

# My first Mandelbrot!

Written by Hans Summers  
Friday, 31 December 2010 10:30 -



I first came across the Mandelbrot set via an article in the July/August edition of Elektor Electronics, 1986. This was the third consecutive year that I had purchased this bumper summer issue of the magazine, containing over 100 projects. Two years earlier the summer issue had led, amongst other things, to my [sidereal clock project](#). The Mandelbrot article wasn't really an electronics project but it captured my interest immediately, particularly as I was the proud new owner of my first computer, an Acorn Electron. (I was forced to remove the article, [Read Why](#)).

## My first Mandelbrot!

Written by Hans Summers

Friday, 31 December 2010 10:30 -

---

The program listing in the article is written in BASIC for the Acorn Electron or BBC home computers, and I had it typed in and running in next to no time. The Electron is very similar to the more powerful BBC, but the BBC contains more I/O and an extra screen graphics mode. The last sentence of the article was somewhat cryptic: "Obviously, the suggested program allows a good deal of further patching and experimenting to arrive at even more attractively styled designs...". Eventually I made sense of this and enhanced the program so that you could use the keyboard to size and locate a zoom-in box on the screen, and the program would calculate the new parameters to enlarge that section of the mandelbrot set.

They were right that it took ages and ages to draw though. If I recall, it took about 15 hours of continuous computing time to draw the basic mandelbrot set, with a depth of 15 iterations. It was fascinating to watch the pixels being drawn one by one, with plenty of time to contemplate the emerging patterns and symmetry. Not that you could do it for 15 hours at a stretch. One thing I was surprised the original programmer had missed: I soon noticed the "**Length=SQR(A+B)**" in line 290 of the program. This is only used to determine the modulus of the complex number Z, in order to stop calculating iterations if it exceeds the value of 2. The squareroot is not necessary, it is equivalent to halt the calculation when  $A+B > 4$ , avoiding the squareroot calculation which I suspected would be a lengthy procedure for the computer (the Electron used an 8-bit processor, the 6502. The 6502 has no floating point support, and can't even multiply. Floating point arithmetic and especially things like squareroots therefore require many 100's of instructions). This suspicion turned out to be correct, when I removed the offending squareroot the Mandelbrot set completed something like 30% more quickly.

Still rather slow though. Well I'd heard that BASIC is an interpreted language. When you run it, every BASIC line must be translated into machine code for the microprocessor to run. A line of BASIC which is in a loop will get translated each time through that loop. This is wasteful both in terms of the inefficiency of translating so many times unnecessarily, and because the translated BASIC line will probably contain many error checks and so on which are probably unnecessary for your particular application. Writing machine code is therefore supposed to speed things up massively. But machine code is gobbledeegook to human eyes. The "Assembly" language is the best compromise: each line of assembly represents one machine code instruction, and translation is carried out only once, during a process known as "assembly".

Fortunately the Electron has a built in assembler. The assembly language program can be included in the BASIC program; when the BASIC program is run, the assembly language gets assembled into machine code, which is then ready in memory to be used as often as required, while the program is running. Equally fortunately, or probably a lot MORE fortunately, the Electron user manual is very comprehensive and even explains machine code and assembly language in great detail. It has a nice example of a subroutine for 32-bit multiplication, which

## My first Mandelbrot!

Written by Hans Summers

Friday, 31 December 2010 10:30 -

---

along the way naturally performs addition as well. Learning the memory mapping of the Electron screen pixels in the Electron's memory is all that is then required to start programming a machine code version of the Mandelbrot set. Plus, a lot of patience and hard work.

The results were impressive, the machine code version of the Mandelbrot set, when I had finally completed it, ran in about 3 hours. It ran in an Electron graphics mode having 320 x 256 pixels and 4 colours. The multiplication routine of the Electron user manual was a simple iterative shift-and-add type, and this was where the program spent most of its time. I did not attempt further optimisation of the program, which doubtless would have greatly reduced the time still further. Later when writing a similar program for the ZX Spectrum, I was able to complete the Mandelbrot set in 7 minutes, and even then further optimisations would have been possible.

This introduction to the Mandelbrot set was the onset of a disease, with which I am still infected to this day: the never-ending fascination of exploring an infinite fractal world where you are soon certain to be viewing a landscape never before seen by any other person.

A picture of my Electron, and more detailed scan of the program listing:

{gallery}mandelbrotfirst/1{/gallery}