Appendix G : repaint(), paint() and update()

Many Java programmers are befuddled by the three methods `repaint()`, `paint(Graphics)`, and `update(Graphics)`. This is because they are designed to work in a wide variety of circumstances, and they interact in a non-obvious fashion. This happens in several contexts in Java, but GUIs are the most obvious. The designers of Java wanted Java programs to be able to run on any machine that had a Java VM. So, a particular program might be running on a desktop machine, or a laptop, or a hand-held machine, like a personal assistant or a phone. This presents quite a challenge for the designer of a abstract windows toolkit (AWT). It also makes the job of a novice programmer more difficult than it might otherwise be. So it goes.

i) public void update(Graphics)

By default update(Graphics) fills the drawable area of a Component with its background color, and then sends paint(Graphics) to the object. Thus, flicker that comes from redrawing the background over and over, can sometimes be fixed by overriding update() (see Code Example 9 on page 238).

ii) public void paint(Graphics)

Every Java Component implements `paint(Graphics)`, which is responsible for painting that component in the Graphics context passed in the parameter. When you extend a Component (like when you write a Applet), if you want to display it differently than its superclass, you override `public void paint(Graphics)`. This was first illustrated in Chapter 3.

iii) repaint()

The `repaint()` method is sent to a Component when it needs to be repainted. This happens when a window is moved, or resized, or unhidden. It also happens when a webpage contains an image and the pixels of the image are arriving slowly down the wire.

When a Container, like a Frame, is painted, all of its Components (Buttons, TextFields, whatever) must be repainted. This is accomplished (roughly), in Java, by sending repaint() to every Component in the Container, in the order they were added to the container.

The action of `repaint()` is to spawn a new Thread (see Chapter 9), which schedules `update(Graphics)` in 100 milliseconds. If another `repaint()` happens before the 100 milliseconds elapses, the previous `update()` is cancelled (since screen flicker is ugly and refreshing over and over (like for every line in an image coming down down from the Net) looks horrible), and a new one is scheduled.

iv) paint(Graphics) or repaint()?

When you wish to redisplay a component, should you send it `paint(Graphics)` or `repaint()`? The answer is almost always `repaint()`. Only use `paint(Graphics)` if you understand what you are doing and have a good reason.