Virtual Muse

Charles O. Hartman

Published by Wesleyan University Press

>>>> / 3 /

THE SINCLAIR ZX81

The first computer I ever owned was a Sinclair ZX81. It was a pretty remarkable machine. If you built it yourself, it cost $49.95; in 1981 a "serious" computer might cost a hundred times that. It had one kilobyte of memory. That means that it could deal with about half a page of data at any one time, or a few hundred computer instructions, or a little bit of each. Desktop computers now hold thousands of times as much. External data storage was on a painfully slow cassette tape; the keyboard was a membrane about the right size for a two-year-old; its output was a fuzzy picture on a television set. Yet the CPU (the calculating heart of the computer) was the powerful Z80 chip, and the clever use of specially designed circuits called gate arrays allowed for a decent built-in BASIC language.

The directions for assembling the ZX81 were good and clear as far as they went, but it was a long afternoon I spent putting mine together. After several hours I figured out the main trouble. The instructions had omitted any step that connected the power supply to the computing circuits themselves. I picked a plausible spot to join them and for the hundredth time tried plugging the machine in. The screen of my old Heathkit black-and-white TV (another gadget I'd built) was still full of snow. I fiddled with dials and wires, trying to get the little white-on-black letter K (the ZX81's cursor) to appear. Quite suddenly what did show up was the beatific Mr. Rogers, crouched down behind his fish tank, wearing a skin diver's mask, and talking in a dreamy voice about all the colors of the fish, which to me looked gray on gray.

Eventually the computer worked as advertised. Besides the built-

in BASIC, you could use low-level assembly-language programming to fit somewhat larger programs into that nutshell of memory. Still, nothing very elaborate was going to come out of the Sinclair. It was a learning computer. And of course it was all mine; it was the first computer I could learn inside out.

One of the last programs I wrote on the ZX81, before I replaced it with a much roomier IBM PC, was a poetry composer. I choose the term with some care. In keeping with the idea of juxtaposition, modern poets sometimes talk about the poem not as a process or speech so much as a composition. Poets who do so are usually thinking in terms of an analogy with the visual arts. The artist's job is to *compose*, to place together in a meaningful arrangement a number of independent elements. Painters compose planes of color, shapes, collaged objects, and so on. The poet's equivalents might be words, lines, phrases, quotations—any pieces of speech that can be treated as separable and rearranged in some poetic "space."

What my poetry composer arranged was lines; the BASIC program was called RanLines. It let you type in twenty short lines, which it stored in an internal array. Then, each time you pressed a key, the computer chose one of the lines at random and printed it on the screen. This is about the simplest possible kind of "computer poem." Yet even this beginning exercise raises a couple of points that remain important in far more sophisticated programs.

One of the Greek oracles, the sibyl at Cumae, used to write the separate words of her prophecies on leaves and then fling them out of the mouth of her cave. It was up to the suppliants to gather the leaves and make what order they could. The products of my first experiment were a little like that:

THE RAMIFYING SUNLIGHT
FOR A NEW NAME AND ADDRESS
DEMANDS MINOR DISCRETIONS
BIRCH BRANCHES
OF A PIECE WITH THE LONG HAUL AND THE TREADMILL
PRETENSE OF URGENCY
KEEPING BEHIND IT ALL, ON TOP OF THE WORLD

AND THE TREE IS LILACS

AND THE TREE IS LILACS

WHATEVER YOUR PLANS, THE AFTERNOON
DEMANDS MINOR DISCRETIONS
THE OVERPLUS OF PLENTY
BIRCH BRANCHES
KEEPING BEHIND IT ALL, ON TOP OF THE WORLD

What the ZX81 program contributed to the act of writing poetry was a simple sort of randomness. This has always been the main contribution that computers have made to the writing of poetry. A little book called *Energy Crisis Poems*, published in 1974 in "an addition of 500 copies" with the subtitle "poetry by program / program by rjs," appears to have been generated in a similar way, as the introduction explains:

> the order of the lines within the poems was selected at random by the program from the input through an explosion of the least significant bits of the interval timer, which is updated every 26.04166 microseconds. The odds against an identical set of poems being created using the same input & parameters are approximately 591924732522520960000000000000-000000 to 1 . . .

What the "input" was, the explanatory note doesn't say. But my guess is that the program, like its predecessor the Auto-Beatnik, was given some syntactical templates—

[nouns] of [adjective] [noun]
[verb] [adverb]
from [adjective] [noun] to [adjective] [noun]

—together with some lists of words and perhaps phrases. The computer would then choose among these lists of parts at random, and each random collection that appealed to the programmer's aesthetic sense would go into the book:

compoundings of senile enlightenment
flagellate emptily
from elemental origins to computer complexities . . .

these frayed lines you see bending
elastically whip and snap
in this ecstatically splintered womb

The principle is the same one used by Steve Allen in the old "Mad Libs" game.

Other early computer poetry reads much like this and feels as though it was produced in the same way. Haiku were especially common. The reasons are clear from the history of modern Imagist poetry. As both poets and programmers have realized, for different reasons, the reader's mind works most actively on sparse materials. We draw the clearest constellations from the fewest stars. So the nonsense factor is low for a tiny collocation of words that can be imbued with imagistic significance. It's hard to put together two words that don't make some kind of sense to the willing reader. If the language goes on longer, we begin to expect more discursive sense, and we more quickly grow suspicious of randomness.

My Sinclair ZX81 version of a poetry composer gave the computer an especially simple version of the random-language task, as befit an especially simple computer. I didn't ask it to fill in any blanks or to make choices on more than one linguistic level (both words and phrases, say). None of its choices depended on other choices. All it did, moment after moment, was to reach into its little bag of lines and pull one out. If it pulled out the one built-in blank line, there was a stanza break—at random. It would often repeat lines, as in the sample output given earlier. It had no memory of what it had just done. It could have no sense of structure in what it was putting together. To put it another way, the program could produce a simplistic kind of poetry forever, but it could never, by itself, produce a poem. All sense of completeness, progress, or implication was strictly a reader's ingenious doing.

Well, not strictly. A clever choice of lines for the input could help. The more discrete and self-contained the syntax of the line (complete clause, complete prepositional phrase), the more easily it joins with lines before and after. Keeping verb tense the same increases the opportunities for coherence. Short sharp images stand alone better than bits of narrative or argument.

The lines used for the sample I gave before favor variety and disjunction. A different store of lines could favor continuity instead. But continuity isn't closure. Only the act of a person, deciding to stop the program, establishes a defining boundary for the poem. The decision to stop (and maybe to select, to edit the output) can be completely arbitrary. The user can decide beforehand to produce exactly fourteen lines or can respond intuitively to the computer's output as it appears. But a human decision, though its motivation may be unknowably complex or obscure, can't be random.

The very simplicity of the program shows especially clearly the fundamental mystery of randomness in the writing of poetry. This is a thread running through all computer poetry, and it's worth examining in more detail.

There are other methods besides computers for introducing randomness into compositional processes. John Cage was a pioneer in the use of dice and other aleatory, or chance, procedures in music composition. Quite a while ago, Jackson Mac Low followed Cage's lead, beginning with music but soon turning back to his own art of poetry. Both men have been practicing Zen Buddhists, though neither seems very interested in what Mac Low calls the "spooky" aspects of Zen. Still, as Buddhists they see the workings of the universe in ways that diverge from the Cartesisan deterministic tradition of Western science.

For determinists, anything random is mere noise, just crud gumming up the clean machine of the universe. But the non-Aristotelian logic of Buddhism (and many other cultural traditions) finds a different place for accident. To someone who believes that the universe is thoroughly coherent and that human action is consistent with that coherence, dice fall the way they must at any particular moment.

Different views of propositional logic entail different ideas about causality. From one scientific perspective, the idea of meaningful coincidence is simply superstition:

1. We threw those virgins into the volcano
2. It stopped erupting
3. We'll act more promptly next time.

(I owe the example to my colleague Julia Genster.) The fact that this fallacy has a Latin name—*post hoc, ergo propter hoc*—shows that we've recognized it for a long time. In contrast, Carl Jung (a Western psychological scientist with Eastern proclivities) proposed the concept of *synchronicity*: that events occurring at the same time are fundamentally connected, and examining one of them can lead to insights about the other. Jung wrote the preface for a famous edition of the *I Ching*, the Chinese book of divination that depends on this kind of faith in the order of things. Casting one particular hexagram at one particular time reveals a truth about that moment.

When two events occurring at the same time are one person's mental events, there's not much doubt that they're connected. In thought there are no accidents. "Random thoughts" are always linked by unconscious motivations. Freud (Jung's mentor but certainly never as radical a critic of determinism) declared that there are no accidents in *any* aspect of our behavior, including actions we disown, including things that simply "happen" to us. That's why there can be a "psychopathology of everyday life" and a psychoanalytic interpretation of dreams, which an earlier science would have written off as witchcraft.

"Happen" comes from a word that means "chance." The idea of synchronicity (and even the Freudian idea of unconscious motivaton) can be seen in two ways. Either nothing occurs at random, or random events are themselves meaningful. It's the latter idea—acknowledging randomness and finding meaning in it—that strikes many Western people as strange, irresponsible, and even frightening.

But for thousands of years people have been consulting chance for advice: throwing the *I Ching*, inspecting birds' entrails, opening the *Aeneid* or the Bible at random, and so on. However severely modern science condemns this as sloppy thinking, it has at least a firm old lineage.

And it turns out that science isn't so single-minded about all this. Einstein wanted to think it was: "I shall never believe that God plays dice with the world." But by rejecting the randomness at the heart of

quantum mechanics, Einstein, who set the course of twentieth-century physics, cut himself off from its progress. Subatomic particles behave in ways that are radically indeterminate and unpredictable, random not just incidentally but in principle. That, the physicists now assure us, is how the world really is.

Attuning themselves to how the world really is, is an old ambition of poets. Our earliest function was to keep the local gods happy by praising them. (It remains our most natural service, I think.) Though the alliance between prophecy and poetry isn't always to the advantage of either, it goes a long way back. And if the religion is Druidism or Buddhism or Quantum Mechanical Scientism, randomness becomes a plausible religious rite and a reasonable method for poetry.

Not many contemporary poets feel comfortable with such a grand reading of their role in the universe. And in any case, this is all off-duty theorizing. What a poet worries about while writing poems is likely to be more practical. At its baldest, the poet's problem is to write poems that will engage the attention of readers.

(A footnote here: What a poet's really conscious of while writing a poem, most of us will insist, is the poem itself, not the audience. The question is complicated, because the mind is always shifting in toward the heart of the poem and out toward the world it grows from and will grow back into. But the fact remains that few poets can keep going for very long on poems they don't think *anyone* will read.)

Poems are partly incubated in the warm matrix of tradition. Poets and readers share a half-tacit knowledge of this background. It supplies a context for the experience of poetry and a basis for communication. But this is a problem as well as a support. The same background of literary history that helps a reader to recognize a poem *as* a poem threatens to determine so much about it that it becomes boringly predictable. As Howard Nemerov puts it, "The poet's task has generally been conceded to be hard, but it may be so described as to make it logically impossible: Make an object recognizable as an individual of the class *p* for poem, but make it in such a way that it resembles no other individual of that class."

So a more direct use of randomness is to *reduce the level of probability* in the poems. If the next word in the line I'm writing comes at random, I can at least be sure that it won't be coming from a cliché.

Part of my hope is to surprise the reader; part of it is to surprise myself. The idea isn't just to make the process of writing more entertaining but to *authenticate* it. If I'm discovering, the reader is more likely to have a sense of discovery. Again, the problem is compounded by history. From our reading and our classrooms we learn classical canons of taste and value. When we're writing more or less classically constructed poems—and most poets still do, at least during a period of apprenticeship—these canons continue to operate more or less well. They tell us what we're doing, what it fits into, what comes next, and what it's worth. But these canons fall silent during our work on other kinds of poems. How do we know whether a given word, line, sentence is the *right* one to add to the poem when the poem is breaking new ground? If it really is "given," or "inspired" as we used to say, how do we know whether to trust the source of it? How do we tell false from true inspiration, dreams through the ivory gate from dreams through the gate of horn?

Then we're flying by the seat of the pants. This also makes revision difficult—not difficult in the old sense of being hard work but difficult to justify, point by point. That's one reason a lot of poets, following the lead of the jazz musicians, have become interested in improvisation. The improvisor can't edit but must fall back on the most basic standard of all: Is this interesting to me, right now? And this implies the hope that I can be, as well as the writer, a good enough representative of the reader to judge for both of us.

If I can't surprise myself, it's very difficult to interest myself (though of course, surprise doesn't guarantee interest). Furthermore, if I can be surprised, that ensures my closer alliance with the reader I'm standing in for. If using randomness makes me a little more passive—a little more obviously a judge than a creator—that's another similarity between me and the reader and another point of contact for the poem. I allow myself to be not perfectly in control.

None of this deliberation about strategy really explains the fascination of programs like the little one running on my ZX81. It doesn't account for the fact that I could fascinatedly watch the program produce a number of "poems" this way but never seriously consider adopting the program as my customary method of writing poems. Other poets who saw the machine's little trick seemed fascinated as well, but none of them ran out to buy computers as a result. Simple randomness won't suffice to shock language into poetry, though chance will play an important part in experiments we'll see later.

In the meantime, a more atavistic pleasure such programs give is that of delegating a human function to something else. We tell ourselves many stories about this: stories of the golem, of Frankenstein's monster, of robots, automatons, and so on. Through these machines we place ourselves in the role of creators as well as creatures.

But this secondary creation tends to make writers (and other "creative" people) nervous. Your self gets tied up in what you make. A computer that becomes too autonomous begins to feel like a usurper. Just who's in charge here after all? For instance, even this first simple program raised questions about authorship. Exactly who wrote the poem I presented earlier? Me? The computer? The program? Myself through the computer?

The title page of *Energy Crisis Poems* proclaims that "anyone with access to an IBM S360 or S370 running under OS or OS/VS can use the program exactly as it exists." We might hear a hint of participatory democracy left over (in 1974) from the sixties. The hint recurs in my fifty-dollar Sinclair computer. A touch of radical democracy, even of anarchy, is implicit in today's ubiquitous desktop computers. True, the proliferation threatens new ways of regimenting workers, and so on. But potentially, the microcomputer revolution of the late seventies extended the social revolutions of the previous decade. (That was hardly the manufacturers' main motive. But it was important to some of the designers, many of whom were hackers with a subversive bent.) In this atmosphere we might expect the privilege and heirophany associated with Authorship and Authority to come

under scrutiny. For Buddhists like John Cage and Jackson Mac Low, that curtailment of the ego's realm amounts to a liberation.

These questions come up in more interesting forms as later programs mediate in more complex ways between the programmer/poet and the final result. In the meantime, my story turns to quite a different use of the computer, a program with a quite different intellectual pedigree.