



Oral History of Monty Newborn

Interviewed by:
Dag Spicer

Recorded: February 28, 2005
Mountain View, CA

Total Running Time: 1:01:08

CHM Reference number: X3109.2005

© 2005 Computer History Museum

Q: --February 28, 2005 in Mountain View, California. Monty is a huge figure, a seminal figure, in computer chess and has been organizing tournaments for 30 plus years. We're really happy to have you here.

Monty: It's my pleasure to be here.

Q: I wanted to start with a really general question for you. Why is computer chess an interesting problem for you, and for AI as well?

Monty: That's two separate questions. For AI, I think that you have to go back and look at the first people interested in computer chess, and their motivation. They were of course interested in the question of how intelligent could computers be. They were looking for a problem on which one could study how intelligent the computers could be and somehow, they backed into chess. David, and of course _____ and Shannon, and Norbert Weiner, and of course _____, even further back, was interested in this. Herb Simon, who won a Nobel Prize, became interested in it in the '50s. John McCarthy, now of Stanford, became interested in the '50s and '60s. Some of the leading names in the history of computers backed into looking at chess as the environment in which to study how smart computers could be at solving a difficult problem. I think that's the original motivation.

Q: It's amazing, the names that you mentioned. Every major figure in computer science, as you said, has had some contact with this particular problem.

Monty: Even Von Neumann. If you look at Von Neumann, he of course got into the theory of games. Von Neumann was fiddling around with the mini manix algorithm. That's a lot of impressive names that all focused on chess and computers.

Q: When you say they "backed into" this problem, what do you mean by that?

Monty: Well I guess, probably, maybe they all played chess a little bit, so maybe that was a common aspect of it. They didn't have to be real good chess players to do it, but chess was always sort of the game where one thought man's intellect was put to the real test. When you had a computer that followed directions and could do calculations relatively quickly, it seemed like chess was a good one to have these computations carried out on, so these early guys were trying to figure out how to program these computers to play chess. Of course, they thought it would unlock some of the secrets of the mind's way of computing. The bigger question, of course, is what does a human mind do that a computer can't do? Where is the distinction? What tells us, for example, that we're alive? A computer doesn't know it's alive. Is the idea of being alive a computation, or is it something more than a computation? We haven't been able to figure out what it is that tells us that we're alive, as far as I know, to this date. That's still an open question. Where in the brain do we figure out that we're alive? These are the big questions.

Q: They are. I'm sorry, I cut you off.

Monty: Going back to my own interest, it combined two of my lifelong interests, which were chess - which I was never a great player, but I was a reasonable player - and computers and programming. I enjoyed both of them. I guess my interest aroused in the '70s. As a scientist, the thing that I enjoyed about working in the environment of computer chess was just that you could put your ideas to a test, and that there's a lot of science in universities that probably should be put more to a test. The exciting thing in this long-term experiment we did was that we sort of weeded out the talk from the action. In the world of science there's also a lot of talk, and we filtered that out quite well. By the time the programs got to be at the level of the world's best chess players, the people involved were a real tough bunch of scientists, and a talented bunch of guys.

Q: We had Ken Thompson in a week or so ago. We did an oral history with him. One thing that really surprised me was his comment that he stopped playing chess in 6th grade. He doesn't play chess very much, and yet he was a major contributor to the field. That was very surprising.

Monty: Well, Shue [ph?] himself, the head of the _____ program, was just a fair chess player. He probably learned by training his computer. When you train the computer, you in turn learn yourself.

Q: Take us back to what got you and David Levy to organize this back in 1970.

Monty: In 1970, the ACM had a conference in New York City. They wanted me to be a part of the conference committee, and to organize the special events at this conference. We decided to organize several special events. I had a number of interests at that time. We organized the first tournament for computers. In addition to that, we organized a computer art contest, and we organized a computer music contest. Our first activity was a combination of a national competition for computer music, and a national competition for computer art, and the chess tournament. We did all three. It was quite exciting. All three were exciting but the chess, of course, maybe captured the imagination of the scientists. All three were exciting, but somehow we wound up sticking with computer chess. I organized the first tournament in New York with Ken King, who was the head of Columbia University's computer center at the time. The second tournament was organized by Ben Mitman [ph?] from Northwestern.

Q: Not David Levy?

Monty: Maybe David helped Ben on the second one. The second one was held in Chicago, where the Northwestern team lived. Ben had it in his own backyard, so to speak. The second tournament then was a big success, and the third tournament I was back involved in organizing as well. For a number of years, Mitman and I organized these tournaments together. I started competing myself in the Boston tournament, the third tournament.

Q: With Ostrich?

Monty: With Ostrich. Ostrich was not a bad program. It never quite reached the top, but it was not bad. Ben and I organized tournaments year after year. David Levy got involved with some. We organized the ACM tournaments yearly, and in addition we organized world championships every third year. This went on pretty consistently right up until the time that Kasparov played the computer. They're still having these tournaments, but my interest as a scientist sort of was fulfilled when Kasparov lost to the computer. As a scientist, I would say that that was the end of the experiment.

Q: How early on was defeating the world chess champion a goal of the computer chess people?

Monty: If you go back to Norbert Weiner, who was one of the first ones that wrote on the subject, the question was whether you could have a program, I think the word he used was play "reasonable" chess. You've got me on the spot on a perfect quote. I think that the real issue at first was whether they could play reasonable. Little by little, they kept upping the ante from reasonable to like the level of a good player, a master player, a grand master, and then could they actually beat the world champion. At every stage, there were those that said, "It'll never get any better. This is as good as they're going to get." Of course, these were, to some extent, grand masters holding their turf, or defending their turf, in some crazy way, in denial more than anything. The computers continually got better and better and better. They'll get better and better from here on out. The computer technology is marching along.

Q: These people that are still in the field, what's their goal now that they've dethroned the best human player in the world?

Monty: Well, those that are interested in artificial intelligence, from the sense of game theory, are looking at the game of Go. Those that are interested in the deeper question of artificial intelligence are back to this issue, as I said before, as to what distinguishes human intelligence from computer intelligence. I think that's the exciting question. The real exciting question is: What does the human mind do that the computer can't do?

Q: Is it fair to say that chess is solved as a problem?

Monty: As a problem, we found that to play chess better than man is a problem of computation. The computers can out-compute the human mind.

Q: This brings us to the big divide between the two approaches as I see them, brute force versus heuristics, or the encoding of game play. Kasparov talked about a harmony of the board that great chess players have. I think we understand what he means by that. Can you comment a bit on that, and the

fusion of those two approaches? I don't think it's all brute computation. There may be some symbolic reasoning that's going on as well.

Monty: Well, the bottom line in general is that, the faster the computer, the more possibilities it will look at. This increase in the number of possibilities eventually overweighs-- Well, the idea of heuristics is to narrow your search. It's to not look at as many possibilities, but to look at the more relevant ones. If you can look at enough possibilities, whether they're relevant or not doesn't matter. A faster computer will eventually catch up with the smarter, slower computer that uses heuristics. The problem with most heuristics as it moves to brute force is that the exciting moves are missed by the heuristics, because they go against the rules. It's like if you train somebody to do something using very simple rules. It's like school children. If you force them not to be creative, they'll solve most of the problems but they won't solve the difficult ones. What you want is some crazy guy whose mind is uncontrolled, that goes out and thinks of something that nobody else has been thinking about. That's the magic solution, so to speak. To some degree, the heuristics only go so far. They miss out the crazy scenarios that they're programmed to filter out. For example, one classic, very simple rule in chess is one of the heuristics in the early programs. It was don't move your knights to the side of the board because you don't want them getting stuck on the side of the board. Well, that rule works 98% of the time, but it's those 2% that it doesn't work that allows the program to do something exciting.

Q: Is that likely to arise only from a human player, or can a computer still come up with that 2%, that really innovative move?

Monty: Well, if you don't force it to follow this heuristic, it will then play those moves but it will possibly not be able to look at other moves more deeply. You have your choice of looking more deeply along high "interesting" lines, or deeper along all kinds of lines which may look totally stupid, but may pan out later. The more you force the computer to use strong heuristics, unless these heuristics are guaranteed to be successful-- I mean, each heuristic has a certain probability of being correct. When you're looking at millions and millions of positions, the odds are that one of those heuristics is not going to really be that relevant. The distinction between brute force and heuristics is not a real hard distinction. The question is: What moves are you giving up shallow in the search to look at others deep in the search? The odds of giving up a move shallow in the search to look at one deeper in the search, you have to be very careful about giving up something shallow in the search in the expectation of being able to search deeper along some line.

Q: How much were the algorithms shared through the 27 years from '70 to '97? Was there any published literature in computer chess that sort of created a discipline and a field?

Monty: There was a lot of sharing of ideas. There were several key ideas, of course. As the great experiment went along, of course the mini max was shared by thousands. That's the first perhaps big idea. The alpha beta algorithm became quickly spread around the community. The idea of using large

hash tables, or transposition tables, got spread around in the late '60s and early '70s. The Greenblat [ph?] program, perhaps, was one of the early ones. The key ideas - mini max, alpha beta, the use of large hash tables - that was then followed by perhaps the key idea, which was Peter Fry's idea of using iterative deepening search rather than straight depth-through search. The introduction of iterative deepening depth-through search moved the programs up from playing about a weak class A to almost master level in about a one or two year period. So, the iterative deepening was a key. Of course, following the iterative deepening, Ken Thompson's Belle came in with the special purpose hardware. As soon as Thompson came in with special purpose hardware, others followed right along. Following the special purpose hardware, one looked into a parallel search using many computers in parallel. All of these ideas were shared by others as one marched along. Furthermore, at our ACM conferences every year, we held panel discussions in which the ideas were pretty well shared, so we had our pretty much yearly panel discussions. There was a certain amount of material published in the major publications. In general, the community was a very close community. We had lots of fun together. Following the games, we generally went out to dinner and had a good time and talked. Early on, of course, there were the Russians involved. It was always exciting, getting together with the Russians and talking chess and politics. It was quite an exciting period.

Q: I was just going to ask you about the Russians. What were their contributions, and what was their environment like for developing chess? Obviously, it was the height of the cold war. 1972 was a big year.

Monty: Of course, chess in Russia is a much bigger thing than it is in North America. In Russia, there were basically two groups that got involved in developing chess programs. There was, of course, Vodvenic [ph?]. Vodvenic was the first. Following Vodvenic, there was the group, IETP, the Institute of Experimental and Theoretical Physics. That group was a talented bunch of guys. Vodvenic's group and the IETP group were in competition with each other. The IETP group eventually wound up playing McCarthy's program in a two game match in the 1966, 1967, I think the years were. The Russian program dominated very strongly. This was really the first test. The Russian program was more of a brute force program, and the McCarthy Kotok [ph?] program was really sort of a selective search program. One of Vodvenic's observations when he looked over the games was that the McCarthy Kotok program threw away the baby with the bathwater, in the sense that it filtered out the good moves that the computer should be playing. That was the first test of brute force versus selective search, and the brute force really dominated. That may not be the only reason, but it was one of the reasons. So the two Russian groups, the Vodvenic and the IETP group, competed, but the IETP group were really strong programmers. Vodvenic, as a chess player, was sort of a theoretician who, I think he expected too much out of his ideas. They were a bit complicated to program. His program never competed to the level that the IETP program did. Of course, IETP was joined by Misha Donskoy [ph?] at some point. When Donskoy joined the IETP group, Donskoy came up with his own variation of the program, called Kaissa [ph?]. Kaissa wound up winning the first world championship. They beat the Northwestern program. It wasn't clear that they were actually better, but sometimes the best team doesn't always win. They were good enough that, following the tournament, there was such interest. The two never played during the tournament. Following the tournament, the organizers decided just to have a one game match to see who would win between

the Northwestern program and Kaissa, and Kaissa won. I hope I'm right on that! You'd better edit that carefully.

Q: We'll check it.

Monty: Anyways, three years later, the Northwestern program was clearly better than Kaissa. Kaissa finished in the next world championship in the middle of the pack. Basically, after that it was never among the top bunch.

Q: When you look at the rankings of the winners in ACM and the world championships, you see that the various programs seemed to dominate for two to four years, then they started to come down. Belle was one. Chess Inversion 4.0 was another. Hitech won it once, and Deep Thought dominated a little bit. Is there any explanation for why the dominance comes from one team for a little while? Is it just better algorithms or better hardware?

Monty: It's energy. I think the design of a chess program takes a tremendous amount of energy. Also, you get tied into your data structures in your program. Once you've got a program up, running, and playing at a certain level, if you want to improve it, if you're going to make a major change, it takes a lot of work. To some degree, that was one issue. Also, for example, you have to look at the cases. Each case was a little different. Thompson's Belle was tied into hardware, so if he wanted to improve it, he needed to come up with better hardware. You have to come up with better hardware, pretty much. Better hardware could involve, for example, parallel. If you could multiply Belle, get 50 or 100 or 500 version of Belle tied together, then have them play as a team. I'm sure that's what was his next step if he had continued but somehow or other, let's say the Shue beat him to it. Of course, Hitech was looking at the same idea of parallelism, or hardware parallelism. Berliner's approach didn't have the potential of Shue's. Shue's was a simpler approach, I guess, and perhaps its simplicity allowed it to materialize. If you had too complicated of an idea, then you might not get it to work because it takes too much work. It was very interesting that there was this succession of people that moved to the top of the pack. You had to keep running at top speed or you wouldn't keep up. The ideas of one team were available to the next bunch. The sharing of ideas and the racing of technology, and the using of, for example, Cray Blitz got onto the Cray computers. When the Cray moved up to the top of the pack, Cray Blitz moved up to the top of the pack. Each group found it hard to stay on top because of this big race. The available information was there for everybody to share. Grabbing new technology was important.

Q: Seymour Cray used to say, "Try not to be first. Try to be second," because the first guy is taking all the risk and, as you say, is locked into his data structures, and his psychic and intellectual investment. That makes it harder to change gears. That second team that comes in can sort of cherry-pick what they like, and come up with a real powerful program. I found it was interesting that Belle could actually beat Cray Blitz. Belle was built for \$5,000 or something, and it beat a \$10 million supercomputer. That was amazing, I think, a very impressive feat.

Monty: The PCs of today are probably as good as Deep Blue was. Well, past Deep Blue there's been these two matches, one with Kramnick [ph?] and one with _____, with much weaker computing facilities than Deep Blue. Yet, they were able to tie the matches with these top world players.

Q: This brings up an interesting question. Now that players are training on computers using programs that you can run on a PC, programs that all but 100 people in the world can't defeat, the style of play has changed. Could you tell us a bit about that?

Monty: Well, I think that the tactics side gets tested much more when you play a computer than some of the positional aspects of the game. The one thing that the computer does is it leads the position into scenarios that the human is less likely to have seen when playing. When two people play each other, their minds are much the same. One of my classic observations is that, if I make a mistake, often my opponent won't see that I made a mistake because his mind is thinking like my mind. When I make a mistake when I'm playing chess, I'm saying to myself, "God, I hope my opponent doesn't see my mistake," and quite often he doesn't because his mind is thinking like mine. But the computer's mind, he's looking around mainly for tactical little things to grab, but he's also very clever positionally. Perhaps the balance between tactics and positional play has shifted more towards the tactical side, but the computers will just always outplay the human tactically.

Q: I forget which game it was, but there was one game in one of the two Kasparov Deep Blue matches where the computer actually made a mistake. It was a programming error. Kasparov thought that it was part of a larger strategy, that there was no way the computer would make such an elementary mistake. He said later that it affected his game play. I wish I had more details. There's an interesting example of the psychology occurring between the two players. I guess to play against a computer that ominous, you're already kind of freaked out about what it's really doing and how it's going to attack you.

Monty: Well, you know when you play a computer that you have to be very, very careful. Of course, the computers are way better than I am at this point. You know that you're in trouble when you play. Basically, it's like an octopus. It's got these tentacles all over the board, ready to grab things when you're not looking, because it reaches its arm back and grabs your piece that you're not imagining. You've got to focus on everything. Its mind is focused on everything. The human mind isn't as flexible in certain respects. We're trained to look for certain things. We're trained to look, for example, at structures where the king is castled. If the king isn't castled, we get to structures on the board that are less familiar. People study pattern recognition. There are certain patterns in chess that we're programmed to watch for. The computer doesn't have these patterns, and it could care less about these patterns. If you want to worry about these patterns, then you're missing out on worrying about other things that perhaps you should be worrying about.

Q: That brings me to the question of endgame databases, for example, or databases of game play generally. How are those are integrated into game play?

Monty: The Deep Blue program had a certain number of endgame databases built into it. These endgames that are usually for four, five, and six pieces on the board. Whenever the game only has four, five, or six pieces on the board, the computer knows just what to play. It will play perfect chess. The problem is that often you get a position of maybe eight pieces on the board. The computer, during its search, will find that if it trades off a few pieces, it winds up with only six pieces on the board after three or four levels of search. It knows, from that point on, the perfect play. Even if it's not in a position with six pieces on the board, it can often play essentially perfect chess with seven or eight pieces on the board because, by the time trades get resolved, you'll be down into its databases.

Q: Does that mean that it's deliberately giving up pieces in order to get to that point?

Monty: No, but it's very happy. If the position looks pretty equal and it's complicated, it doesn't mind getting into these endgames at all.

Q: What impact did Ken Thompson's endgame databases have on computer chess?

Monty: Well, his endgame databases of course fit into Deep Blue. One of the questions was whether they helped Deep Blue. There was one game where they were a small factor. Of course, from the standpoint of the rules of chess, they shook them up. Chess has a rule that says that, after 50 moves, if nobody can take a piece off the board, the game's considered a draw because pieces are just dancing around the board. Ken's database showed that, in many endgames, you could still win the game but you had to dance around the board for more than 50 moves. The chess community had sort of said, "Well, if you can't win a game in 50 moves, it's a draw," then had to come back and argue that you can win a lot of games when you have to dance around for 50 moves. I've lost track of how they've resolved that, but that was a problem in the rules of chess created by Ken.

Q: Is it fair to guess that they raised the number?

Monty: No, I don't think they raised the number. I've lost track of what they've done.

End of CHMO-63