Geek Nation

Table of Contents

GEEK NATION - Introduction
I was born a teenage '386

Well, not really. Intel released the 80386 CPU in **1986**, when I was 21. By then, the company that invented the microprocessor had released the 4004, the 8008, the 8080, and the processors upon which the entire Personal Computer industry would be based: the 8086 and the 8088. These product numbers are actually electronic component part numbers, like the simpler 'chips' that are still used in electronics today. By the time Intel released the 80586 processor, they realized you could not trademark a number, so they renamed the 80586 the Pentium. Finally, computers had become powerful enough to be useful to the average Joe. Before the coming of Intel, students of electrical engineering, physics, and mathematics separately pioneered the computer industry by assembling their own makeshift computers with simpler components such as vacuum tubes, transistors and relays.

The history of computers this is not. I just know my place in the evolution of things. The period of 1977 to 1984 were defining years in the computer industry. By 1980, the "essentials" of personal computing had been established, and the next generation
of Analysts, Programmers and Engineers (**APEs**) were able to create the next wave in computing. This technology wave, in turn, enabled the Average Joe (**AJ**) to use these same machines for productive work. In 1978, the Radio Shack electronics company released the TRS-80 microcomputer. My High School bought two. The Physics teacher was knighted the Computer teacher and the curriculum modified to give juniors a choice between computer programming or physics. I should have taken the physics, but the computer classes had me hooked.

Most kids who took the computer classes fell into two groups: the Nerds and the Game Players. The Nerds probably had their own Commodore 64, Commodore Pet, Apple II, Timex-Sinclair 1000, Texas Instruments programmable calculator or like me, a Radio Shack Pocket Computer. Nerds took this new fad seriously. We *had* to have one for ourselves.

Your *own* computer! It waited for us when we got home from school. Ready to give us the attention we craved, even when Suzy girl down the street wouldn't come a calling. Hormones. Who needed those? I had 8 bits of raw computing power in my hands! Nerds had gone through the telescopes, microscopes, home brew go-carts, and CB radio fads of the 1970s and were looking for a new hobby. Who knew it would lead to a career or an industry? Not I. Nor any adults I knew. Just a fad? Perhaps computers were just that.

The machines of the day had the computing power of a gnat, but that didn't stop us from learning how to program. Using languages such as BASIC, Pascal, C and Assembler, APEs learned how to create file systems, draw pictures, interface with mice, play with joysticks, control cassette recorders and write programs to do almost any task. The idea of having your own personal computer had taken root years before, in the mid 1970s. But now, computers had become affordable to any teenager with a summer job or a rich daddy.

The Radio Shack TRS-80 had a character-only screen with 64 columns by 24 lines. Just barely enough screen real estate to tell the user that there really wasn't enough screen real estate to put up a worthwhile message. While my high school class assignments revolved around database programming and simple file operations, I stayed after school learning how to program games. I wanted to play my own games. At the time, 1979-1980, the first console (coin operated) video games hit the streets in back rooms of kiddie soda-jerk shops which were later to be renamed "arcades". The games Asteroids and PacMan ruled the day during the turn of the decade.

Back on the TRS-80, I figured out how to draw a maze like PacMan's maze. I also figured out how to make the game monsters. I also modeled their ability to move, munch, eat dots and die. Later I created a game called Circus-Circus which was pattered after a board game I played once as a kid. Its main premise had a main character jumping on moving trampolines into a final bucket. The kicker was that I 'rented' these games to the kids in lower school for 25 cents a play. I was a much bigger guido back then I guess.

My cute brace-face girlfriend had her OWN TRS-80 at home. Her dad was a well-to-do gynecologist who wanted the best for his daughter; which, I was later to find out, didn't include me. One day the computer class sponsored a programming contest. My girlfriend created a crossword puzzle that apparently created a unique board each time you played it. I learned later that she typed in the code from a book.

My program involved my other hobby, Shortwave Radio. I created a program that controlled a cassette player that held an audio tape I narrated, along with live broadcast clips from other countries. I drew an on-screen representation of my Sony ICF-2010 shortwave radio. As the tape was playing, the simulated radio "display" would change the frequency and radio "controls" as I talked about the different stations one could hear from around the world. My speaking and the commands to change the visualization were insync. Of course, the crossword puzzle won first place.

My program would later be classified as machine automation. I had created "kiosk" software and didn't even know it. I was teaching and training using a computer. What a novel idea.

The jockstraps and rich kids were the other bunch who took computer programming and had no intention of learning the subject matter. It's quite something to have other students in a class you signed up for taunting and teasing you because you **could** figure something out. Perhaps I should go to soccer practice and tell the jocks "you're a sorry excuse for an athlete" just to get even.

As part of our senior yearbook, students were allowed to create a "last will and testament" to leave to the school. One of the sports jocks who got stuck in computer class left the computer teacher an "error in line 10" as a joke. In the TRS-80 BASIC programming language, line 10 usually was the first line of a program. To this day, I try to leave my "errors in line 10". After you learn to write programs for a living, your creative juices (mischievous intentions) manifest themselves as strange comments in source code. I guess it's a geek thing.

These "Easter Eggs" are subconscious going-away presents to all those underpaying employers who later sell our jobs to the lowest bidder from the poorest country. Our errors in line 10 come as messages, errors or code blocks ranging from the sublime to the destructive. Oops. How did that get in there? What part of the nuke plant does this control? I can't bring up payroll today. Hmmm. Usually manifesting themselves on the day after March 31st, these joyful bundles of programming misery can now be found in everything from business applications, to PS/2 games, to DVDs.

When you create a video game from scratch, even a simple one, you are actually learning Modeling. You are translating an idea into a representation and/or presentation of that idea. The games I created were either copies of existing games or an original idea of my own. An analyst analyzes an idea and figures out how to represent it in the machine language of the computer. Along the way, APEs found out that some ideas are harder to represent than others, things like 3D.

How does a PacMan move? (Add/Subtract to the object's X/Y coordinates)
How does he know he was eaten? (Do two objects coordinates cross each other)
How does a dot know it was eaten? (Coordinate collisions)
How do the monsters seem to know the maze better than you, when you can see it?
All very good questions.
All very good brain teasers for an adolescent.
Just as complicated as physics or math, computer programming requires the cognitive ability to take a problem and find a solution:
Y = MX + B?
How about MOV EAX,0001 instead?
But I'm getting ahead of myself (Chapter 34)

Of course, APEs had already discovered the amazing ability of the computer.
When a refrigerator is plugged in, it still does the same thing 10 years later.
But a computer is a machine whose function can be changed by swapping out its program code over and over.

It's Neil Armstrong all over again!
The wonder of unlimited potential.
I could make this machine do anything.
One day it's a calculator, the next,
It's a spell checker.
What can we invent now?
But now it's ME walking on the moon.
Nobody had been HERE before.
"My version is the first version of
PacMan for the TRS-80!"
Dexter is born.

For a kid, Model Rockets are modeling real rockets. To construct one 12 feet long on a teenager's budget is impossible. But using my computer, I could make a rocket flying game for free. All it took was time. Our Moms used to say that 'Patience is a virtue'. APEs have lots of virtue. We are the kind of simian who doesn't mind elevators, because we know they will eventually arrive. Maybe that's also why we don't get married until our thirties. Check Please.

My other adolescent hobby was listening to Shortwave Radio. There were two Ham Radio operators that lived in my neighborhood who encouraged me in this. They would let me speak to people on the radio occasionally. One neighbor was Ed Passow. Ed helped develop the NTSC color TV standard in America. Retired, Ed called CQ and downed the coffee, talking to old friends and new from his own real "radio shack".

I myself was a radio listener only. I used to track the stock broadcasts from the BBC daily. I was trying to spot trends in hopes of striking it rich. Trouble was, I couldn't open a brokerage account at 15. I didn't get the BIG picture on many things like that until later. So I tried to program my Radio Shack Pocket Computer to track these stocks. I soon had typed in thousands of stock prices, trying to find my magical trends. That's when a problem developed that turned into an epiphany for me. My stock list was not sorted and the Pocket Computer I used had a really slow search function. This meant that the computer had to search through each item, one at a time, checking for a match. Searching from the first item to the last was slow and inefficient. There had to be a better way.

It was real cool to have all those stocks in your back pocket. Of course, none of the kids in my neighborhood were impressed when I whipped out the Nerdly Extraordinaire 5000 with nine months worth of IBM stocks in it. Strange looks always followed.

Anyway, it took my dainty Pocket Computer a few minutes to search for a certain stock entry.
What can I do? This thing is too slow. That's when it hit me:

- How do people normally find things?
- The phone book is alphabetically ordered.
- The library has the Dewey system.
- Maybe I should order my lists too.
- But then, would this be any better at finding an item in the middle of the list versus an item at the end of the list?

By repeatedly halving the list and comparing the search item value against the new first and last entries, I "independently discovered" the binary search method without knowing anything about database theory. I later learned that items inserted into a "Binary Tree" structure was to become the most efficient way to create ordered lists that leant themselves to rapid searches. But at 15, I never figured out how to balance the tree. Today's corporate data warehouses are built on this search and sorting algorithm.

Thusly, I studied computer programming during my junior and senior years of high school and was amazed to learn that the high and mighty IBM was announcing they would release their own version of a computer. "The computer for the rest of us." In August 1981, the IBM Personal Computer (PC) was born. My Dad eyeballed one for his spreadsheet and word processing needs. (After all, those were the ONLY needs back then.) Dad had used the IBM mini computers (the size of Grandpa's Garage) of the 1970s to balance the books of small and large corporations. But this machine was different: your *own* computer. Able to create recipes and track home expenses! Ooh baby. Don't step in the drool.

Kids back then and kids of today are the same: "Dad, the computer is a game playing machine".

Funny thing, while I despised my High School computer curriculum for its stodgy "Database Programming" focus, I later would make a living doing just that: Databases. As an adult there was no time for games when the code had to ship. I never did write another game after my 1982 summer school camp at Duke University. Here is why.

My mom was pushing me into college ever since I was 5 years old, maybe sooner. By junior year, even with good grades, I was no closer it seemed to making a career choice than I was at five. She managed to see an advertisement for a computer camp for High School kids. Since I was a bright kid (but not bright enough to pick a major), she suggested the Duke University camp in the summer of 1982. The 2 week course was one of the first in the country. The first Nerd Camp. Remember to rinse before you spit.

Duke had a roomful of the new IBM Personal Computers to be used by graduate students to teach the next generation of nerds how to become APEs. After daily lessons, the class was broken up into four-packs of craniums that had to solve a shared problem. The computers were running the UCSD-Pascal Operating System with a Pascal Compiler. I was later to realize what a good choice Duke made in NOT teaching the BASIC language.
Pascal teaches structured programming habits, code reuse, object oriented design, team programming. UCSD Pascal was also a compiler, so any programs written in it were many times faster than BASIC code, still are today. UCSD Pascal and the OS were linked together. There was no PASCAL Compiler for MS-DOS as of then, so Pascal on the UCSD OS was what we learned.

It wasn't until the next-to-last day of camp that I realized Software Design was to be the career for me. The Duke Camp took us on a field trip to N.C. State. The engineering school had a computer room worth visiting. Boy was it ever! In this one room, two sets of Graduate students and one PhD were showing off their achievements to an astonished group of onlookers, including me.

The first project belonged to a woman who was a crystallographer. She was staring through a funky pair of goggles at a vector display. I wanted to know why. A vector display can only draw lines, not dots. She explained that crystallographers create molecules with stick figures. Her team was trying to "virtualize" them by drawing three-dimensional figures of basic chemical compounds on the display twice; and one version was slightly offset from the other. The second image was also made to vibrate on screen, Doing this, one could have a true 3-D representation of the molecule and spin it in three dimensions. What I didn't know was that the goggles had a simple card spinning inside it that alternately blocked and revealed your view. The frequency of the spinning card and the second vibrating wire-frame crystal molecule on-screen were both moving at 60 times per second. When you looked through the goggles, voila! Three dimensions.

The next student team had a green-screen only monitor mounted on wheeled floor cage. The monitor was bolted to a huge bass speaker cone underneath that was silent. When you stood over the monitor, all one could see were a bunch of dots, in a blob, not very interesting. When the switch was thrown, the speaker started to hum at 60Hz and

the image literally "went deep" as another 3-D effect was brought to life. The image was that of a spinal cord and backbone. Another scientific medical imaging tool. Wow.

The last project was assigned to a PhD student. His professor gave him a library of CAT scan slice images of a human heart. His thesis was to create a 3D model of the heart that could be rotated and the "slices" inserted and removed at will. I was sold.

All these people were doing things I never thought possible. **Imagination. Dedication. Improvisation. Creativity. And lots of study.**

Later that year I applied to N.C. State and got accepted, based mostly on my S.A.T. scores. Although I signed up for Electrical Engineering, I was placed in Industrial Engineering because EE was full. I actually never thought about a Computer Programming or software major, but that is where I would spend the next 20 years of my life. And although I looked for it many times, I never did find that computer lab again.

So, to get back to my original point, I would say yeah, there's a need for another tell-all book. I mean, why should the brainless, I mean Hollywood types, get all the checkout stand real estate? There's got to be some room in there for us geeky, nerdy APEs. We need some attention ($$$) too.

THE AMAZON (Chapter

After all, its not everyday you get paid to sit in a chair for 10 hours, slaying mudwumps and drooling over Amazons like the guys in the I.T. business. Where fresh air comes in cans and the company Christmas party consists of a memo.

Besides, the Information Technology (I.T.) industry needs a little exposure for putting us all out of work post-why-two-kay(Y2K)(2000).
(We are still wondering who to thank for that.)

---

www.computerjobs.com
Tuesday, August 12, 2003
6,532 Active Jobs!
1,988 new jobs in the last week
Over 350,000 active, pre-screened IT resumes
1,230,000 registered users

1.2 million geeks fighting for 2000 jobs.
Hmm…

(Is that 53:1 or 188:1 odds?)
here is a better 53:1

I am obviously biased towards the tools, methods and design choices I made in my career creating the computer programs that I did. I'm not saying those choices will guarantee me a future in I.T., nor a cult following, nor were they even the best way to skin my cats, but they worked for me and I was hooked on those phonics.

And, yes, I still wonder "are they using it" long after I am gone. Even though the I.T. mantra is 'its old before it ships", and most ideas or systems get canned before they can be brought to life, we were young and naive, expecting some kind of continued employment for all our hard work and loyalty. All those sleepless nights smoking away on a hot keyboard, constantly studying, begging, borrowing and working just too hard for unappreciative employers. We now journey down the street towards the unemployment office. What color is your parachute?

It's like the stupid machine from the first Terminator movie:
In the end, the metal press squeezed the life out of the pressed-metal man.
One wonders if the T-800 learned his lesson:
"T-800: system log: don't crawl thru ten-ton press trying to smoke your target"

Ahh, yes, the glamorous IT industry:
We could all smell the silicon with our 14-year old Stridex smiles:

"I can make a computer do what I say and you can't!"

When once the stories of east-coast school systems handing out I.T. certifications called CNEs, CCNEs and MSCEs to eager high-school students faded, these same students, who rushed out to make 40 to 70k per year ***more*** than their blue-collared pensioned parents, are now moving back in with them; 40, broken, broke, with a broken-down signing bonus of a broken BMW.

A new chant from the smoke of our once glorious careers is heard nationwide:
"I think Ill take another career and benefits to go, please!"

Singing like Sammy Hagar without Van Halen, Sting with no Police: employed, but not cranking it.

The mighty American knowledge worker has fallen and can't get up.
Can we call some kind of national 911?

"Vhat is thee na-chur dove yur ebergencee?"
"Uhh, my company like, sold me out..heh heh"
"Hey, Butthead this new job sucks"
"Dees iz tek shuporrt. Can I delp you?"

Instead of compiling code, America's best minds are compiling scrapbooks of digital pictures of their past lives, always wondering if that Indian or Russian hacker who got their job really works on a wooden keyboard from a mud hut.

Did you say a Management position?
Yes Mr. Smith. But you realize the starting salary at Pizza Hut is seventeen-five?
Uhh, is that per hour?
No sir, Per year.
Oh.