

[Home](#) | [Products](#) | [Order](#) | [Contact](#) | [Gallery](#)

The Graphing Calculator Story

Copyright © 2004 Ron Avitzur.

Pacific Tech's Graphing Calculator has a long history. I began the work in 1985 while in school. That became [Milo](#), and later became part of [FrameMaker](#). Over the last twenty years, many people have contributed to it. Graphing Calculator 1.0, which Apple bundled with the original PowerPC computers, originated under unique circumstances.

I used to be a contractor for Apple, working on a secret project. Unfortunately, the computer we were building never saw the light of day. The project was so plagued by politics and ego that when the engineers requested technical oversight, our manager hired a psychologist instead. In August 1993, the project was canceled. A year of my work evaporated, my contract ended, and I was unemployed.

I was frustrated by all the wasted effort, so I decided to uncanceled my small part of the project. I had been paid to do a job, and I wanted to finish it. My electronic badge still opened Apple's doors, so I just kept showing up.

I had many sympathizers. Apple's engineers thought what I was doing was cool. Whenever I gave demos, my colleagues said, "I wish I'd had that when I was in school." Those working on Apple's project to change the microprocessor in its computers to the IBM PowerPC were especially supportive. They thought my software would show off the speed of their new machine. None of them was able to hire me, however, so I worked unofficially, in classic "skunkworks" fashion.

I knew nothing about the PowerPC and had no idea how to modify my software to run on it. One August night, after dinner, two guys showed up to announce that they would camp out in my office until the modification was done. The three of us spent the next six hours editing fifty thousand lines of code. The work was delicate surgery requiring arcane knowledge of the MacOS, the PowerPC, and my own software. It would have taken weeks for any one of us working alone.

At 1:00 a.m., we trekked to an office that had a PowerPC prototype. We looked at each other, took a deep breath, and launched the application. The monitor burst into flames. We calmly carried it outside to avoid setting off smoke detectors, plugged in another monitor, and tried again. The software hadn't caused the fire; the monitor had just chosen that moment to malfunction. The software ran over fifty times faster than it had run on the old microprocessor. We played with it for a while and agreed, "This doesn't suck" (high praise in Apple lingo). We had an impressive demo, but it would take months of hard work to turn it into a product.

I asked my friend Greg Robbins to help me. His contract in another division at Apple had just ended, so he told his manager that he would start reporting to me. She didn't ask who I was and let him keep his office and badge. In turn, I told people that I was reporting to him. Since that left no managers in the loop, we had no meetings and could be extremely productive. We worked twelve hours a day, seven days a week.

Greg had unlimited energy and a perfectionist's attention to detail. He usually stayed behind closed doors programming all day, while I spent much of my time talking with other engineers. Since I had asked him to help as a personal favor, I had to keep pace with him. Thanks to an uncurtained east-facing window in my bedroom, I woke with the dawn and usually arrived ten minutes before Greg did. He would think I had been working for hours and feel obliged to work late to stay on par. I in turn felt obliged to stay as late as he did. This feedback loop created an ever-increasing spiral of productivity.

People around the Apple campus saw us all the time and assumed we belonged. Few asked who we were or what we were doing. When someone did ask me, I never lied, but relied on the power of corporate apathy. The conversations usually went like this:

Q: *Do you work here?*

A: *No.*

Q: *You mean you're a contractor?*

A: *Actually, no.*

Q: *But then who's paying you?*

A: *No one.*

Q: *How do you live?*

A: *I live simply.*

Q: (Incredulously) *What are you doing here?!*

At that point I'd give a demo and explain that the project had been canceled but that I was staying to finish it anyway. Since I had neither a mortgage nor a family, I could afford to live off savings. Most engineers at Apple had been through many canceled projects and completely understood my motivation.

Apple at that time had a strong tradition of skunkworks projects, in which engineers continued to work on canceled projects in hopes of producing demos that would inspire management to revive them. On occasion, they succeeded. One project, appropriately code-named Spectre, was canceled and restarted no fewer than five times. Engineers worked after hours on their skunkworks, in addition to working full time on their assigned projects. Greg and I, as nonemployees who had no daytime responsibilities, were merely extending this tradition to the next level.

In September, Apple Facilities tried to move people into our officially empty offices. They noticed us. The Facilities woman assumed that I had merely changed projects and had not yet moved to my new group, something that happened all the time. She asked what group I worked in, since it would be that group's responsibility to find me space. When I told her the truth, she was not amused. She called Security, had them cancel our badges, and told us in no uncertain terms to leave the premises.

We were saved by the layoffs that began that month. Twenty percent of Apple's fifteen thousand workers lost their jobs, but Greg and I were safe because we weren't on the books in the first place and didn't officially exist. Afterwards, there were plenty of empty offices. We found two and started sneaking into the building every day, waiting out in front for real employees to arrive and casually tailgating them through the door. Lots of people knew us and no one asked questions, since we wore our old badges as decoys.

We were making great progress, but we couldn't get it done alone. Creating sophisticated software requires a team effort. One person can use smoke and

mirrors to make a demo that dazzles an audience. But shipping that to a million customers will expose its flaws and leave everyone looking bad. It is a cliché in our business that the first 90 percent of the work is easy, the second 90 percent wears you down, and the last 90 percent - the attention to detail - makes a good product. Making software that is simultaneously easy to learn, easy to use, friendly, useful, and powerful takes people with an incredible combination of skills, talent, and artistry working together with intensity and patience. Greg and I could do the core engineering, but that was a far cry from creating a finished product.

Among other things, we needed professional quality assurance (QA), the difficult and time-consuming testing that would show us the design flaws and implementation bugs we couldn't see in our own work. Out of nowhere, two QA guys we had never met approached us, having heard about our venture through the rumor mill. (We had become a kind of underground cause célèbre.) Their day job, QA-ing system software, was mind-numbingly boring. They volunteered to help us, saying, "Let's not tell our boss about this, OK?" One guy had a Ph.D. in mathematics; the other had previously written mathematical software himself. They were a godsend. They started right away.

Next, we needed help writing software to draw the three-dimensional images that our software produced. A friend with expertise in this area took a weekend off from his startup company to write all of this software. He did in two days what would have taken me a month.

My skunkworks project was beginning to look real with help from these professionals as well as others in graphic design, documentation, programming, mathematics, and user interface. The secret to programming is not intelligence, though of course that helps. It is not hard work or experience, though they help, too. The secret to programming is having smart friends.

There was one last pressing question: How could we get this thing included with the system software when the new machines shipped? The thought that we might fail to do this terrified me far more than the possibility of criminal prosecution for trespass. All the sweat that Greg and I had put in, all the clandestine aid from the friends, acquaintances, and strangers on whom I had shamelessly imposed, all the donations of time, expertise, hardware, soft drinks, and junk food would be wasted.

Once again, my sanity was saved by the kindness of a stranger. At 2:00 one morning, a visitor appeared in my office: the engineer responsible for making the PowerPC system disk master. He explained things this way: "Apple is a hardware company. There are factories far away building Apple computers. One of the final steps of their assembly line is to copy all of the system software from the 'Golden Master' hard disk onto each computer's hard disk. I create the Golden Master and FedEx it to the manufacturing plant. In a very real and pragmatic sense, I decide what software does and does not ship." He told me that if I gave him our software the day before the production run began, it could appear on the Golden Master disk. Then, before anyone realized it was there, thirty thousand units with our software on the disks would be boxed in a warehouse. (In retrospect, he may have been joking. But we didn't know that, so it allowed us to move forward with confidence.)

Once we had a plausible way to ship, Apple became the ideal work environment. Every engineer we knew was willing to help us. We got resources that would never have been available to us had we been on the payroll. For example, at that time only about two hundred PowerPC chips existed in the world. Most of those at Apple were

being used by the hardware design engineers. Only a few dozen coveted PowerPC machines were even available in System Software for people working on the operating system. We had two. Engineers would come to our offices at midnight and practically slip machines under the door. One said, "Officially, this machine doesn't exist, you didn't get it from me, and I don't know you. Make sure it doesn't leave the building."

In October, when we thought we were almost finished, engineers who had been helping us had me demonstrate our software to their managers. A dozen people packed into my office. I didn't expect their support, but I felt obliged to make a good-faith effort to go through their official channels. I gave a twenty-minute demonstration, eliciting "oohs" and "ahhs." Afterward, they asked, "Who do you report to? What group are you in? Why haven't we seen this earlier?" I explained that I had been sneaking into the building and that the project didn't exist. They laughed, until they realized I was serious. Then they told me, "Don't repeat this story."

The director of PowerPC software was an academic on leave from Dartmouth. The director of PowerPC marketing was the son of a math teacher. Seeing the value of putting this educational software on every Macintosh in every school, they promptly adopted us.

Then things got really weird. The QA manager assigned people to test our product. (I didn't tell him that those people were already working on it.) The localization group assigned people to translate it into twenty languages. The human interface group ran a formal usability study. I was at the center of a whirlwind of activity. Nevertheless, Greg and I still had to sneak into the building. The people in charge of the PowerPC project, upon which the company's future depended, couldn't get us badges without a purchase order. They couldn't get a purchase order without a signed contract. They couldn't get a contract without approval from Legal, and if Legal heard the truth, we'd be escorted out of the building.

Greg was lurking outside one day, trying to act casual, when another engineer accosted him and said, "I'm sick and tired of you guys loitering in front of the building every day!" Later he phoned the appropriate bureaucrats on our behalf. I listened to his side of the conversation for twenty minutes: "No, there is no PO, because we're not paying them. No, there is no contract, because they are not contractors. No, they are not employees; we have no intention of hiring them. Yes, they must have building access because they are shipping code on our box. No, we don't have a PO number. There is no PO, because we're not paying them." Finally, he wore them down. They said to use the standard form to apply for badges, but to cross out *Contractor* and write in *Vendor*. Where it asked for a PO number, we were to use the magic words "No dollar contract." We got badges the next day. They were orange Vendor badges, the same kind the people working in the cafeteria, watering the plants, and fixing the photocopy machines had.

Official recognition made life exciting. Suddenly even more people became enthusiastically involved. When formal usability testing with students and teachers began, we discovered, again, that we were far from being done.

I had long been proud of the elegance and simplicity of our design. I wanted our program to ship with every Macintosh, so I had designed it for all users, even those who know little about computers and hate math. I wanted to make mathematics as easy and enjoyable as playing a game. In a classroom, any time spent frustrated with the computer is time taken away from teaching. Sitting behind a two-way mirror,

watching first-time users struggle with our software, reminded me that programmers are the least qualified people to design software for novices. Humbled after five days of this, Greg and I went back and painstakingly added feedback to the software, as if we were standing next to users, explaining it ourselves.

Our recognition made life interesting in other ways since we could no longer remain a well kept secret. After a demo to outside developers, one person called Apple claiming that we infringed his patent, causing a fire drill until I could show prior art. Another company, the makers of Mathematica™, simply demanded that our product be pulled. Apple very politely declined. One week we were evading security, the next week Apple is rising to our defense.

By November, we were in full crunch mode, working sixteen hours a day, seven days a week, and feeling the pressure. The home stretch was a blur - wake up, grab a bagel, eat it while driving, work till we drop, sleep, repeat. If this story were a movie, you would now see the clock hand spinning and the calendar pages blowing away in the wind.

We finished in January 1994. Graphing Calculator has been part of the Macintosh ever since. Teachers around the world use it as an animated blackboard to illustrate abstract concepts visually. It shipped on more than twenty million machines. It never officially existed.

Why did Greg and I do something so ludicrous as sneaking into an eight-billion-dollar corporation to do volunteer work? Apple was having financial troubles then, so we joked that we were volunteering for a nonprofit organization. In reality, our motivation was complex. Partly, the PowerPC was an awesome machine, and we wanted to show off what could be done with it; in the Spinal Tap idiom, we said, "OK, this one goes to eleven." Partly, we were thinking of the storytelling value. Partly, it was a macho computer guy thing - we had never shipped a million copies of software before. Mostly, Greg and I felt that creating quality educational software was a public service. We were doing it to help kids learn math. Public schools are too poor to buy software, so the most effective way to deliver it is to install it at the factory.

Beyond this lies another set of questions, both psychological and political. Was I doing this out of bitterness that my project had been canceled? Was I subversively coopting the resources of a multinational corporation for my own ends? Or was I naive, manipulated by the system into working incredibly hard for its benefit? Was I a loose cannon, driven by arrogance and ego, or was I just devoted to furthering the cause of education?

I view the events as an experiment in subverting power structures. I had none of the traditional power over others that is inherent to the structure of corporations and bureaucracies. I had neither budget nor headcount. I answered to no one, and no one had to do anything I asked. Dozens of people collaborated spontaneously, motivated by loyalty, friendship, or the love of craftsmanship. We were hackers, creating something for the sheer joy of making it work.

After six months of grueling unpaid labor, Greg couldn't explain to his parents what he had done. They didn't use computers, and the only periodical they read was the New York Times. So as the project was winding down, I asked Greg if he wanted his photo in the Times so his parents would know what he was up to. He gave the only

possible response: "Yeah, right." We made a bet for dinner at Le Mouton Noir, a fine French restaurant in Saratoga. To be honest, I expected to lose, but I made a phone call. Greg doesn't bet against me any more: On March 11, 1994, the front page of the Times business section contained an article on the alliance among Apple, IBM, and Motorola, picturing Greg and me in my front yard with a view of the Santa Cruz Mountains. Someone I knew in Apple Public Relations was livid. I had asked if she wanted to send someone for the interview, but she had said that engineers are not allowed to talk with the press. It's hard to enforce that kind of thing with people who can't be fired. It was positive press for Apple, though, and our parents were pleased.

We wanted to release a Windows version as part of Windows 98, but sadly, Microsoft has effective building security.

Postscript: After the events described, we made everything retroactively legitimate by licensing the software to Apple for distribution. Pacific Tech started a few years later, and continued to develop Graphing Calculator, both in new free versions that Apple bundled with Mac OS 8 and Mac OS 9, and commercial releases. Visit [here](#) to download the software.

[Download](#) Graphing Calculator Viewer for [Mac OS X](#) or [Windows](#)

[Software design lessons of the Graphing Calculator](#)

[Pacific Tech on the Mac App Store](#)

[Slashdot](#), [AppleTurns](#), [Newsweek](#), [This American Life](#) and [Google Tech Talk](#)



Graphing Calculator

Like 598