

File created: 5-Dec-2020 16:36:45 {DSK}<Users>arunwelch>SKYDRIVE>DOCUMENTS>UNIX>LISP>LDE>ROOM
S>MEDLEY-35>SCREENPAPER.;3

previous date: 17-Aug-90 14:46:25 {DSK}<Users>arunwelch>SKYDRIVE>DOCUMENTS>UNIX>LISP>LDE>ROOMS>MEDLEY-35>SCREENP
APER.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;
;; Copyright (c) 1901, 1986, 1988, 1990, 2020 by Venue & Xerox Corporation. All rights reserved.

(RPAQQ **SCREENPAPERCOMS**

((FNS SCREENPAPER SCREENPAPERNEWREGIONFN KALSHOW DOPOINT MAPN)
[ADDVARS (IDLE.FUNCTIONS ("Screen wallpaper" 'SCREENPAPER]

;;; faster versions of editbitmap functions

(FNS INVERT.BITMAP.HORIZONTALLY INVERT.BITMAP.VERTICALLY ROTATE.BITMAP)
(VARS SCREENPAPERSIZE SCREENPERIOD SCREENREPEAT))

(DEFINEQ

(**SCREENPAPER**

[LAMBDA (WINDOW REGION.OR.SIZE OPTION) ; Edited 9-Sep-88 17:05 by bmw
(OR WINDOW (SETQ WINDOW (CREATEW)))
(OR REGION.OR.SIZE (SETQ REGION.OR.SIZE (if (EQ OPTION 'PICK)
then (GETREGION 16 16 NIL (FUNCTION SCREENPAPERNEWREGIONFN))
else SCREENPAPERSIZE)))
(LET ((SIZE (if (REGIONP REGION.OR.SIZE)
then (fetch (REGION WIDTH)
REGION.OR.SIZE)
else REGION.OR.SIZE))
TRIANGLE STREAM BUF1 2SIZE BIGBUF PBT BUF1A BUF2 BUF3 BUF4 CX CY (CNT SCREENPERIOD))
(**DECLARE** (SPECVARS TRIANGLE STREAM BUF1 2SIZE BIGBUF PBT BUF1A BUF2 BUF3 BUF4 CX CY CNT))
(SETQ TRIANGLE (BITMAPCREATE SIZE SIZE))
(SETQ BUF1 (BITMAPCREATE SIZE SIZE))
(SETQ STREAM (DSPCREATE TRIANGLE))
(FILLPOLYGON (LIST '(-1 . -1)
(CONS SIZE SIZE)
(CONS -1 SIZE))
BLACKSHADE STREAM)
(SETQ BUF2 (BITMAPCREATE SIZE SIZE))
(SETQ BUF3 (BITMAPCREATE SIZE SIZE))
(SETQ 2SIZE (PLUS SIZE SIZE))
(SETQ BIGBUF (BITMAPCREATE 2SIZE 2SIZE))
(SETQ PBT (**create** PILOTBBT))
(DSPDESTINATION BUF1 STREAM)
(if (EQ OPTION 'PICK)
then (bind POS do [RESETFORM (CURSOR CROSSHAIRS)
(until (MOUSESTATE (OR LEFT MIDDLE RIGHT])
(if (LASTMOUSESTATE (ONLY MIDDLE))
then (RETURN BIGBUF)
elseif (LASTMOUSESTATE (ONLY RIGHT))
then (RETURN NIL)
elseif (REGIONP REGION.OR.SIZE)
then (SETQ POS (CONS (fetch (REGION LEFT)
REGION.OR.SIZE)
(fetch (REGION BOTTOM)
REGION.OR.SIZE)))
(SETQ REGION.OR.SIZE)
else (SETQ POS (GETBOXPOSITION SIZE SIZE)))
(BITBLT (SCREENBITMAP)
(CAR POS)
(CDR POS)
BUF1 0 0 SIZE SIZE)
(KALSHOW BUF1 WINDOW SIZE (if (SHIFTDOWNP 'SHIFT)
then 'INVERT
else NIL)))
else (MAPN WINDOW (FUNCTION (LAMBDA (X Y)
(BITBLT (WINDOWPROP WINDOW 'IMAGECOVERED)
X Y BUF1 0 0 SIZE SIZE)
(DRAWLINE (SUB1 SIZE)
0
(RAND 0 (SUB1 SIZE))
(RAND 0 (SUB1 SIZE))
1
'INVERT STREAM)
(KALSHOW BUF1 WINDOW SIZE (if (VIDEOCOLOR)
then NIL
else 'INVERT)))
(if (LEQ (add CNT -1)

```

0)
then (SETQ CNT SCREENPERIOD)
      (to SCREENREPEAT do (BITBLT WINDOW 0 0 BUF1)
                          (KALSHOW BUF1 WINDOW SIZE))

```

(SCREENPAPERNEWREGIONFN

(* BN "17-Sep-84 10:40")

```

[LAMBDA (FP MP)
(COND
  [MP (with POSITION MP (PROG [(DX (IDIFFERENCE XCOORD (fetch (POSITION XCOORD) of FP)))
                              (DY (IDIFFERENCE YCOORD (fetch (POSITION YCOORD) of FP))
                              [COND
                                [(IGREATERP (IABS DX)
                                              (IABS DY))
                                 (SETQ YCOORD (IPLUS (fetch (POSITION YCOORD) of FP)
                                                       (ITIMES DX (COND
                                                         ((MINUSP (ITIMES DX DY))
                                                         -1)
                                                         (T 1]
                                                       (T (SETQ XCOORD (IPLUS (fetch (POSITION XCOORD) of FP)
                                                       (ITIMES DY (COND
                                                         ((MINUSP (ITIMES DX DY))
                                                         -1)
                                                         (T 1]
                                                       (RETURN MP]
(T FP))

```

(KALSHOW

```

[LAMBDA (BUF1 WINDOW SIZE MODE)
  (BITBLT TRIANGLE NIL NIL BUF1 NIL NIL NIL NIL 'ERASE)
  (ROTATE.BITMAP BUF1 BUF2 PBT)
  (INVERT.BITMAP.VERTICALLY BUF2 BUF3 PBT)
  (BITBLT BUF3 NIL NIL BUF1 NIL NIL NIL NIL 'PAINT)
  (LET (CX CY)
    (BITBLT BUF1 NIL NIL BIGBUF 0 SIZE)
    (INVERT.BITMAP.HORIZONTALLY BUF1 BUF2 PBT)
    (BITBLT BUF2 NIL NIL BIGBUF SIZE SIZE)
    (INVERT.BITMAP.VERTICALLY BUF1 BUF3 PBT)
    (BITBLT BUF3 NIL NIL BIGBUF 0 0)
    (INVERT.BITMAP.HORIZONTALLY BUF3 BUF2 PBT)
    (BITBLT BUF2 NIL NIL BIGBUF SIZE 0)
    (SETQ CX (QUOTIENT (WINDOWPROP WINDOW 'WIDTH)
                      2))
    (SETQ CY (QUOTIENT (WINDOWPROP WINDOW 'HEIGHT)
                      2))
    (for I from 0 while (LESSP I (QUOTIENT (PLUS 2SIZE (MAX CX CY)
                                              2SIZE))
      do (for J from 0 while (LEQ J I) do (DOPOINT [FUNCTION (LAMBDA (X Y)
                                                             (BITBLT BIGBUF NIL NIL WINDOW
                                                             (PLUS CX (TIMES X 2SIZE))
                                                             (PLUS CY (TIMES Y 2SIZE))
                                                             NIL NIL MODE 'REPLACE]
                                                             J I)))
    (BLOCK])

```

; Edited 5-Aug-88 11:54 by drc:

(* THAT ERASED ALL BUT THE TRIANGLE)

(DOPOINT

(* edited%: "31-Dec-00 16:08")

```

[LAMBDA (FN X Y)
  (if (LESSP X Y)
    then (DOPOINT FN Y X))
  (APPLY* FN X Y 1)
  (APPLY* FN (DIFFERENCE -1 X)
            Y 1)
  (APPLY* FN X (DIFFERENCE -1 Y)
            1)
  (APPLY* FN (DIFFERENCE -1 X)
            (DIFFERENCE -1 Y)
            1))

```

(MAPN

(* edited%: " 1-Jan-01 00:09")

```

[LAMBDA (WINDOW FN)
  (LET ((MAXX (DIFFERENCE (WINDOWPROP WINDOW 'WIDTH)
                          SIZE))
        (MAXY (DIFFERENCE (WINDOWPROP WINDOW 'HEIGHT)
                          SIZE))
        X Y NX NY STEPS)
    (SETQ X (RAND 0 MAXX))
    (SETQ Y (RAND 0 MAXY))
    (while T do (SETQ NX (RAND 0 MAXX))
                (SETQ NY (RAND 0 MAXY))
                (SETQ STEPS (QUOTIENT (PLUS (ABS (DIFFERENCE NX X))
                                              (ABS (DIFFERENCE NY Y)))
                                      4))
    (if (NEQ STEPS 0)
      then [for I from 1 to STEPS do (APPLY* FN (PLUS X (QUOTIENT (TIMES (DIFFERENCE NX X)

```

```

                                I)
                                STEPS))
                                (PLUS Y (QUOTIENT (TIMES (DIFFERENCE NY Y)
                                I)
                                STEPS]
                                (SETQ X NX)
                                (SETQ Y NY])
)

(ADDTovar IDLE.FUNCTIONS ("Screen wallpaper" 'SCREENPAPER))

;;; faster versions of editbitmap functions

(DEFINEQ

(INVERT.BITMAP.HORIZONTALALLY
  [LAMBDA (BITMAP BM2 PBT)
    (OR BM2 (SETQ BM2 (BITMAPCOPY BITMAP)))
    (OR PBT (SETQ PBT (create PILOTBBT)))
    (with PILOTBBT PBT (SETQ PBTDESTLO (ffetch BitMapLoLoc BM2))
      (SETQ PBTDESTHI (ffetch BitMapHiLoc BM2))
      (SETQ PBTSOURCELO (ffetch BitMapLoLoc BITMAP))
      (SETQ PBTSOURCEHI (ffetch BitMapHiLoc BITMAP))
      (SETQ PBTDESTBPL (TIMES 16 (ffetch BITMAPRASTERWIDTH BM2)))
      (SETQ PBTSOURCEBPL (TIMES 16 (ffetch BITMAPRASTERWIDTH BITMAP)))
      (SETQ PBTFLAGS 16384)
      (SETQ PBTHEIGHT (ffetch BITMAPHEIGHT BITMAP))
      (SETQ PBTWIDTH 1)
      (for I from 0 while (LESSP I (ffetch BITMAPWIDTH BITMAP)) do (SETQ PBTSOURCEBIT I)
        (SETQ PBTDESTBIT
          (DIFFERENCE (SUB1 (ffetch BITMAPWIDTH BITMAP)
          )
          I))
        (\PILOTBITBLT PBT 0)))
    BM2])
  (* edited%: "31-Dec-00 17:15")

(INVERT.BITMAP.VERTICALLY
  [LAMBDA (BITMAP BM2 PBT)
    (OR BM2 (SETQ BM2 (BITMAPCOPY BITMAP)))
    (OR PBT (SETQ PBT (create PILOTBBT)))
    [with PILOTBBT PBT
      (SETQ PBTDESTHI (ffetch BitMapHiLoc BM2))
      [SETQ PBTDESTLO (PLUS (ffetch BitMapLoLoc BM2)
        (TIMES (SUB1 (ffetch BITMAPHEIGHT BITMAP))
        (ffetch BITMAPRASTERWIDTH BM2]
      (SETQ PBTSOURCELO (ffetch BitMapLoLoc BITMAP))
      (SETQ PBTSOURCEHI (ffetch BitMapHiLoc BITMAP))
      (SETQ PBTDESTBPL (TIMES 16 (ffetch BITMAPRASTERWIDTH BM2)))
      (SETQ PBTSOURCEBPL (TIMES 16 (ffetch BITMAPRASTERWIDTH BITMAP)))
      (SETQ PBTSOURCEBIT 0)
      (SETQ PBTDESTBIT 0)
      (SETQ PBTFLAGS 16384)
      (SETQ PBTHEIGHT 1)
      (SETQ PBTWIDTH (ffetch BITMAPWIDTH BITMAP))
      (for I from 0 while (LESSP I (ffetch BITMAPHEIGHT BITMAP)) do (\PILOTBITBLT PBT 0)
        (add PBTSOURCELO (ffetch BITMAPRASTERWIDTH
          BITMAP))
        (add PBTDESTLO (MINUS (ffetch
          BITMAPRASTERWIDTH
          BM2]
      BM2])
    ]
  (*
  (* by experiment, disjoint replace)

(ROTATE.BITMAP
  [LAMBDA (BITMAP BM2 PBT)
    [OR BM2 (SETQ BM2 (BITMAPCREATE (ffetch BITMAPHEIGHT BITMAP)
      (ffetch BITMAPWIDTH BITMAP)
    (OR PBT (SETQ PBT (create PILOTBBT)))
    [with PILOTBBT PBT
      (SETQ PBTDESTHI (ffetch BitMapHiLoc BM2))
      (SETQ PBTDESTLO (ffetch BitMapLoLoc BM2))
      (SETQ PBTSOURCELO (ffetch BitMapLoLoc BITMAP))
      (SETQ PBTSOURCEHI (ffetch BitMapHiLoc BITMAP))
      (SETQ PBTDESTBPL (TIMES 16 (ffetch BITMAPRASTERWIDTH BM2)))
      (SETQ PBTSOURCEBPL 1)
      (SETQ PBTSOURCEBIT 0)
      (SETQ PBTDESTBIT (ffetch BITMAPWIDTH BM2))
      (SETQ PBTFLAGS 0)
      (SETQ PBTHEIGHT (ffetch BITMAPHEIGHT BM2))
      (SETQ PBTWIDTH 1)
      (for I from 0 while (LESSP I (ffetch BITMAPHEIGHT BITMAP)) do (add PBTDESTBIT -1)
        (\PILOTBITBLT PBT 0)
        (add PBTSOURCELO (ffetch BITMAPRASTERWIDTH
          BITMAP]
    ]
  (* edited%: "31-Dec-00 16:24")
  (*
  (* by experiment, disjoint replace)

```

```
{MEDLEY}<rooms>SCREENPAPER.;1  (ROTATE.BITMAP cont.)
```

Page 4

```
    BM2])
```

```
)
```

```
(RPAQQ SCREENPAPERSIZE 64)
```

```
(RPAQQ SCREENPERIOD 100)
```

```
(RPAQQ SCREENREPEAT 0)
```

```
(PUTPROPS SCREENPAPER COPYRIGHT ("Venue & Xerox Corporation" 1901 1986 1988 1990 2020))
```

FUNCTION INDEX

DOPOINT	2	KALSHOW	2	SCREENPAPER	1
INVERT.BITMAP.HORIZONTALLY	3	MAPN	2	SCREENPAPERNEWREGIONFN	2
INVERT.BITMAP.VERTICALLY	3	ROTATE.BITMAP	3		

VARIABLE INDEX

IDLE.FUNCTIONS	3	SCREENPAPERSIZE	4	SCREENPERIOD	4	SCREENREPEAT	4
----------------------	---	-----------------------	---	--------------------	---	--------------------	---