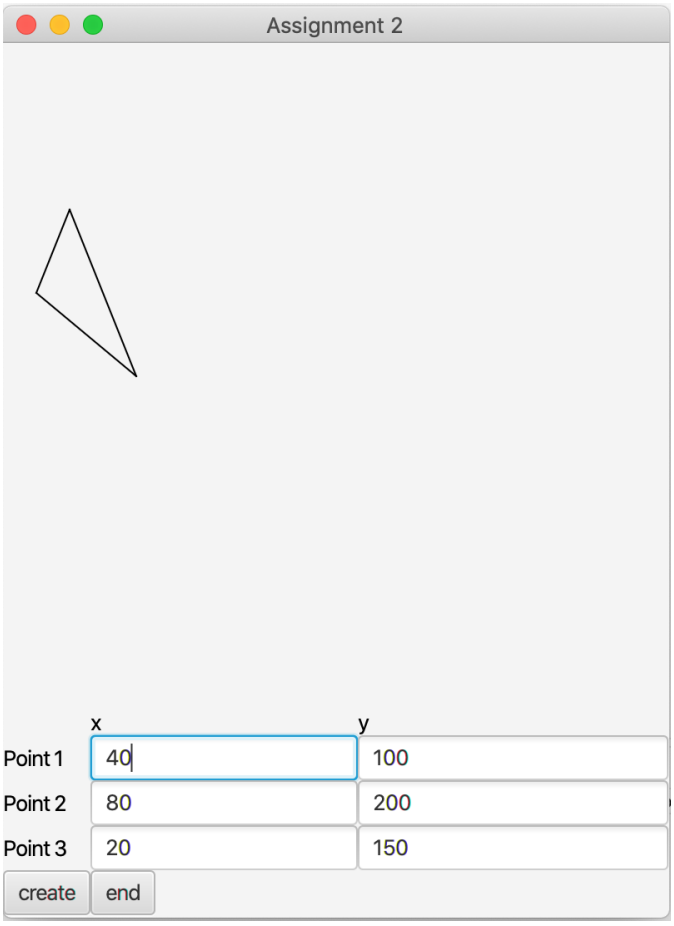
Objective

To learn GUI programming using JavaFX.

Problem Description

You are asked to provide GUI support for creating `Triangle` (from Assignment 1) objects and rendering them on the screen. The interface should look as below. (There can be some variations, but the look and feel should essentially be the same.)

The area for drawing the triangles is 400 pixels wide and 400 pixels high.



The user inputs the x and y coordinates of three points. You can assume that the user only enters integer values. As in the above interface, it should be clear what each entry (TextField) is: (which point and which coordinate it is. So, label the rows and columns properly.) Don't worry about validating whether the input is negative or greater than 399.

After entering the six coordinates, the user clicks the create button. The program draws the triangle based on the specified coordinates.

The user may draw any number of triangles.

Clicking on the close button in the title bar or the end button ends the program.

The title bar should display "Assignment 2" as shown in the picture.

The layout should be structured as above, although there can be slight variations (for example, the sizes of buttons). Your program should not have drastic deviations such as rendering the triangles below the buttons or putting the buttons in a separate window.

All components (drawing area, buttons, etc.) must be completely visible when the program starts.

The program must be written using JavaFX.

Some Ideas on How to Approach the Problem

Refer to the JavaFX examples posted on D2L. Project5.java should be useful as a starting point.

Create a Java project with the proper name (last name followed by 372 and then Assignment2).

1. Copy the `Point` and `Triangle` classes from Assignment 1 to this project.
2. Create a class that extends `Application`. (Unless otherwise specified, rest of this section refers to this class that extends `Application`.)
3. Declare two `Button` fields, six `TextField` fields, and a `Canvas` field (400 by 400). for clicking commands, entering points, and drawing, respectively.
4. In `the start()` method, create a container such as `GridPane`. Add the `Canvas`, `TextField`, and other objects into the container in the correct order. The `GridPane` class has methods like

   ```
   add(node, columnNumber, rowNumber)
   add(node, columnNumber, rowNumber, columnSpan, rowSpan)
   ```

5. Set this class as listener for the two buttons.
6. Process the button clicks appropriately.

To draw on the `Canvas` object, get its `GraphicsContext` and use the `strokeLine()` method

(https://docs.oracle.com/javase/8/javafx/api/javafx/scene/canvas/GraphicsContext.html#strokeLine-double-double-double-double-) to draw the three lines of a triangle.

Grading

Your grade in this assignment is based on the following:

- Your submission meets specifications as described above.
- The program is robust with no runtime errors or problems.
- You follow the good programming style as shown in the D2L document CodingStandards.pdf located in the Assignments folder.
- Follow the submission instructions.

Specific Grading Criteria.

1. GUI looks as shown in this handout. 10 points
2. The triangles are drawn correctly. 12 points
3. The program works correctly when the end and window close buttons are clicked. 10 points
4. The GUI class is properly structured. 8 points
5. The program follows all coding conventions. 10 points

Submission Instructions

Follow the steps given below to upload your code to D2L:

- Create a java project and call it `<yourLastName>Assignment2` (e.g., mine will be called `DathanAssignment2`). If you wish, you can add a qualifier to add the course name to have the form `<yourLastName>372Assignment2`
- In the dialog for a module name, choose not to create a module. If you submit a project with a module and I am unable to execute it, you will lose 10% of the grade.
- Create the `.java` files to implement the classes as described above.
- Archive the project into **one zip** file using Eclipse using the following steps:
  - In Eclipse Project Explorer, right click on the **project folder** of the project and click on `Export`.
  - Choose `General` then `Archive File` and click `Next`.
  - Use the `Browse` key to choose a folder to store the archive file on your hard drive and give the file the same name as your project (e.g., `DathanAssginment2.zip`), then click `Save`, then click `Finish`.
  - Upload the **.zip** file you created to the D2L folder called `Assignment 2`.
  - The most recent submission will be graded, unless you indicate so in the submission page.