

Why does the Internet Explorer animated logo arrange its frame vertically?



Raymond Chen

If you ever tried to build a custom animated logo for Internet Explorer, you certainly noticed that the frames of the animation are arranged vertically rather than horizontally. Why is that?

Because it's much more efficient.

Recall that bitmaps are stored as a series of rows of pixels. In other words, if you number the pixels of a bitmap like this:

123
456
789

then the pixels are stored in memory in the order 123456789. (Note: I'm assuming a top-down bitmap, but the same principle applies to bottom-up bitmaps.) Now observe what happens if you store your animation strip horizontally:

12	34	56	78
AB	CD	EF	GH

These pixels are stored in memory in the order 12345678ABCDEF GH. To draw the first frame requires pixels 1, 2, A and B. The second frame takes 3, 4, C, and D. And so on. Observe that the pixels required for each frame are not contiguous in memory. This means that they occupy different cache lines at least, and for a bitmap of any significant size, they also span multiple memory pages.

Now consider a vertically-arranged animation strip:

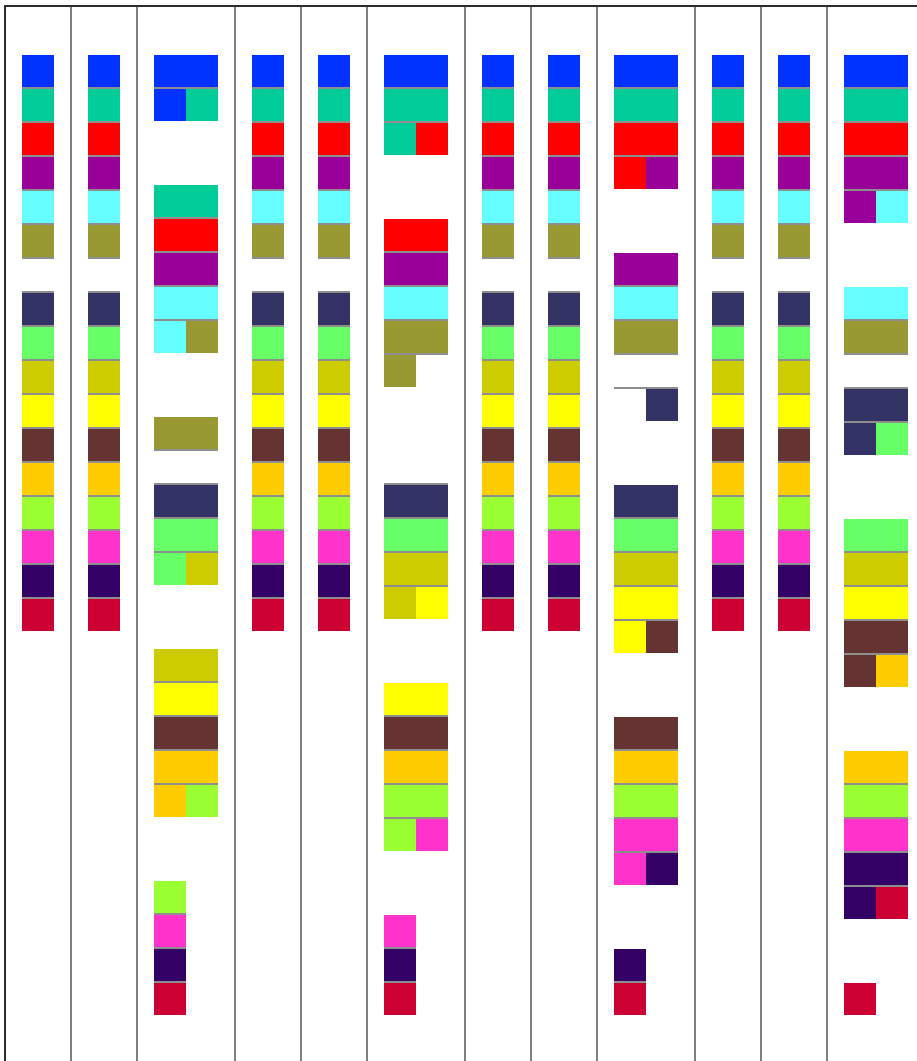
AB
CD

EF
GH

Again, the pixels are stored in memory in the order 12345678ABCDEFGH, [typo fixed, 15 Aug] but this time, the pixels of the first frame are 1, 2, 3 and 4; the second frame consists of 5, 6, 7, and 8; and so on. This time, all the pixels for a single frame are adjacent in memory. This means that they can be packed into a small number of cache lines, and reading the pixels for a single image will not force you to jump across multiple pages.

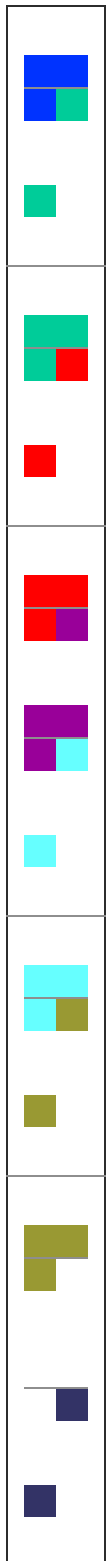
Let's illustrate with some pictures: Let's say that the large animation is a series of twelve 38x38 frames, for a total bitmap dimension of 38x456. Let's assume further, for the sake of example, that it's a 32bpp bitmap and that the page size is 4KB.

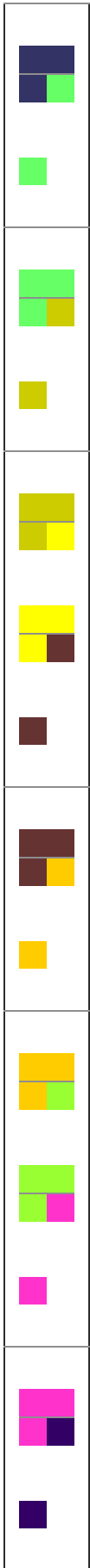
If the bitmap were stored as a horizontal strip (456x38), then the memory layout would look like this, where I've color-coded each memory page.

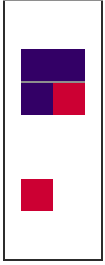


Observe that no matter which frame you draw, you will have to touch every single page since each frame contains a few bytes from each page.

Storing the bitmap vertically, on the other hand, arranges the pixels like so:







Notice that with the vertical strip, each frame touches only two or three pages; compare the horizontal strip, where each frame touches seventeen pages. This is quite a savings especially when you realize that most of the time, the only frame being drawn is the first one. The other frames are used only during animation. In other words, this simple change trimmed 60KB out of the normal working set.

Raymond Chen

Follow

