GUI Assignment

2022
Semester 1

2012940 Adarsh Mungra
2012397 Jeevesh Rishi Dindyal
2012925 Akshit Sarju Tooshalduth
2012274 Fadil Khodadin
2012345 Farhaan Mohungur

Table of Contents

2012940 Adarsh Mungra
2012397 Jeevesh Rishi Dindyal
2012925 Akshit Sarju Tooshalduth
2012274 Fadil Khodadin
2012345 Farhaan Mohungur

Introduction

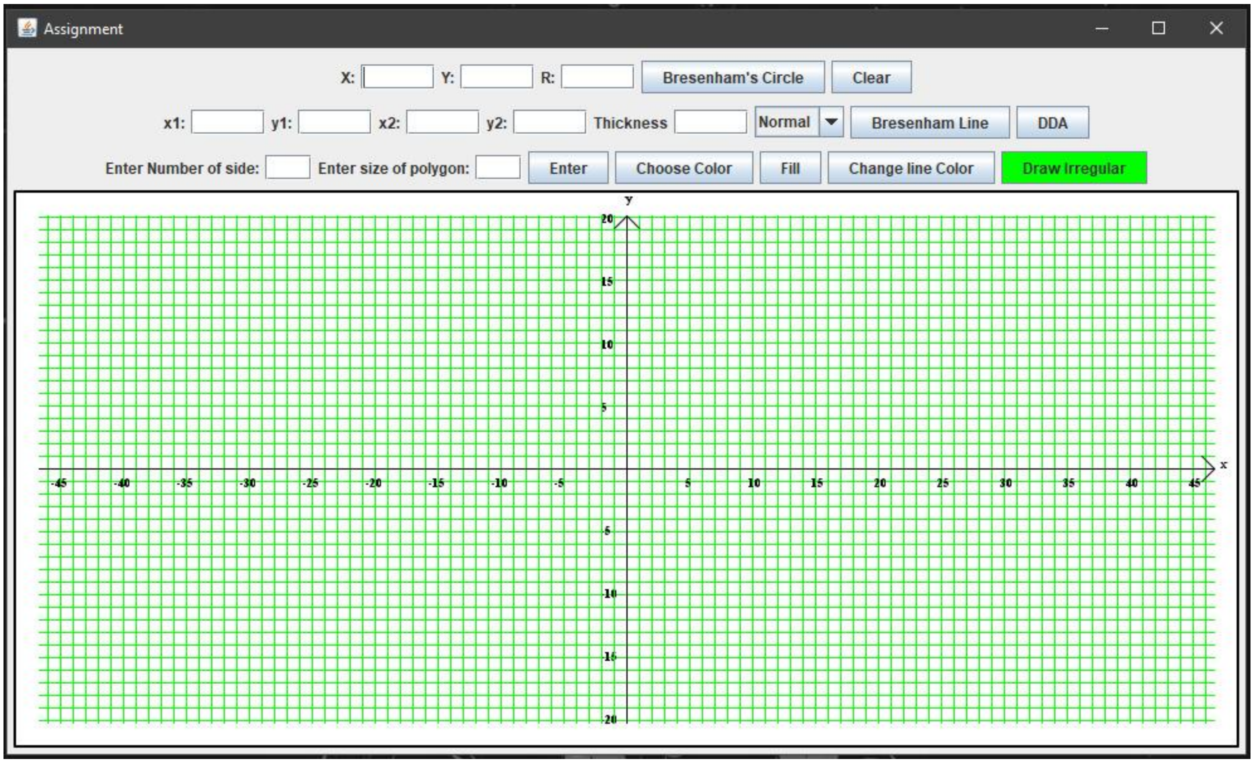
A program needs to have an exemplary user interface so that potential users and technical persons can interact, understand and manipulate the program. This report describes the Graphical User Interface and explains all the functionalities incorporated in the system. The functionalities include:

- Drawing a line using the DDA line algorithm and Bresenham Line algorithm.
- Draw a regular or irregular polygon.
- Fill drawn shapes with colour.
- Draw a circle using the Bresenham Mid-Point Circle Algorithm.

Description of User Interface Design

Main interface

Upon running the program, this interface is presented to the user. The user can perform multiple tasks such as drawing polygons, transforming, filling colour in shapes and other integrated functionalities.



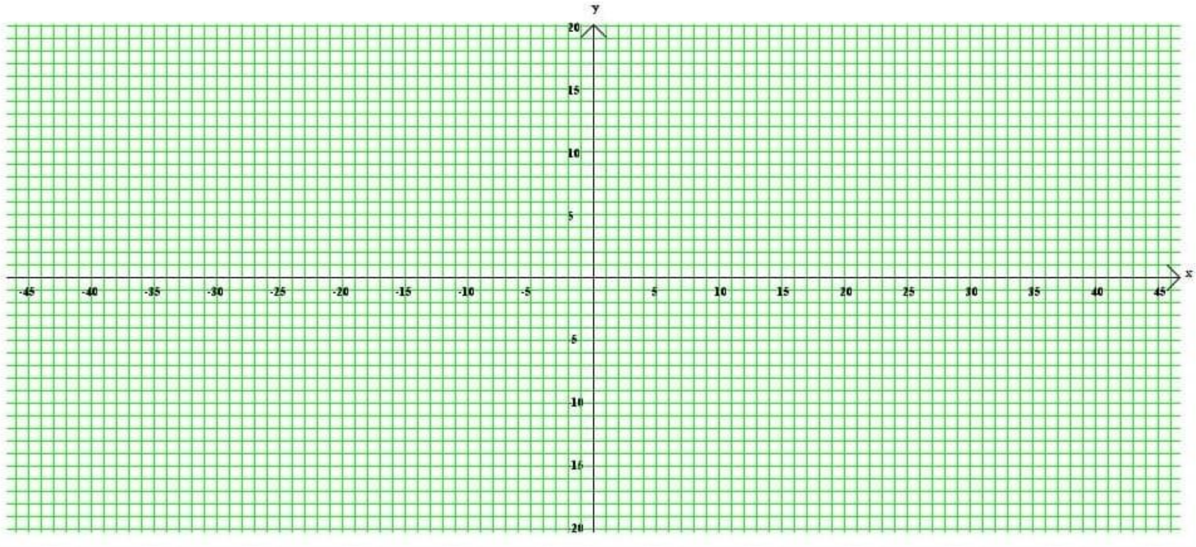
JFrame and ContentPane

A JFrame was used as a container, providing a base for the panels. A panel was defined as the content pane to accommodate all the components. For example, the three tool panels provided by polyswing, bresenham(circle) and bresenhampanel and the drawingpanel are an instance of the scale_coordinate class.

Drawing Panel

Each class uses this panel to perform their respective operations. The panel allows a minimum adequate size to plot from 0 to 45 on the x-axis and from 0 to 20 on the y-axis. If the user is required to draw beyond those limits, The user can increase the size, and the scale coordinate class will generate more drawing area. paintComponent drew the axes dark colour to stand out from the background. paintComponent drew the scale five units apart to find a balance between the range of values input by the user and the screen width.

The drawing panel has to override the paintComponent for each scale rendering. It provides responsiveness to the enlargement of the frame.

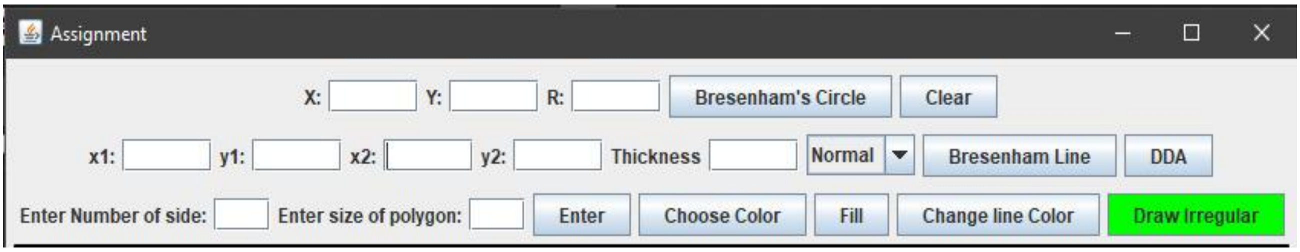


Tool Panel

- The tool panel is a combination of three different panels.
- The first line represents the Bresenham Mid-Point Circle Algorithm used for drawing circles.
- The second line represents the DDA line algorithm and Bresenham Line

algorithm used for drawing lines

- The third panel is used to draw regular and irregular polygons and fill colours in shapes.



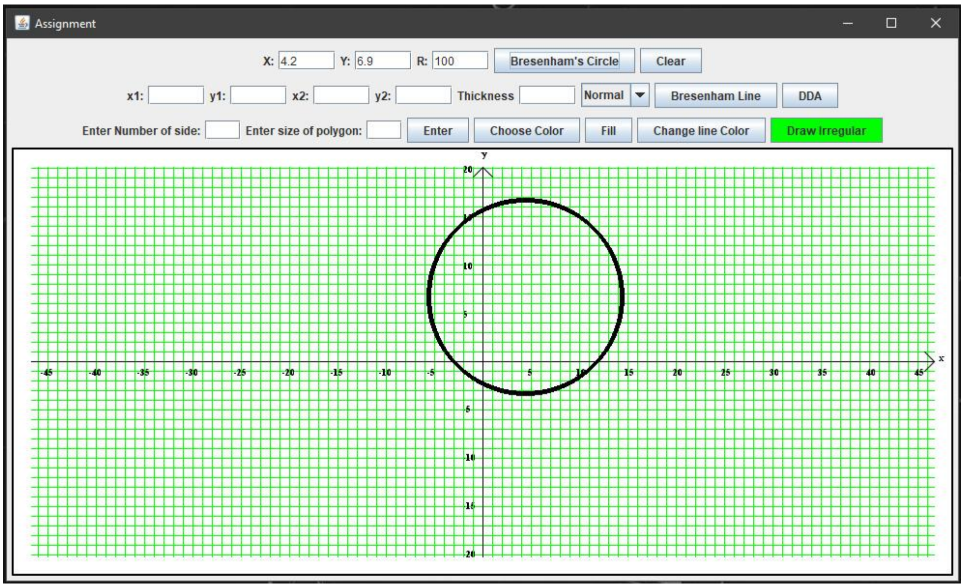
The primary colour of the panel is grey. We decided to make it the top menu as it is more convenient for users to select options on the top. It thus promotes usability and adaptability. The user can insert values for x-coordinates, y-coordinates, radius, line thickness, size and number of sides of polygons in the boxes.

Brasenham's Mid-point Circle algorithm

Steps to draw a circle using the program on the first panel of the tool panel:

- Input the value of the x-coordinate in the X box.
- Input the value of the y-coordinate in the Y box.
- Input the radius of the circle
- Click on Bresenham's Circle.

In the example below, the Bresenham Circle algorithm drew a circle with centre x-coordinates 4.2 and y-coordinates 6.9 with 100 pixels.



X: 4.2 Y: 6.9 R: 100

Bresenham circle algorithm was implemented by extending a JPanel to get the tool panel on the first line.

After each entry, fill the bresenham class calculate the circle path and draw an oval filled in black to simulate dots.

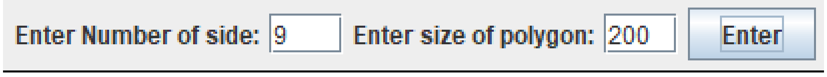
The dots are drawn on the drawingpanel passed to the constructor of the Bresenham class.

Drawing of a regular or irregular polygon

Regular polygon

There are many ways to draw a regular polygon: measuring one side and approximating the centre using the one side size of the polygon. The easiest one is to draw the regular polygon in a circle. We chose the easiest way to implement where we have an error of +-1 pixel not noticeable with eyes.

The number of sides is used to define the angle between each polygon point. Coordinate of the point is deduced using the angle. The coordinate is then added to the polygon object.



Enter Number of side: 9 Enter size of polygon: 200

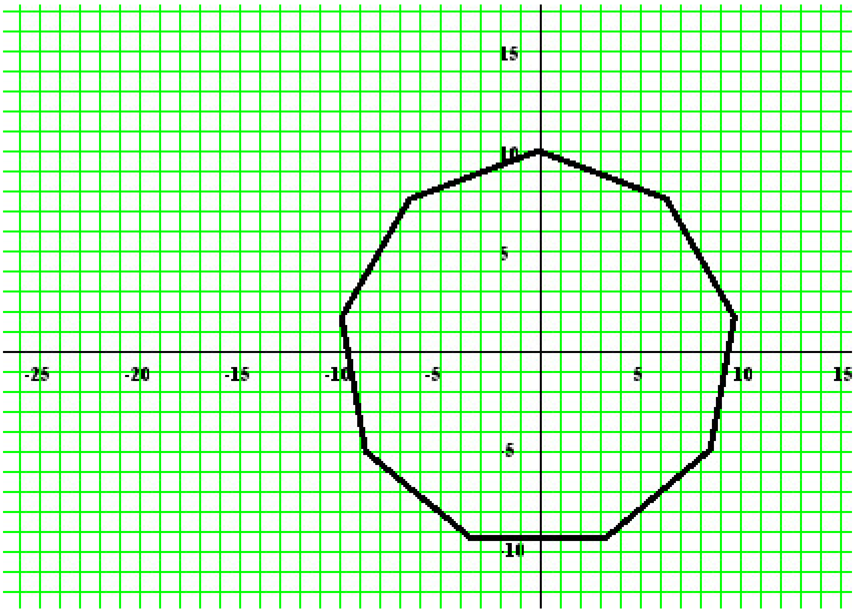
The first field is used to input the number of the side for a regular polygon, and the second is used to input the size of the circle in which The DrawPoly will draw.

The unit is the same as Bresenham's circle algorithm, 1 unit = 1 pixel. In the picture above, the first field is 9, indicating nonagon and 200 to indicate a circle of diameter 200 pixels or 20 units.

The regular polygon was implemented using the polygon shape in the awt.

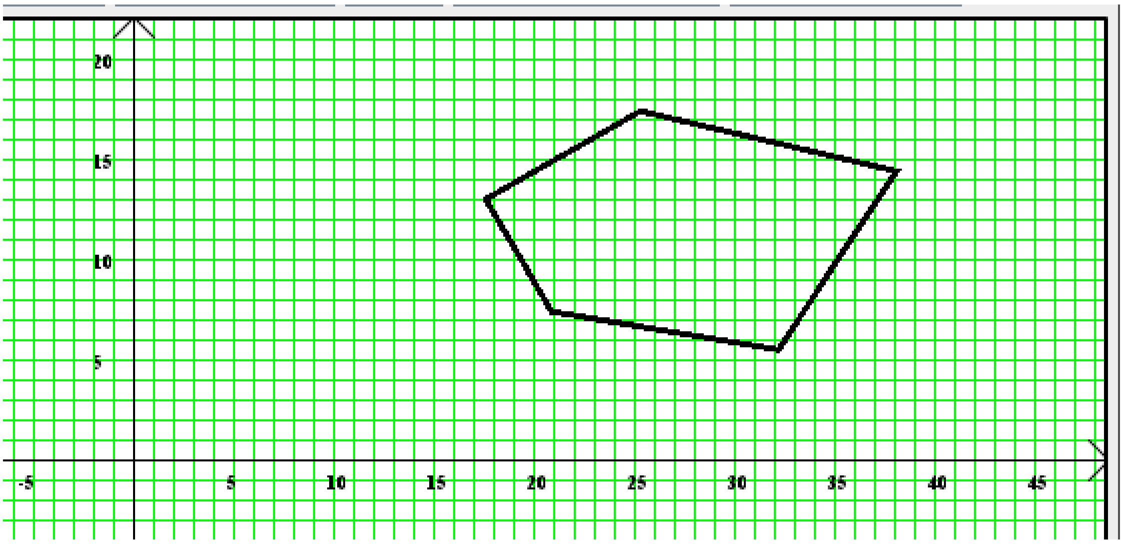
The regular polygon has a maximum of 72 points input, and the same limit stands for the irregular polygon.

Polygon shape provides a fillpolygon() method to facilitate the fill of the polygon with colour.



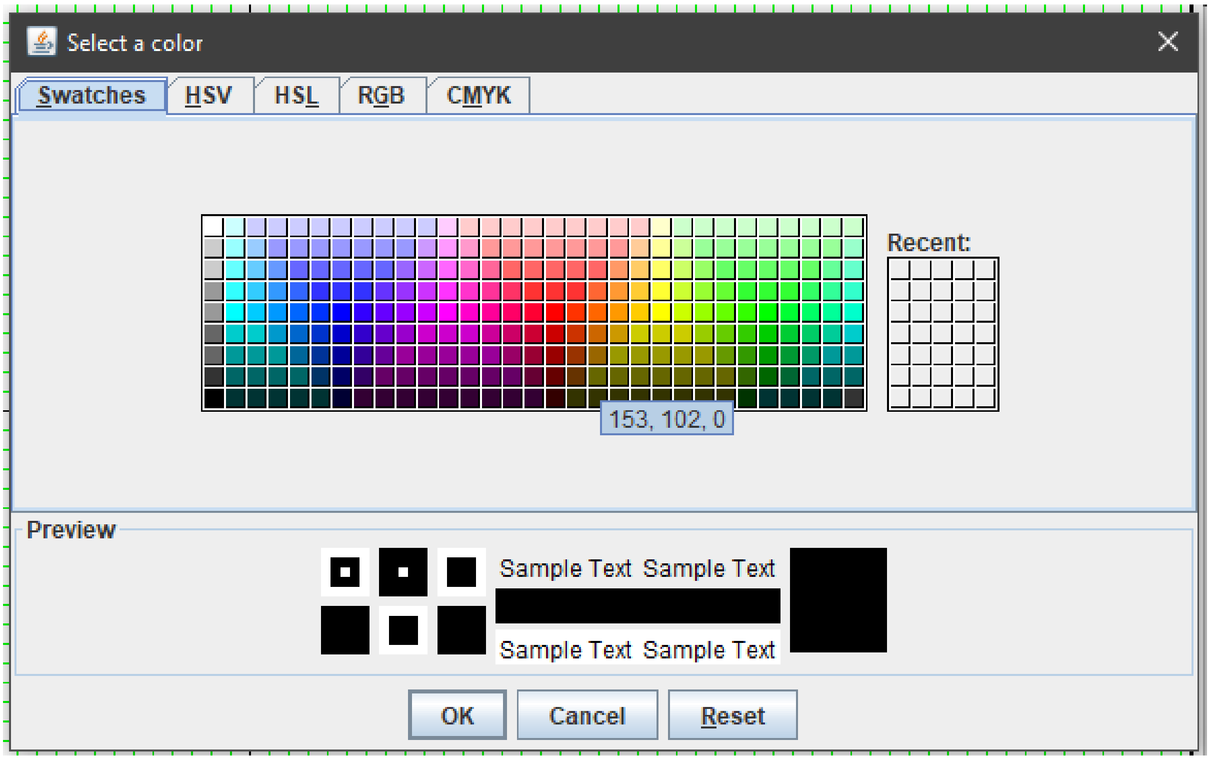
Irregular Polygon

The irregular polygon was implemented using a button that activates when clicked and records every click on the drawing panel, and stores it in an array list. The array list is then converted to an array since the polygon from awt only accept an array.
ArrayList was used for efficient usage of memory.



Choose Color button

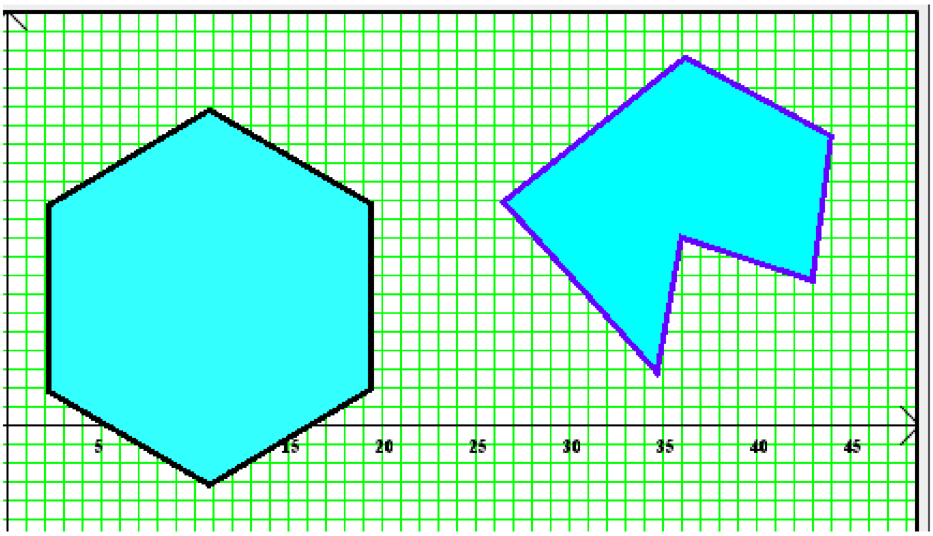
The "Choose colour" button was implemented using a dialog box with the jcolorchooser method in the dialog box, which allow the user to pick a colour from its wide range.
The jcolorchooser has an initial colour of black.
The same colour chooser button is used to choose the line's colour and to fill the polygon.
The dialog box is already built-in java, and we do not need to build a new colour chooser. It facilitates and decreases our work time.



Filling Line and Polygon

Two-button was chosen to fulfil these two tasks, fill and fill line button.
After selecting a colour in the colour chooser dialog box, the button press will retrieve that colour and fill it in the line if the fill line button is clicked or fill the polygon interior if the button fill is clicked.

The fill and fill Line button modified the last drawn polygon only.

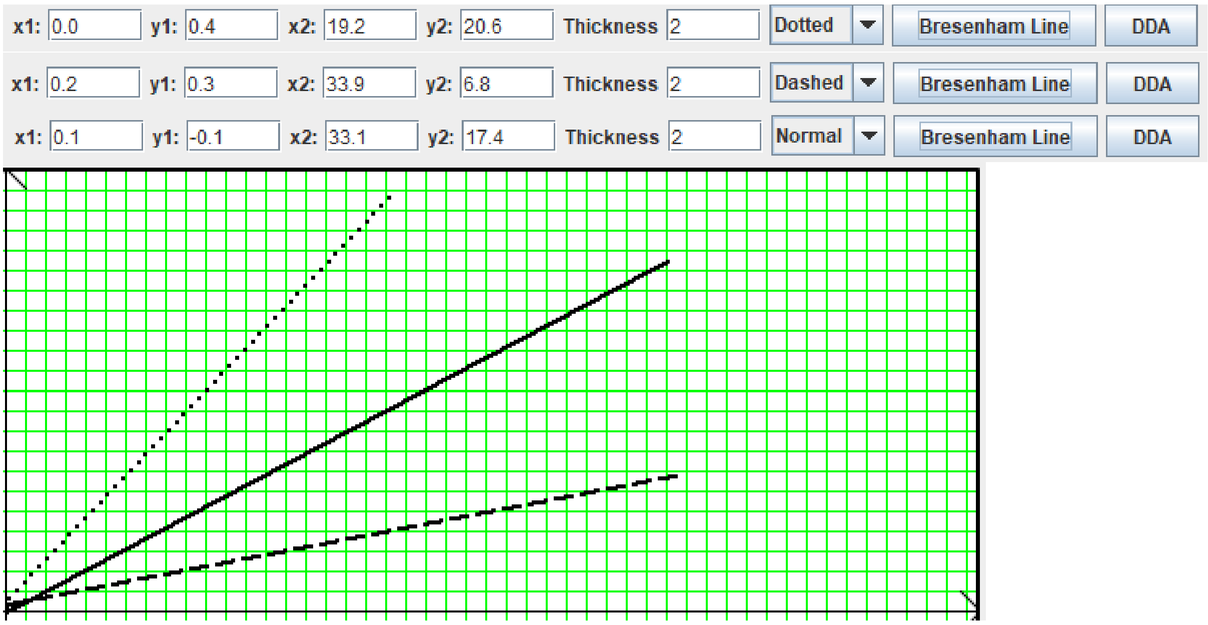


Bresenham and Digital deferential analyzer line Algorithm

The bresenhampanel class return the 2nd panel in the tool panel. Mouse listener is used to getting the coordinate at 2 points, or the two coordinates can be input in the text field to draw the line.

Users can enter the thickness in the text field near the label thickness, and the line can be normal(solid) dashed or dotted. Users can select these attributes in the drop-down menu. The bresenham and dda buttons specify which algorithm to use on the given input.
The dda and bresenham algorithms use lines to simulate the dot on one pixel. Using a line from awt, we can edit the thickness by using the newStroke(thickness) instance and the attribute dashed and dotted are simulated using calculation.

Using Bresenham's Algorithm

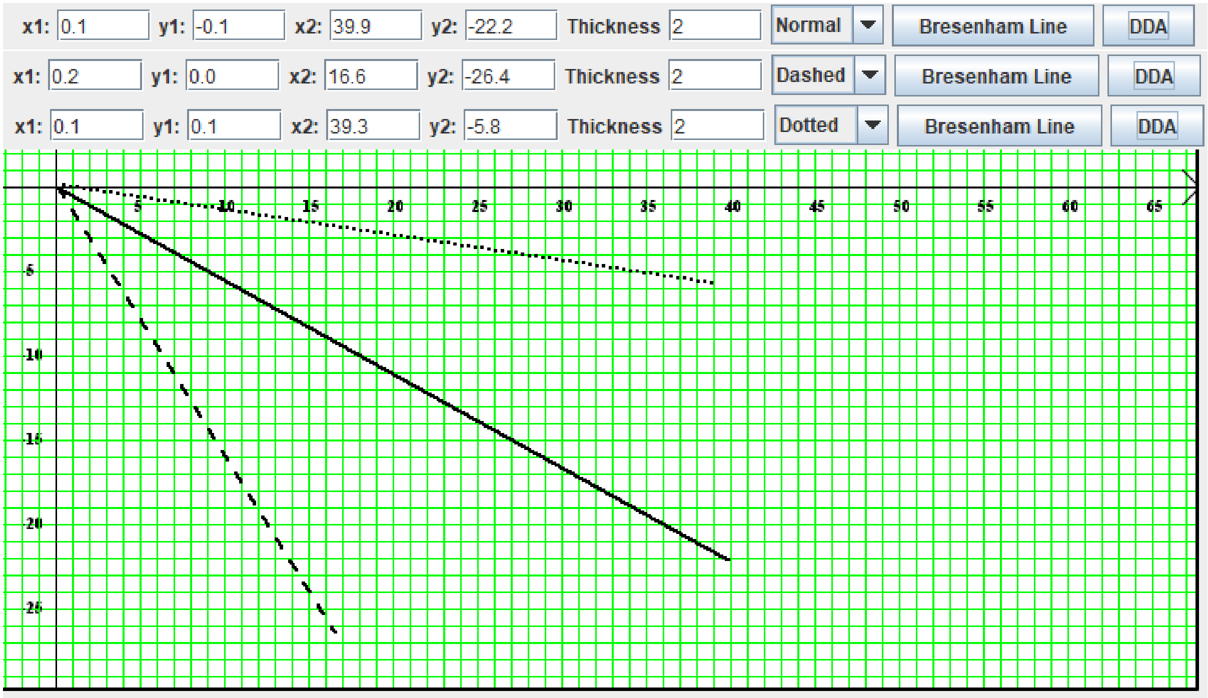


x1: 0.0 y1: 0.4 x2: 19.2 y2: 20.6 Thickness 2 Dotted

x1: 0.2 y1: 0.3 x2: 33.9 y2: 6.8 Thickness 2 Dashed

x1: 0.1 y1: -0.1 x2: 33.1 y2: 17.4 Thickness 2 Normal

Using DDA algorithm



x1: 0.1 y1: -0.1 x2: 39.9 y2: -22.2 Thickness 2 Normal

x1: 0.2 y1: 0.0 x2: 16.6 y2: -26.4 Thickness 2 Dashed

x1: 0.1 y1: 0.1 x2: 39.3 y2: -5.8 Thickness 2 Dotted