

Re: Great SWT Program

Source: <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2007-10/msg03298.html>

- *From:* nebulous99@xxxxxxxxxx
 - *Date:* Sat, 27 Oct 2007 02:38:52 -0000
-

On Oct 26, 9:04 am, blm...@xxxxxxxxxxxxxx <blm...@xxxxxxxxxxxxxx> wrote:

The ones that involve more typing and might be interrupted --
I can't think of any that don't show you what you're doing as
you're doing it.

That's better than I expected of it, but it must gobble precious
screen real-estate. Devoting one line to some sort of command line
means losing about 4% of the total screen area from being available
for document editing. (A GUI app would produce a popup with an input
field, only when needed, instead of cluttering up the window's
interior with always-present widgets you are 99.7% of the time not
using.)

I can't quite imagine why being able to display application-produced
text in the user's language of choice is a feature of GUIs and not
text-mode applications. Remember that discussion of how the old
text-mode applications managed to work with a variety of terminals,
without themselves including the specifics of what particular
terminals needed in the way of escape sequences? Something
similar could be done with whatever text the application needed
to display, no?

Perhaps, but that would be like putting life vests with parachutes
into the seatbacks on an Edsel. Since it won't be making
transcontinental flights at 38,000 feet, they'd be somewhat
superfluous there, I think you'll agree.

What it means is that mouse support is useless on software designed
for a text terminal environment.

Uh-huh. So, the fact that someone new to vim can use the mouse to
select and paste text in a way that will seem fairly familiar --
useless? "Whatever", I guess.

Re: Great SWT Program

It's not putting mouse support into an editor running on modern hardware that bothers me. It's putting vi into an editor running on modern hardware that bothers me. It's like trying to build a transcontinental jet by modifying that Edsel instead of designing it from scratch as a jet. :P

"Got me" how? What do you want here? Do you want me to agree that the tools I like are out of date, and I'm an idiot for sticking with them? or do you just want me to shut up and allow your [insult deleted] claims to stand

I don't know, actually. But those tools seem to be ... not so much out of date, as half-out-of-date and half not. Someone didn't merely keep their 1960s-era wide-body Ford with the tail fins. They didn't even merely keep it and lovingly restore it to factory-new condition. They didn't even merely do all this and *actually drive it as their main vehicle for getting around in*. They embarked on a program to try to convert it for spaceflight and have the curious notion that it might actually fly them to the moon!

Only one issue with the analogy: that 1960s Ford didn't require hand-cranking to start the engine.

Or something like that.

People keep rabbiting on about this mythical power that lurks in the keyboard and is only accessible by mastering the magical meta key somehow, and is somehow impossible to implement in a modern UI.

One more time: No one is telling you that "modern" UIs *can't* provide good support for people who prefer to use the keyboard. What more than one person has said is that the ones we know about *don't*.

Maybe the problem is the "prefer to use the keyboard" bit. Use the mouse. Love the mouse. Then, and only then, can you be as productive in a GUI (with a minimum of effort) as you *maybe* can *eventually* be with things like vi (and six years of intensive training and experience). Insist on using only the keyboard and the GUI will be slow and awkward. It's not a fair comparison, though. GUIs weren't designed with that pattern of use in mind.

Strangely enough, that "small number of orthogonalized features that can be used in a large number of combinations" — that's

Re: Great SWT Program

more or less what I like about *my* tools. You will probably disagree, but since you don't seem to know very much about these tools, how can you be sure?

Well, there's three approaches to user interface design and command orthogonalization, it seems. Compare them to street layouts:

Emacs -- Lots of narrow alleys and oddly angled streets and intersections, with dead ends, cul-de-sacs, and one-way streets, like your typical old downtown area. If you memorize the map you can find your way around and there's one or two particular places with shortcuts between them, but if you're new to the town, God help you. Oh, and did I mention there are no street signs?

Modern GUI -- A nice, simple grid. Even a newcomer can quickly figure out how to get to places like 1st and E street. Clear signage anyway.

Vi -- A grid, but it's nighttime and the streetlights don't work. Neither does your car (mouse), so you have to walk (hjkl) or ride the bus (jump around using search).

Vim -- car is fixed, except for the headlights. Also, street signs have been posted. Now if you could just read in the dark...

That you're describing something highly implausible, and if it actually does work as advertised, it probably does so only for people with memorization and related skills bordering on savantism.

Whether something is possible and/or true seems to me to be a separate issue from whether "normal" people (whatever that means) can use it easily.

If something is possible in the sense that memorizing a whole phone book is possible, I don't see the fact that it's *technically* possible as necessarily having all that much *practical* value.

Strange. I don't recall any such matter arising in and therefore being relevant to this particular thread.

So, we're not allowed to mention content from one thread in another? Okay. New rule to me!

Well, isn't this thread OT enough, without confusingly changing the subject further inside it?

Re: Great SWT Program

Re: Great SWT Program

I do too. Why, did you think I had to think about where all the keys were for some reason?

Why not? You seem to think that's what I do.

When using them for something unnatural, yes.

That does not make sense. There's clearly an extra translation step involved, and that has to have a cost in cycles and memory use. You just may not be consciously aware of it except when you're discussing the subject itself at the meta-level as we are here and therefore are consciously paying attention to it in a way that you normally don't.

I'm not sure how we could design an experiment that would settle this one way or another. What you seem to be saying here is that you know better than I do what's going on in my head.

It's a guess, but given that you had to think for a bit to find the key to press when actually discussing it, I'm guessing you have to whenever, and the fact that you were actually discussing it merely made you consciously aware of the fact where you otherwise weren't. Since it makes more sense than the alternative, which is that hitting your "up" key requires more thinking when you're discussing "up" keys than simply when you're editing text and moving up in it. Why would it be context sensitive how much thought is required in translating the imperative to "hit the up key" into a motor action?

I guess that's possible, but — wouldn't you reject a claim that I know better than you do what's going on in your head?

If the claim was false, or at least highly implausible.

OK, OK, time out.

Just how many fucking modes does this monstrosity have, anyway? :P

None, as far as I know. There's an insert mode, and command mode,

Re: Great SWT Program

and none for the activity you mention.

Cute. (I wonder how many emacs has?)

Must be a pain when it comes time to de-indent a section then. Eight times as many del/bksp keypresses per line. Ouch!

That would indeed be a pain. I'm glad I don't have to do it.
Didn't I explain that later in the post to which you're replying

Didn't I explain that this whole thing is "you don't like tabs" followed by assorted awkward kludgy workarounds for using spaces when the logical structure of what you're editing would use tabs? Bandaging your foot because you shot it and then limping everywhere and blaming the limp on having a hole in your foot without acknowledging that it was self-inflicted seems a bit ... odd. OTOH, I'm not sure that what's going on isn't more analogous to not even admitting that the limp exists, perhaps even to yourself ... hmm ...

Well, no, because I don't edit makefiles with the "turn tab characters into spaces" feature on.

And you never forget to toggle it?
Never, ever?

Not even just once?

This is all much easier than you seem to be imagining,
but I'm getting tired of providing long explanations that don't seem to have any result except more confusion and argument.

Nothing about this sounds "easy". It sounds like the software equivalent of having to use a patch board instead of a keyboard to enter data.

Of course, if you have myLongFileName001.jpg through myLongFileName999.jpg it won't know which of these you mean ... saving the ".jpg" by reducing it from four keystrokes to one isn't a big gain percentage-wise in this instance. :)

Again, the feature doesn't work the way you seem to be imagining.

Re: Great SWT Program

Admittedly I didn't spell out all the details. In the scenario you describe, pressing tab would fill in as much of the filename as possible, turning "myL" into "myLongFileName" and leaving one in a state in which one could then type the rest of the name, and/or press tab again to get a list of choices.

So it has a special case for every file in the directory having the same long prefix. Not much use when many of them do, and many don't, e.g. there's also a myLisaBoylePictureArchive.zip in there. :)

Except that you do. You have eight spaces to delete instead of one tab character. A child can do this math correctly!

A child who doesn't realize that one can delete more than one character at a time.

So you can bind some key to "delete eight characters at a time" or something? What a crafty workaround.

What do you have against tabs, anyway?

There are several ways in vim, and emacs if I remember right, to delete multiple characters with fewer keypresses than there are characters. vim also allows repeating the previous command with one keypress (where the previous command could be "delete eight characters starting at the current cursor position").

Sounds dangerous, especially when applied to anything with the word "delete" in its description.

(I think there actually are several faster ways to do this in vim, but apparently I can't be bothered to learn the proper incantations. I don't seem to need this feature much these days, since vim will do automatic reindentation of code.)

Feep! Feep! IDE feature detected in "no-frills" text editor! Feep! Feep!

The closest thing I can think of in Windows editors is the ability to select text to delete using the mouse, or using keys such as Home and End to help select text. I don't know how much this

Re: Great SWT Program

helps you in making a lot of similar changes (meaning "I don't know", not "I think it's not possible").

Well, manually unindenting a few lines is as simple for me as up, del, up, del...

Manually unindenting them with spaces instead of tabs would be nastier. Unindenting one is as simple as getting to the eighth character and hitting shift-home delete, but the repositioning to the eighth character on the next line is the spoiler. Just another reason to use real tabs, or for that matter a real IDE with autoindent functionality.

(1) Use search-and-replace to remove a selected number of spaces from the start of each line, applied to a selected range of lines.

That's not possible.

Right. Something I do routinely isn't possible. I'm lying? Deluded?

You expect me to just believe everything you say, even when it's the equivalent of "I've invented a perpetual motion machine!" or "I spent the week at Alpha Centauri — took two whole days to get there and two to get back so I was gone for 11 days total"?

Didn't I previously mention being able to insert text at the beginning of each of a selected range of lines, using a regular expression that matches "start of line"? Well, perhaps it's not obvious from that discussion that it would also be easy to delete eight spaces at the start of each of a selected range of lines.

"Easy"? It can't possibly be, because that isn't a query you can possibly simply type in literally. You'd have to specify it as some sort of code, which basically means writing a short, novel program. That program will contain a bug in its first draft. And then you turn it loose deleting things ...

Perhaps it would help clarify things [*] to mention that I'm not talking here about a combined search / search-and-replace feature like what you get when you press control-F in a typical

Re: Great SWT Program

Windows application. Search and search-and-replace are distinct things in vim (and emacs).

How cruffy.

[*] Or not. I don't seem to be having much success here at communicating a sense of how this application actually works, as opposed to how you seem to think it works.

A textual medium does not in general do justice to attempts to convey the detailed functioning of Rube Goldberg apparatuses.

Use the what? Control-I is 0x09, the tab character. Are you saying you have it set up to convert tab keypresses into spaces, then do an end-run around this to insert literal tabs anyway? This seems to be even more seriously broken behavior, and this time it's not software behavior that violates least surprise I'm talking about, either, but behavior of the component that goes between the keyboard and the chair...

NO — I'm saying that vim and emacs have a feature comparable to what happens in Eclipse when you select some text and then press control-I. I didn't mean to suggest that this functionality was literally invoked by pressing control-I. (It's not.)

Oh. Of course it's not. Anything that inserts a literal tab in one of these unix tools will be bound to some key other than control-I, because making it so that control-I (and thus the tab key) did it would be making things too easy, wouldn't it? :) So tab (and control-I, both produce ASCII 0x09 after all) produces eight spaces and something else produces tabs, of course; perfectly logical. :)

No-one else who looked at the code would have to do anything of the sort — everyone could use their own preferred indent-per-tab when viewing and editing, which is another argument strongly in favor of real tabs.

Well, except that in my experience that doesn't always work very well, for example if people use tabs to position inline comments at the right of a line of code. <shrug>

I never do that. If I had two tabs and 31 characters of code, a space,

Re: Great SWT Program

\\, and a comment, I'd continue it on the next line with two tabs, 32 spaces, \\, and the continuation.

As best I know, there's not a global setting that means "show all tabs in text files as four spaces". I'm not sure how useful it would be. <shrug>

Strange, considering how they architected the rest of the system.

Are you *trying* to make this more confrontational? Why? Or am I meant to interpret this as humor, or -- well, I hear that men are able to have heated discussions about technical topics without taking it personally, and maybe that's what this is meant to be. <shrug>

Well, I am getting tired of your only rarely and grudgingly admitting that I might be right about GUI tools being (with appropriate, not too hard to acquire proficiency) better to work with ... :)

And (this may offend) sometimes it seems to me that women are unable to have discussions about anything without getting emotionally invested. :P :)

a) All that hand-hacking of configuration files does take extra effort and translates into a reduced willingness to use non-default settings, even when they'd only have to be set once per app.

Not for me.

Your own lack of desire to configure all your stuff to four-character tabs seems to belie that claim.

Heck, it's not like you'd have to configure anything but vi anyway, since the one editor gets used for everything on unix, you said. There's no global tab-width setting because the editor you use has one that should amount to global as far as you're concerned anyway.

I could perfectly well have a nice wide 200-character-wide text box to edit in, if I wanted one. I don't, though, mostly because I don't have a wide enough monitor to display 200 characters in a font I'd find readable.

Re: Great SWT Program

Hrm ... unusually narrow monitor or terrible eyesight?

And I seem to remember reading somewhere that humans actually find shorter line lengths more readable — isn't there a reason newspapers, magazines, etc., are printed in columns?

There's undoubtedly a reason, yes. What it is is anybody's guess.

Regardless, I find lines of code to be far more readable when they DON'T wrap. I'd take a single 150-character long line over

```
int foo (boolean quux, PogglegigglerIterator pwi,  
int[] fooArray, String description,  
List<Arsehole> whoToFlameNext, Nut  
lastInsultSource, Fruit bumbleSnatch) {
```

Especially if forced to work in an environment where the latter covers a full 1/6 of the screen height.

Yes, that **is** nice. Fortunately, vim and emacs provide similar functionality.

Feep! Feep! More IDE-specific features in a supposedly general-purpose text editor! Feep! Feep!

Nice for writing HTML too. And LaTeX source. And other things where some notion of indenting things makes sense.

But not plain English prose, such as email and news posts.

And there's no one-size-fits-all for those anyway. HTML, LaTeX, and Algol-derived languages are indented using three different cues; indenting based on { } nesting will only be appropriate for the third, for instance. So no matter what the editor uses to decide, it will get it wrong for at least two of those three. Specialized IDEs for each of the three will, however, be able to apply one that's sensible for their particular input, because it's known at programming-time what that input will be (e.g. Java source when programming a Java IDE).

"The set ... you're accustomed to and like". Perhaps they provide a set, but it's as alien to you as the ONLY set of features provided by

Re: Great SWT Program

your tools are to me?

Perhaps they do. Perhaps overall I would like your preferred platform better, once I got past the learning curve. I've learned, in this thread, about a few things I didn't realize were possible. (That would seem to counter your claims about your platform being more intuitive, but perhaps you didn't actually claim that *everything* about it would be obvious to the beginner.)

No, but enough to actually get by while learning the rest, which is a distinct improvement over "the beginner will be forever bumping into heavy furniture in the dark, unable to make any progress whatsoever". :)

Then again, based on this discussion it seems to me that you're as unfamiliar with my preferred platform as I am with yours. So it's hard to compare their features. Maybe we're about done here

Well, I'd say it's like comparing apples and oranges, but I rather suspect it's more like comparing something inedible and covered in prickly thorns with something like fine Belgian chocolate. Only in the case of Windoze, the chocolate's been left in the sun too long, and in the case of the Mac, some genius in the factory misread the recipe and put a quart of sugar into a mixing vat rather than an ounce.

.