

File created: 24-Oct-88 21:40:18 {ERIS}<BANE>LISP>EYECON.;18

changes to: (VARS EYECONCOMS EYECON.CONTROL.POINTS)
(FNS EYECON.TRACK EYECON.FIND.POINT)

previous date: 24-Oct-88 21:32:12 {ERIS}<BANE>LISP>EYECON.;17

Read Table: XCL

Package: INTERLISP

Format: XCCS

; Copyright (c) 1988 by ENVOS Corporation. All rights reserved.

(RPAQQ **EYECONCOMS**

```
((CONSTANTS EYECON.PUPIL.SIZE EYECON.LEFT.CENTER EYECON.RIGHT.CENTER EYECON.HEIGHT.OFFSET
  EYECON.RIGHT.WINK EYECON.WINK.WIDTH)
(VARS EYECON.PUPIL EYECON.MASK EYECON.OPEN EYECON.CLOSED EYECON.ICON EYECON.ICON.MASK EYECON.LEFT
  EYECON.BOTTOM)
(FNS EYECON EYECON.DIE EYECON.DRAW.PUPIL EYECON.FIND.POINT EYECON.ICON EYECON.OPEN EYECON.TRACK
  EYECON.WINK)
```

:: Check screen size and compute appropriate scale factor

:: All computations are scaled to keep them in smallp range

```
(VARS EYECON.SCALE.FACTOR EYECON.CONTROL.POINTS)
(P (BIND (MINSF _ 256)
  (RANGE _ (MAX SCREENHEIGHT SCREENWIDTH))
  FIRST
  (SETQ EYECON.SCALE.FACTOR 0)
  WHILE
  (ILESSP MINSF RANGE)
  DO
  (ADD EYECON.SCALE.FACTOR 1)
  (SETQ MINSF (LLSH MINSF 1)))
(SETQ EYECON.SLOPE.LIST (FOR P IN (CDR EYECON.CONTROL.POINTS)
  COLLECT
  (IQUOTIENT (LSH (CDR P)
    8)
    (CAR P))))
(PUSH EYECON.SLOPE.LIST (IPLUS (CAR EYECON.SLOPE.LIST)
  1))))
```

(DECLARE\ : EVAL@COMPILE

(RPAQQ **EYECON.PUPIL.SIZE** 10)

(RPAQQ **EYECON.LEFT.CENTER** 37)

(RPAQQ **EYECON.RIGHT.CENTER** 97)

(RPAQQ **EYECON.HEIGHT.OFFSET** 12)

(RPAQQ **EYECON.RIGHT.WINK** 82)

(RPAQQ **EYECON.WINK.WIDTH** 50)

```
(CONSTANTS EYECON.PUPIL.SIZE EYECON.LEFT.CENTER EYECON.RIGHT.CENTER EYECON.HEIGHT.OFFSET EYECON.RIGHT.WINK
  EYECON.WINK.WIDTH)
)
```

(RPAQQ **EYECON.PUPIL** )

(RPAQQ **EYECON.MASK** )

(RPAQQ **EYECON.OPEN** )

(RPAQQ **EYECON.CLOSED** )

(RPAQQ **EYECON.ICON** )

```
(RPAQQ EYECON.ICON.MASK )
```

```
(RPAQQ EYECON.LEFT 200)
```

```
(RPAQQ EYECON.BOTTOM 200)
```

```
(DEFINEQ
```

```
(EYECON
```

```
  (LAMBDA (EW) ; Edited 8-Oct-88 23:02 by jrb:
    (BITBLT EYECON.OPEN NIL NIL EW)
    (EYECON.DRAW.PUPIL EYECON.PUPIL EW EYECON.LEFT.CENTER EYECON.HEIGHT.OFFSET)
    (EYECON.DRAW.PUPIL EYECON.PUPIL EW EYECON.RIGHT.CENTER EYECON.HEIGHT.OFFSET)
    (WINDOWPROP EW 'PROCESS (ADD.PROCESS `(EYECON.TRACK ',EW))))
```

```
(EYECON.DIE
```

```
  (LAMBDA (EW) ; Edited 2-Oct-88 20:00 by jrb:
    (DEL.PROCESS (WINDOWPROP EW 'PROCESS))))
```

```
(EYECON.DRAW.PUPIL
```

```
  (LAMBDA (PM WINDOW DX DY MASK?) ; Edited 9-Oct-88 16:22 by jrb:
```

```
    ;; Draw the pupil
```

```
    (LET* ((EYECON.PUPIL.OFFSET (IQUOTIENT EYECON.PUPIL.SIZE 2))
           (BDX (IDIFFERENCE DX EYECON.PUPIL.OFFSET))
           (BDY (IDIFFERENCE DY EYECON.PUPIL.OFFSET)))
      (if MASK?
          [then] (BITBLT EYECON.PUPIL NIL NIL PM)
                (BITBLT EYECON.MASK BDX BDY PM 0 0 EYECON.PUPIL.SIZE EYECON.PUPIL.SIZE 'INPUT 'ERASE)
                (BITBLT PM NIL NIL WINDOW BDX BDY NIL NIL 'INPUT 'PAINT)
          [else] (BITBLT PM NIL NIL WINDOW BDX BDY NIL NIL 'INPUT 'INVERT))))
```

```
(EYECON.FIND.POINT
```

```
  (LAMBDA (DX DY) ; Edited 24-Oct-88 21:13 by jrb:
```

```
    (SETQ DX (IABS DX)) ; Edited 24-Oct-88 20:34 by jrb:
    (SETQ DY (IABS DY))
```

```
    (LET ((SLOPE (IF (EQ DX 0)
                     THEN (CAR EYECON.SLOPE.LIST)
                     ELSE (IQUOTIENT (LSH DY 8)
                                       DX))))
      (FOR S IN EYECON.SLOPE.LIST AS P IN EYECON.CONTROL.POINTS WHEN (ILEQ S SLOPE)
        DO (RETURN P))))
```

```
(EYECON.ICON
```

```
  (LAMBDA (EW OLDICON) ; Edited 8-Oct-88 21:37 by jrb:
```

```
    (LET ((EP (WINDOWPROP EW 'PROCESS)))
      (IF EP
          THEN (DEL.PROCESS EP)
               (WINDOWPROP EW 'PROCESS NIL))
      (OR OLDICON (ICONW EYECON.ICON EYECON.ICON.MASK))))
```

```
(EYECON.OPEN
```

```
  (LAMBDA (LEFT BOTTOM) ; Edited 8-Oct-88 23:01 by jrb:
```

```
    (LET ((EW (CREATEW ([create] REGION
                        WIDTH _ (BITMAPWIDTH EYECON.OPEN)
                        HEIGHT _ (BITMAPHEIGHT EYECON.OPEN)
                        LEFT _ (OR LEFT EYECON.LEFT)
                        BOTTOM _ (OR BOTTOM EYECON.BOTTOM))
            NIL 4)))
      (OPENW EW)
      (WINDOWPROP EW 'CLOSEFN 'EYECON.DIE)
      (WINDOWPROP EW 'RESHAPEFN 'DON'T)
      (WINDOWPROP EW 'ICONFN 'EYECON.ICON)
      (WINDOWPROP EW 'EXPANDFN 'EYECON)
      (EYECON EW)))
```

```
(EYECON.TRACK
```

```
  (LAMBDA (EW) ; Edited 24-Oct-88 21:02 by jrb:
```

```
    ;; Bind lots of variables
```

```
    (LET (EWR (ELX 0)
          (ERX 0)
          (EY 0)
          (OLDLX OLDLY OLDLB (LEFTLID T)
                (RIGHTLID T)
                (ELPX 0)
                (ERPX 0)
                (ELPY 0)
                (ERPY 0)
```

```

(ELPMASK (BITMAPCOPY EYECON.PUPIL))
(ERPMASK (BITMAPCOPY EYECON.PUPIL))
(OLPX 0)
(ORPX 0)
(OLPY 0)
(ORPY 0)
(ELDX ERDX EDY CP)
(WHILE T DO (BLOCK)
  ;; Refresh our knowledge of where the window is if it's been moved (relying on the hack that the region windowprop gets
  ;; changed when windows get moved)
  (IF (NOT (EQ EWR (WINDOWPROP EW 'REGION)))
    THEN (SETQ EWR (WINDOWPROP EW 'REGION))
          (SETQ ERX (SETQ ELX (FETCH (REGION LEFT) OF EWR)))
          (SETQ EY (IPLUS (FETCH (REGION BOTTOM) OF EWR)
                           EYECON.HEIGHT.OFFSET))
          (ADD ELX EYECON.LEFT.CENTER)
          (ADD ERX EYECON.RIGHT.CENTER)
          ;; Clobber OLDLX so we'll fix the pupils for sure after moving the window
          (SETQ OLDLX NIL))
  ;; See if mouse has changed and mung eyes
  (IF (OR (NOT (EQ OLDLX LASTMOUSEX))
          (NOT (EQ OLDLY LASTMOUSEY))
          (NOT (EQ OLDLB LASTMOUSEBUTTONS)))
    THEN (SETQ OLDLX LASTMOUSEX)
          (SETQ OLDLY LASTMOUSEY)
          (SETQ OLDLB LASTMOUSEBUTTONS)
          ;; first check the button state, as we can skip some stuff if eyes are closed
          (|if| (NOT (EQ 0 (LOGAND 1 OLDLB)))
            |then| ; Both are closed
              (|if| LEFTLID
                |then| ; Closing left now
                  (EYECON.WINK EYECON.CLOSED 0 EW)
                  (SETQ LEFTLID NIL))
              (|if| RIGHTLID
                |then| ; Closing right now
                  (EYECON.WINK EYECON.CLOSED EYECON.RIGHT.WINK EW)
                  (SETQ RIGHTLID NIL))
            |else| (|if| (EQ 0 (LOGAND 4 OLDLB))
              |then| ; Left is open
                (|if| (NOT LEFTLID)
                  |then| ; Opening it now
                    (EYECON.WINK EYECON.OPEN 0 EW)
                    (SETQ LEFTLID 'NOW))
                |else| (|if| LEFTLID
                  |then| ; Closing left now
                    (EYECON.WINK EYECON.CLOSED 0 EW)
                    (SETQ LEFTLID NIL)))
              (|if| (EQ 0 (LOGAND 2 OLDLB))
                |then| ; Right is open
                  (|if| (NOT RIGHTLID)
                    |then| ; Opening it now
                      (EYECON.WINK EYECON.OPEN EYECON.RIGHT.WINK EW)
                      (SETQ RIGHTLID 'NOW))
                  |else| (|if| RIGHTLID
                    |then| ; Closing right now
                      (EYECON.WINK EYECON.CLOSED EYECON.RIGHT.WINK EW)
                      (SETQ RIGHTLID NIL))))
              (SETQ ELDX (IDIFFERENCE OLDLX ELX))
              (SETQ ERDX (IDIFFERENCE OLDLX ERX))
              (SETQ EDY (IDIFFERENCE OLDLY EY))
          ;; See if we need to scale and do so
          (IF (OR (IGREATERP (IABS OLDLX)
                           255)
                  (IGREATERP (IABS OLDLY)
                           255))
            THEN (SETQ ELDX (RSH ELDX EYECON.SCALE.FACTOR))
                  (SETQ ERDX (RSH ERDX EYECON.SCALE.FACTOR))
                  (SETQ EDY (RSH EDY EYECON.SCALE.FACTOR)))
          ;; find out where on the magic circle the pupils need to land
          (|if| LEFTLID
            |then| (SETQ CP (EYECON.FIND.POINT ELDX EDY))
                  (IF (ILESSP ELDX 0)
                    THEN (SETQ ELPX (IMINUS (CAR CP)))
                    ELSE (SETQ ELPX (CAR CP)))
                  (IF (ILESSP EDY 0)
                    THEN (SETQ ELPY (IMINUS (CDR CP)))
                    ELSE (SETQ ELPY (CDR CP)))
                  (IF (OR (EQ LEFTLID 'NOW)
                          (NOT (EQ ELPX OLPX))
                          (NOT (EQ ELPY OLPY)))
                    THEN (|if| (EQ LEFTLID 'NOW)

```

```

      |then| (SETQ LEFTLID T)
    |else| (EYECON.DRAW.PUPIL ELPMASK EW (IPLUS
                                         EYECON.LEFT.CENTER
                                         OLPX)
          (IPLUS EYECON.HEIGHT.OFFSET OLPY)))
    (SETQ OLPX ELPX)
    (SETQ OLPY ELPY)
    (EYECON.DRAW.PUPIL ELPMASK EW (IPLUS EYECON.LEFT.CENTER ELPX)
      (IPLUS EYECON.HEIGHT.OFFSET ELPY)
      T)))
  (|if| RIGHTLID
    |then| (SETQ CP (EYECON.FIND.POINT ERDX EDY))
    (IF (ILESSP ERDX 0)
      THEN (SETQ ERPX (IMINUS (CAR CP)))
      ELSE (SETQ ERPX (CAR CP)))
    (IF (ILESSP EDY 0)
      THEN (SETQ ERPY (IMINUS (CDR CP)))
      ELSE (SETQ ERPY (CDR CP)))
    (IF (OR (EQ RIGHTLID 'NOW)
      (NOT (EQ ORPX ERPX))
      (NOT (EQ ORPY ERPY)))
      THEN (|if| (EQ RIGHTLID 'NOW)
        |then| (SETQ RIGHTLID T)
        |else| (EYECON.DRAW.PUPIL ERPMASK EW (IPLUS
                                         EYECON.RIGHT.CENTER
                                         ORPX)
          (IPLUS EYECON.HEIGHT.OFFSET ORPY)))
        (SETQ ORPX ERPX)
        (SETQ ORPY ERPY)
        (EYECON.DRAW.PUPIL ERPMASK EW (IPLUS EYECON.RIGHT.CENTER ERPX)
          )
          (IPLUS EYECON.HEIGHT.OFFSET ERPY)
          T))))))

```

(EYECON.WINK

```

  (LAMBDA (SOURCE LEFT WINDOW) ; Edited 8-Oct-88 23:23 by jrb:
    (BITBLT SOURCE LEFT 0 WINDOW LEFT 0 EYECON.WINK.WIDTH (BITMAPHEIGHT EYECON.OPEN)))

```

```

)

```

```

;; Check screen size and compute appropriate scale factor

```

```

;; All computations are scaled to keep them in smallp range

```

```

(RPAQQ EYECON.SCALE.FACTOR 3)

```

```

(RPAQQ EYECON.CONTROL.POINTS ((0 . 4)
                               (1 . 4)
                               (2 . 4)
                               (3 . 4)
                               (4 . 3)
                               (5 . 3)
                               (6 . 2)
                               (7 . 1)
                               (7 . 0)))

```

```

(BIND (MINSF _ 256)
  (RANGE _ (MAX SCREENHEIGHT SCREENWIDTH)) FIRST (SETQ EYECON.SCALE.FACTOR 0) WHILE (ILESSP MINSF RANGE)
  DO (ADD EYECON.SCALE.FACTOR 1)
  (SETQ MINSF (LLSH MINSF 1)))

```

```

(SETQ EYECON.SLOPE.LIST (FOR P IN (CDR EYECON.CONTROL.POINTS) COLLECT (IQUOTIENT (LSH (CDR P)
                                                                                       8)
                                         (CAR P))))

```

```

(PUSH EYECON.SLOPE.LIST (IPLUS (CAR EYECON.SLOPE.LIST)
                               1))

```

```

(PUTPROPS EYECON COPYRIGHT ("ENVOS Corporation" 1988))

```

FUNCTION INDEX

EYECON	2	EYECON.DRAW.PUPIL	2	EYECON.ICON	2	EYECON.TRACK	2
EYECON.DIE	2	EYECON.FIND.POINT	2	EYECON.OPEN	2	EYECON.WINK	4

VARIABLE INDEX

EYECON.BOTTOM	2	EYECON.ICON	1	EYECON.MASK	1	EYECON.SCALE.FACTOR	4
EYECON.CLOSED	1	EYECON.ICON.MASK	2	EYECON.OPEN	1		
EYECON.CONTROL.POINTS ...	4	EYECON.LEFT	2	EYECON.PUPIL	1		

CONSTANT INDEX

EYECON.HEIGHT.OFFSET	1	EYECON.PUPIL.SIZE	1	EYECON.RIGHT.WINK	1
EYECON.LEFT.CENTER	1	EYECON.RIGHT.CENTER	1	EYECON.WINK.WIDTH	1