Background

These notes describe a suite of (small) Java programs which are used to illustrate some threading/concurrency issues. They assume that you already have had a general introduction to java.

Many of the examples are used to expose and then investigate typical bugs/errors. These are followed up in lectures/seminars along with various issues that we will come across as a result. Some of the bugs/errors are as a result of small, code-level faults; some from OO-level problems. Examine the code carefully, be critical.

The examples are accessed through a couple of programs which allow you to select which example to run. The structure of the front end program used in this suite of examples is different to that used in the next suite which is different to the one after that, etc. There are good & bad points about each of them – appraise them, be critical.

The descriptions below are meant to be brief reminders – this is not an independent learning resource. In particular where you see [*1] – that means you will need to turn up to a lecture/seminar for a proper explanation.

BasicMotion

(primary program BasicMotion.java)

We start investigating threading using an animated ball (ok, big over-statement – a moving blob), the first few example programs consider various ways to move the blob.

the basics

most of the moving of the blob is based around some kind of simple loop structure where (i) the blob is a filled oval (ii) the x coordinate of the blob is incremented through the loop (iii) some kind of delay – using `Defs.snooze(` *`milli-secs`* `)`

```
for( int x=50; x < frameWidth-50; x += step )
{       g.fillOval( x, y, size, size );
        Defs.snooze( 200 );
}
```

The next sections give a brief explanation of each of the (sub)programs used.

GraphicInConstructor

Simple approach where the redraw-blob loop is called directly from the JFrame constructor circumventing all the normal Java mechanisms for handling windows & window updating. Redrawing the blob in the constructor causes various problems...

1. the blob does not animate because we are not erasing/removing blobs draw before – this issue is not thread-related but we will deal with it soon;
2. you cannot interrupt the drawing (ie: close it before it has finished) – this *is* a threading problem, we will sort this out once we get on to threads;
3. once the drawing has finished the screen goes blank – this can happen if we do drawing in the constructor [*1].

GraphicInPaintLoop1

The redraw loop is in the paint method which is a better idea than doing it in the constructor. Compared to the last version issues are...

1. blob does not animate because we are not erasing/removing – still a problem but have not bothered to fix it yet;
2. you cannot interrupt the drawing – still a problem because have not got into threads yet;
3. once the drawing has finished the screen goes blank – sorted by using paint method rather than constructor.

GraphicInPaintLoop2

Redraw loop still in paint method. Compared to the previous version...

1. blob now animates (but the code for the for-loop is clunky enough to be error prone & not immediately obvious to some programmers who have not worked on these kinds of problem before;
2. still cannot interrupt the drawing – not into threads yet;
3. screen doesn't go blank anymore but the animation happens twice [*1].

GraphicInPaintLoop3

Redrawing approach similar to the last example but...

1. program is better structured by splitting some of its function into additional methods – a downside to this is the increased need for class level (global-ish) variables;
2. we use *Graphics.clearRect* to erase the blob (there is not a *clearOval*).

GraphicViaRepaintCalls

Redraw loop now indirects through calls to repaint. It is worth having some discussion about this [*1] because the use of repaint is consistent with Java's way of doing things but the apparent recursing through paint-repaint is not such a good idea. Main points...

1. graphic only happens once now but...
2. it all flickers a bit [*1]
3. the redrawing can now be interrupted – why?

RedrawInBusyPaintLoop

This is almost the same as the previous version but there are some extra graphics included to emphasise the flickering – it makes the current approach doubtful as a long-term solution. Issues...

1. everything (including the background) flickers unacceptably.

XOR Mode

XOR mode graphics paint by logically alternating the background color with the selected foreground color.

The following examples all exhibit some problems (i) the drawing cannot be interrupted (ii) the window is blank for a while before drawing starts. The 'animation' is smoother though.

UsingXORsimple

Graphics look better, the flicker has gone. There is a delay before drawing starts, after that the graphics mostly seem to work ok but still a bit jumpy (this is easily fixed with next example).

UsingXORsimpleSmooth

Smooth moving version of previous example. Blob moves shorter distances each time (just a couple of pixels) & the delay is shortened so the overall movement does not appear to slow.

NB: this program only differs from the last by changing a couple of variable values.

UsingXORmultiBall 1, 2 & 3

Three different versions on using XOR mode with multiple blobs – only the draw loop is different (& mostly the differences are small syntactically) – but the effects are significant (the last one is included because it has been suggested as a solution a couple of times, it isn’t a solution but it is worth looking at).