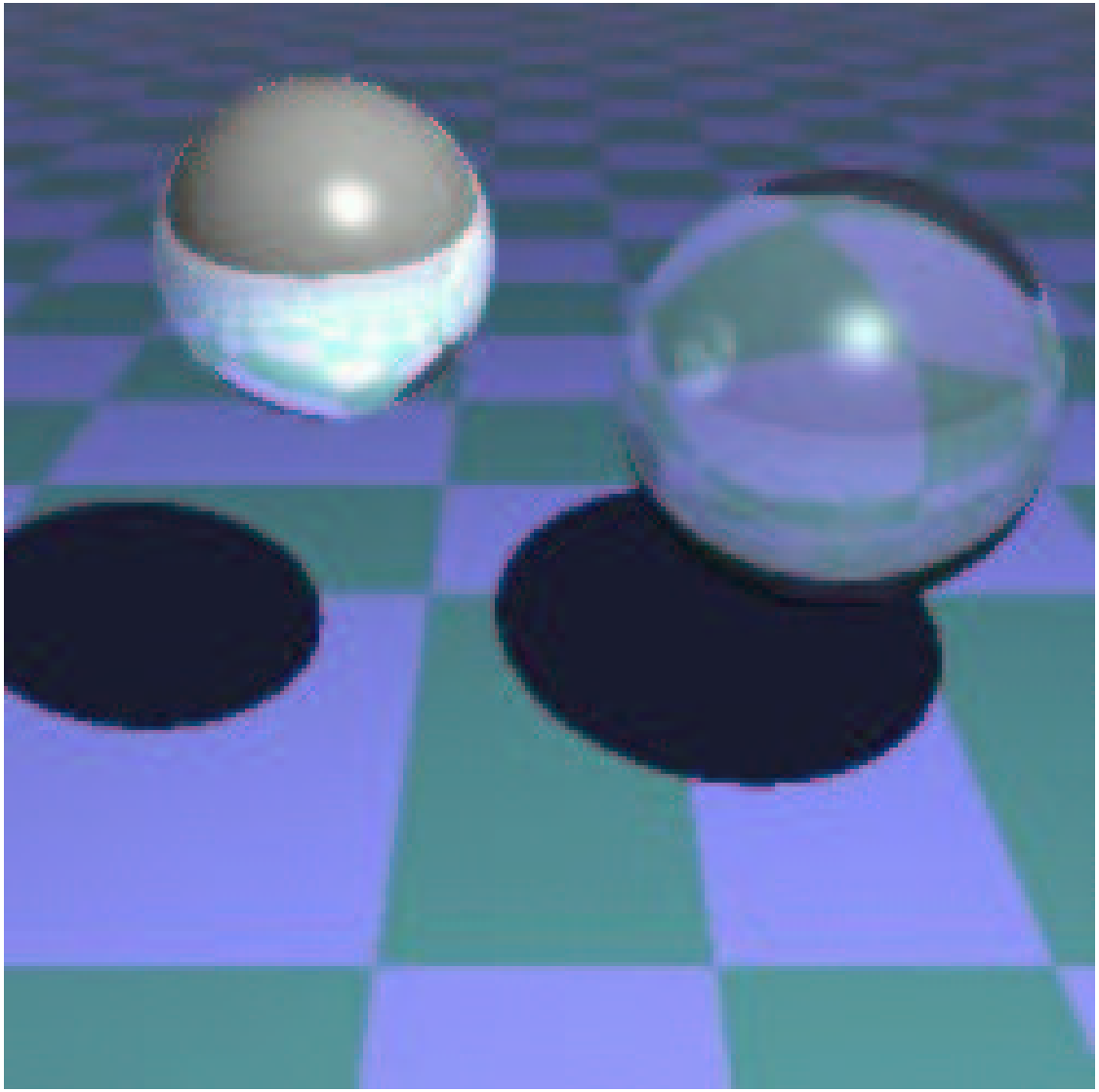
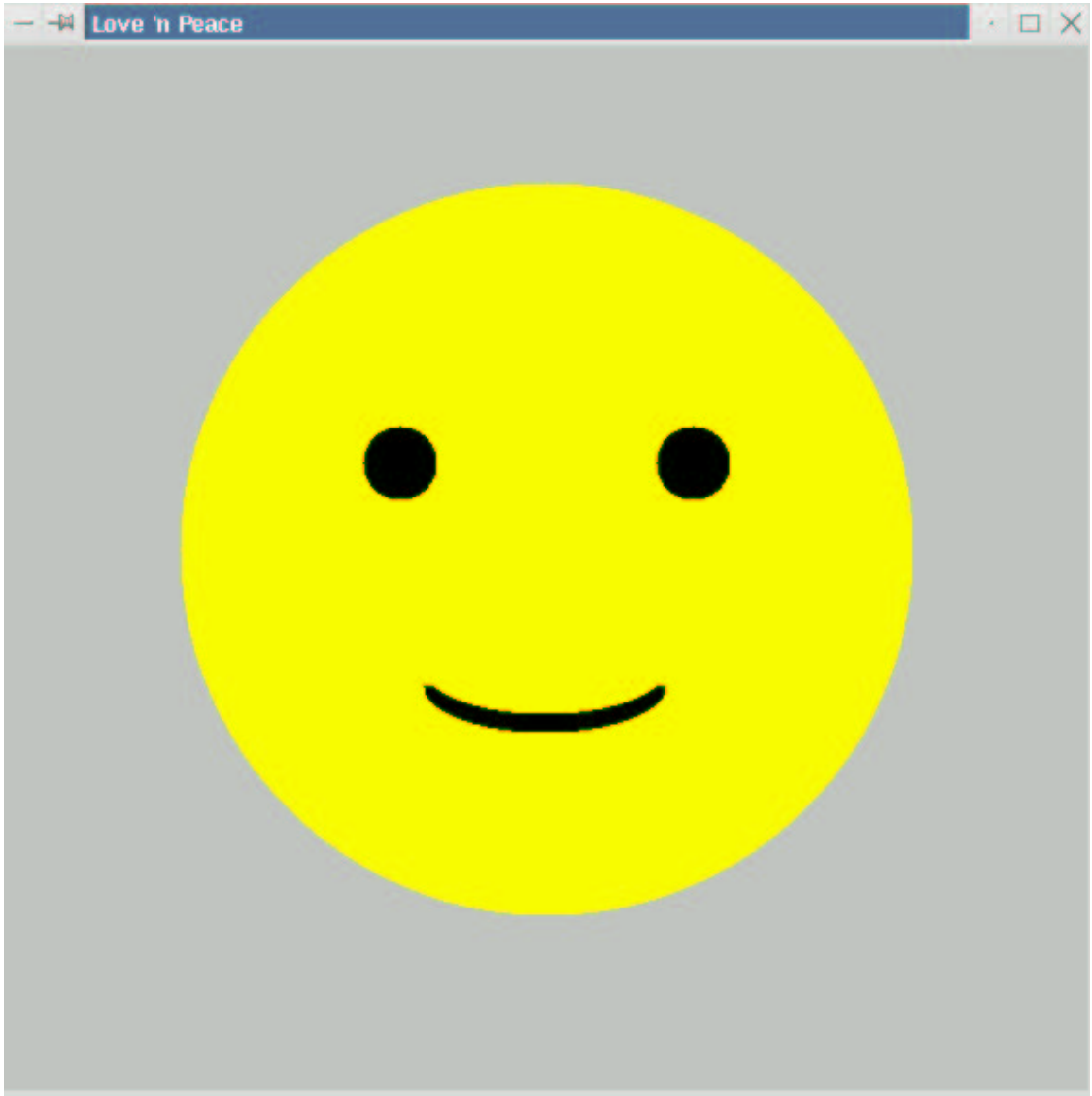
Graphics

Murray Cole



Graphics

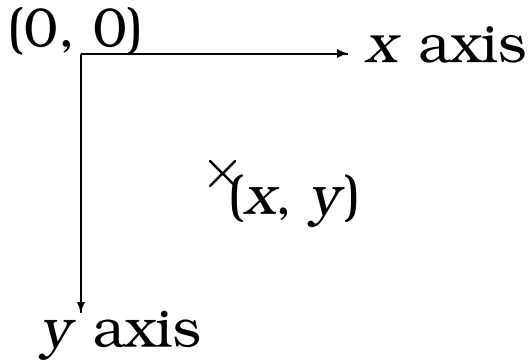

TOY
STORY


Love 'n Peace

```
class Smiley extends JFrame {
 public Smiley() {
  super ("Love 'n Peace");
  setSize (600, 600);
  show();
 }
 public void paint (Graphics g) {
  g.setColor(Color.yellow);
  g.fillOval(100,100,400,400);
  g.setColor(Color.black);
  g.fillArc(233, 350, 132, 50, 0, -180);
  g.fillOval(200, 233, 40, 40);
  g.fillOval(360, 233, 40, 40);
  g.setColor(Color.yellow);
  g.fillArc(233, 340, 132, 50, 0, -180);
 }
```

Basics

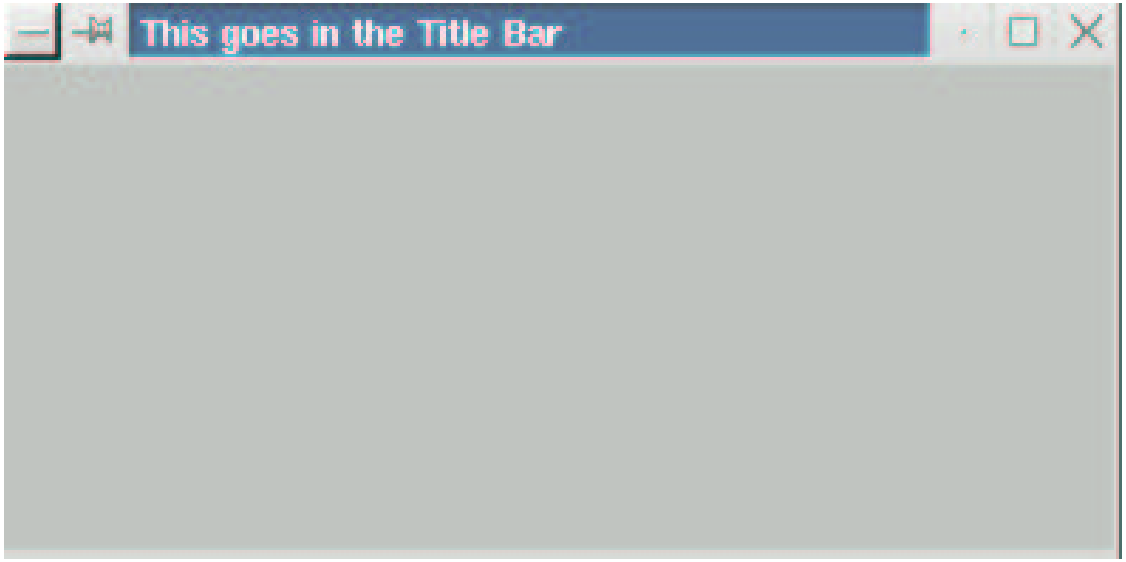
- package `javax.swing` contains basic classes (there are others)
- graphical objects displayed in a drawing area called a `JFrame`



A Basic Frame

```
import javax.swing.*;

class BasicFrame {
  public static void main (String[] args) {
    JFrame f = new JFrame("This goes in the Title Bar");
    f.setSize (400, 200);
    f.show();
  }
}
```


This goes in the Title Bar

Painting

- `JFrame` defines a method `paint` which is called by the system when the frame needs to be drawn or redrawn

```
public void paint (Graphics g)
```

- its parameter `g` is supplied by the system and is our drawing toolkit

- by default, paint does nothing, but we can override it if we define our own class to extend `JFrame`

```
import javax.swing.*;

class Smiley extends JFrame {
 public Smiley() {
  super ("Love 'n Peace");
  setSize (600, 600);
  show();
 }


 public static void main (String[] args) {
  Smiley s = new Smiley();
 }
```

```
 public void paint (Graphics g) {
  g.setColor(Color.yellow);
  g.fillOval(100,100,400,400);
  g.setColor(Color.black);
  g.fillArc(233, 350, 132, 50, 0, -180);
  g.fillOval(200, 233, 40, 40);
  g.fillOval(360, 233, 40, 40);
  g.setColor(Color.yellow);
  g.fillArc(233, 340, 132, 50, 0, -180);
 }
}
```

Graphics context objects

The `Graphics` object has many methods for setting colours, fonts, sizes and drawing various shapes, with various effects, in the `JFrame` to which it corresponds

```
void setColor (Color c) //Predefined or roll your own

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

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

void drawString(String s, int x, int y)
```

Colours

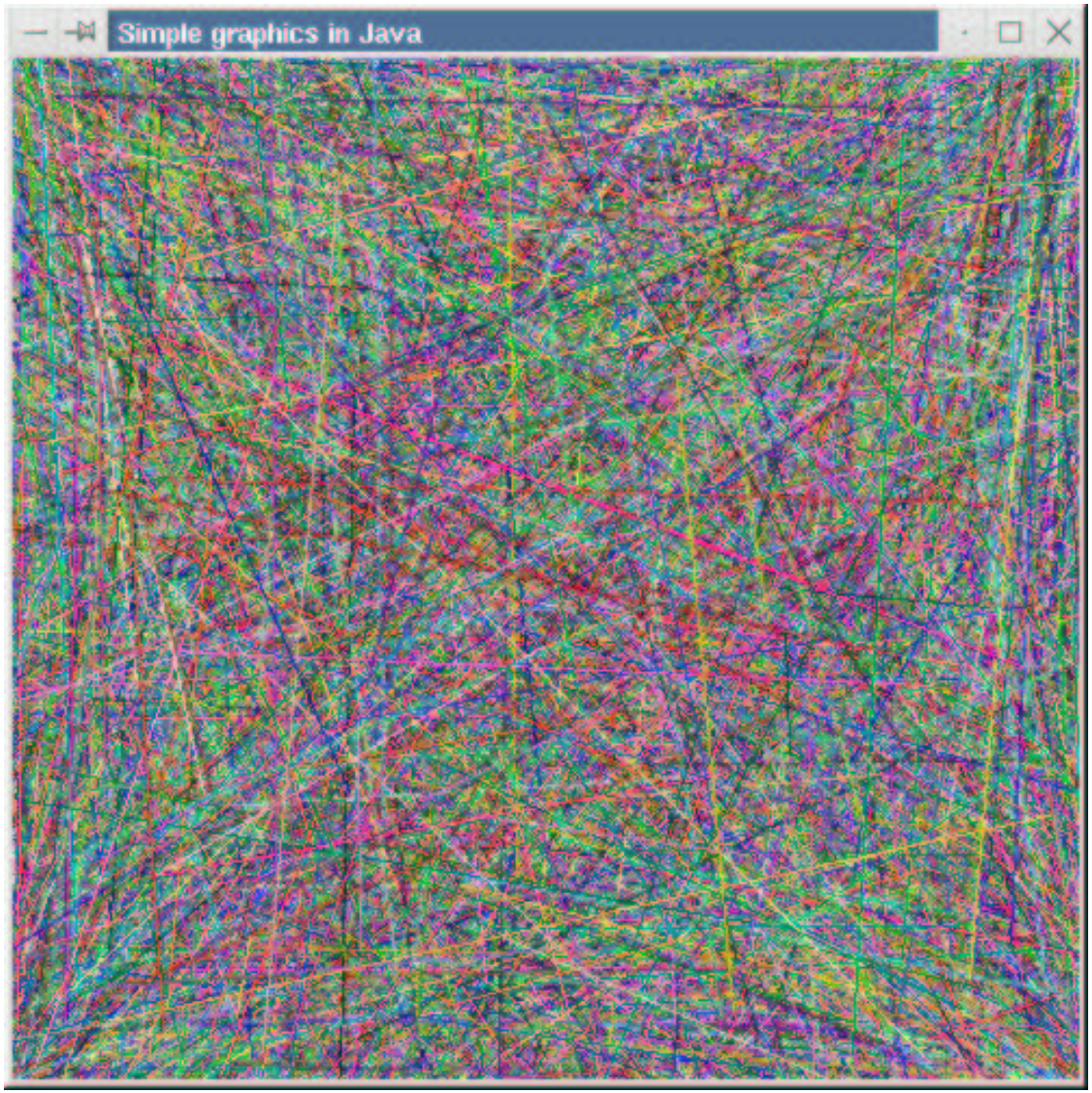
- the `Color` class defines colours as combinations of red, green and blue on a scale of 0 (none) to 255 (lots)
- some predefined constants (`Color.red`, `Color.magenta` ...)
- or build your own

```
Color c = new Color (255, 255, 0);  // which is yellow
```

An example - random lines

```
// A method to return random numbers between 0 and N
int rnd (int N) {
  return (int) (N * Math.random());
}

public void paint (Graphics g) {
  while (true) {
    g.setColor (new Color(rnd(255), rnd(255), rnd(255)));
    g.drawLine (rnd(500), rnd(500), rnd(500), rnd(500));
  }
}
```

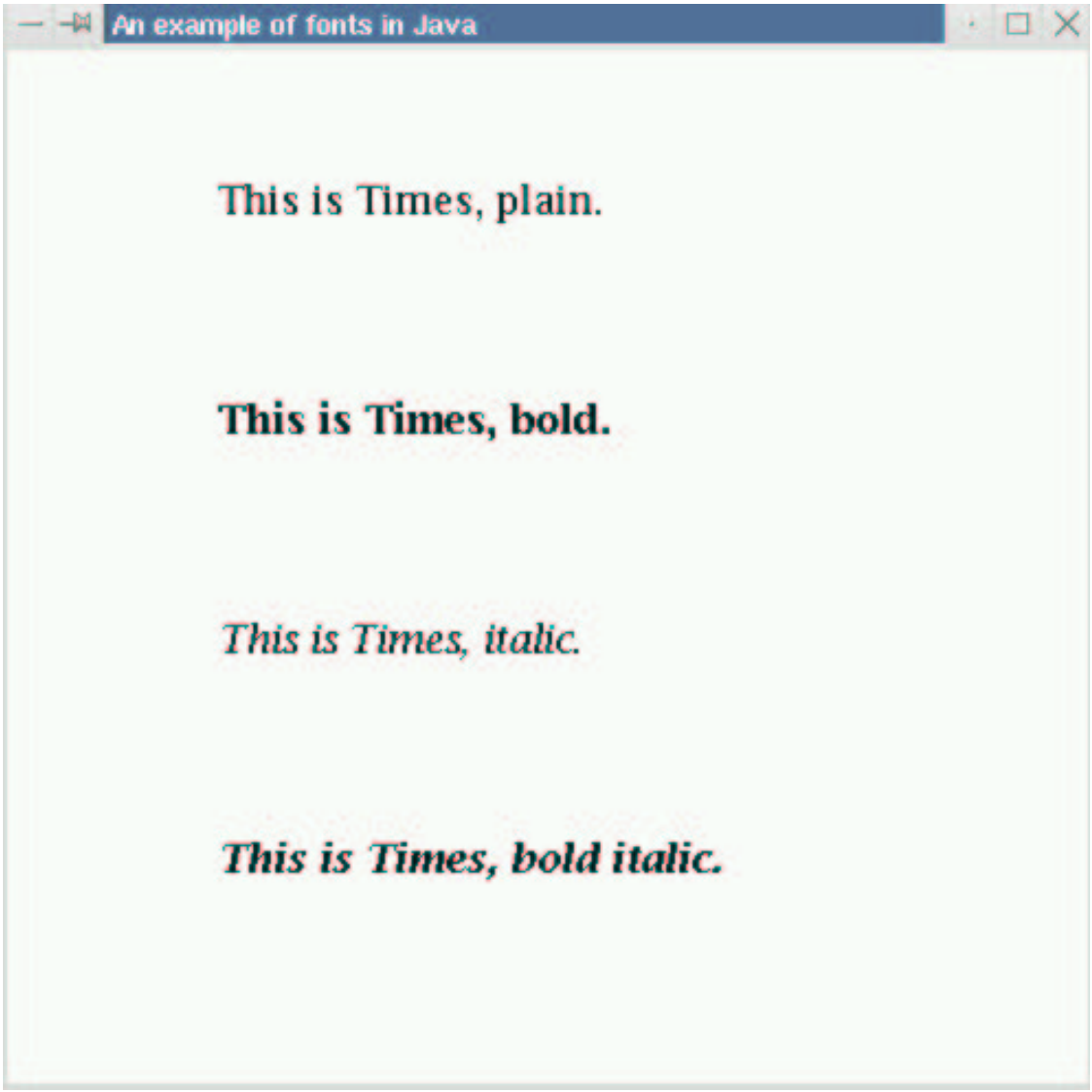

Simple graphics in Java

Fonts

- the `Font` allows us to choose fonts in different styles and sizes

```
Font bold = new Font("TimesRoman", Font.BOLD, 18);
g.setFont(bold);
g.drawString("This is Times, bold.", 100, 200);

Font italic = new Font("TimesRoman", Font.ITALIC, 18);
g.setFont(italic);
g.drawString("This is Times, italic.", 100, 300);
```


An example of fonts in Java
This is Times, plain.
This is Times, bold.
This is Times, italic.
This is Times, bold italic.

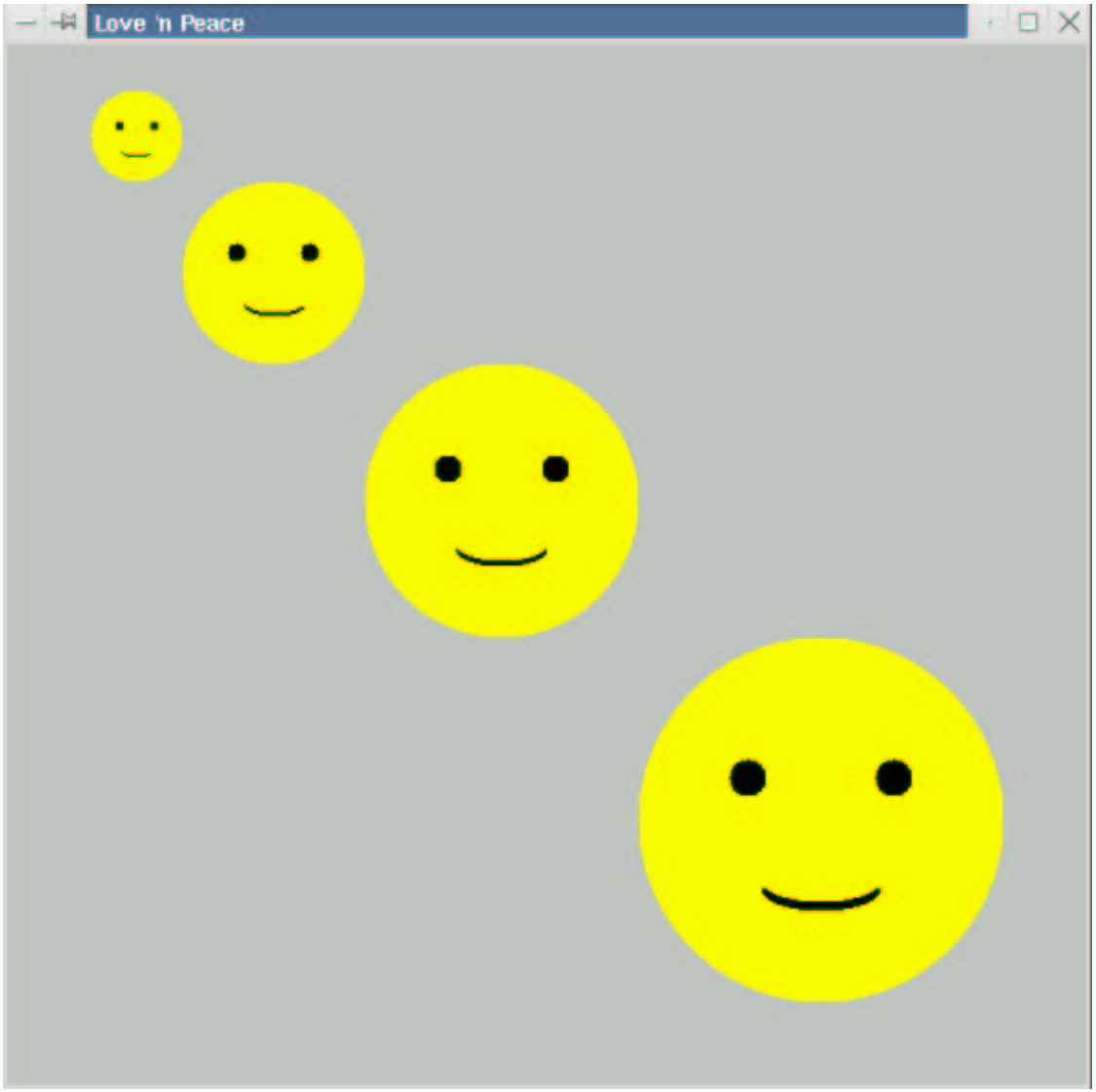
Relatively Easy?

```
public void drawSmiley (Graphics g, int x, int y, int r) {
    g.setColor(Color.yellow);
    g.fillOval(x-r, y-r,2*r,2*r);
    g.setColor(Color.black);
    g.fillArc(x-r/3, y+r/4, 2*r/3, r/4, 0, -180);
    g.fillOval(x-r/2, y-r/3, r/5, r/5);
    g.fillOval(x+3*r/10, y-r/3, r/5, r/5);
    g.setColor(Color.yellow);
    g.fillArc(x-r/3, y+r/5, 2*r/3, r/4, 0, -180);
}
```

```
public void paint (Graphics g) {
 int i;

 final int radius = 25;

 for (i=1; i<5; i++) {
    drawSmiley(g, (i*i+2)*radius, (i*i+2)*radius, i*radius);
 }
}
```


Love 'n Peace

Step Aside Pixar...

Animation involves

- a sequence of similar images (use `Graphics g`)
- displayed at suitable intervals (use `paint()` and `Thread.sleep()`)
- a few million pounds to make it look good (err ...)

```
public void paint (Graphics g) {
  g.setColor(getBackground());
  g.fillRect(0, 0, 600, 600);
  drawSmiley(g, x, y, radius);
}
public static void main (String[] args)
                          throws java.lang.InterruptedException
  SmileyAnim s = new SmileyAnim (600, 600);
  s.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  Thread.sleep(2000);
  while (x<600-radius-10) {
    x++; Thread.sleep(1); s.repaint();
  }
}
```