

computers are bad

You are receiving this facsimile because you signed up for fax delivery of this newsletter. To stop delivery, contact Computers Are Bad by email or fax.

<https://computer.rip> - me@computer.rip - fax: +1 (505) 926-5492

2021-06-07 building the first guis

The modern GUI, as we understand it, can be attributed almost entirely to the work of Douglas Engelbart.

Engelbart

In fact, it is rather surprising to me that so much can be attributed to one person. I have said before that the technology industry moved so quickly that nearly every significant innovation can be attributed to multiple, parallel efforts. In fact this is probably true of the GUI, but any parallel efforts have been deeply forgotten in comparison to Engelbart's pioneering work.

In 1968, Engelbart presented to a conference a demonstration of a project he had built while at SRI. Generally based on Vannevar Bush's [1] conceptual design for the "memex," Engelbart's effort put together nearly all of the major aspects of a modern GUI system. There was a mouse, there were windows, buttons, hyperlinks, menus, everything you could want. The GUI, to a remarkable degree, was just invented all at once.

Of course Engelbart was not precognizant. He made a number of missteps, many of which would be repeated by the XPARC work on the Alto which was closely based on Engelbart's demonstration. Most amusingly, Engelbart found it unlikely that computer users would want to use a mouse with one hand when the keyboard requires both. As a solution he proposed (and used) a one-handed, chord-based keyboard. Despite the best efforts of many dweebs, one handed text entry has never caught on [2].

More profoundly, though, Engelbart failed to anticipate the complete lack of interest in actually implementing the concepts he demonstrated. Despite the amazing impact of his demonstration on the audience, the technology was complex and difficult to build, and bore little resemblance to the text-mode, command-oriented environment which was the respected norm in business computing.

Engelbart invented the modern GUI in 1968. It would not be available on the market until 1981.

From our comfortable position today it is hard to imagine how this could be. GUIs seem to be the obvious progression in computer interfaces. Yet, during Engelbart's work his vision was regarded as largely academic, not practical. GUIs as a concept were closely tied to cybernetics and artificial intelligence, fields which attracted a

great deal of graduate students but very few actual users. The GUI was *cool*, it was *interesting*, but it was not practical.

This situation is perhaps most exemplified by Smalltalk. Smalltalk was developed at XPARC (that's the Xerox Palo Alto Research Center) as a teaching language, and was best known for being an early object-oriented language and for its frequent implementation in highly GUI-centric virtual machines. Major implementations like Squeak couple Smalltalk with graphical development and debugging environments which are surprisingly cutting edge, and yet completely unused.

You see, Smalltalk, despite its innovations, has basically always been constrained to academia. Most CS students are exposed to Smalltalk at some point (probably in a programming language theory course), but no one actually uses it for anything. The situation was largely the same for all graphical environments through the course of the '70s and to a good degree into the '80s.

Many new technologies fall into this trap to some degree, being the subject of a great deal of excited research but never bridging the gap into wide-scale implementation. For example, basically the entire field of computer usability.

Apple

What unstuck the GUI and pushed it into the world of industry? Basically Steve Jobs, although he too suffered a few false starts. The Lisa was technically advanced but a commercial failure, the Macintosh was a commercial success but relatively primitive. Nonetheless, the Macintosh was essentially the next major step from Engelbart's demo, and it established many of the norms for GUIs for years to come.

The relative success of the Macintosh compared to the costly but significantly superior Lisa is a rather unfortunate situation. For the most part, the Lisa was the more innovative and capable machine. The Macintosh was essentially a compromise, stripping out the most interesting features of the Lisa to achieve a low price and more gentle learning process. To be quite honest, the Macintosh sucked, which is why we far more often talk about its various successors.

I will probably devote an entire post to this, because I want to do the topic justice and did not intend to take it on here. But the Lisa was a *document* interface, while the Macintosh was a *program* interface.

This is actually the same paradigm we discussed in a previous post, of functional vs object-oriented user interfaces. Graphical operating systems that we use today are nearly entirely functional, with the operating system's role fundamentally being the launching and management of programs. It might be hard to picture anything else. But most early GUI research actually *did* envision something else, a fully object-oriented interface that is nearly entirely structured around documents and data. The Lisa was document-oriented, and Microsoft made various efforts towards a document-oriented Windows experience. But document-oriented interfaces were ultimately unsuccessful, and none survive today [3].

Despite the disappointing compromise of the Macintosh, it set the trend for most GUI systems to follow. The Macintosh interface was WIMP (Windows, Icons, Menu, Pointer), it had drag-and-drop file management (although it opened a new window for every folder the user descended into, an especially irritating element of early GUI operating

systems that was fortunately cast off by the new millennium), and it used icons on a desktop as the primary entry point at menus at the top for access to commands.

Windows?

In the eyes of most, the next major step from the Macintosh was Microsoft Windows. Windows was introduced in its first version only a year after the Macintosh and a few years after the Lisa. Early releases of Windows, and to a degree all releases of Windows outside of NT, were simply applications which ran on top of DOS. This was a logical decision at the time, to build GUIs on top of a better established foundation, but it also imposed significant limitations.

In part as a result, the early versions of Windows were primitive and simply not that interesting. They were correspondingly unsuccessful, which is why you virtually never hear any mention of Windows 1.0 or 2.0.

The reason for the poor performance of Windows 1 and 2 is actually a surprisingly interesting and surprising one. It wasn't because Windows was inferior to the Macintosh; this was a factor to a degree but the Apple world was already highly differentiated from the PC world and the PC world had a formidable hold in the business world that ought to have conferred a big advantage on PC software.

It was more that early releases of Windows failed because they were inferior to *other DOS GUIs*.

The '80s PC world

Before we can get into the history of PC GUIs, we ought to devote some discussion to the context in which they were developed. Although IBM and others developed multiple operating systems for various generations of their personal computers, and thus for their many clones, by the '80s there was a high degree of consolidation on CP/M (for non-IBM small computers) and DOS (for IBM small computers and their clones). CP/M bears mentioning more so than other non-IBM operating systems of the time because, as a result of happenstance, CP/M was highly influential on the design of DOS which was intended to have a high degree of similarity to ease transition from one to the other.

We could almost say that DOS was a new version of CP/M, but the process was politically and technically rocky and various features of CP/M fell off the truck on the way to DOS. In the same way, some features of CP/M were carried into DOS even though they probably shouldn't have been. A number of DOS's oddities can be attributed to its origin as Microsoft Imitation CP/M Product.

So of the early non-Apple GUIs, most (but not all!) were intended to run on top of CP/M or DOS.

The thing is, CP/M and DOS were both primitive operating systems by modern standards. CP/M and DOS were not multi-tasking. They did not employ virtual memory, but instead addressed all memory directly. As a natural result of these two prior facts, they provided no isolation between running programs, and so the primitive "multitasking-like" behavior that could be implemented was very prone to problems.

If we were presented with this situation today, we might declare that development of a

GUI environment on top of these operating systems is simply impossible. And yet...

[1] If the name Vannevar Bush is familiar to you, there could be any number of reasons as he had a prominent career. Perhaps most notably, as director of the OSRD, he was a major figure in the early development of nuclear weapons.

[2] The obvious solution to this problem, of integrating the mouse *into* the keyboard, was popular on '90s laptops but is largely forgotten today. A small group of trackstick devotees have managed to keep them on “business” laptops, a great benefit to myself. I cannot imagine life without a trackstick mouse, the only civilized way to move the cursor with both hands on the home row.

[3] In fact, Apple launched several different independent GUI operating systems in a span of a few years in the early '80s, the Macintosh being the only one that survived. One day I will write about these.