2D Graphics Programming – Rendered Graphics

Zbigniew Dendzik
Institute of Physics, University of Silesia

2025

Introduction

The Java2D library, which is part of the Java SE platform, is used for 2D graphics programming. In this set you will find examples and exercises on rendering graphic objects and using colours, gradients, textures, transparency and fonts available in the local graphic environment.

Example I

Below you will find the definition of the `Board` class extending the library class `javax.swing.JPanel`, containing the field `shape` of type `java.awt.Shape`, and the corresponding `Program` class used to test the example in a graphical interface window (`javax.swing.JFrame`).

```
1  import javax.swing.*;
2  import java.awt.*;
3  import java.awt.geom.*;
4
5  class Board extends JPanel
6  {
7     Shape shape;
8
9     Board(Shape shape)
10    {
11       this.shape=shape;
12    }
13
14    public void paintComponent(Graphics g)
15    {
16       super.paintComponent(g);
17       Graphics2D g2d=(Graphics2D)g;
18
19       g2d.draw(shape);
20    }
21 }
22
23 public class Program
24 {
25    public static void main(String[] args)
```

```
26      {
27          Shape obj1;
28          obj1=new Rectangle2D.Float(100,100,140,100);
29
30          Board p;
31          p=new Board(obj1);
32
33          JFrame jf=new JFrame();
34          jf.add(p);
35
36          jf.setTitle("Graphics test");
37          jf.setSize(400,400);
38          jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
39          jf.setVisible(true);
40      }
41  }
```

Listing 1: Program.java

Ex. 8.1

Identify the class members, analyse the purpose of their use and the syntax, and analyse the documentation for each of the library classes used. Prepare an appropriate compilation unit and test the example.

Ex. 8.2

Create a board with a rectangle represented by a `Rectangle2D.Float` object and an ellipse represented by an `Ellipse2D.Float` object. Using the documentation, find an appropriate method or methods to check whether these figures intersect. For the test, use an `if` statement and the `void drawString(String str, float x, float y)` method of the `java.awt.Graphics2D` class (you may need to use appropriate casting).

Ex. 8.3

Create a board with a rectangle represented by a `Rectangle2D.Float` object and an ellipse represented by an `Ellipse2D.Float` object. Using the documentation, find an appropriate method or methods to check whether these figures contain each other.

Ex. 8.4

Create a board with a rectangle represented by an `Ellipse2D.Float` object and a point represented by a `Point2D.Float` object. Using the documentation, find an appropriate method or methods to check whether this point lies inside the rectangle.

Ex. 8.5

Using the documentation of the `java.awt.Graphics2D` class and its methods, write a program that will render a green rectangle with fill.

Ex. 8.6

Using the documentation of the `java.awt.Graphics2D` class and its methods, write a program that will render an ellipse covered with a gradient according to the figure in the instructions.

Ex. 8.7

Using the documentation of the `java.awt.Graphics2D` class and its methods, write a program that will render an ellipse covered with a radial gradient.

Ex. 8.8

Using the documentation of the `java.awt.Graphics2D` class and its methods, write a program that will render a rectangle and cover it with a texture loaded from any jpg file.

A `BufferedImage` object can be created as follows:

```
import java.io.*;
import java.awt.image.BufferedImage;
import javax.imageio.ImageIO;
// (...)

String fileName="file.jpg";
BufferedImage img;
try
{
   File f=new File(fileName);
   img=ImageIO.read(f);
}
catch(IOException e)
{
   System.err.println("File problem");
}
```

Listing 2: BufferedImage – skeleton

Ex. 8.9

Using the documentation of the `java.awt.Graphics2D` class and its methods, write a program that will render two intersecting ellipses shown in the figure in the instructions.

An `AlphaComposite` object can be created for example as follows:

```
AlphaComposite a = AlphaComposite.getInstance(AlphaComposite.
   SRC_OVER,0.5f);
```

Listing 3: AlphaComposite – example

Ex. 8.10

Using the documentation of the `java.awt.Graphics2D` class and its methods, write a program that will render an arc segment (`Arc2D`) shown in the figure in the instructions.

Ex. 8.11

Using the documentation of the `java.awt.Graphics2D` class and its methods, write a program that will render the lines shown in the figure in the instructions. Pay attention to the line caps.

Ex. 8.12

Using the documentation of the `java.awt.Graphics2D` class and its methods, write a program that will render the shapes shown in the figure (`Polygon`) and, using appropriate methods of the graphics context, demonstrate translation, rotation and scaling operations. Pay attention to the way lines are joined.

Ex. 8.13*

Extend the calculator from the previous set by adding graphical calculator functionality – your calculator should have the functionality of a normal calculator and should draw graphs of functions (for this you will also have to solve the problem of parsing mathematical expressions; you may limit yourself to a subset of expressions, for example polynomials or rational functions).