

Infocom Takes Off

Around the time Infocom began, big, expensive mainframe computers dominated the market. In 1979, that all changed when Dan Bricklin and Bob Frankston released *VisiCalc*, the first electronic spreadsheet and “killer app” for the personal computer. People purchased personal computers just for the sole purpose of using *VisiCalc*. Apple, Commodore, Atari, IBM, and Radio Shack had a whole range of models available, but each one garnered a small percentage of the market.

Personal computers cost over a thousand dollars each, and owners tended to be wealthy and well educated. Doctors, businessmen, lawyers, and other professionals could afford the price tag. The owners of the personal computers also tended to enjoy reading - a factor that would prove critical to the success of Infocom’s games.

The emerging personal computer market presented a golden opportunity for Infocom, but its founders faced a difficult challenge: How could they make *Zork*, which barely ran on a PDP-10, work on comparatively tiny microcomputers?

Making Zork Fit On Personal Computers

Berez and Blank spent one month in Pittsburgh together discussing the feasibility of porting the complex and addictive world of *Zork* to smaller computers. It was a daunting technical task - they would be faced with limitations on all fronts, from memory size to processor power. And to complicate matters, they would have to support multiple platforms, since there was no clear market leader in the home computer business.

Blank initially thought that compressing *Zork* to fit the stringent memory and code size limitations of a personal computer would be impossible. The mainframe version of *Zork* ran on a PDP-10 with 512 kilobytes of memory and required a megabyte of code. In contrast, the TRS-80 Model I and Apple II had 64 kilobytes of memory at most with an optional floppy drive that could hold 80 kilobytes. Motivated by Blank’s doubts, Berez felt there had to be a way to make *Zork* smaller. He looked at *UCSD Pascal*, a language used to compile *Pascal* to platform-independent byte-codes (called P-codes), which were then interpreted by a virtual machine. While *UCSD Pascal* made code portable, it would not make *Zork* much smaller. Taking the idea one step further, Berez came up with the idea of making a design for a virtual machine specifically designed for text-adventure games, which he called the Z-machine.

Berez and Blank sketched out the details of the Z-machine to convince themselves that it would allow *Zork* to run on a computer that had 32 kilobytes of memory and a floppy drive. They saw that they could minimize the code size by tailoring the machine’s instruction set for the specific operations of the game. In addition, Berez and Blank realized that it wasn’t absolutely necessary to keep all the program code

loaded in main memory, thereby reducing the memory requirements. Instead, the Z-machine could leave the bulk of the code on disk and load certain sections into memory whenever the program called for it. This idea drove Berez and Blank to employ one of the earliest virtual memory managers for the personal computer.

The Z-machine was especially suited for games like *Zork* because its design revolved around an object tree structure representation. Objects represented the things in a game, such as rooms, items, players, enemies, and weapons. Furthermore, objects had attributes that described what they could do. For example, an object could be “takeable,” which meant that the player could carry it around the game. Each object also had parents, siblings, and children to represent their relationship to other objects. For example, suppose a game had a glass bottle and a knife in a kitchen. The kitchen would be the parent of the glass bottle and the knife, and the glass bottle would be the sibling of the knife. This hierarchical relationship made it simple to describe where objects resided.

Figure 2 shows a diagram of the Z-machine design.¹⁵ The Z-machine had two types of storage: memory and stack. The memory contains the code and object tree for the program, while the stack holds local variables and temporary values. The Z-machine executed one instruction at a time starting at the program counter (PC). Its basic instructions included:

- Manipulating an object tree (e.g. moving objects, testing for hierarchical relationships) Calling other routines
- Accessing global or local variables
- Printing information to the screen
- Reading input from the console
- Jumping to another instruction
- Computing simple arithmetic (e.g. add, subtract, mod)

The Z-machine made it easy to express adventure games compactly by providing instructions for common operations. The mainframe version of *Zork* duplicated code for tasks like moving objects from one place to another, checking objects for certain properties, modifying properties, testing for hierarchical relationships. In the Z-machine, these operations could be expressed in several bytes of code - a huge savings factor. Table 2 shows a listing of some of the basic instructions of the Z-machine.

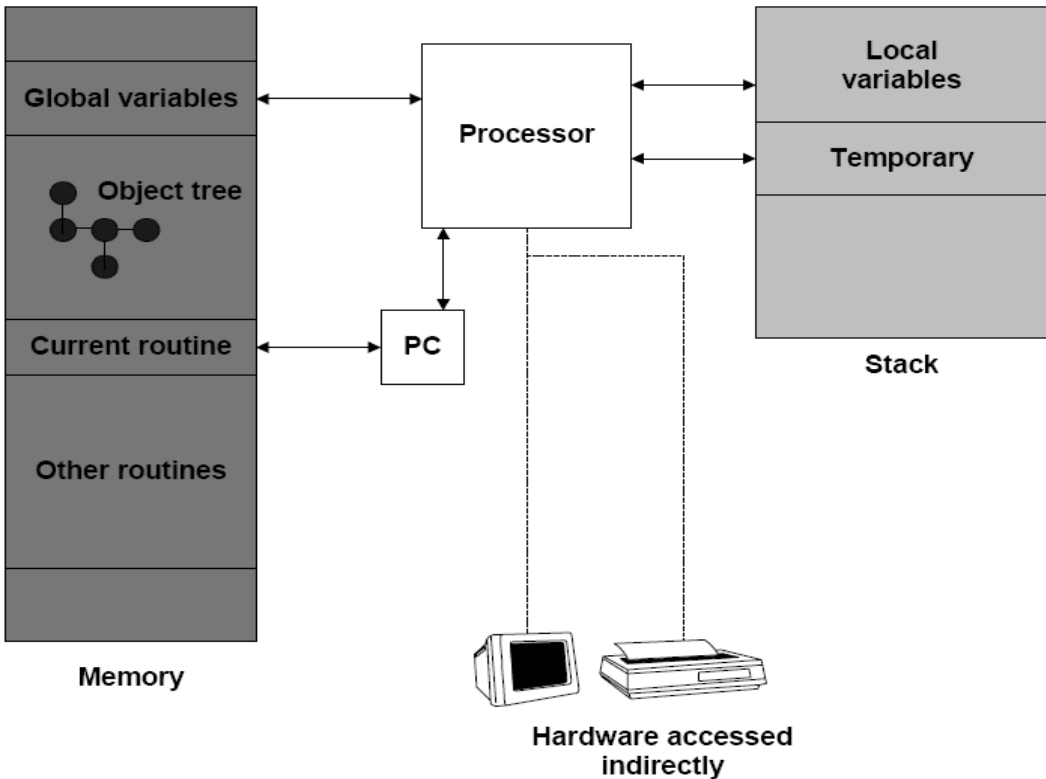


Figure 2. The Z-machine design. The memory contains global variables, the object tree, and the code for the program. The stack contains local and temporary variables. Z-machine instructions provide ways of indirectly accessing the hardware (e.g. printing to screen, reading input from console)

Instruction	Description
MOVE	Move object X into object Y
REMOVE	Remove object X from everything (e.g. if it is destroyed)
FSET?	Test for attribute X in object Y
NEXT?	Get sibling for object X
FIRST?	Get child for object X
LOC	Get parent for object X

Table 2. Sample Z-machine instructions and their descriptions.

Several other techniques were used to make *Zork* even smaller. The Z-machine compressed text, which constituted a large portion of the data, by using a representation that required approximately five and a half bits per character instead of the usual eight.¹⁶ In addition, Infocom stripped out many unnecessary features of MDL, such as associative storage, and created a language called the *Zork* Implementation Language (ZIL). Realizing the mainframe version of *Zork* would still be too big to fit on a floppy, Lebling looked at a map of *Zork* and divided

the game up into three sections. The latter two would be used for sequels to the original *Zork*.

With the Z-machine design completed, Berez and Blank began writing a two-stage compiler that would convert the high-level ZIL code first to assembly code and then to Z-machine byte-codes. Blank then wrote a Z-machine software emulator - which came to be known as a Z-machine Interpretive Program (ZIP)—for the DECsystem-20.¹⁷ In 1980, Scott Cutler, a member of the DM group who went on to work in New York, finished writing a ZIP for the TRS-80 Model I. And in 1981, working remotely from California, Bruce Daniels completed a ZIP for the Apple II.

The portability facilitated by the Z-machine design proved to be an important asset. The game files were stored as Z-machine byte codes, which could then be interpreted by a ZIP. To make all its software to run on a platform, all Infocom had to do was write one ZIP. This was especially important because Atari, Apple, Commodore, IBM, NEC, Radio Shack, and other companies had models of personal computers out of the market. While the Apple II constituted over 50% of the market by 1982, the other platforms shared the remaining percentage.

Infocom released *Zork* for the TRS-80 Model I in 1980, beginning its entry into the software entertainment industry. In the beginning, sales were slow. The TRS-80 version sold over 1500 copies, but *Zork* really became a hit after the Apple II version sold over 6000 copies. With two supported platforms and more on the way, Infocom began to take off.

Company Culture

In 1982, Infocom moved to an office on 55 Wheeler Street in Cambridge, Massachusetts, and soon a jocular, freewheeling company culture emerged. Most of Infocom's twenty-one employees were in their mid-20's, and they came from diverse backgrounds. Berez and Anderson had majored in electrical engineering and computer science. Blank had studied biology. Lebling came to LCS with a degree in political science, while Galley graduated from Caltech University with a physics degree. Mike Dornbrook, who was a friend of Berez and Blank, joined the company after receiving his MBA from Harvard Business School. Dornbrook's roommate, Steve Meretzky, also joined. (Meretzky originally detested computers, but he grew to love *Zork* after Dornbrook brought home an Apple II.) Donning Hawaiian shirts, Dave "Hollywood" Anderson came from Southern California as a product tester.

The youthful energy of Infocom created a general feeling of excitement within the company. The employees loved applying their creative skills to make computer games and they came into work with a sense of humor and a laid-back attitude. With sales exceeding \$160,000 in 1981, they had every reason to feel good.