MandelNet 1.0 by Roy Wood

©1992 Silicon Angst Software
122 Britannia Avenue, London, Ontario, CANADA, N6H 2J5

MandelNet is yet another Mandelbrot generator. However, the difference between MandelNet and all the other Mandelbrot generators is that MandelNet is designed to utilize networked computers to generate the Mandelbrot set in parallel.

Basically, MandelNet runs in two modes: slave mode and master mode. In master mode, the program scans the AppleTalk (or is that LocalTalk? I can never keep that straight) network and determines the number of slave "nodes" available. Then, the master node assigns subsections of the Mandelbrot image to each node. After each slave node has generated its image, the master reassembles the complete Mandelbrot image. N'est-ce pas? As for slave mode— well, you should have figured that out by now.

To use the program efficiently, find yourself a network of Macs, fire them all up with MandelNet, put one into master mode, then tell the master to "Draw Mandlebrot Set." With luck, the master will assign a section of the image to each slave on the network, and assemble the complete image.

Via the various menus, you can control things like shading, pixel size, number of iterations, graph limits, slave-polling frequency, etc. The meaning of each menu option is very obvious, so I won't say much more here. The only obscure option is "Master Participation," which controls whether the master node helps to generate the image, or just assigns the job to the slaves.

When MandelNet starts up, it defaults to slave mode. You can override this by holding down the option key as you start-up the program; this forces MandelNet to start up in master mode.

Moving the cursor around in the graph window shows the coordinates of the crosshair. You can also zoom in on a region of the Mandelbrot image by click-dragging in the graph. When you zoom this way, the image is expanded crudely, and you'll probably want to regenerate the image to get a better picture.

The Mandelbrot set is calculated via the standard, oh-so-slow, iterate-every-point algorithm. I know there are faster methods, but I'm not enough of a fractal junkie to really care about implementing them. I wrote this program because I was interested in the distributed-processing aspect of it. That's also why the program doesn't allow you to do anything like saving, loading, or printing images.

If you like the program, send $10 to the address listed in the about box; If you don't like it, send $20. Feel free to write me, too. I like getting mail. Just as a reminder, here's the address: Roy Wood, 122 Britannia Avenue, London, Ontario, Canada, N6H 2J5. (519) 438-3177.

"Live by the harmless lies that make you brave and kind, happy and healthy."

— from the Books of Bokonon.