Alex Groce (agroce@gmail.com), Northern Arizona University

Martin Gardner's *The Last Recreations: Hydras, Eggs, and Other Mathematical Mystifications* collects the final set of Gardner's Scientific American "Mathematical Games" columns (in their final form, with addenda bringing readers up to the collection's publication date on each topic) for our delight and edification. It stands, in this *Passages* column, for all of Gardner's columns published between 1957 and 1981. These columns sometimes directly present ideas from computer science, Gardner's successors included computer scientists Douglas R. Hofstadter and A. K. Dewdney, and there are 5 entries under the name "Knuth, Donald E." in the index of this collection alone. However, it is not for the columns' relevance to computing, *per se*, that *Passages* is devoting this month to a kind of Gardner-Fest; rather it is because *Gardner's columns magically create software engineers and computer scientists out of otherwise perhaps quite normal people.*

In preparation for writing this column, I spoke to numerous software and security engineers and researchers, and other computer scientists (some retired, rich, from Google, even) and asked if they read Gardner. Some had not, of course, but a remarkably high percent, perhaps higher than for any other set of written works one might mention, immediately began explaining how, in high school, or elementary school, or their freshman year of college (precociousness varies without clear relation to future success of the child), they devoured Gardner, adored Gardner, feasted on Gardner, geeked out on Gardner. Tenured full professors, Haskell-coding smart contract analysts at Trail of Bits, startup warriors, ski bums who know too much assembly, and your own humble columnist: all fell under the spell of these brief write-ups of amusing mathematical ideas. In this column, we look at the recreational mathematics columns of Martin Gardner, and try to understand why.

The initiated don't really need this column, though it can be fun to indulge in nostalgia, and reading the column might inspire you to grab one of the collections and read, or re-read, it. I read *The Last Recreations*, the only collection I hadn't read (my high school and university libraries didn't have it) and had a very good time of it.

What does Gardner write about, ask the uninitiated? Everything under the sun, but with a core of (recreational) mathematical relevance. The collection considered here covers eggs, parabolas, symmetry, knots, directed graphs, group theory, the pigeonhole principle, prime numbers, minimal Steiner trees, and seemingly-infinite, but actually just really long, tasks, among other things. Gardner likes to quote Lewis Carroll, and created the beloved *Annotated Alice*, and it's certainly true that with Gardner the time has always come to talk ships, and shoes, and sealing wax, and cabbages and kings (checkers and chess show up at length here).

Gardner's quotations are not limited to Alice-whimsy. Part of Gardner's appeal is that he was a man of vast and humane reading, with a background in philosophy, not mathematics (in fact, Gardner never took formal mathematical training past calculus, and it is in part this that makes his columns readable for the excited/excitable 7th grader). Reading Gardner "turned me on" to

Boswell's *Life of Johnson*, the experimental literary works of the Oulipo, the poems of Longfellow, the fideism of Miguel de Unamuno, and the multifarious works of G. K. Chesterton (Gardner also wrote the *Annotated Man Who Was Thursday*).

Gardner has certain key sources for his topics. The mathematician John Horton Conway has 11 entries in this book's index, for example, a typical Conway-count. Gardner popularized some of the ideas for which Conway is best known: the Game of Life (a Turing-universal 2D cellular automata that became an all-consuming obsession for many "computer people"), Hackenbush, surreal numbers (about which Knuth wrote a very odd, but surprisingly readable, little novel), Penrose tilings (devised by Penrose, but much advanced by Conway), and the Monster Group (again, not actually found by Conway, but related to his wider group theory work, and the topic of his famous "Moonshine" conjectures that were later proven true, though not to Conway's satisfaction). Gardner, and the Gathering for Gardner, feature prominently in a fascinating recent biography of the eccentric and brilliant, suicidal and cheerful, Conway, *Genius at Play: the Curious Mind of John Horton Conway*. The Conway biography is written in an unusual, experimental style (mixing in many direct comments from Conway, who participated in the writing and sometimes frustrated it with his oddities), by Siobhan Roberts. While not suitable for this column, I strongly recommend it to Gardner-inclined readers. Other members of what was called Gardner's "Mathematical Grapevine" included Solomon W. Golomb, Bill Gosper (famed MIT hacker), Elwyn Berlekamp, Ronald Graham, Benoit Mandelbrot, Piet Hein, Scott Kim, Penn & Teller, M. C. Escher, Gregory Chaitin, H. S. M. Coxeter, Rudy Rucker, Ron Rivest, Raymond Smullyan, and, of course, Donald Knuth. Many of these names should be familiar to most widely-read software engineers.

This kind of general commentary and capsule biography (we could talk about how Gardner hated travel, so only attended the first two events in the on-going series of "Gatherings for Gardner," or how he challenged Reinhold Neibuhr) doesn't really give much flavor of reading Gardner, however. Let's dive into one column from this book, the examination of "Bulgarian Solitaire and Other Seemingly Endless Tasks." Termination (or lack thereof) is of great theoretical interest to the computer scientist, and occasional practical interest to the practicing software engineer. However, Turing's famous impossibility proof is of less practical relevance than might be imagined; a task that requires your computer to produce all possible bit-patterns in a 500MB block of memory takes a finite amount of time, but do you really care? This column looks at tasks that appear to be endless, impossible to complete, but actually just take a ludicrously long (perhaps unbounded) time to finish. One of the first examples presented is from a paper by logician Raymond Smullyan. Consider an infinite supply of pool balls, each labeled with a positive integer. There are an infinite number of balls with each label. You have a box containing a finite number of positive-integer-labeled balls. You are allowed to remove a ball, and replace it with *any* finite number of balls of lower number. Removing a 1-ball just removes it, of course. Now, it's easy to empty the box fairly quickly (linear removals in number of initial balls), but if you wish to delay, can you put off emptying the box forever? At any step you could take a single ball labeled *n* and replace it with a number of balls labeled *n-1* that is so large that it requires some novel notation for big numbers, something dwarfing Knuth's arrow

notation. Nonetheless, you must eventually empty the box, take it ever so long.

Note that this is forced (really delayed) termination, not merely probable, converging, termination. R. A. Lafferty wrote a story, “Been a Long, Long Time,” about a vacillating angel tasked with awaiting the proverbial six monkeys with typewriters producing, randomly, the works of Shakespeare. Over infinite time, the probability approaches 1, but it does so exceedingly slow; however, the termination is not forced; in principle randomness *could* avoid Shakespeare “forever,” and at any moment in that extraordinary duration, if Shakespeare has not been produced yet, there’s no sense in which it is closer to being produced; random has no state. Smullyan’s task, though of similarly mind-bogglingly long duration with a willful ball-remover, does have a measure that steadily moves towards “done,” though in a very peculiar way.

Is this relevant to computer science or software engineering concerns? Perhaps thinking of these tasks will help us understand better semi-algorithms or potentially long-running automated testing methods? No, I don’t think so. One of the things most impressed upon my mind by reading Gardner early in life is a passage from Boswell’s *Life of Johnson*:

> *“We then fell into a disquisition whether there is any beauty independent of utility. The General maintained there was not. Dr. Johnson maintained that there was; and he instanced a coffee-cup which he held in his hand, the painting of which was of no real use, as the cup would hold the coffee equally well if plain; yet the painting was beautiful.”*

Gardner sometimes talks about things that are useful to know (reading his *Penrose Tiles to Trapdoor Ciphers* isn’t the worst way to learn about RSA encryption), but most of the columns are truly concerned with *recreational* mathematics, “useless” math, at least for most software engineers. Yet reading Gardner in youth seems likely to spark an interest in the wide world of structure, beauty, complexity, and forge delight that paves the way to success in computational endeavors. Forget Knuth’s points about the utility of theory to practitioners, and vice versa; one thing that may not be essential to software engineering, but often accompanies the best of it, is a spirit of enjoyment, fun, and the thrill of thinking of complex, beautiful things. Martin Gardner did more to produce such a spirit in young, and old, minds that almost anyone else in the last century, and to read or re-read his columns can light or re-light that fire, any day you choose to pick up the book.