

Today's topics

Java Applications
Graphics

Upcoming
Simulation

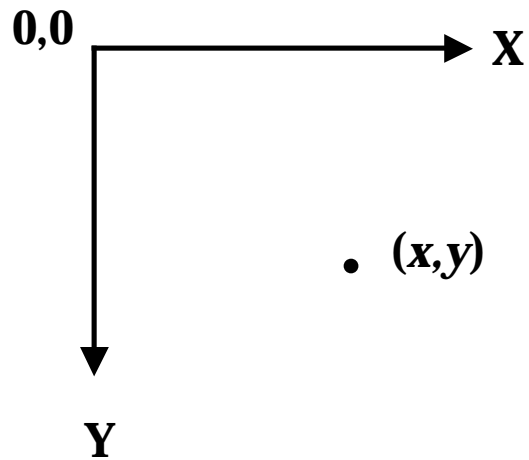
Reading
Great Ideas, Chapters 5

Graphics

- **Java comes with drawing package**
 - **Fairly limited**
 - **Not hard to use**
- **Use several new classes**
 - **Canvas Class**
 - **Graphics Class**
- **Basics ideas**
 - **Pixels**
 - **Coordinates**

Coordinate System

- **Origin at top left!**



- **Coordinates specified by giving x and y values**
 - **x is to the right**
 - **y is downward**

Setting up Graphics

- **Declare Graphics and Canvas**

```
Graphics g;
```

```
Canvas c;
```

- **Create Objects (in `init()`)**

```
c = new Canvas();
```

```
c.setSize(200, 200);
```

```
add(c);
```

```
g = c.getGraphics();
```

- **Can now create graphical figures**

```
g.DrawLine(50, 50, 150, 100);
```

draws line from (50,50) to (150,100) , a diagonal line

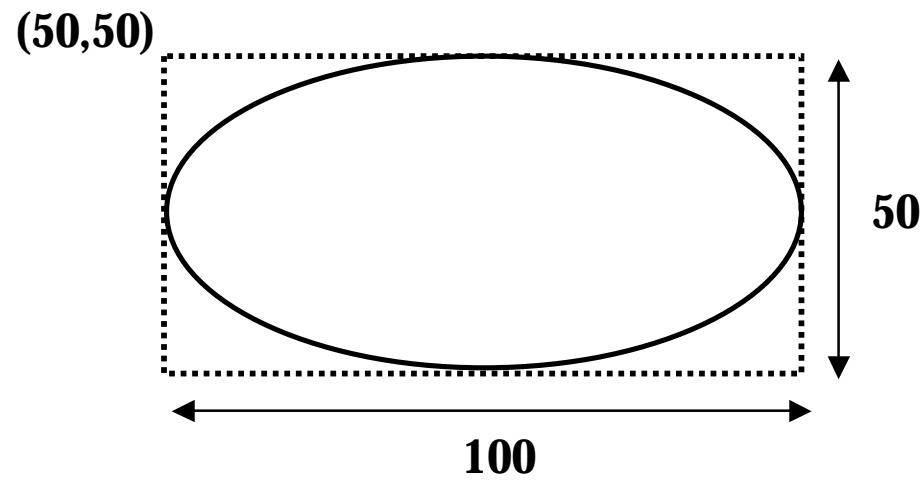
Specifying Location and Size

- A (straight) line is located by its endpoints
- A rectangle is located by its *top left corner* and its *width* and *height*

```
g.drawRect(50, 100, 100, 50);
```

- An oval is located by its
 - This is a rectangle into which an oval fits tightly
 - The bounding box is specified like a rectangle

```
g.drawOval(50, 50, 100, 50);
```



Graphics Demo

```
public class GDemo extends java.applet.Applet implements
                                   ActionListener
{
    Graphics g;
    Canvas c;
    Button b1, b2, b3, b4;

    public void init()
    {
        c = new Canvas();
        c.setSize(200, 200);
        add(c);
        g = c.getGraphics();
        b1 = new Button("Start");
        b2 = new Button("Line");
        b3 = new Button("Oval");
        b4 = new Button("Rectangle");
    }
}
```

Graphics Demo.2

```
b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
b4.addActionListener(this);
add(b1); add(b2); add(b3); add(b4);
}

public void actionPerformed(ActionEvent event)
{
    Object cause = event.getSource();

    if (cause == b1)
    {
        g.setColor(Color.white); // Color the whole canvas white
        g.fillRect(0, 0, 200, 200);
    }
}
```

Graphics Demo.3

```
    if (cause == b2)
    {
        g.setColor(Color.blue);
        g.drawLine(50, 50, 150, 100); // Demonstrate drawLine
    }
    if (cause == b3)
    {
        g.setColor(Color.green);
        g.drawOval(50, 50, 100, 50); // Demonstrate drawOval
    }
    if (cause == b4)
    {
        g.setColor(Color.red);
        g.drawRect(50, 100, 100, 50); // Demonstrate drawRect
    }
}
}
```

The Graphics Methods

- Used some additional methods
- Summary:

- **Basic:**

```
void drawLine(int x1, int y1, int x2, int y2)
```

```
void drawRect(int x, int y, int width, int height)
```

```
void drawOval(int x, int y, int width, int height)
```

- **Set color (before use):**

```
void setColor(Color c)
```

- Color.white, Color.red, Color.blue, Color.green

- **Versions *filled* with color:**

```
void fillRect(int x, int y, int width, int height)
```

```
void fillOval(int x, int y, int width, int height)
```

Flexibility

- **Everything was fixed in first example**
- **Count (Should!) use variables**
 - **Allow Coordinates to be read in**
 - **Allow (pre programmed) shift for each button press**
 - **Shown in next program**
- **Could do much more**
- **Buttons and input could control:**
 - **(in addition to position)**
 - **Color**
 - **Height and Width**
 - **Filled or Not**

Graphics with offsets

```
public class GDemo2 extends java.applet.Applet implements
                                   ActionListener

{Graphics g;
  Canvas c;
  IntField iX, iY;
  Button b1, b2, b3, b4, b5;
  int x, y;

public void init()
{ c = new Canvas();
  c.setSize(200, 200);
  add(c);
  g = c.getGraphics();
  b1 = new Button("Start"); b2 = new Button("Oval");
  b3 = new Button("Rectangle"); b4 = new Button("SetCorner");
  b5 = new Button("Shift");
  iX = new IntField(20); iY = new IntField(20);
```

Graphics with offsets.2

```
b1.addActionListener(this); b2.addActionListener(this);
b3.addActionListener(this); b5.addActionListener(this);
add(iX); add(iY); add(b1); add(b2); add(b3); add(b4);
add(b5);
}

public void actionPerformed(ActionEvent event)
{
    Object cause = event.getSource();

    if (cause == b1)
    {
        g.setColor(Color.white); // Color the whole canvas white
        g.fillRect(0, 0, 200, 200);
    }
}
```

Graphics with offsets.3

```
    if (cause == b2)
    { g.setColor(Color.green);
      g.drawOval(x, y, 100, 50); // Demonstrate drawOval
    }
    if (cause == b3)
    { g.setColor(Color.red);
      g.drawRect(x, y, 100, 50); // Demonstrate drawRect
    }
    if (cause == b4)
    { x = iX.getInt();
      y = iY.getInt();
    }
    if (cause == b5)
    { x = x + 5;
      y = y + 5;
    }
  }
}
```

Array Defined Graphical Figure

```
public class GDemo3 extends java.applet.Applet implements
                                   ActionListener
{
    Graphics g;
    Canvas c;
    Button b1, b2;
    int x[] = {45, 95, 130, 130, 95, 45, 10, 10};
    int y[] = {10, 10, 45, 95, 130, 130, 95, 45};
    int n;

    public void init()
    {
        c = new Canvas();
        c.setSize(200, 200);
        add(c);
        g = c.getGraphics();
        b1 = new Button("Start");  b2 = new Button("StringFig");
    }
}
```

Array Defined Graphical Figure.2

```
b1.addActionListener(this);
b2.addActionListener(this);
add(b1); add(b2);
n = 8;
}

public void actionPerformed(ActionEvent event)
{
    Object cause = event.getSource();

    if (cause == b1)
    {
        g.setColor(Color.white); // Color the whole canvas white
        g.fillRect(0, 0, 200, 200);
    }
}
```

Array Defined Graphical Figure.3

```
if (cause == b2)
{
    int k, j;
    k = 0;
    g.setColor(Color.blue);
    while (k < n)
    {
        j = 0;
        while (j < n)
        {
            g.drawLine(x[k], y[k], x[j], y[j]);
            j = j + 1;
        }
        k = k + 1;
    }
}
}
```