

Code Feedback Wanted (Generating more garbage)

Source: <http://coding.derkeiler.com/Archive/Lisp/comp.lang.lisp/2004-03/2231.html>

From: David Steuber (david.steuber_at_verizon.net)

Date: 03/24/04

Date: Wed, 24 Mar 2004 10:57:22 GMT

I've done a bit more hacking on my Lisp code for generating the Mandelbrot Set. I've not been working on it constantly, that's why I'm not further along than might be expected from the last time I posted. I'm not using rationals anymore. I'm going with ints as a way to do fixed radix math. I've not implemented certain features yet like adjusting the number of bits used based on zoom factor. I'm also doing a fairly brute-force approach to calculating the pixels. I am considering using an array to hold the escape value data so that I can do edge detection on the set so that I don't have to calculate pixels in the interior. Anyway, here is the Lisp that I have:

```
;;; fractal.lisp --- an exploration of the Mandelbrot Set using
;;; arbitrary sized integer math.
;;;
;;; By David Steuber with feedback from comp.lang.lisp
;;;
;;; This software is put in the public domain by David Steuber
;;;
;;; Just for the record, I hate global variables because they
;;; break modularity by being non-local.

;; global variables related to the fractional bits in our fixed radix numbers
(defvar *fractional-bits* 16
  "Number of fractional bits in our number")
(defvar *escape-value* (ash 4 *fractional-bits*)
  "Numbers larger than this have escaped and are not part of the Mandelbrot Set")
(defvar *my-two* (ash 2 *fractional-bits*))

(defun set-fractional-bits (fb)
  "A nice way to set both *fractional-bits* and *escape-value* at the same time"
  (setf *escape-value* (ash 4 fb)
        *my-two* (ash 2 fb)
        *fractional-bits* fb))

(defun shifted-multiply (&rest args)
  "Multiplies the args as integers left-shifted by *fractional-bits* and right-shifts accordingly"
```

comp.lang.lisp: Code Feedback Wanted (Generating more garbage)

```
(let ((n (apply #'* args)))
  (ash n (- (* (max 0 (- (list-length args) 1)) *fractional-bits*))))

(defun mandelbrot-escape-fn (c-real c-imag &key (max-iterations 10000))
  "Return iterations for c-real + c-imag i as integers left shifted by *fractional-bits*"
  (loop for iterations below max-iterations
        for z-imag = 0 then (+ (shifted-multiply z-real z-imag *my-two*) c-imag)
        for z-real = 0 then (+ (- z-real-squared z-imag-squared) c-real)
        for z-real-squared = 0 then (shifted-multiply z-real z-real))
```