

Example	Description
Animator	This applet performs one of four separate animations.
ArcTest	This applet demonstrates drawing arcs. You can interact with the applet to change attributes of the arc that is displayed.
BarChart	This applet draws a simple bar chart.
Blink	This applet displays blinking text in different colors.
CardTest	This applet demonstrates several GUI components and a variety of ways in which GUI components can be arranged on the screen (the arrangement of GUI components is also known as the <i>layout</i> of the GUI components).
Clock	This applet draws a clock with rotating “hands,” the current date and the current time. The clock is updated once per second.
DitherTest	This applet demonstrates drawing with a graphics technique known as dithering that allows gradual transformation from one color to another.
DrawTest	This applet allows the user to drag the mouse to draw lines and points on the applet in different colors.
Fractal	This applet draws a fractal. Fractals typically require complex calculations to determine how they are displayed.
GraphicsTest	This applet draws a variety of shapes to illustrate graphics capabilities.
GraphLayout	This applet draws a graph consisting of many nodes (represented as rectangles) connected by lines. Drag a node to see the other nodes in the graph adjust on the screen and demonstrate complex graphical interactions.
ImageMap	This applet demonstrates an image with <i>hot spots</i> . Positioning the mouse pointer over certain areas of the image highlights the area and a message is displayed in the lower-left corner of the appletviewer window. Position over the mouth in the image to hear the applet say “hi.”
JumpingBox	This applet moves a rectangle randomly around the screen. Try to catch it by clicking it with the mouse!
MoleculeViewer	This applet present a three-dimensional view of several different chemical molecules. Drag the mouse to view the molecule from different angles.
NervousText	This applet draws text that jumps around the screen.
SimpleGraph	This applet draws a complex curve.
SortDemo	This applet compares three sorting techniques. Sorting (described in Chapter 7) arranges information in order—like alphabetizing words. When you execute the applet, three appletviewer windows appear. Click in each one to start the sort. Notice that the sorts all operate at different speeds.
SpreadSheet	This applet demonstrates a simple spreadsheet of rows and columns.
SymbolTest	This applet draws characters from the Java character set.
TicTacToe	This applet allows the user to play Tic-Tac-Toe against the computer.
WireFrame	This applet draws a three-dimensional shape as a wire frame. Drag the mouse to view the shape from different angles.

Fig. 3.1 The examples from the **applets** directory.

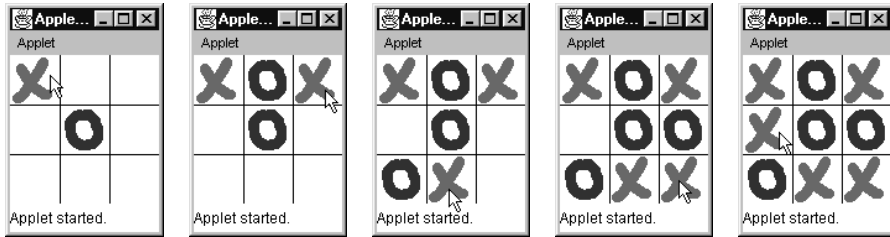


Fig. 3.2 Sample execution of the **TicTacToe** applet.

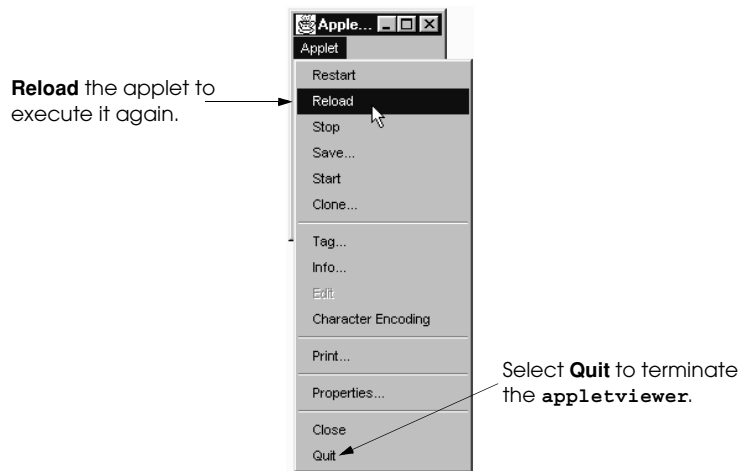


Fig. 3.3 Selecting **Reload** from the **appletviewer**'s **Applet** menu.

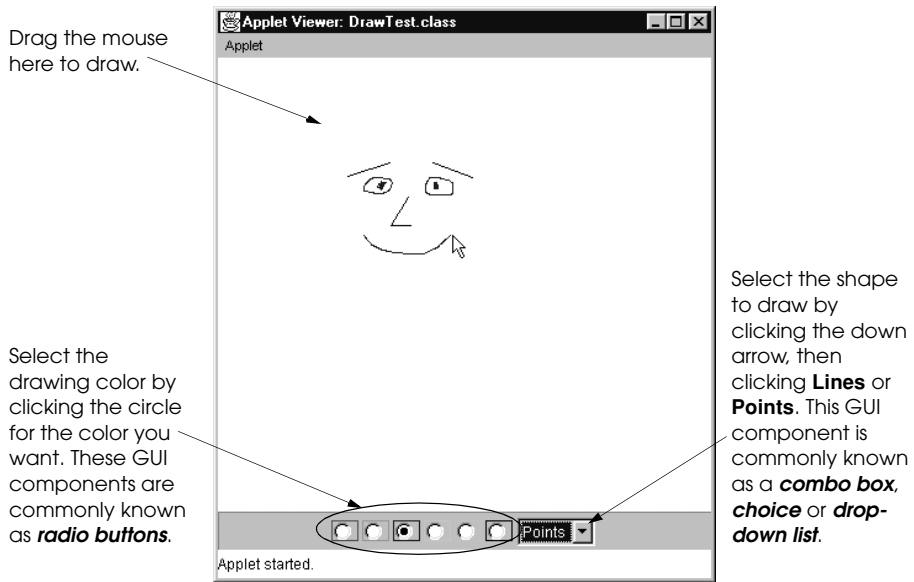


Fig. 3.4 Sample execution of the **DrawTest** applet.

Click a tab to select a two-dimensional graphics demo.

Try changing the options to see their effect on the demonstration.

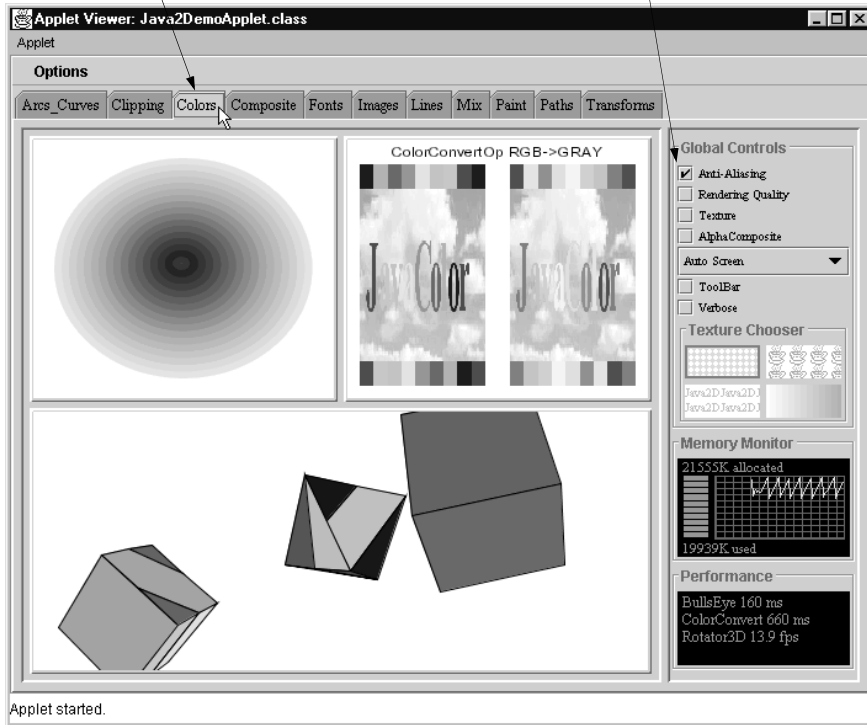


Fig. 3.5 Sample execution of the **Java2D** applet.

```

1 // Fig. 3.6: WelcomeApplet.java
2 // A first applet in Java
3 import javax.swing.JApplet; // import class JApplet
4 import java.awt.Graphics; // import class Graphics
5
6 public class WelcomeApplet extends JApplet {
7     public void paint( Graphics g )
8     {
9         g.drawString( "Welcome to Java Programming!", 25, 25 );
10    }
11 }

```

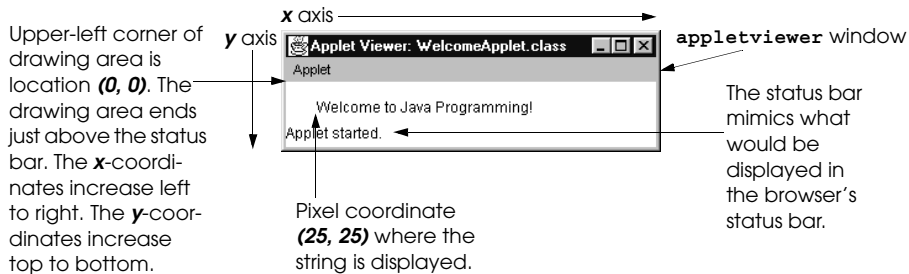


Fig. 3.6 A first program in Java and the program's screen output.

```
1 <html>
2 <applet code="WelcomeApplet.class" width=300 height=30>
3 </applet>
4 </html>
```

Fig. 3.7 The **WelcomeApplet.html** file, which loads the **WelcomeApplet** class of Fig. 3.6 into the **appletviewer**.

```
1 // Fig. 3.8: WelcomeApplet2.java
2 // Displaying multiple strings
3 import javax.swing.JApplet; // import class JApplet
4 import java.awt.Graphics; // import class Graphics
5
6 public class WelcomeApplet2 extends JApplet {
7     public void paint( Graphics g )
8     {
9         g.drawString( "Welcome to", 25, 25 );
10        g.drawString( "Java Programming!", 25, 40 );
11    }
12 }
```

Pixel coordinate **(25, 25)**, where
Welcome to is displayed

Pixel coordinate **(25, 40)**, where
Java Programming! is displayed

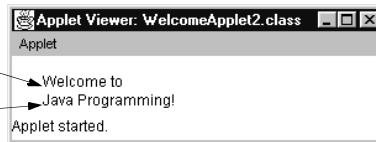


Fig. 3.8 Displaying multiple strings.


```
1 <html>
2 <applet code="WelcomeApplet2.class" width=300 height=45>
3 </applet>
4 </html>
```

Fig. 3.9 The `WelcomeApplet2.html` file, which loads the `WelcomeApplet2` class of Fig. 3.8 into the `appletviewer`.

```
1 // Fig. 3.10: WelcomeLines.java
2 // Displaying text and lines
3 import javax.swing.JApplet; // import class JApplet
4 import java.awt.Graphics; // import class Graphics
5
6 public class WelcomeLines extends JApplet {
7     public void paint( Graphics g )
8     {
9         g.drawLine( 15, 10, 210, 10 );
10        g.drawLine( 15, 30, 210, 30 );
11        g.drawString( "Welcome to Java Programming!", 25, 25 );
12    }
13 }
```



Fig. 3.10 Drawing strings and lines.

```
1 <html>
2 <applet code="WelcomeLines.class" width=300 height=40>
3 </applet>
4 </html>
```

Fig. 3.11 The **WelcomeLines.html** file, which loads the **WelcomeLines** class of Fig. 3.10 into the **appletviewer**.

```

1 // Fig. 3.12: AdditionApplet.java
2 // Adding two floating-point numbers
3 import java.awt.Graphics; // import class Graphics
4 import javax.swing.*; // import package javax.swing
5
6 public class AdditionApplet extends JApplet {
7     double sum; // sum of the values entered by the user
8
9     public void init()
10    {
11        String firstNumber, // first string entered by user
12            secondNumber; // second string entered by user
13        double number1, // first number to add
14            number2; // second number to add
15
16        // read in first number from user
17        firstNumber =
18            JOptionPane.showInputDialog(
19                "Enter first floating-point value" );
20
21        // read in second number from user
22        secondNumber =
23            JOptionPane.showInputDialog(
24                "Enter second floating-point value" );
25

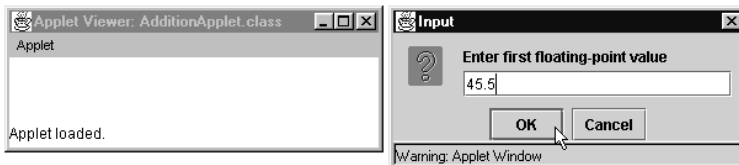
```

Fig. 3.12 An addition program "in action" (part 1 of 2).

```

26        // convert numbers from type String to type double
27        number1 = Double.parseDouble( firstNumber );
28        number2 = Double.parseDouble( secondNumber );
29
30        // add the numbers
31        sum = number1 + number2;
32    }
33
34    public void paint( Graphics g )
35    {
36        // draw the results with g.drawString
37        g.drawRect( 15, 10, 270, 20 );
38        g.drawString( "The sum is " + sum, 25, 25 );
39    }
40 }

```



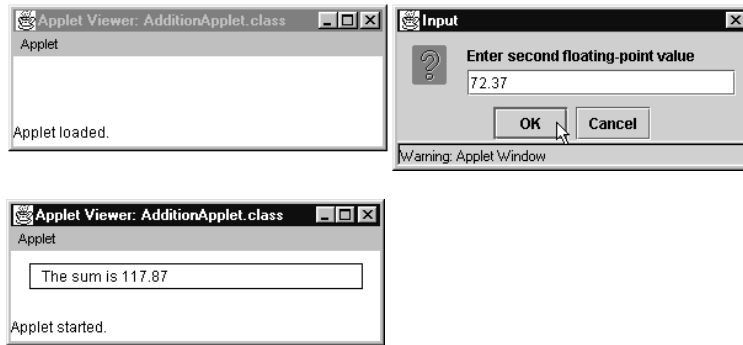


Fig. 3.12 An addition program "in action" (part 2 of 2).

```
1 <html>
2 <applet code="AdditionApplet.class" width=300 height=50>
3 </applet>
4 </html>
```

Fig. 3.13 The `AdditionApplet.html` file, which loads the `AdditionApplet` class of Fig. 3.12 into the `appletviewer`.