

## 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](mailto:me@computer.rip) - fax: +1 (505) 926-5492

---

### 2021-06-19 The Visi On Vision

First, after lengthy research and development I have finally followed through on my original vision of making *Computers Are Bad* available via Gopher. Check it out at <gopher://waffle.tech/computer.rip>.

Let's talk a bit more about GUIs. I would like to begin by noting that I am intentionally keeping a somewhat narrow focus for this series of posts. While there were many interesting GUI projects across a range of early microcomputer platforms, I am focusing almost exclusively on those GUIs offered for CP/M and DOS. I am keeping this focus for two reasons: First, these are the microcomputer platforms I am personally most interested in. Second, I think the landscape of early CP/M and DOS GUIs are an important part of the history of Windows, because these are the GUIs with which Windows directly competed. A real portion of the failure of Windows 1 and 2 can be attributed to Microsoft's lackluster effort compared to independent software vendors--something quite surprising from the modern perspective of very close coupling between the OS and the GUI [1].

Let's talk, then, about my personal favorite GUI system, and one of the most significant examples of stretching the boundary between operating system and application by implementing basic system features on top of an OS that lacks them... but first, we need to take a step back to perhaps the vintage software I mention most often.

VisiCalc is, for most intents and purposes, the first spreadsheet. There were "spreadsheet-like" applications available well before VisiCalc, but they were generally non-interactive, using something like a compiled language for formulas and then updating data files offline. VisiCalc was the first on the market to display tabular data and allow the definition of formulas within cells, which were then automatically evaluated as the data they depended on changed. It was the first time that you could change one number in a spreadsheet and then watch all the others change in response.

This is, of course, generally regarded as the most powerful feature of a computer spreadsheet... because it allows for the use of a spreadsheet not just as a means of recording and calculation but as a means of *simulation*. You can punch in different numbers just to see what happens. For the most part, VisiCalc was the first time that computers allowed a user to "play with numbers" in a quick and easy way, and nearly overnight it became a standard practice in many fields of business and engineering.

Released in 1979, VisiCalc was one of the greatest innovations in the history of the computer. VisiCalc is widely discussed as being the "killer app" for PCs, responsible for the introduction of microcomputers to the business world which had formerly

eschewed them. I would go one further, by saying that VisiCalc was a killer app for the GUI as a concept. VisiCalc was one of the first programs to truly display the power of direct manipulation and object-oriented interface design, and it wasn't even graphical. It ran in text mode.

We have already, then, identified VisiCalc's creator Dan Bricklin and his company VisiCorp [2] as a pioneer of the GUI. It is no surprise, then, that this investment in the GUI goes beyond just the spreadsheet... and yet it would surprise many to hear that VisiCorp was also the creator of one of the first complete GUIs for DOS, one that was in many ways superior to GUIs developed well after.

By 1983, VisiCorp had expanded from spreadsheets to the broader world of what we would now refer to as productivity software. Alongside VisiCalc were VisiTrend/VisiPlot for regression and plotting [3], word processor VisiWord, spell checker VisiSpell, and proto-desktop database VisiFile. The problem was this: each of these software packages were fully independent, any interoperation (such as spell checking a document or plotting data) requiring saving, launching a new program, and opening.

Of course this was a hassle on a non-multitasking operating system, although multitasking within the scope of a user was sufficiently uncommon at the time that it was not necessarily an *extreme* limitation. Nonetheless, the tides were turning in the direction of integrated software suites that allowed simultaneous interoperation of programs. In order to do this effectively, a new paradigm for computer interface would be required.

In fact this idea of interoperation of productivity software is an important through-line in GUI software, with most productivity suite developers struggling with the same problem. It tended to lead to highly object-oriented, document-based, componentized software. Major examples of these efforts are the Apple Lisa (and the descendent OpenDoc framework) and Microsoft's OLE, as employed in Office. On the whole, none of these have been very successful, and this remains an unsolved problem in modern software. There is still a great deal of saving the output of one program to open in another. I will probably have a whole message on just this topic in the future.

In any case, VisiCorp realized that seamless interoperation of *Visi applications would require the ability to run multiple Visi applications easily, preferably simultaneously*. This required a GUI, and fortunately for VisiCorp, the GUI market was just beginning to truly take off.

In order to build a WIMP GUI there are certain fundamental complexities you must address. First, GUI environments are more or less synonymous with multitasking, and so there must be some type of process scheduling arrangement, which had been quite absent from DOS. Second, both multitasking and interprocess communication (which is nearly a requirement for a multitasking GUI) all but require virtual memory. Multitasking and virtual memory management are today considered core features of operating systems, but at this point in time they were unavailable on many operating systems and so anyone aiming for a windowed environment was responsible for implementing these themselves.

Released late 1983, VisiCorp's Visi On GUI environment featured both of these. Multitasking was not at all new and as far as I can tell Visi On multitasking was cooperative (it is very possible I am wrong on this point, it is hard to find a straight answer to this question), so the multitasking capability was not especially cutting edge. What was quite impressive is Visi On's implementation of virtual memory

complete with page swapping, which made it practical to have multiple applications running even if they were heavy applications like VisiCorp productivity tools.

Beyond its implementation of multitasking and virtual memory, Visi On was a graphics mode application (i.e. raster display) and supported a mouse. The mouse was used to operate a fundamentally WIMP UI with windows in frames, drop-down menus at the top of windows, and a cursor... fundamentally similar to both pioneering GUIs such as the Alto and the environments that we use today. Visi On allowed multiple windows to overlap, which sounds simple but was not to be taken for granted at the time.

Perhaps the most intriguing feature of Visi On is that it was intended to make software portable. Visi On applications, written in a language called Visi C, targeted a virtual machine called the Visi Machine. The Visi Machine could in theory be ported to other architectures and operating systems, making Visi On development a safer bet for software vendors and adoption of Visi On software a safer bet for users. This feature was itself quite innovative, reminiscent of what Java much later.

For the many things that Visi On was, there were several things that it was not. For one, Visi On did not embrace the raster display as much as even other contemporary GUIs. There was virtually no use of icons in Visi On, although it ran in graphics mode it was, visually, very similar to VisiCorp's legacy of text-mode software with GUI-like features.

One of the most significant limitations of Visi On is reflective of the basic problem with GUI environments running on existing operating systems. Visi On was not capable of running DOS software.

This sounds sort of bizarre considering that Visi On itself was a DOS application. Technically, it make sense, though. DOS was a non-multitasking operating system with direct memory addressing and no hardware abstraction. As a result, all DOS programs were essentially free to assume that they had complete control of the system. As a result, DOS applications would freely write to memory anywhere they pleased, and never yielded control back to the system [4]. In short, they were terrible neighbors.

While some GUI systems found ways to coexist with at least some DOS applications (notably, Windows), Visi On did not even make the attempt. Visi On was only capable of running applications specifically built for it, and all other applications required that the user exit Visi On back to plain old DOS. If you wonder why you have never heard of such a revolutionary software package as Visi On, this is one major reason: Visi On's incompatibility with the existing stable of DOS applications made it unappealing to most users, who did not want to live a life of only VisiCorp products.

The other big problem with Visi On was the price. Visi On was expensive to begin with, retailing at \$495. It had particularly high system requirements in addition. Notably, the use of virtual memory and swapping required something to swap to... Visi On required a hard drive, which was not yet common on PCs. All in all, a system capable of running Visi On would be a huge expense compared to typical PCs and even other GUI systems that emerged not long after.

Visi On had a number of other intriguing limitations to boot. Because it was released for DOS 2 which used FAT12, it could only be run on a FAT12 system even as DOS 3 made the jump to FAT16... among the many things Visi On had to implement to enable multitasking was direct interaction with the storage. VisiCorp required a Mouse Systems mouse, which was standard as of release but was soon after obsoleted (for most purposes) by the Microsoft mouse standard, so even obtaining a mouse that worked with

Visi On could be a hassle.

In the end, Visi On's problems were at least as great as its innovations... cost of a working system most of all. Visi On was the first proper GUI environment to market for the IBM PC, but many others followed very quickly after, including Microsoft's own Windows (which was, debatably, directly inspired by Visi On). More significantly at the time, the Macintosh was released shortly after Visi On. The Macintosh was a lemon in many ways, but did gain appreciable market share by fixing the price issues with the Lisa (admittedly partially through reduced functionality and a less ambitious interface).

The combination of Visi On's high price, limitations, and new competition were too much for VisiCorp to bear. Perhaps VisiCorp could have built on its early release to remain a technical leader in the space, but there were substantial internal issues within VisiCorp that prevented Visi On receiving care and attention after its release. It became obsolete very quickly, and this coincided with VisiCalc encountering the same trouble: ironically, Lotus 1-2-3 was far more successful in taking advantage of the raster display (by being available for common hardware configurations unlike Visi On), which lead to VisiCalc itself becoming obsolete.

Shortly after release, in 1984, VisiCorp sold Visi On to CDC. CDC didn't really have much interest in the software, and neither enhanced it nor marketed it. Visi On died an ignominious death, not even a year after its release... and that was the end of the first GUI for the IBM PC. Of course, there would be many more.

[1] Of course you may be aware that non-NT Windows releases (up to Millennium Edition) similarly consisted basically of Windows running as an application on DOS, although the coupling became tighter and tighter with each release. This is widely viewed as one of the real downfalls of these operating systems because they necessarily inherited parts of DOS's non-multitasking nature, including an if-in-doubt-bail-out approach to error handling in the "kernel." Imagine how much worse that was in these very early GUIs!

[2] The Corporate Entity Behind VisiCalc went through various names through its history, including some acquisitions and partnerships. I am always referring to the whole organization behind VisiCalc as VisiCorp for simplicity and because it's the best name out of all of them.

[3] This view of regression and plotting as coupled features separate from the actual spreadsheet is still seen today in spreadsheets such as Excel, where regression and projection are mostly clearly exposed through the plotting tool. This could be said to be the main differentiation between spreadsheets and statistical tooling such as Minitab: spreadsheets do not view operations on vectors as a core feature. Nonetheless, Excel's inability to produce a simple histogram without a plugin for decades was rather surprising.

[4] There were DOS applications that produced a vestige of multitasking, called TSRs for Terminate and Stay Resident. These were not multitasking in any meaningful way, though, as the TSR had to set an interrupt handler and hope the running application did not change it. The TSR could only gain control via an interrupt. When the interrupt occurred, the TSR became the sole running task. Of course, these limitations made the "multitasking-like" TSRs that existed all the more interesting.