Motivation (extended)

tl;dr: QWERTY is superior in issuing commands using letter keys, and it is not that far inferior in typing prose. However, it falls short in typing symbols. How do we address this? By using QWERTY's letters, but providing it an easier way to type symbols, such as through a symbols layer like the one provided by `3l-symbols`.

Maybe someone cares. Maybe I like talking. Regardless, here is why I think you should use `3l-symbols` over any other layout:

Why do we type? We type in order to make our thoughts reality on our computer, without having to consciously think about how we make it happen. This can happen in one of two ways: by recording information, such as by typing up an email to send, writing code, or typing a URL in the browser URL bar; or by issuing some kind of command to interact with previously recorded information, such as by sending that email, navigating that code, or copying and pasting that URL.

So how do different layouts stack up at these various tasks? Let's look at recording information first. We can look at two basic kinds of information: prose information, such as class notes or an essay for a class–this includes lots of words and basic punctuation (periods and commas) and few other symbols; and code, which I'm using as a broad category that also encompasses other symbol-heavy documents like LaTeXsource–code includes lots of symbols of various types.

For prose information, QWERTY is acutally basically fine. It doesn't really feel bad unless you type a whole bunch. It is perfectly fast with practice. This may be anecdotal, but I have never met anyone who was competent in QWERTY who is significantly faster at typing regular prose in any alternative layout. Things like finger travel distance or raw speed are often poor metrics anyway because they do not reflect the behavior of many actual typists, who commonly do things like taking pauses to think or moving their fingers around anyway, independent of what is theoretically optimal. While alternative layouts may feel better, I have not seen any evidence that they significantly improve actual performance in the real world. My verdict here is that while QWERTY is not the best, it is not as bad as it is made out to be.

QWERTY is significantly worse than alternative layouts for code. I can't deny this. It is physically painful to type LaTeXsource, with all its backslashes, in QWERTY. Other symbols, such as the parentheses and various types of brackets, are also excessively cumbersome for characters that see more use than many actual letters inside code. Backspacing is also a point of difficulty for typing in QWERTY, in both prose and code. I definitely would prefer an alternative to searching around the top row or the right-pinkie wasteland for all of these keys.

So now let's look at issuing commands. There are two factors in play here: muscle memory of other keyboard layouts and programs/hotkeys designed specifically

for the QWERTY layout.

Muscle memory of QWERTY bindings is actually why I stopped trying to learn `3l`. I could type prose and symbols fine, but it was incredibly difficult for me to issue the correct commands in my text editor, which I have configured to use vim keybindings. I don't actually think about what keys I use to accomplish things–I just press the keys I know will give me the result I want. That means that to switch to `3l`, I had to stop, think about what key I wanted, think about where that key was in `3l`, and then press that key. This was very hard, because I issue a lot of editor commands and issuing them in this way was very slow. While this is not inherently problematic for those who are determined enough to switch or for those who are not extensively familiar with the keystrokes for their editor, it makes learning any alternative layout much more difficult for those of us who do know our editor, and adds a big learning curve that would otherwise not need to exist. Fundamentally, until the keybindings are re-memorized, it breaks the idea that we type to quickly and simply convert our thoughts into reality.

The other thing alternative layouts are up against here is that programs are designed assuming the user will be on a QWERTY layout. For example, the universal copy/paste shortcuts, C^x, C^c, and C^v are all in a row for the same hand on QWERTY, and vim's basic navigation keys are hjkl, which form a line on home row for the same hand in QWERTY. Game controls are even more extreme in their adherence to QWERTY. This means that things that are conceptually easy and convenient to type in QWERTY, such as navigating a file in vim or copying and pasting a selection by holding down Ctrl and moving one finger across the bottom row, while the other hand is on the mouse selecting the text in question, become extremely confusing and/or inconvenient in most alternative layouts (Colemak maybe excepted). You have to have both hands on the keyboard when it doesn't otherwise make sense, you have to move your hand down to scroll up, or other problems occur. Additionally, many programs place these commands in such a way that they are convenient to use, despite the fact that the keys they are on are uncommon letters. This means that these keys get shunted off to remote corners of the keyboard in alternative layouts, meaning that simple things like changing which line of a file we're on are hard to do because they are now associated with inconvenient keys.

You might think that you could rebind your keys to solve this, but think for a moment how many config files you would have to change to accomplish that. Your editor, your browser, yes. But also, depending what kinds of things you use special keybindings for, you may also want to modify the configurations for your pager, your word processor, your shell, your file manager, and so on. I wouldn't want to touch it.

You might also think that you could just get software companies to support your layout, but I have a couple of sad facts for you:

- Dvorak has been around since 1930 and you don't really see a whole lot of

support for it yet.

- Linux has been around since 1991 and you're only starting to see any kind of widespread support for it from any software company whose developers do not themselves use Linux. Linux is not itself a keyboard layout, but more people use Linux than any alternative keyboard layout, and they haven't been very effective yet.

Especially if your alternative layout of choice is not Dvorak, what chance do you think your layout really stands of being supported?

So ultimately, QWERTY is superior in issuing commands using letter keys, and it is not that far inferior in typing prose. However, it falls short in typing symbols. How do we address this? By using QWERTY's letters, but providing it an easier way to type symbols, such as through a symbols layer like the one provided by `3l-symbols`.