Homework – Due Monday

- Write a Scheme program using MrEd that accepts a list of atoms consisting of f, +, and -.
- Create an 800 x 500 pixel frame.
- Start drawing directly up from the middle of the frame.
- Draw each f as a 20 pixel line.
- Each + and – changes the direction of drawing by +90 degrees or -90 degrees.
  - Note: since these are all right angles, you do not need to use sin() and cos().
MrEd

MrEd is a Scheme implementation based on MzScheme

MrEd embeds MzScheme and extends it with a graphical user interface (GUI) toolbox.

GUI applications written with MrEd run without modification under Windows, Mac OS X, and Unix/X.
;Make a 300 × 300 frame
(define frame
  (instantiate frame%
    ("Drawing Example")
    (width 300)
    (height 300)
  )
)

;Show the frame
;("send" calls methods from outside a class.
(send frame show #t)
Canvas with Paint-callback

;Drawing in MrEd requires a device context (dc).

(define canvas

  (instantiate canvas% (frame)
    (paint-callback
      (lambda (canvas dc) (draw-lines dc))
    )
  )
)
Create a Pen

A pen is a drawing tool with a color, width, and style.

(define red-pen
  (instantiate pen% "RED" "RED" 2 'solid))
A Pen's Style

A pen's style is one of the following:

- 'transparent -- Draws with no effect.
- 'solid -- Draws using the pen's color.
- 'xor -- The pen's color is xor-ed with existing destination pixel values. The 'xor mapping is unspecified for arbitrary color combinations, but performing the same drawing operation twice in a row with 'xor is guaranteed to be equivalent to a no-op.
- 'dot
- 'long-dash
- 'short-dash
- 'dot-dash
- 'xor-dot
- 'xor-long-dash
- 'xor-short-dash
- 'xor-dot-dash
A Pen's Size

- A pen of size 0 uses the minimum line size for the destination drawing context.
- In (unscaled) canvases and bitmaps in unsmoothed mode, a zero-width pen behaves the nearly same as a pen of size 1.
- In a smoothing mode, a pen of size 0 draws a line thinner than a pen of size 1.
set-smoothing

Enables or disables anti-aliased smoothing of lines, curves, rectangles, rounded rectangles, ellipses, polygons, paths, and clear operations.

Text smoothing is not affected by this method, and is instead controlled through the font% object.
(define (draw-lines dc)
    (send dc set-smoothing 'smoothed)

    (send dc set-pen red-pen)
    (send dc draw-line 0 0 300 300)
    (send dc draw-line 0 300 300 0)

    (do ((i 0 (+ i 10))) ((>= i 300))
        (send dc draw-line 0 0 i 300)
    )
)
Test Cast 1

(drawSystem
  '(f + f - f + f - f + f))
Test Cast 2

(drawSystem
  '(f - f - f - f f f + f + f + f))

[Diagram of a square with two smaller squares inside it, representing the system drawn in the text.]
Test Cast 3

(drawSystem
  ' (f - f - f + f -
    f - f + f + f)
  )
Test Cast 4

(drawSystem

' (f - f - f + f - f - f + f + f - f - f - f - f + f + f - f + f + f)

)
Test Cast 5

(drawSystem

' (f f – f – f f +
    f f – f – f f +
    f f – f – f f +
    f f – f – f f )

)
Grading Rubric for HW-9

15 Points Total

[5 points]:
- Assignment appears to have good effort,
- ALL source is included in the submission including very standard helpers like atom?

[2 points each]:
- Given the input in test cases 1 through 5 on the previous slides, the output should match the given image with the smallest unit length being 20 pixels.
- The foreground and background colors can be anything (except both the same).