Lab 6

[valid 2020-2021]

GeometryDrawing
Create an application with graphical user interface for creating images (layouts) containing standard or custom geometric figures: diamonds, trapezes, regular polygons, snow flakes, etc.
You may use either Swing or JavaFX.

The main specifications of the application are:

Compulsory (1p)

Create the following components:

- The *main frame* of the application.
- A *configuration panel* for introducing parameters regarding the shapes that will be drawn: the size, the number of sides, the stroke, etc.
  The panel must be placed at the *top* part of the frame. The panel must contain at least one label and one input component for specifying the size of the component.
- A *canvas (drawing panel)* for drawing various types of shapes. You must implement at least one shape type at your own choice. This panel must be placed in the *center* part of the frame.
  When the users execute *mouse pressed* operation, a shape must be drawn at the mouse location. You must use the properties defined in the configuration panel (at least one) and generate random values for others (color, etc.).
- A *control panel* for managing the image being created. This panel will contains the buttons: *Load*, *Save*, *Reset*, *Exit* and it will be placed at the *bottom* part of the frame.
- Use a *file chooser* in order to specify the file where the image will be saved (or load).

Optional (2p)

- Implement a *retained mode* drawing and add support for deleting shapes.

- Add support for drawing multiple types of components. Consider creating a new panel, containing a *list* of available shapes.
  The configuration panel must adapt according to the type of the selected shape.
  Implement at least two types of shapes.
- Implement *free drawing* and a simple *shape recognition* algorithm, capable of recognizing at least lines and circles.

Bonus (2p)

- Create a simple grammar in order to specify commands for drawing geometric shapes, for example `fill circle name,x,y,radius,color`.
  The commands will be specified in a text area component. **Important**: parsing the strings using regular expressions or other "custom" methods is not accepted.
- Use ANTLR(or a similar library) to generate a parser for your grammar, in order to evaluate the syntax and the semantics of your commands.
- Implement various commands at your own choice, for example *draw*, *fill*, *delete*, *etc*.
  An additional bonus may be given for *looping commands*, like *for*, in order to perform a drawing several times.

Note: This lab could be extended to a project, by adding support for multiple 2D, 3D shapes, drawing important lines (perpendiculars, medians, etc), graphs of functions, geometric animations, etc. (similar to GeoGebra).
Or, parsing geometry problems written in natural language using a NLP library (Apache OpenNLP, Stanford CoreNLP, etc).
Or, you may consider developing a tool for creating TikZ images.

Resources

- Slides
- Creating a GUI With JFC/Swing
- The Java Tutorials: 2D Graphics
- Getting Started with JavaFX
- JavaFX Scene Builder

Objectives

- Get familiar with the basic elements of design involved in creating a GUI.
- Understand the concepts of *component*, *container*, *layout manager*.

- Get acquainted with various libraries for creating a GUI application, such as AWT, Swing, SWT, Java FX.
- Write *event listeners* to handle events.
- Understand how Swing components are *painted*.
- Create custom components using Java2D that coexist with standard Swing components.
- Understand the concept of *graphic context.*
- Get familiar with Java2D basic *geometric* primitives, use *colours, fonts, images*
- Get acquainted with JavaFX technology and understand the differences between Swing and JavaFX.