<http://www.newyorker.com/online/blogs/elements/2014/03/the-electronic-holy-war.html?mobify=0>

March 25, 2014

# The Electronic Holy War

Posted by *[Patrick House](http://www.newyorker.com/magazine/bios/patrick_house/search?contributorName=Patrick%20House" \o "search site for content by Patrick House)*

In May, 1997, I.B.M.’s Deep Blue supercomputer prevailed over [Garry Kasparov](http://www.newyorker.com/reporting/2007/10/01/071001fa_fact_remnick) in a series of six chess games, becoming the first computer to defeat a world-champion chess player. Two months later, the *Times* [offered machines another challenge](http://www.nytimes.com/1997/07/29/science/to-test-a-powerful-computer-play-an-ancient-game.html?emc=eta1) on behalf of a wounded humanity: the two-thousand-year-old Chinese board game *wei qi*, known in the West as Go. The article said that computers had little chance of success: “It may be a hundred years before a computer beats humans at Go—maybe even longer.”

Last March, sixteen years later, a computer program named Crazy Stone defeated Yoshio Ishida, a professional Go player and a five-time Japanese champion. The match took place during the first annual *Densei-sen,* or “electronic holy war,” tournament, in Tokyo, where the best Go programs in the world play against one of the best humans. Ishida, who earned the nickname “the Computer” in the nineteen-seventies because of his exact and calculated playing style, described Crazy Stone as “[genius](http://gogameguru.com/crazy-stone-computer-go-ishida-yoshio-4-stones/).”

The victory was not quite a Deep Blue moment; Crazy Stone was given a small handicap, and Ishida is no longer in his prime. But it was an impressive feat. As with computer chess in the nineteen-eighties, computer Go is dominated by individual programmers and small teams. Crazy Stone, for example, is programmed by one man, Rémi Coulom, a professor of computer science at [Université Lille 3](http://www.univ-lille3.fr/), in France.

No large company has invested yet in computer Go the way that I.B.M. did with chess. Peter Norvig, a director at Google Research and one of the founders of modern A.I., told me that, even if Google or I.B.M. hired a cadre of experts, invested “one hundred times more hardware than anyone else had ever applied to the problem,” and was “very clever about the system-design architecture, the exact machine learning algorithms, and the insights from neuroscience,” he doesn’t know if this would be enough to make the equivalent of Deep Blue for Go.

Go sounds simple. The board is a square with nineteen vertical and nineteen horizontal lines that cross, creating three hundred and sixty-one intersecting points. It starts empty, and two players take turns placing circular stone pieces on vacant intersections until the game is finished—either when both sides agree to end it or one player withdraws. Players score by surrounding intersections or by capturing the opponent’s pieces, each worth one point. Why is this so hard for computers? In chess, it takes fifteen moves for the number of possible game states to equal the number of stars in the universe. Go gets there in ten moves. At that point, the average chess game is nearly half over, but Go is only beginning: the average game still has a hundred and forty moves left, each adding a new universe of possibilities.

To say that Go is more complex than chess, though, is a little like saying that one infinity is [larger](http://www.businessinsider.com/the-different-sizes-of-infinity-2013-11) than another. While technically true—and mathematically possible—it does not fully explain why computers, which can’t fully compute chess or Go, have become good at one and not the other. “A few hundred orders of magnitude don’t matter when you’re up in the ten to the one hundred and twenty,” Murray Campbell, a member of the I.B.M. Deep Blue team, told me. The key lies in Go’s structure. Deep Blue was able to exploit a weakness in chess’s armor: at the grandmaster level, to tell who is winning, you add up the pieces on the board and consider their positions. Campbell explained that, to win, you just stay ahead the whole time, vastly reducing the number of moves for computers to consider.

*Wei qi* translates to “encircling game,” and one of its few rules is that, when a piece is surrounded by an opponent’s, it is removed from the board. The rule is transitive, so large groups of pieces can surround other large groups, and may themselves be surrounded. Part of the difficulty for computers—and humans—is that it is often hard to determine at any given time whether a group of pieces is being surrounded or doing the surrounding, and thus who is ahead. Feng-Hsiung Hsu, another member of the Deep Blue team, [writes](http://spectrum.ieee.org/computing/software/cracking-go) that the distinction is particularly hard for computers to grasp because “the status of one group can affect that of its neighbors—like a cowboy who points a revolver at another cowboy, only to find himself covered by a rifleman on a roof.” Without a clear understanding of who is ahead, programs like Deep Blue stutter. “All the machinery that was built up for computer chess is pretty useless,” Campbell said.

The first computer Go attempts were based on what can only be described as computational exegesis: Go strategies through the millennia are collected in aphorisms and proverbs, and some of them—“Never cut a bamboo joint,” “Don’t go fishing when your house is on fire,” “Never chase a dragon”—can be easily translated into lines of code. For example, “Don’t go fishing when your house is on fire” is interpreted for the computer as “prioritize local responses versus global search,” according to David Fotland, who has been writing Go programs since 1981. Other early Go programs used the human brain as inspiration: neuroscientists in Japan, China, and Korea put Go professionals [into](http://www.ncbi.nlm.nih.gov/pubmed/15820625) [brain](http://sheng-lab.psych.umn.edu/pdf_files/Chen_fMRI_GO.pdf) [scanners](http://www.ncbi.nlm.nih.gov/pubmed/15820625), but the very tasks at which the human brain excels—pattern recognition, learning, intuition—are some of the hardest unsolved problems in A.I. Two decades of early efforts amounted to programs that were effective on a small scale, for tactical fighting but little else.

Coulom’s Crazy Stone program was the first to successfully use what are known as “Monte Carlo” algorithms, initially developed seventy years ago as part of the Manhattan Project. Monte Carlo, like its casino namesake, the roulette wheel, depends on randomness to simulate possible worlds: when considering a move in Go, it starts with that move and then plays through hundreds of millions of random games that might follow. The program then selects the move that’s most likely to lead to one of its simulated victories. Google’s Norvig explained to me why the Monte Carlo algorithms were such an important innovation: “We can’t cut off a Go game after twenty moves and say who is winning with any certainty. So we use Monte Carlo and play the game all the way to the end. Then we know for sure who won. We repeat that process millions of times, and each time the choices we make along the way are better because they are informed by the successes or failures of the previous times.”

Crazy Stone won the first tournament it entered. Monte Carlo has since become the de facto algorithm for the best computer Go programs, quickly outpacing earlier, proverb-based software. The better the programs got, the less they resembled how humans play: during the game with Ishida, for example, Crazy Stone played through, from beginning to end, approximately three hundred and sixty million randomized games. At this pace, it takes Crazy Stone just a few days to play more Go games than humans collectively ever have. “I have to be honest: I still find it kind of magical, that it works as well as it does,” Campbell said. Ten years ago, he added, if someone had described Monte Carlo to him, “I would say that would never work.”

The “electronic holy war” will run once a year in Tokyo until 2017. This past weekend, at the second annual tournament, Crazy Stone faced Norimoto Yoda, a Japanese professional who has a reputation for slamming pieces onto the board—sometimes shattering them—to intimidate his opponent. Crazy Stone was given a four-move head start and, lacking the eyes and ears through which another player might have been intimidated, won by two and a half points. After the match, Yoda, through a translator, told me that he was grateful for Crazy Stone because it eased up at the end and allowed the game to be closer than it actually was: the result of randomness—or, perhaps, of the beginnings of pity.

*Correction: An earlier version of this post omitted chess pieces’ positions as a factor in determining who is winning a grandmaster-level chess game.*

*Patrick House is a Ph.D. candidate in neuroscience at Stanford University.*