Graphics in Scmutils

We provide low-level support for plotting graphs and making simple drawings.

One can make a window, draw lines and points in the window, clear the window and close it.

A window is a data object that is made with the procedure frame. So, for example, one may make a window and give it the name win1 as follows:

```
(define win1 (frame 0 7 -2 2))
```

The window so constructed will have horizontal coordinates that range from 0 (inclusive) to 7 (exclusive) and vertical coordinates that range from -2 (inclusive) to 2 (exclusive).

Execution of the frame procedure will construct the window and put it up on your screen. However, you must give it a name so that you can refer to the window to draw in it.

Given such a window, you can use it to plot a function:

```
(plot-function win1 sin 0 7 .01)
```

This will plot in the window win1 the curve described by (sin theta), in the interval from theta=0 to theta=7, sampling at intervals of delta-theta=.01.

The general pattern is

```
(plot-function <window> <procedure> <x-min> <x-max> <delta-x>)
```

where <procedure> takes one numerical argument and produces a numerical value.

We can overlay other plots in the same window:

```
(plot-function win1 cos 0 7 .01)
```

If we want, we can clear the window:

```
(graphics-clear win1)
```

And we can make the window go away:

```
(graphics-close win1)
```

After a window is closed it is no longer useful for plotting so it is necessary to make a new one using frame if you want to plot further.

There are other useful procedures for plotting.

```
(plot-point <window> <x> <y>)

  drops a point at the coordinates (x, y) in the window.


(plot-line <window> <x0> <y0> <x1> <y1>)

  draws a line segment from (x0, y0) to (x1, y1) in the window.


(plot-parametric <window> <procedure> <t-min> <t-max> <delta-t>)

  draws a parametric curve.  The <procedure> must implement a
  function of one real argument (the parameter) and must return the
  cons pair of two numbers, the x and the y value for the given
  value of the parameter.
```

One can use the pointing device (mouse) to indicate a position. The procedure to interrogate the pointing device is:

```
(get-pointer-coordinates <window> <continuation>)
```

where <continuation> is a procedure that is called when the pointing device is positioned and a button is pressed. The continuation takes 3 arguments, the x-coordinate of the hit, the y-coordinate of the hit, and a designator of which mouse button was pressed.

For example:

```
(get-pointer-coordinates win1 list)
;Value: (.16791979949874686 .5037593984962406 0)
```

The value returned indicates that the left mouse button was pressed when the pointer was placed at the coordinates .1679... .5037...

The frame procedure takes a large number of optional arguments, allowing one to tailor a window to particular specifications. The default values shown below are for the X window system used with Unix.

```
(frame xmin             ;minimum x coordinate.        0.0
       xmax             ;maximum x coordinate.        1.0
       ymin             ;minimum y coordinate.        0.0
       ymax             ;maximum y coordinate.        1.0
       frame-width      ;width of window              400 pixels
       frame-height     ;height of window             400 pixels
       frame-x          ;horizontal screen position   750 pixels
                        ; of left edge of window.
       frame-y          ;vertical screen position       0 pixels
                        ; of top edge of window
       )
```