August 83

MICRO CHESS

UNIVERSITY CHALLENGE

Tony Harrington hears how Martin Bryant made the transition from novice computer programmer to circumspect businessman.

There are a growing number of chess programs for home computers and the latest will shortly be released for the BBC Microcomputer. Its author, Martin Bryant, first became interested in computer chess as a computer science student at Manchester University.

Bryant took a course in Pascal programming as a standard part of his degree course in 1977. He had been school chess champion and, within a few months of beginning the degree, it struck him that writing a chess program might be more interesting than simply making a blob move about a screen.

'I wrote my first program half way through my first year,' he explained. 'It didn't work because I didn't know anything about the theory of writing chess programs. I didn't know what minimax theory was or what scoring functions or move generators were. The program that I came up with didn't even play legal chess. I lost interest for a while after that.'

What reawakened his interest was the discovery that the university mainframe, a Cyber 72, had a US chess program on it as one of its programs. This particular program was a few years old at that stage but had won the US 1971 Association for Computing Machinery computer chess tournament for mainframe programs. (Every year the ACM holds a tournament at one centre or other in the US.) 'It took me a while to stumble on the program because it was hidden away in one of the systems programmer's storage files,' he said. 'But the programmer left a listing of it lying around and I found it. I thought it looked interesting and that I could pick up some tips from it. I also bought a book by Monro Newborn called *Computer Chess*. That taught me all about minimax, scoring functions and all the other good things you need to know about to put a decent chess program together. I wrote White Knight Mk 2, my second chess program, incorporating these things. It was a better program and it played legal chess — not very well but it was at least up and running.'

That took him to the end of his first year. During the summer vacation he didn't have the use of the university machine, but he used the time to rewrite the program on paper. This time the program included all the frills expected of chess programs, such as castling and en passant pawn captures. 'The usual way to design a chess program is to worry about getting the move generators and the other essential bits and pieces working. You can always put in the extra bits later — and this was the approach I had taken with the Mk 2 version,' Bryant said.

When term started again the new, Mk 3, version was typed into the computer. This was the first program Bryant felt confident enough of to release on the University system for his fellow students to play against. Most of them did badly against the machine. This pleased him but, as he himself put it, their losses didn't exactly prove that the program was brilliant, since their chess was fairly poor.

This version lasted six months. It had a few flaws, in that it didn't understand draw by repetition. 'I also built more chess knowledge into the scoring function. For example, I told it little things that I hadn't put in before, such as that rooks on the seventh rank would be strong in most positions.'

This helped on the program a little as far as playing strength was concerned. The modified version, naturally enough, became the Mk 4. It was the first of his programs that could beat him. By the end of Bryant's second year this version, too, was ready for a rewrite. All this makes it sound as if he did nothing at university except beaver away at his chess programs, but he did manage to keep passing his exams, although he admits that it was hard going at times.

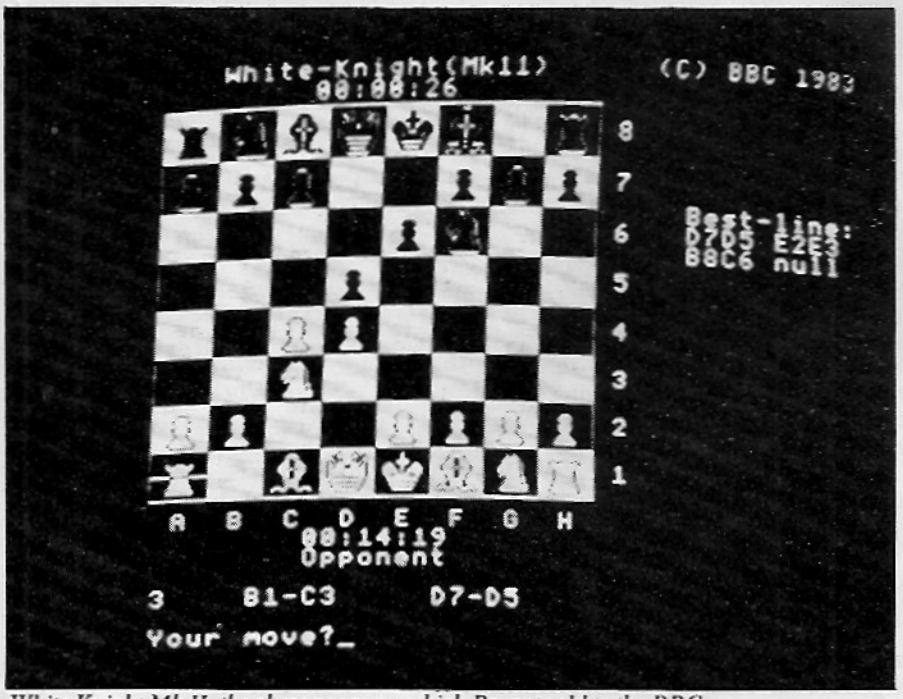
Bryant explained: 'I was totally dedicated to the chess program. I don't think that I am unique in this, because other chess programmers that I know tend to take the same attitude. Perhaps we are a weird bunch, but designing a chess program is a pretty obsessive thing. I would work at it six or seven hours a day as well as going out and also fitting in some study. It was like having a job on top of everything else. When I say that it is an obsessive business, I mean that you might find yourself leaving it alone for four months, then suddenly you start waking up in the middle of the night with new ideas and the work starts again.'

The attraction of trying to improve the Mk 4 proved just as irresistible as it had on the earlier models. But at this stage the resulting Mk 5 version meant that after two years of evolution the program was a mass of amended routines. Bryant decided that he had to throw away the program and start again from scratch, incorporating all that he had learned so far.

The summer holidays at the end of the second year were spent in the same way as the previous year — rewriting the program on paper from the bottom up. At the start of his third year, Bryant once again typed in his new version. The Mk 6 benefited from the rewrite. The program was much more efficient and ran a great deal faster. 'We had limited access to the mainframe as students. There was no time for full, tournament length games against the program. Five second chess was all we had time for and even then we ran into trouble from time to time for hogging machine time,' Bryant said.

The new Mk 6 had a deeper program search function. Computer chess programs tend to search on a brute force basis for the first few moves and then follow up high scoring lines in more depth. The Mk 6 could carry out more extensive searches than the previous model. It was released on the system and it did reasonably well against those students who felt like taking it on.

At this stage Bryant took a break from writing chess programs. He decided to try something different — like writing a chess learning program. 'I typed in 50 positions from grandmaster games and told it the moves the grandmasters had made from there, on the assumption that these would



White Knight Mk II, the chess program which Bryant sold to the BBC.

be the best moves in the circumstances. I told the scoring function to change its parameters to bring them into line with the grandmaster move (ie, if it would have previously given that move a low scoring value, it was instructed to replace that value with a higher value). The scoring function did succeed in changing its values. There is very little theory on learning programs. Most chess programmers tend to feel that chess learning programs are too slow a way of teaching chess programs the more esoteric points of chess. Certainly, trying to have a chess program learn from its own games would be a painfully slow way of doing things. But this seemed to me to be a valid short cut.'

Bryant reckons that the program took him two months to write. At the end of that time, he took the new 'educated' scoring function and used it to replace the Mk 6 scoring function. He then played a series of eight games between the old style Mk 6 and the new Mk 7, complete with its grandmaster scoring function. The results were six-and-a-half to one-and-a-half in favour of the Mk 7, which Bryant saw as reasonable proof of the new scoring function having some effect.

'The results of the learning program were rather surprising though. Some of the moves the program came up with did not seem very sensible to me, but it still beat the old style Mk 6. It might be that the moves simply looked odd to me because of my limited skills as a chess player,' he commented.

This took him to the end of his third year. That summer holiday he once more rewrote the program, improving some of the functions and changing some facets of it. In the fourth year, besides typing in the Mk 8 version of White Knight, he also joined the University chess club.

'I had spent the last three years playing nothing except computers, and the difference between their play and human chess was immediately brought home to me. Computers don't play attacking chess right from the opening. They tend to like a quiet position. The chess club reminded me how much my chess had gone off. At school I reckon that I was around 140. Now, although my chess has picked up a bit again, I reckon that it is still no better than 120.'

Despite joining the chess club, Bryant decided in his final year that finishing his degree was marginally more important than getting out yet another version of White Knight. Very little work was done on the chess computer that year. Towards the end of his time at university, though, it occurred to him that he would soon be out of reach of the Cyber 72. 'I decided to buy an Apple II. I also taught myself 6502 assembler code.'

After graduating, he went to work for a tool manufacturing company in Basingstoke as a programmer in the DP department.

While working at Basingstoke, he

Martin Bryant – a fervent computer chess supporter

UNIVERSITY CHALLENGE

started planning his first tournament entry. 'I was keen on entering a program into the 1981 *PCW* show. I translated the program from Pascal to assembler, but in the process I had to cut a great chunk out of the algorithm to squeeze it onto the micro. Other aspects of the program suffered as well.'

Bryant was so keen on getting his program into shape for the tournament that he left his job to have more time to work on it. The resulting program wasn't as good as the Mk 8, but he never had the chance to see just how it compared to it.

The Mk 9 went into the *PCW* show, but its performance was less than great. It came 10th out of 12. 'I was a bit disappointed,' he said. 'As far as a mainframe program was concerned, White Knight was rather good and I expected it to be better than most micro programs even in its cut down form. But that year Cyrus appeared, a program written by Richard Lang, and it beat everything in sight and won with a score of five out of five.'

As it turned out though, the *PCW* show gave Bryant something to replace his lost job at the machine tool factory. 'I met David and Kevin from Intelligent Software at the tournament. They were looking for chess programmers and offered me a job. I went to work for them in October 1981. Richard Lang was already working for them. Over that next year we pooled our knowledge. Our approaches to computer chess programming were very different but there were areas inside both our programs where we could draw on one another's work. Lang developed Cyrus II while I went on to produce Mk 10. This was designed specifically to fit on a micro. I rewrote the algorithms and the scoring function specifically with a microcomputer in mind. In the end it fitted inside 36k, while the Mk 9 had barely squeezed inside a 48k Apple.'

Bryant entered the Mk 10 into the 1982 *PCW* show. The machine did reasonably well and he collected the prize for the second best amateur program, with a score of three-and-a-half out of seven. The winning amateur entry ran on a much more powerful computer, so there was no real comparison between the two programs.

More important than the final placing though was the fact that at the 1982 show, Meyer Solomon, publications manager at the BBC, contacted Bryant and told him that he was interested in finding a chess program for the BBC micro. Bryant, naturally, was interested.

After further talks in October a contract was drawn up and Bryant began working on a translation (and an improvement) of White Knight Mk 10 for the BBC. The BBC micro was in many ways an ideal machine for his program. The Apple has a 1 MHz 6502 processor, while the BBC micro has a 2MHz processor. 'That was very exciting. Chess programmers slave away to try and get a ten per cent increase in speed in their programs, and to get a 100 per cent increase just by translating the program was incredible,' he commented.

The new version, called Mk 11, had a better scoring function and some minor changes were made to the algorithm. It also had a range of functions added to it. 'The final version which I sold to the BBC can forward step or backward step through the whole game to a maximum of 120 moves. Any legal chess position can be set up and the program will run an automatic check on the legality of the position — it won't let you play on without kings on the board, for example. It is also about five times faster, in problem solving mode, than any of the commercial programs I have seen.'

The screen display is light blue on black. One nice feature is that there are clocks for both sides incorporated in the program. There are no levels to be set up, since the program can be handicapped by giving it less time to complete the game.

One thing the program does not have is an openings book. 'This is a long standing argument in computer chess programming. An openings book tends to be of value in actual play only if it is very well set up. The BBC micro simply did not have enough space in its 32k for me to incorporate an openings book. 10k of the BBC's 32k goes to handle the screen while other functions also take a bite out of the available memory. I wrote the program to run inside 20k,' Bryant said.

Since selling his White Knight program, Bryant has been hard at work writing another program to compete in this year's *PCW* show. He intends running a program called Collosus on an Apple with a 3,85 MHz accelerator board.

'I have to design the whole thing from scratch, since I obviously can't just translate the old White Knight algorithms. I have to find a new approach and new algorithms. Whether the program will be ready or not in time for the show I don't know — but I am working on it!'

I hope he succeeds in his aim.

Games section

White: Micromurks; Black: White Knight; *PCW* Tournament, London 1982; French Defence; Notes by David Levy.

1	e2-e4	e7-e6
2	d2-d4	Ng8-f6

(Unusual, but in a computer v computer game it is not easy to refute irregular openings variations.)

3	e4-e5	Nf6-e4

(The knight cannot be trapped, for example, 4 f2-f3?? Qd8-h4+ 5 g2-g3 Ne4xg3.)

4	Nb1-d2	Bf8-b4?

(4 . . .Ne4xd2 5 Bc1xd2 c7-c5, would give Black an easy game.)

5	Qd1-g4?	

(This allows black to strand the enemy king in the centre.)

5	. . .	Ne4xd2
6	Bc1xd2	Bb 4xd2+
7	Ke1xd2	0-0
8	Ng1-f3	d7-d5
9	Bf1-d3	Nb 8-c6?

(White could now win with 10 Bd3xh7+ Kg8xh7 11 Qg4-h5+ Kh7-g8 12 Nf3-g5 Rf8-e8 13 Qh5xf7+ Kg8-h8 14 Qf7-g6, and Black can resign because of the simultaneous threats of mate on h7 and winning the queen by Ng5-f7+.)

10	c2-c3	f7-f5
11	e5xf6 e.p.	Rf8xf6
12	Ra1-e1?	

(Overlooking the full force of Black's reply.)

12	. . .	e6-e5!

(Winning material, because of the threat to White's queen and the threat of e5-e4, forking two pieces.)

13	Bd3xh7+	

(The best chance, but not good enough.)

13	. . .	Kg8xh7
14	Qg4-h4+	Kh7-g8
15	d4xe5	Rf6-f5
16	Qh4xd8+	

(White would do better to try to keep queens on the board with 16 Qh4-g3. Now Black's material advantage will be decisive.)

16	. . .	Nc6xd8
17	Rh1-f1	Nd8-e6
18	Kd2-c2	Bc8-d7
19	Rf1-g1	Ra8-f8
20	Kc2-b1	Bd7-a4
21	h2-h4	Ne6-c5
22	e5-e6	Kg8-h8
23	b2-b3	Ba4-b5
24	e6-e7	Bb5-d3+
25	Kb1-a1	Rf8-e8
26	Re1-e5	

Again allowing Black to trade pieces. It is surprising how many programs do not know that when you are behind in material you should try to avoid exchanging pieces!)

26	. . .	Rf5xe5
27	Nf3xe5	Bd3-e4
28	f2-f3	Re8xe7
29	f3xe4	Re7xe5
30	e4xd5	Re5xd5
31	Ka1-b2	Rd5-d2+!

(Black starts to mop up in the most efficient manner.)

32	Kb2-a3	Nc5-e4!
33	c3-c4	Ne4-c3
34	Rg1-c1	

(White cannot save both the a-pawn and the g-pawn.)

34	. . .	Nc3xa2
35	Rc1-a1	Na2-c3
36	Ka3-b4	Rd2-a2
37	Ra1-f1	Nc3-e4
38	Rf1-f8+	Kh8-h7
39	g2-g4	c7-c6
40	c4-c5?	Ra2-c2
41	Rf8-a8	a7-a6
42	Ra8-b8	Ne4xc5
43	h4-h5	a6-a5+
44	Kb4xa5	b7-b5!
45	Rb8-e8	Nc5xb3+
46	Ka5-b6	Nb3-d4
47	Re8-e4	Rc2-c4
48	Kb6-c7	b5-b4!
49	Kc7-d6	b4-b3
50	Re4-e1	Nd4-b5+
51	Kd6-d7	b3-b2
52	Re1-b1	Rc4-c2
53	Rb1xb2	

(White realised that other moves allow . . . Nb5-c3, etc)

53	. . .	Rc2xb2
54	Resigns	

END