File created: 28-Apr-2021 18:48:27 **{DSK}<mnt>c>Users>Larry>home>ilisp>medley>LISPUSERS>FILLREGION.;4**

changes to: (VARS FILLREGIONCOMS)

previous date: 9-Mar-87 18:02:40 {DSK}<mnt>c>Users>Larry>home>ilisp>medley>LISPUSERS>FILLREGION.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
;;
;; Copyright (c) 1985, 1987, 2021 by Xerox Corporation.

(RPAQQ FILLREGIONCOMS
       [(DECLARE%: EVAL@COMPILE DONTCOPY
              (MACROS ADD.TASK CIRCLE.ABOUT DEC.X DEC.Y FINISH.ORIENTATION INC.X INC.Y NEXT.POINT.ON.CURVE
                     POP.TASK RIGHT.BIT SEARCH SEARCH.AND.FILL SET.BIT TEST.BIT WANT.TO.EXTEND)
              (RECORDS TASK))
        (FNS * \FILLREGION.FNS)
        (BLOCKS * (LIST (APPEND '(FILLREGION)
                                \FILLREGION.FNS
                                '((ENTRIES AUTO.FILL FILL.REGION)
                                  (GLOBALVARS BITMASKARRAY)
                                  (LOCALVARS . T])

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

(PUTPROPS ADD.TASK MACRO [(STARTW STARTM)
                          (if FREELIST
                              then (create TASK
                                           W _ STARTW
                                           M _ STARTM smashing (PROG1 (CAR FREELIST)
                                                                      (psetf FREELIST (CDR FREELIST)
                                                                             (CDR FREELIST)
                                                                             AGENDA AGENDA FREELIST)))
                            else (push AGENDA (create TASK
                                                      W _ STARTW
                                                      M _ STARTM])

(PUTPROPS CIRCLE.ABOUT MACRO ((START.W START.M)
                              (first (SETQ CIRCLE.WORD START.W)
                                     (SETQ CIRCLE.MASK START.M)
                                     (NEXT.POINT.ON.CURVE CIRCLE.WORD CIRCLE.MASK 6 CIRCLE.THIS.DIR)
                                     (SETQ CIRCLE.EXTEND NIL)
                                eachtime (SETQ CIRCLE.PREV.W CIRCLE.WORD)
                                       (SETQ CIRCLE.PREV.M CIRCLE.MASK)
                                       (NEXT.POINT.ON.CURVE CIRCLE.WORD CIRCLE.MASK CIRCLE.THIS.DIR
                                              CIRCLE.NEXT.DIR)
                                       (if (WANT.TO.EXTEND CIRCLE.THIS.DIR CIRCLE.NEXT.DIR)
                                           then (if (NOT CIRCLE.EXTEND)
                                                    then (SETQ CIRCLE.EXTEND T)
                                                         (ADD.TASK CIRCLE.PREV.W CIRCLE.PREV.M))
                                         else (SETQ CIRCLE.EXTEND NIL))
                                       (SET.BIT DEST.BASE CIRCLE.PREV.W CIRCLE.PREV.M)
                                until (AND (EQ CIRCLE.PREV.W START.W)
                                           (EQ CIRCLE.PREV.M START.M)
                                           (FINISH.ORIENTATION CIRCLE.THIS.DIR CIRCLE.NEXT.DIR))
                                do (SETQ CIRCLE.THIS.DIR CIRCLE.NEXT.DIR))))

(PUTPROPS DEC.X MACRO [(WORD MASK)
                       (SETQ MASK (if (EQ MASK (CONSTANT (MASK.1'S 15 1)))
                                      then (add WORD -1)
                                           1
                                    else (LLSH MASK 1])

(PUTPROPS DEC.Y MACRO ((WORD MASK)
                       (add WORD RASTERWIDTH)))

(PUTPROPS FINISH.ORIENTATION MACRO ((THIS.DIR NEXT.DIR)
                                    (IGEQ THIS.DIR (LOGXOR NEXT.DIR 4))))

(PUTPROPS INC.X MACRO [(WORD MASK)
                       (SETQ MASK (if (EQ MASK 1)
                                      then (add WORD 1)
                                           (CONSTANT (MASK.1'S 15 1))
                                    else (LRSH MASK 1])

(PUTPROPS INC.Y MACRO ((WORD MASK)
                       (SETQ WORD (IDIFFERENCE WORD RASTERWIDTH))))

(PUTPROPS NEXT.POINT.ON.CURVE MACRO ((WORD MASK DIN DOUT)
```

```
                                    (PROG NIL
                                          [if (IGEQ DIN 5)
                                              then (if (EQ DIN 7)
                                                       then     (* DIN = 7)
                                                            (INC.Y WORD MASK)
                                                            (GO L6)
                                                     else       (* DIN = 5 or 6)
                                                          (INC.X WORD MASK)
                                                          (GO L4))
                                            else (if (IGEQ DIN 3)
                                                     then       (* DIN = 3 or 4)
                                                          (DEC.Y WORD MASK)
                                                          (GO L2)
                                                   else (if (NEQ DIN 0)
                                                            then
                                                                (* DIN = 1 or 2)
                                                                  (DEC.X WORD MASK)
                                                                  (GO L0)
                                                          else   (* DIN = 0)
                                                               (INC.Y WORD MASK)
                                                               (GO L6]
                                      L0  (DEC.Y WORD MASK)
                                          (if (NEQ 0 (TEST.BIT SRCE.BASE WORD MASK))
                                              then (SETQ DOUT 0)
                                                   (RETURN))
                                          (INC.X WORD MASK)
                                          (if (NEQ 0 (TEST.BIT SRCE.BASE WORD MASK))
                                              then (SETQ DOUT 1)
                                                   (RETURN))
                                      L2  (INC.X WORD MASK)
                                          (if (NEQ 0 (TEST.BIT SRCE.BASE WORD MASK))
                                              then (SETQ DOUT 2)
                                                   (RETURN))
                                          (INC.Y WORD MASK)
                                          (if (NEQ 0 (TEST.BIT SRCE.BASE WORD MASK))
                                              then (SETQ DOUT 3)
                                                   (RETURN))
                                      L4  (INC.Y WORD MASK)
                                          (if (NEQ 0 (TEST.BIT SRCE.BASE WORD MASK))
                                              then (SETQ DOUT 4)
                                                   (RETURN))
                                          (DEC.X WORD MASK)
                                          (if (NEQ 0 (TEST.BIT SRCE.BASE WORD MASK))
                                              then (SETQ DOUT 5)
                                                   (RETURN))
                                      L6  (DEC.X WORD MASK)
                                          (if (NEQ 0 (TEST.BIT SRCE.BASE WORD MASK))
                                              then (SETQ DOUT 6)
                                                   (RETURN))
                                          (DEC.Y WORD MASK)
                                          (if (NEQ 0 (TEST.BIT SRCE.BASE WORD MASK))
                                              then (SETQ DOUT 7)
                                                   (RETURN))
                                          (GO L0))))

(PUTPROPS POP.TASK MACRO (NIL (PROG1 (CAR AGENDA)
                                     (CL:PSETF AGENDA (CDR AGENDA)
                                            (CDR AGENDA)
                                           FREELIST FREELIST AGENDA))))

(PUTPROPS RIGHT.BIT MACRO ((X)
                           (LOGAND X (IMINUS X))))

(PUTPROPS SEARCH MACRO [(WORD MASK)
                        (PROGN (SETQ SEARCH.BITS (\GETBASE SRCE.BASE WORD))
                               (for old WORD
                                  first (if (NEQ MASK 1)
                                            then (if (NEQ [SETQ SEARCH.BITS (LOGAND SEARCH.BITS
                                                                                    (LOGNOT (SUB1 MASK]
                                                          0)
                                                     then (SETQ MASK (RIGHT.BIT SEARCH.BITS))
                                                          (RETURN)
                                                   else (add WORD -1)))
                                  when (NEQ (SETQ SEARCH.BITS (\GETBASE SRCE.BASE WORD))
                                            0)
                                  do (SETQ MASK (RIGHT.BIT SEARCH.BITS))
                                     (RETURN)
                                  by (SUB1 WORD])

(PUTPROPS SEARCH.AND.FILL MACRO
          [(WORD MASK)
           (PROGN (SETQ SEARCH.BITS (\GETBASE SRCE.BASE WORD))
                  (for old WORD first [if (NEQ MASK 1)
                                          then [SETQ SEARCH.MASKEDBITS (LOGAND SEARCH.BITS
                                                                               (SETQ SEARCH.EXTENDEDMASK
                                                                                (LOGXOR (SUB1 MASK)
                                                                                       (CONSTANT (MASK.1'S 0 16]
```

```
                                        (if (NEQ SEARCH.MASKEDBITS 0)
                                            then [\PUTBASE DEST.BASE WORD (LOGOR (SETQ FILL.BITS
                                                                                     (\GETBASE DEST.BASE WORD))
                                                                                 (LOGAND
                                                                                  SEARCH.EXTENDEDMASK
                                                                                  (SUB1 (SETQ MASK
                                                                                              (RIGHT.BIT
                                                                                                  SEARCH.MASKEDBITS
                                                                                                     ]
                                                  (RETURN (LOGAND FILL.BITS MASK))
                                              else (\PUTBASE DEST.BASE WORD (LOGOR (\GETBASE DEST.BASE WORD)
                                                                                   SEARCH.EXTENDEDMASK))
                                                  (SETQ WORD (SUB1 WORD]
                        when (NEQ (SETQ SEARCH.BITS (\GETBASE SRCE.BASE WORD))
                                  0)
                        do [\PUTBASE DEST.BASE WORD (LOGOR (SETQ FILL.BITS (\GETBASE DEST.BASE WORD))
                                                           (SUB1 (SETQ MASK (RIGHT.BIT SEARCH.BITS]
                           (RETURN (LOGAND FILL.BITS MASK))
                        by (PROGN (\PUTBASE DEST.BASE WORD (CONSTANT (MASK.1'S 0 16)))
                                  (SUB1 WORD])

(PUTPROPS SET.BIT MACRO ((BASE WORD MASK)
                         (change (fetch (BITMAPWORD BITS) of (\ADDBASE BASE WORD))
                                 (LOGOR DATUM MASK))))

(PUTPROPS TEST.BIT MACRO ((BASE WORD MASK)
                          (LOGAND (\GETBASE BASE WORD)
                                  MASK)))

(PUTPROPS WANT.TO.EXTEND MACRO ((THIS.DIR NEXT.DIR)
                                (IGREATERP (LOGAND (IDIFFERENCE THIS.DIR 3)
                                                   7)
                                           NEXT.DIR)))
)

(DECLARE%: EVAL@COMPILE

(RECORD TASK (W . M))
)
)

(RPAQQ \FILLREGION.FNS (AUTO.FILL FILL.KERNEL FILL.REGION REMOVE.SINGLE.POINTS))

(DEFINEQ
```

(AUTO.FILL

```
  [LAMBDA (SHADE)                                          (* JWogulis "28-Feb-85 09:03")
    (PROG ((W (WHICHW)))
          (GETMOUSESTATE)
          (RETURN (FILL.REGION W (CONS (LASTMOUSEX W)
                                       (LASTMOUSEY W))
                                 SHADE])
```

(FILL.KERNEL

```
  [LAMBDA (SRCE.BASE DEST.BASE WORDNUM BITMASK RASTERWIDTH)      (* mgb%: "15-Mar-85 01:54")

          (* * Appalling PROG structure instead of CLISP is to permit only one expansion of CIRCLE.ABOUT macro)

    (PROG (TASK.W TASK.M AGENDA FREELIST TASK CIRCLE.PREV.W CIRCLE.PREV.M CIRCLE.WORD CIRCLE.MASK
                  CIRCLE.THIS.DIR CIRCLE.NEXT.DIR CIRCLE.EXTEND SEARCH.BITS SEARCH.EXTENDEDMASK SEARCH.MASKEDBITS
                  FILL.BITS KERNEL.WORD KERNEL.MASK KERNEL.THIS.DIR KERNEL.NEXT.DIR INITIALIZED)
          (SETQ KERNEL.WORD WORDNUM)
          (SETQ KERNEL.MASK BITMASK)
          (INC.X KERNEL.WORD KERNEL.MASK)
          (SEARCH KERNEL.WORD KERNEL.MASK)
          (GO DO.CIRCLE)
      TASK.LOOP
          (if (NULL AGENDA)
              then (RETURN))
          (SETQ TASK (POP.TASK))
          (SETQ TASK.W (fetch (TASK W) of TASK))
          (SETQ TASK.M (fetch (TASK M) of TASK))
          (SETQ KERNEL.NEXT.DIR 2)
      SEED.LOOP
          (SETQ KERNEL.THIS.DIR KERNEL.NEXT.DIR)
          (SETQ KERNEL.WORD TASK.W)
          (SETQ KERNEL.MASK TASK.M)
          (NEXT.POINT.ON.CURVE TASK.W TASK.M KERNEL.THIS.DIR KERNEL.NEXT.DIR)
          (if [NOT (AND (WANT.TO.EXTEND KERNEL.THIS.DIR KERNEL.NEXT.DIR)
                        (PROGN (DEC.X KERNEL.WORD KERNEL.MASK)
                               (EQ 0 (TEST.BIT DEST.BASE KERNEL.WORD KERNEL.MASK]
              then (GO TASK.LOOP))
          (if (NEQ 0 (SEARCH.AND.FILL KERNEL.WORD KERNEL.MASK))
              then (GO SEED.LOOP))
      DO.CIRCLE
          (CIRCLE.ABOUT KERNEL.WORD KERNEL.MASK)
```

```
          (if INITIALIZED
              then (GO SEED.LOOP)
            else (SETQ INITIALIZED T)
                 (GO TASK.LOOP])
```

(FILL.REGION

```
  [LAMBDA (WINDOW.OR.BM INTERIOR.POS SHADE)                    (* mgb%: "15-Mar-85 01:51")

          (* * This function has been "optimised" for performance. Any resemblance to structured programming is purely coincidental.)

    [PROG ((X (CAR INTERIOR.POS))
           (Y (CDR INTERIOR.POS))
           WIDTH HEIGHT SRCE.BM SRCE.BASE DEST.BM DEST.BASE INVERTFLG? RASTERWIDTH TEMP.SHADE)
          (if (WINDOWP WINDOW.OR.BM)
              then (SETQ WIDTH (WINDOWPROP WINDOW.OR.BM 'WIDTH))
                   (SETQ HEIGHT (WINDOWPROP WINDOW.OR.BM 'HEIGHT))
            elseif (BITMAPP WINDOW.OR.BM)
              then (SETQ WIDTH (BITMAPWIDTH WINDOW.OR.BM))
                   (SETQ HEIGHT (BITMAPHEIGHT WINDOW.OR.BM))
            else (RETURN (ERROR "Must be either window or bitmap:" WINDOW.OR.BM)))
          (if (OR (LESSP X 0)
                  (LESSP Y 0)
                  (IGEQ X WIDTH)
                  (IGEQ Y HEIGHT))
              then (RETURN (ERROR "Outside of the window or bitmap:" INTERIOR.POS)))
          (add WIDTH 2)
          (add HEIGHT 2)
          (SETQ SRCE.BM (BITMAPCREATE WIDTH HEIGHT))
          (SETQ DEST.BM (BITMAPCREATE WIDTH HEIGHT))
          (SETQ INVERTFLG? (NEQ 0 (BITMAPBIT WINDOW.OR.BM X Y)))

          (* Above%: INVERTFLG? is T if we start on a black pixel instead of a white one.
          XPOS will be the x position of the bitmap that is the first point on the edge of the figure to be filled.)

          (SETQ SRCE.BASE (ffetch BITMAPBASE of SRCE.BM))
          (SETQ RASTERWIDTH (ffetch BITMAPRASTERWIDTH of SRCE.BM))
          (SETQ DEST.BASE (ffetch BITMAPBASE of DEST.BM))
          (BITBLT WINDOW.OR.BM 0 0 SRCE.BM 1 1 NIL NIL (if INVERTFLG?
                                                           then 'INVERT))

          (* This will remove all the points in the window that have no one adjacent.
          This is a special case and be dealt with more easily at thislevel than in the program.)

          (REMOVE.SINGLE.POINTS SRCE.BM DEST.BM)
          (BITBLT DEST.BM 0 0 SRCE.BM 0 0 NIL NIL 'INPUT 'ERASE)

          (* NOW DO SOMETHING REALLY UGLY. Here we put a 1 bit border around the whole window
          (which has been enlarged to hold it) so that we never run off the edge and the window gets filled up.
          This also makes the fast versions of BITMAPBIT fast, i.e. no checking for within the boundaries.)

          (BITBLT NIL NIL NIL SRCE.BM 0 0 1 NIL 'TEXTURE NIL 65535)
          (BITBLT NIL NIL NIL SRCE.BM 0 0 NIL 1 'TEXTURE NIL 65535)
          (BITBLT NIL NIL NIL SRCE.BM 0 (SUB1 HEIGHT)
                 NIL 1 'TEXTURE NIL 65535)
          (BITBLT NIL NIL NIL SRCE.BM (SUB1 WIDTH)
                 0 1 NIL 'TEXTURE NIL 65535)
          (FILL.KERNEL SRCE.BASE DEST.BASE (IPLUS (LRSH X 4)
                                                  (ITIMES (SUB1 (IDIFFERENCE HEIGHT (ADD1 Y)))
                                                         RASTERWIDTH))
                 (ELT BITMASKARRAY (LOGAND X 15))
                 RASTERWIDTH)
          (BITBLT DEST.BM 1 1 WINDOW.OR.BM 0 0 NIL NIL 'MERGE (if INVERTFLG?
                                                                  then 'ERASE
                                                                else 'PAINT)
                 (if (NOT INVERTFLG?)
                     then SHADE
                   elseif (BITMAPP SHADE)
                     then (SETQ TEMP.SHADE (BITMAPCREATE (BITMAPWIDTH SHADE)
                                                         (BITMAPHEIGHT SHADE)))
                          (BITBLT SHADE NIL NIL TEMP.SHADE NIL NIL NIL NIL 'INVERT)
                          TEMP.SHADE
                     else (LOGNOT SHADE]
    WINDOW.OR.BM])
```

(REMOVE.SINGLE.POINTS

```
  [LAMBDA (BITMAP RESULT.BM)                                   (* edited%: " 8-Mar-85 01:22")
    (BITBLT BITMAP 0 0 RESULT.BM)
    (for X from -1 to 1 do (for Y from -1 to 1
                              do (if (NOT (AND (EQ 0 X)
                                               (EQ 0 Y)))
                                     then (BITBLT BITMAP X Y RESULT.BM 0 0 NIL NIL 'INPUT 'ERASE])

)

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY
```

```
(BLOCK%: FILLREGION AUTO.FILL FILL.KERNEL FILL.REGION REMOVE.SINGLE.POINTS (ENTRIES AUTO.FILL FILL.REGION)
       (GLOBALVARS BITMASKARRAY)
       (LOCALVARS . T))
)

(PUTPROPS FILLREGION COPYRIGHT ("Xerox Corporation" 1985 1987 2021))
```

FUNCTION INDEX

MACRO INDEX

VARIABLE INDEX

RECORD INDEX