23. SIMPLE INTERACTIONS WITH THE CURSOR, A BITMAP, AND A WINDOW

The purpose of this chapter is to show you how to build a moderately tricky interactive interface with the various Medley display facilities. In particular how to move a large bitmap (larger than 16 x 16 pixels) around inside a window. To do this, you will change the CURSORINFN and CURSOROUTFN properties of the window. If you would also like to then set the bitmap in place in the window, you must reset the BUTTONEVENTFN. This chapter explains how to create the mobile bitmap.

GETMOUSESTATE Example Function

One function that you will use to "trace the cursor" (have a bitmap follow the cursor around in a window) is GETMOUSESTATE. This function finds the current state of the. mouse, and resets global system variables, such as LASTMOUSEX and LASTMOUSEY.

As an example of how this function works, create a window by typing

```
(SETQ EXAMPLE.WINDOW (CREATEW))
```

into the Executive Window, and sweeping out a window. Now, type in the function

```
(DEFINEQ (PRINTCOORDS (W)
     (PROMPTPRINT "(" LASTMOUSEX ", "LASTMOUSEY ")")
     (BLOCK)
     (GETMOUSESTATE)))
```

This function calls GETMOUSESTATE and then prints the new values of LASTMOUSEX and LASTMOUSEY in the promptwindow. To use it, type

```
(WINDOWPROP EXAMPLE.WINDOW ’CURSORMOVEDFN ’PRINTCOORDS)
```

The window property CURSORMOVEDFN, used in this example, will evaluate the function PRINTCOORDS each time the cursor is moved when it is inside the window. The position coordinates of the mouse cursor will appear in the prompt window. (See Figure 23.1.)

Figure 23.1. Current Position Coordinates of Mouse Cursor in Prompt Window

Advising GETMOUSESTATE

For the bitmap to follow the moving mouse cursor, the function GETMOUSESTATE is advised. When you advise a function, you can add new commands to the function without knowing how it is actually implemented. The syntax for advise is

(ADVISE *fn when where what*)

fn is the name of the function to be augmented. *when* and *where* are optional arguments. *when* specifies whether the change should be made before, after, or around the body of the function. The values expected are BEFORE, AFTER, or AROUND.

what specifies the additional code.

In the example, the additional code, *what*, moves the bitmap to the position of the mouse cursor. The function GETNOUSESTATE will be ADVISEd when the mouse moves into the window. This will cause the bitmap to follow the mouse cursor. ADVISE will be undone when the mouse leaves the window or when a mouse button is pushed. The ADVISEing will be done and undone by changing the CURSORINFN, CURSOROUTFN, and BUTTONEVENTFN for the window.

Changing the Cursor

One last part of the example, to give the impression that a bitmap is dragged around a window, the original cursor should disappear. Try typing:

```
(CURSOR (CURSORCREATE (BITMAPCREATE 1 l) 1 1]
```

into the Executive Window. This causes the original cursor to disappear. It reappears when you type

```
(CURSOR T)
```

When the cursor is invisible, and the bitmap moves as the cursor moves, the illusion is given that the bitmap is dragged around the window.

Functions for Tracing the Cursor

To actually have a bitmap trace (follow) the cursor, the environment must be set up so that when the cursor enters the tracing region the trace is turned on, and when the cursor leaves the tracing region the trace is turned off. The function Establish/Trace/Data will do this. Type it in as it appears (include comments that will help you remember what the function does).

```
(DEFINEQ (Establish/Trace/Data
    [LAMBDA (wnd tracebitmap cursor/rightoffset cursor/heightoffset
      GCGAGP)

      (* * This function is called to establish the data to trace
      the desired bitmap. "wnd" is the window in which the tracing
      is to take place, "tracebitmap" is the tracing bitmap,
      "cursor/rightoffset" and "cursor/heightoffset" are integers
      which detemine the hotspot of the tracing bitmap.
      As "cursor/heightoffset and "cursor/rightoffset" increase
      the cursor hotspot moves up and to the right.
      If GCGAGP is non-NIL, GCGAG will be disabled.)


(PROG NIL

      (if (OR (NULL wnd)
              (NULL tracebitmap))
          then (PLAYTUNE (LIST (CONS 1000 4000)))
               (RETURN))
      (if GCGAGP
          then (GCGAG))

        (* * Create a blank cursor.)

      (SETQ *BLANKCURSOR*(BITMAPCREATE 16 16))
      (SETQ *BLANKTRACECURSOR*(CURSORCREATE *BLANKCURSOR*))
```

```
          (* * Set the CURSOR IN and OUT FNS for wnd to the
          following:)

        (WINDOWPROP wnd (QUOTE CURSORINFN)
                    (FUNCTION SETUP/TRACE))
        (WINDOWPROP wnd (QUOTE CURSOROUTFN)
                    (FUNCTION UNTRACE/CURSOR))

          (* * To allow the bitmap to be set down in the window by
           pressing a mouse button, include this line.
           Otherwise, it is not needed)

        (WINDOWPROP wnd (QUOTE BUTTONEVENTFN)
                    (FUNCTION PLACE/BITMAP/IN/WINDOW))
        (WINDOWPROP wnd (QUOTE CURSOROUTFN)

          (* * Set up Global Variables for the tracing operation)

        (SETQ *TRACEBITMAP* tracebitmap
        (SETQ *RIGHTTRACE/OFFSET*(OR cursor/rightoffset 0))
        (SETQ *HEIGHTTRACE/OFFSET*(OR cxursor heightoffset 0))
        (SETQ *OLDBITMAPPOSITION*(BITMAPCREATE (BITMAPWIDTH
           tracebitmap)
                                              (BITMAPHEIGHT
           tracebitmap)))
        (SETQ *TRACEWINDOW* wnd]))
```

When the function `Establish/Trace/Data` is called, the functions `SETUP/TRACE` and `UNTRACE/CURSOR` will be installed as the values of the window's `WlNDOWPROPS`, and will be used to turn the trace on and off. Those functions should be typed in, then:

```
(DEFINEQ (SETUP/TRACE
    [LAMBDA (wnd)

      (* * This function is wnd’s CURSORINFN.
      It simply resets the last trace position and the current
      tracing region.  It also readvises GETMOUSESTATE to perform
      the trace function after each call.)

    (if *TRACEBITMAP*
      then (SETQ *LAST-TRACE-XPOS* -2000)
          (SETQ *LAST-TRACE-YPOS* -2000)
          (SETQ *WNDREGION* (WINDOWPROP wnd (QUOTE REGION)))
          (WINDOWPROP wnd (QUOTE TRACING)
              T)

      (* * make the cursor disappear)

      (CURSOR *BLANKTRACECURSOR*)
      (ADVISE (QUOTE GETMOUSESTATE)
            (QUOTE AFTER)
            NIL
            (QUOTE (TRACE/CURSOR]))

(DEFINEQ (UNTRACE/CURSOR
 [LAMBDA (wnd)

      (* * This function is wnd’s CURSOROUTFN. The function first
      checks if the cursor is currently being traced;  if so, it
      replaces the tracing bitmap with what is under it and then
      turns tracing off by unadvising GETMOUSESTATE and setting the
      TRACING window property of *TRACEWINDOW* to NIL.)

 (if (WINDOWPROP *TRACEWINDOW*(QUOTE TRACING))
      then (BITBLT *OLDBITMAPPOSITION* 0 0 (SCREENBITMAP)
              (IPLUS (CAR *WNDREGION*)*LAST-TRACE-XPOS*)
              (IPLUS (CADR *WNDREGION*)*LAST-TRACE-YPOS*))
          (WINDOWPROP *TRACEWINDOW*(QUOTE TRACING)
                                  NIL))

      (* * replace the original cursor shape)

 (CURSOR T)

      (* * unadvise GETMOUSESTATE)

 (UNADVISE (QUOTE GETMOUSESTATE]))
```

The function SETUP/TRACE has a helper function that you must also type in. It is TRACE/CURSOR:

```
(DEFINEQ (TRACE/CURSOR
 [LAMBDA NIL

      (* * This function does the actual BITBLTing of the tracing
      bitmap.  This function is called after a GETMOUSESTATE, while
      tracing.)

 (PROG ((xpos (IDIFFERENCE (LASTMOUSEX *TRACEWINDOW*)
              *RIGHTTRACE/OFFSET*))

        (ypos (IDIFFERENCE (LASTMOUSEY *TRACEWINDOW*)
              *HEIGHTTRACE/OFFSET*))

      (* * If there is an error in the function, press the right
      button to unadvise the function.  This will keep the machine
      from locking up.)

      (if (LASTMOUSESTATE RIGHT)
          then (UNADVISE (QUOTE GETMOUSESTATE)))
      (if (AND (NEQ xpos *LAST-TRACE-XPOS*)
               (NEQ ypos *LAST-TRACE-YPOS*))
          then

      (* * Restore what was under the old position of the trace
      bitmap)

          (BITBLT *OLDBITMAPPOSITION* 0 0 (SCREENBITMAP)
              (IPLUS (CAR *WNDREGION*)*LAST-TRACE-XPOS*)
              (IPLUS (CADR *WNDREGION*)*LAST-TRACE-YPOS*))

      (* * Save what will be under the position of the new trace
      bitmap)

          (BITBLT (SCREENBITMAP)
              (IPLUS (CAR *WNDREGION*)
                  xpos)
              (IPLUS (CADR *WNDREGION*)
                  ypos)*OLDBITMAPPOSITION* 0 0)

      (* * BITBLT the trace bitmap onto the new position of the
      mouse)

          (BITBLT *TRACEBITMAP* 0 0 (SCREENBITMAP)
              (IPLUS (CAR *WNDREGION*)
                  xpos)
              (IPLUS (CADR *WNDREGION*)
                  ypos)
              NIL NIL (QUOTE INPUT)
              (QUOTE PAINT))

      (* * Save the current position as the last trace position.)

          (SETQ *LAST-TRACE-XPOS* xpos)
        (SETQ *LAST-TRACE-YPOS* ypos]))
```

The helper function for UNTRACE/CURSOR, called UNDO/TRACE/DATA, must also be added to the environment:

```
(DEFINEQ (UNDO/TRACE/DATA
    [LAMBDA NIL

           (* * The purpose of this function is to turn tracing off
            and to free up the global variables used to trace the
            bitmap so that they can be garbage collected.)

            (* * Check if the cursor is currently being traced.
            It so, turn it off.)

      (UNTRACE/CURSOR)
      (WINDOWPROP *TRACEWINDOW*(QUOTE CURSORINFN)
                  NIL)
      (WINDOWPROP *TRACEWINDOW*(QUOTE CURSOROUTFN)
                  NIL)
      (SETQ *TRACEBITMAP* NIL)
      (SETQ *RIGHTTRACE/OFFSET* NIL)
      (SETQ *HEIGHTTRACE/OFFSET* NIL)
      (SETQ *OLDBITMAPPOSITION* NIL)
      (SETQ *TRACEWINDOW* NIL)

           (* * Turn GCGAG on)

      (GCGAG T]))
```

Finally, if you included the WlNDOWPROP to allow the user to place the bitmap in the window by pressing a mouse button, you must also type this function:

```
(DEFINEQ (PLACE/BITMAP/IN/WINDOW
    [LAMBDA (wnd)

      (UNADVISE (GETMOUSESTATE))
      (BITBLT *TRACEBITMAP* 0 0 (SCREENBITMAP)
           (IPLUS (CAR *WNDREGION*)
                xpos)
           (IPLUS (CADR *WNDREGION*)
                ypos)
           NIL NIL (QUOTE INPUT)
           (QUOTE PAINT]
```

That’s all the functions!

Running the Functions

To run the functions you just typed in, first set a variable to a window by typing something like

```
(SETQ EXAMPLE.WINDOW (CREATEW))
```

into the Executive Window, and sweeping out a new window. Now, set a variable to a bitmap, by typing, perhaps,

```
(SETQ EXAMPLE.BTM (EDITBM))
```

Type

```
      (Estab1ish/Trace/Data EXAMPLE.WINDOW EXAMPLE.BTM))
```

When you move the cursor into the window, the cursor will drag the bitmap.

(If you want to be able to make menu selections while tracing the cursor, make sure that the hotspot of the cursor is set to the extreme right of the bitmap. Otherwise, the menu will be destroyed by the `BITBLT`s of the trace functions.)

To stop tracing, do one of the following:

- Move the mouse cursor out of the window
- Press the right mouse button
- Call the function `UNTRACE/CURSOR`