

riffing with acme

Sevki
s@sevki.org

ABSTRACT

adding keyboard support to an editor that will not want one



I get asked "what is that weird arse editor you're using?" by people .I a lot bespoke^{usea}

Developers tend to argue about stylistic choices quite often; things like tabs vs spaces, snake case vs camel case (python vs perl??), requiring semi-colons/trailing commas vs no semicolons/trailing commas... things that don't impact function, just form. This is because most developers suffer from a condition that requires them to a) always be right and b) do the least amount of work possible, which as we all know the is to offer unsolicited advice and be shitty when the advised doesn't immediately fall in love with the advice they didn't ask for.

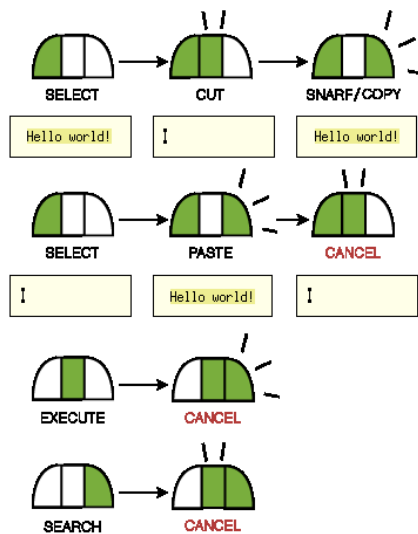
How Acme got it right but got it wrong

The advent of windowed terminals has given each user what amounts to an array of teletypes, a limited and unimaginative use of the powers of bitmap displays and mice. Systems like the Macintosh that do involve

the mouse as an integral part of the interaction are geared towards general users, not experts, and certainly not programmers. Software developers, at least on time-sharing systems, have been left behind.

source: acme

At the time it was introduced acme went all in on the mouse, not any old mouse a very specific 3 button old mouse, which was the ubiquitous pointing device.



source: acme mouse

The reason I put such a clickbaity heading at the beginning of this section is because, I can take the statement above, update a couple of technologies with more current ones and the general statement still holds true.

The advent of "cloud computing" has given each user what amounts to an array of mobile devices, a limited and unimaginative use of the powers of infinite compute and multi-touch displays. Systems like the iPad that do involve the multitouch as an integral part of the interaction are geared towards general users, not experts, and certainly not programmers. Software developers, at least on traditional desktop systems, have been left behind.

So I would suggest what acme didn't get quite get right was the specific input device type, but exploring a alternative input device was I think the right call.

At the time, there was really one pointing device and had Rob Pike predicted multitouch would be this prevalent, I suspect he would have retired to a private island in the caribbean.

The work I would like to do is removing device specific input methods from Acme. If keyboard to mouse was a paradigm shift, mouse to multitouch should be another one.

Riff

Namesake of this blogpost refers to a psuedo programming language/instruction set that I affectionately call **riff Riff** seemed like the next logical step. In the context of acme a **riff** refers to "brief, relaxed chords repeated over changing scales".

```
/mnt/acme
|-- 1
|   |-- addr
|   |-- body
|   |-- ctl
|   |-- data
|   |-- editout
|   |-- errors
|   |-- event
|   |-- rdssel
|   |-- tag
|   |-- wrssel
|   `-- xdata
|-- acme
|-- cons
|-- consctl
|-- draw
|-- editout
|-- index
|-- label
|-- log
|-- new
|   |-- addr
|   |-- body
|   |-- ctl
|   |-- data
|   |-- editout
|   |-- errors
|   |-- event
|   |-- rdssel
|   |-- tag
|   |-- wrssel
|   `-- xdata
```