

# Roberto Capodieci: an IT Artisan



## 1980s

I was born in 1974, one of the first generation of acne-ridden, computer-loving geeks. I am old enough to have seen the birth of home computing, and young enough to belong to the group they call "Digital Natives." At school I learned to write in my (paper) notebook and, at home, to write code on my computer keyboard. I have belonged to the digital era since a time when Information Technology was unknown to most people.

I would spend hours sitting in a darkened closet at my home, lit only by a green monochrome monitor. My inspirations were "TrOn" and "War Games", only to be viewed in movie theaters--VCRs were rare in those days. At that time, nobody could imagine that computers could become one of the supporting pillars of our world's economy and that, one day, they would become essential for human beings to survive on this planet. Nobody would have imagined that the mix of electronic paraphernalia, at the time still considered a toy, would become essential to the functioning of our modern society.



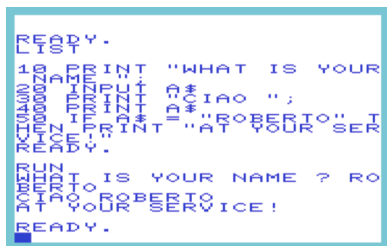
**My first programming experience at age 6 on my dad's Sharp PC-1211**

Today there are legions of IT "experts". But at that time, when home computing was still in its infancy, the IT community numbered only a few people who shared the same "new" hobby of deepening their knowledge of computers. These circles of enthusiasts, many of whom today are prominent specialists, were for a long time members of a small and restricted elite, even though computers had already started to gain ground in our daily lives.

Two very different groups of computer owners naturally formed: a majority which used their equipment only to play games or to do word processing, and a minority that, curious and eager, went on daily "expeditions" inside the microprocessor. This latter group gave birth to an underground community of IT pioneers devoted to the discovery of uncharted virtual and cyber territories, gaining ground through daily experiments with new lines of code.

### The Identikit of the first little IT geniuses

It required the right mindset, humility and capacity for extreme patience, paired with curiosity and maybe an undiagnosed light form of Asperger's syndrome, to remain isolated in this digital world, with only a monitor and a keyboard for company, sometimes for several hours of daytime and, as often, the entire night.



A short program and its execution on the emulator of a Commodore Vic 20, the computer that, at age 10, I used to develop my first saleable videogame.

Dealing with a very obedient, intelligent and yet stupid machine, supported by barely any documentation and just a few code examples to learn from, required a methodological and daring approach. Only those who understood "the rules" on how to deal with a computer would eventually succeed. The beginning required the humility of an apprentice, having a teacher that spoke a different language. Whatever you would say, the answer would always be: "ERROR"!

Having the patience to learn to communicate with the computer was essential, and keystroke after keystroke, the master of a few words would become a very obedient pupil, fulfilling every request without ever committing the slightest mistake. With time, learning the language to communicate concepts to the machine and, most importantly, understanding the machine's psychology (the electronic psyche of the machine's main brain) was the necessary step to create incredible and fascinating new worlds.

Reaching this stage would usually bring the programmer into a heightened divine state: he could define the rules of his digital world through lines of code that were considered commandments -- more than mere commands. Being able to find refuge in a world where one person could assume the role of God was definitely a good reason for always trying to gain more control over it.



A scene from the movie "The Matrix"

There was a constant need for experiment, trial and error, and hours of study in understanding and analyzing secrets held by another programmer's code. There was a habit to search and reach the limits imposed by computers capacities, to demonstrate the complete domination over the digital environment. Every secret had to be unveiled, every obstacle had to be prevailed over. A better, deeper, broader knowledge would be the ultimate proof of one's superiority in the digital world and over other programmers.

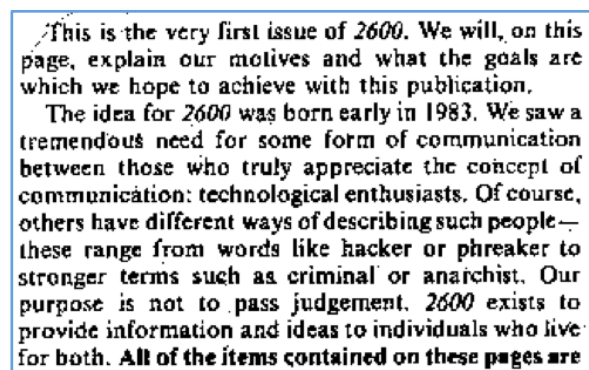
This is why the extreme challenge presented by cracking encrypted files, breaking into password-protected environments and overcoming protection systems has often been a preference for IT explorers. To breach a security system was, for those who succeeded, the demonstration of their superiority over the system's creator. The theft of protected data was hardly the main goal, but a proof of the ability to infiltrate someone else's system.

## The birth of the Hacker community

At that time, there were no books or schools teaching the notions necessary to become a programmer. When somebody discovered something new or wrote an interesting piece of software, they would simply share it with other IT enthusiasts.

Unfortunately, finding a reliable source of information that would inspire and enrich one's knowledge was a real challenge - it was as difficult as finding a way to share successful experiences. On the top of it, given that we were all living in our own secluded digital worlds, getting in touch with another IT enthusiasts to share something was barely possible.

The first "broad" connection between programmers of the first generation was rendered possible by scarcely available printed publications in the English language (giving an additional challenge to non-English speakers such as myself at the time). The year 1984 saw not only my 10th birthday, but also the birth of *2600*, a controversial quarterly magazine collecting and re-sharing tricks and tips about IT and phone networks. *2600* is still published to this day.



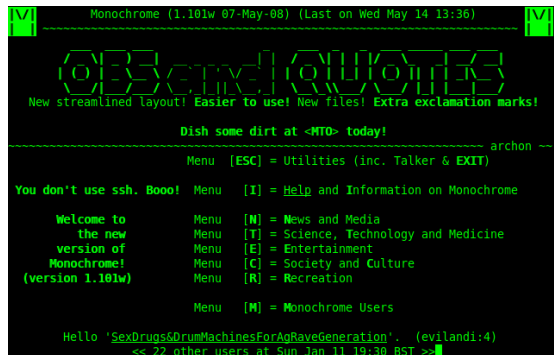
This is the very first issue of 2600. We will, on this page, explain our motives and what the goals are which we hope to achieve with this publication. The idea for 2600 was born early in 1983. We saw a tremendous need for some form of communication between those who truly appreciate the concept of communication: technological enthusiasts. Of course, others have different ways of describing such people—these range from words like hacker or phreaker to stronger terms such as criminal or anarchist. Our purpose is not to pass judgement. 2600 exists to provide information and ideas to individuals who live for both. All of the items contained on these pages are

*From Wikipedia: "The magazine's name comes from the phreaker discovery in the 1960s that the transmission of a 2600 hertz tone (which could be produced perfectly with a plastic toy whistle given away free with Cap'n Crunch cereal [...]) over a long-distance [phone] trunk connection gained access to "operator mode" and allowed the user to explore aspects of the telephone system that were not otherwise accessible"*

**Left: part of the first page of the first edition of the fanzine "2600" printed in 1984**

With the growth of IT communities around the world, very shyly, other magazines addressing similar topics both in English and in other languages began publication. Unfortunately, these ones were only for beginners and were of interest only to a new generation of programmers. In the meanwhile, years before the Internet arose, and thanks to the birth of modems (*MOD*ulators-*DEM*odulators, devices capable of transforming data into sounds and vice versa), computers started learning to exchange data with each other over the phone network.

By leaving a home computer always powered on and ready to answer a phone call through a modem, the first virtual meeting points and document repositories were born. Data connection via the phone network opened a new way for getting to know and confronting others and learning new ideas, thanks to information sharing that was almost in real time.



**A vintage BBS, accessible via Internet, mimics the old monochromatic UI used to run Bulletin Board Systems**

modems, allowing more users to be online at the same moment. The SysOp was the "God" of his own computer and BBS, and those who were granted access were under his power. With no more than a key stroke, he could allow the users to consult archived documents, use messaging systems that were the precursors to today's emails, as well as kick users out of the system and bar them from future access.

Thanks to the BBS that also provided the first electronic mailboxes, the first hackers started to consolidate their friendships and form groups with members that were often living in different cities. Concurrently, these new network systems also served as channels in distributing illegal materials, such as credit card information and stolen access accounts (username and passwords) to systems with sensitive data. The circle of curious and enthusiasts then spread, growing through several BBS networks (the biggest of which still in function is Fidonet), reaching out all around the world. A thin line separated the good and bad hackers, and the first police operations against the first cybercrimes had already started taking place.



**1983:** The FBI raids more than a dozen homes in six states, confiscating Telnet passwords, at least one Apple II+ , a modem and several other computers. An article in InfoWorld refers to an increase in hacker activity following the release of the popular film "WarGames," which portrayed a high school student who was able to hack into the computer system at the North American Defense Command in Colorado Springs, Colo.

In the years following, new big private and public data packet networks were put in place, and connections between these networks and the creations of "networks of networks" became the genesis of ARPANET, which in turn evolved into the Internet. During those years, a small community of people witnessed the birth of something that, today, belongs to everybody's life.

**Twenty years ago bits and bytes were the interest of a few; today they are vital to everyone's life.**

Until the birth of the Internet the IT world was merely about techniques and technologies catching the interest of a few programmers and small groups of users (the pioneers of word processing and electronic spreadsheets) and arguments over bits and Bytes were definitely not of public interest.

That has all changed. Today discussions about IT are of interest to everybody. These are the years of reality check in the IT world: We are at a point where abstract entities such as bits and Bytes (the data) are so important that governments all around the world are--sometimes from positions of deep ignorance--trying to regulate them.

This is because everything is now bits and Bytes: Music, as well as movies, books, money, reservations, documents, signatures, certificates--practically everything is moving from paper, film, tapes, into cyberspace. Commerce has become e-commerce; our lives are moving toward this new place and this new dimension. Digital information is becoming more important as our existence is in the process of being bit-ified.

In this process of evolution, we trusted that transforming everything into digital information, and the place called cyberspace in which it happens, is nobody's land. Land where there are no laws, no extraditions, and no taxes. "This is bad," those who govern us must have thought as soon as they started to grasp what was happening. The knowledge that everything could be transformed into bits and Bytes, and transferred into cyberspace, must have put the highest authorities on alert. When even ordinary people understood what was happening, and the advantages that arrived with this new era, the powers that be decided: "We must intervene."

And so started a crazy regulation race. The problem, sometimes, is that rules are made by those with no idea of what they are ruling upon, not even knowing that these rules are applied to something that is constantly changing and growing everyday at an incredible rate, a rate that scares even people like me, who have been in the field since the beginning. The legislators trying to regulate IT remind me of Don Quixote running against the windmills, making much dust and confusion for nothing. Cyberspace, in the meantime, remains nobody's land.

### **The industry of software development in the hands of the new "experts"**

During this unavoidable rapid change of scenery, this rapid evolution that has overturned out planet's social and economic life, a new generation of experts has emerged. They venture into the IT world for mostly economic reasons. Compared to the geeks of my generation, these new white collars are expert at something else: They understand their client's problems better. But the same can't be said about their knowledge of a microprocessor's ways. Since the late 90's, as IT systems became easier to use, it is no longer necessary to know how a computer works to become a computer programmer.

This allows the new recruits to use pre-made tools for software development to create, in very short time, systems that they don't even fully understand. As a direct consequence the old generation of programmers has found itself nonplussed. The circle of IT experts has grown, from the small elite I mentioned earlier, into a full army of hundreds of thousands of new IT guys that a starving market called "the new economy" wants to hire the most urgently.

Today, to satisfy clients who regard programmers of the new and the old generation as the same, the only requirement is to provide fast software solutions at the expense of security and stability. The new generation of geeks, info-sapiens from the least to the most expert, is born in an era where many things have already been done and many streets have already been paved. And because market needs are different, they find themselves covering a different function: no more forgers of new code, but fast solvers of specific problems.

They are an evolved, though not necessarily a better version, of those who were there at the eve of programming for home computing. This new generation army of white-collar experts has inherited all the fruits of the efforts of their predecessors. New recruits are now able to learn without the need of the baby steps that allowed the first generation of programmers to get acquainted with the finer gears of information systems. To acquire this kind of heritage has both positive and negative sides: The new "experts" certainly have many qualities the old ones lack, but they have lost something that we did have: the ethics with which we've been working for many years. What got lost with this excessive ease in programming computers is a love for the detail, the capacity of optimizing code, and the passion in making robust software. What happened throughout IT's history is the same thing that happened in the Industrial Revolution: craft got replaced by industry. Quantity has won over quality.

### **The IT craftsmen and the IT industry**

I am an IT craftsman; I do things with love--handmade, done well and with precision. I know them down to the tiniest detail, down to the heart of the smallest bit. I know why each thing works, even if there is no apparent need to know. I cannot be asked to work on a production chain, not even to use industrial tools. Today's programmers, and it's not their fault, are working with self-assembling, pre-fabricated kits. They use, and know how to use, programmer's tools without knowing the "*hows and whys*" of their functioning. They only know what result they can get. This is why it's not their fault if often the software they develop has problems, doesn't work well or has security breaches.

Today's programmers can develop applications without making any mistakes. But if their tools are flawed, the end product is sold without any warranty of trust. The advantage is that today's programmers can deliver finished products in two days, things that an old school programmer cannot achieve in any way. The old school programmer can, on the other hand, ensure the quality of his product, because he knows even the smallest details and effects of the instructions and commands that he used to write his software.

The paradox of all this is that, today, the need to guarantee the safety of information should be higher compared to the past, since we are moving into a cyberspace of sensitive data, and because we rely on frameworks that can't be trusted for vital infrastructure. To repeat: Our money, our names, our secrets and our life today are all in the hands of computers that are managed by software applications which, potentially, are not secure.

Daily, without anyone noticing it, cyber thieves are stealing information and breaking into companies' IT systems by using security flaws rendered possible by forgetful frameworks, right before the eyes of unsuspecting users. Beyond the act of stealing per se, this situation can ultimately have very serious repercussions on our economic stability.

We must move away from a cyberspace built on the fly, cobbled together by unknowing hands on a tight budget, to one that is secure, where the commitment is not just to the bottom line but to quality--built and maintained by those who know intimately the tools and the raw material they are using. Even if it takes a little longer. Otherwise we may lose more than we bargained for.