DRAWING LINES AND RECTANGLES

DRAWING LINES

When drawing graphics, you must work with the "Cartesian Coordinate System", whereby the coordinates (i.e. **x** and **y**) of the graphical object must be specified.

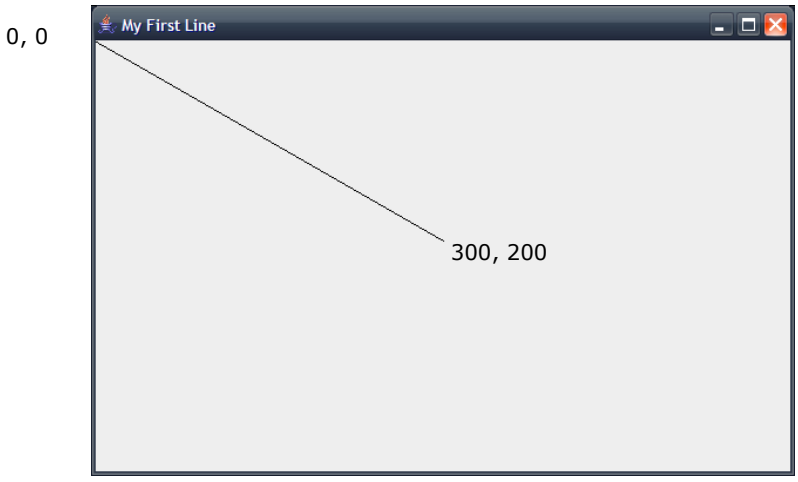
The most basic graphic shape that we will be drawing first is a line. The syntax for drawing a line is:

```
g2.draw(new Line2D.Double(x1, y1, x2, y2));
```

This command draws a line from the starting point (x1, y1) to the ending point (x2, y2).

The following line of code will draw a diagonal line from the top left-hand corner of the frame to the center of the frame (in this example I am using a 600 x 400 frame):

```
g2.draw(new Line2D.Double(0, 0, 300, 200));
```



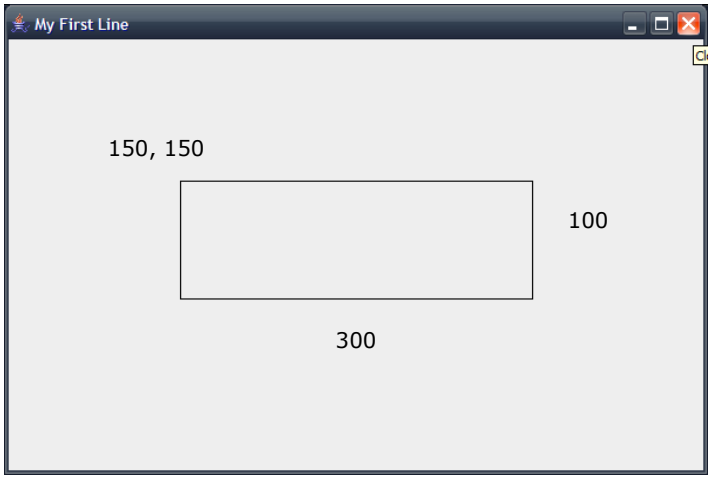
DRAWING RECTANGLES

When drawing a rectangle, you need to specify the upper-left coordinate of the rectangle (x, y) and then the width and height of the rectangle (measured in pixels) as follows:

```
g2.draw(new Rectangle2D.Double(x, y, width, height));
```

The following line of code will draw a 300 x 100 rectangle at the center of the screen:

```
g2.draw(new Rectangle2D.Double(150, 150, 300, 100));
```

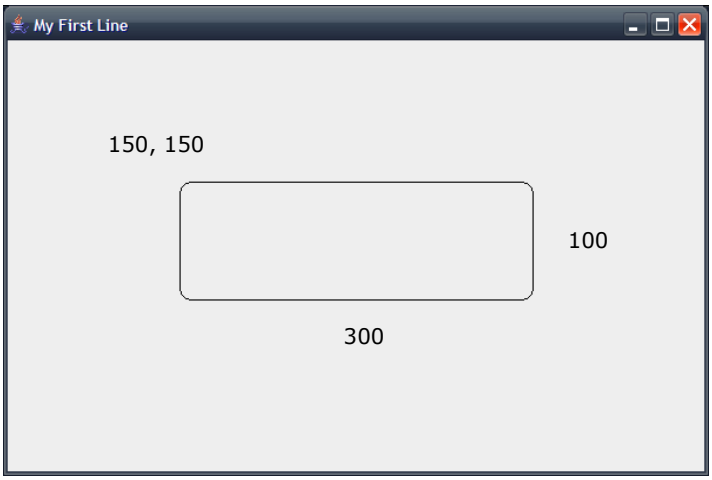


You can also use the **RoundRectangle2D.Double** class to draw a rectangle with rounded corners. This method requires you to specify the upper-left coordinate of the rectangle (x, y), the width and height (in pixels), and the width and height of the arcs that make up the corners:

```
g2.draw(new RoundRectangle2D.Double(x, y, width,
        height, arcWidth, arcHeight));
```

The following line of code will draw the same rectangle as our previous example except with rounded corners:

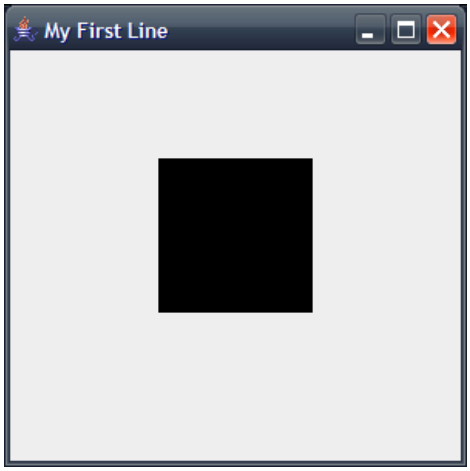
```
g2.draw(new RoundRectangle2D.Double(150, 150, 300, 100, 8,
   10));
```



We will look at one other method that we can use to create a rectangle. The **fill()** method can be used to draw a filled rectangle and a filled rounded rectangle respectively. The parameters that are used are the same as the draw methods; the only difference is that the shape is filled in with the same colour as the shape's outline.

The following command will draw a filled rectangle in the shape of a square:

```
g2.fill(new Rectangle2D.Double(100, 100, 100, 100));
```



SETTING COLOURS

In order to set colours in Java, you need to use the **Color** class, which is part of the **java.awt** package, and the **setColor()** or **setPaint()** method included in the **Graphics2D** class.

One way of setting the colour of an object is by using of the many default colours included in the **Color** class. The following is a list of default colours:

BLACK	BLUE	CYAN	DARK GRAY	GRAY	GREEN
MAGENTA	ORANGE	PINK	RED	WHITE	YELLOW

The following lines of code will set the colour of my rectangle to the default colour pink:

```
g2.setColor(Color.PINK);
g2.fill(new Rectangle2D.Double(100, 100, 100, 100));
```

Of course, you are not limited to Java's default colours. You can specify any colour you wish by simply indicating the RGB values of the colour you want to use. So, for example, if I wanted to use a specific shade of blue, I would do the following:

```
Color myBlue = new Color(0, 0, 122);
g2.setColor(myBlue);
g2.fill(new Rectangle2D.Double(100, 100, 100, 100));
```

In the above example, I've declared a variable called **myBlue** and the RGB values of the colour I'm using are **0, 0, 122**.

The following is an alternative way of setting a color using just one line of code:

```
g2.setColor(new Color (0, 0, 122));
```

To find colour codes, you can simply open a program like Paint, Photoshop or even Microsoft Word, go into the colour picker menu, choose the colour you want and write down the RGB values.