

File created: 25-May-2023 22:14:28 {DSK}<home>larry>il>medley>sources>XXFILL.;9

edit by: lmm

changes to: (FNS POLYSHADE.SCAN.IP)

previous date: 2-May-2023 15:46:54 {DSK}<home>larry>il>medley>sources>XXFILL.;7

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

(RPAQQ **XXFILLCOMS**  
( (COMS

;;; Filled Polygons

```
(FNS SCAN.LESSP CRIT.LESSP)
(DECLARE%: EVAL@COMPILE DONTCOPY (RECORDS SCAN))
(INITRECORDS SCAN)
(VARS FILL.WRULE \FILL.DEBUG)
(DECLARE%: EVAL@COMPILE DONTCOPY (GLOBALVARS \FILL.DEBUG)
 (SPECVARS \FILL.WRULE)
 (MACROS \NORMSECT \DrawScanList.Blt \DrawScanList.Display \DrawScanList.XScan.IP
 \DrawScanList.YScan.IP))
(FNS MAKESCAN SHEDSCAN NORMSECT CRITSECT)
(FNS \POLYSHADE.BLT \POLYSHADE.DISPLAY \POLYSHADE.XSCAN.IP \POLYSHADE.YSCAN.IP)
(FNS POLYSHADE.BLT POLYSHADE.DISPLAY POLYSHADE.IP POLYSHADE.SCAN.IP)
(FNS FILLTRIANGLE))
(COMS
```

;;; Filled Circles

```
(FNS \CIRCSHADE.BLT \CIRCSHADE.DISPLAY \CIRCSHADE.IP \CIRCSHADE.XSCAN.IP)
(FNS CIRCSHADE.BLT CIRCSHADE.DISPLAY CIRCSHADE.IP)
(FNS FILLCIRCLE.IP FILLNGON.IP)
(P (MOVD 'FILLCIRCLE.IP 'CIRCSHADE.IP))
```

;;; Considering scan direction of the printer

;;; You must set these vars before opening IMAGESTREAM

```
(COMS (VARS (PRINTER.DEFAULT.SCAN.DIRECTION 'Y)
 (PRINTER.SCAN.DIRECTIONS.LIST))
(COMS
```

;;; PBBT Optimized routines

```
(VARS PBBT.PANEL)
(DECLARE%: EVAL@COMPILE DONTCOPY (RECORDS FILL.TBLE))
(INITRECORDS FILL.TBLE)
(FNS FILL.INITTBLE FILL.INCY FILL.LINE)
(FNS FILL.TEST FILL.XPER FILL.CONT))
(DECLARE%: DOEVAL@COMPILE DONTCOPY (FILES (LOADCOMP)
 INTERPRESS XXGEOM)))
```

;;; Filled Polygons

(DEFINEQ

(**SCAN.LESSP**

[LAMBDA (SCAN1 SCAN2)

(\* FS " 9-Jul-85 15:24")

(\* Is scan segment 1 less than 2, in scanline sense.)

```
(ILESSP (fetch (SCAN LX) of SCAN1)
 (fetch (SCAN LX) of SCAN2])
```

(**CRIT.LESSP**

[LAMBDA (SCAN1 SCAN2)

(\* FS " 8-Jul-85 15:47")

(\* Is scan segment 1 less than 2, in critical pt sense)

```
(ILESSP (fetch (SCAN BY) of SCAN1)
 (fetch (SCAN BY) of SCAN2])
```

)

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

```

(DATATYPE SCAN (LX RX GEOM TY BY WC REST)
)
(/DECLAREDATATYPE 'SCAN ' (POINTER POINTER POINTER POINTER POINTER POINTER POINTER)
  ;; ---field descriptor list elided by lister---
  '14)
)
(/DECLAREDATATYPE 'SCAN ' (POINTER POINTER POINTER POINTER POINTER POINTER POINTER)
  ;; ---field descriptor list elided by lister---
  '14)
)
(RPAQQ FILL.WRULE 1)
(RPAQQ \FILL.DEBUG NIL)
(DECLARE%: EVAL@COMPILE DONTCOPY
)
(DECLARE%: DOEVAL@COMPILE DONTCOPY
)
(GLOBALVARS \FILL.DEBUG)
)
(DECLARE%: DOEVAL@COMPILE DONTCOPY
)
(SPECVARS \FILL.WRULE)
)
(DECLARE%: EVAL@COMPILE
)
(PUTPROPS \NORMSECT DMACRO ((SELF Y)
  (LET (XL XR (GEOM (ffetch (SCAN GEOM) of SELF)))
    (BRESSTEP GEOM Y XL XR)
    (freplace (SCAN LX) of SELF with XL)
    (freplace (SCAN RX) of SELF with XR))))
)
(PUTPROPS \DrawScanList.Bit DMACRO [(strm scanlist scany)
  ;; count, fill.shade fill.wrule fill.frule are dynamically scoped outside macro
  (SETQ count 0)
  [IF (EQ FILL.WRULE 1)
    THEN (for scan in scanlist
      do (SETQ count (IPLUS count 1))
        (if (ODDP count)
          then (SETQ ex (IPLUS (ffetch (SCAN RX) of scan)
            1))
          else (BLTSHADE FILL.SHADE strm ex scany
            (IDIFFERENCE (ffetch (SCAN LX) of scan)
              ex)
            1 FILL.FRULE)]
    (IF (EQ FILL.WRULE 0)
      THEN (for scan in scanlist
        do [if (EQ count 0)
          then (SETQ ex (ADD1 (ffetch (SCAN RX) of scan)
            (SETQ count (IPLUS count (ffetch (SCAN WC) of scan))))
          (if (EQ count 0)
            then (BLTSHADE FILL.SHADE strm ex scany
              (IDIFFERENCE (ffetch (SCAN LX) of scan)
                ex)
              1 FILL.FRULE)]
        ]
    ]
)
(PUTPROPS \DrawScanList.Display DMACRO [(strm scanlist scany)
  ;; count, fill.shade fill.wrule fill.frule, table are dynamically scoped outside macro. Since if fill.wrule is 1
  ;; count is merely a toggle, use T and NIL.
  [IF (EQ FILL.WRULE 1)
    THEN (SETQ count NIL)
      (for scan in scanlist
        do (SETQ count (NOT count))
          (if count
            then (SETQ ex (ADD1 (ffetch (SCAN RX) of scan)))
            else (FILL.LINE table scany ex (ffetch (SCAN LX)
              of scan)]
    (IF (EQ FILL.WRULE 0)
      THEN (SETQ count 0)
        (for scan in scanlist
          do [if (EQ count 0)
            then (SETQ ex (ADD1 (ffetch (SCAN RX) of scan)
              (SETQ count (IPLUS count (ffetch (SCAN WC) of scan))))
            (if (EQ count 0)
              then (FILL.LINE table scany ex (ffetch (SCAN LX)
                of scan)]
          ]
    ]
)
(PUTPROPS \DrawScanList.XScan.IP DMACRO [(strm scanlist scany)

```

```

;; count, fill.shade fill.wrule fill.frule are dynamically scoped outside macro
(SETQ count 0)
[IF (EQ FILL.WRULE 1)
  THEN (for scan in scanlist
    do (SETQ ix (fetch (SCAN LX) of scan))
      (SETQ tx (ffetch (SCAN RX) of scan))
      (SETQ count (IPLUS count 1))
      (if (ODDP count)
        then (SETQ ex (IPLUS tx 1))
        else (FILLRECTANGLE.IP strm ex scany (IDIFFERENCE ix ex)
              1))
    )
  (IF (EQ FILL.WRULE 0)
    THEN (for scan in scanlist
      do (SETQ ix (fetch (SCAN LX) of scan))
        (SETQ tx (ffetch (SCAN RX) of scan))
        (if (EQ count 0)
          then (SETQ ex (IPLUS tx 1)))
        (SETQ count (IPLUS count (ffetch (SCAN WC) of scan)))
        (if (EQ count 0)
          then (FILLRECTANGLE.IP strm ex scany (IDIFFERENCE
            ix ex)
            1))
      )
    )
  )
]

```

```

(PUTPROPS \DrawScanList.YScan.IP DMACRO [(strm scanlist scany)
  ;; count, fill.shade fill.wrule fill.frule are dynamically scoped outside macro * *
  (SETQ count 0)
  [IF (EQ FILL.WRULE 1)
    THEN (for scan in scanlist
      do (SETQ ix (fetch (SCAN LX) of scan))
        (SETQ tx (fetch (SCAN RX) of scan))
        (SETQ count (IPLUS count 1))
        (if (ODDP count)
          then (SETQ ex (IPLUS tx 1))
          else ;; Unreflect coordinates back
              (FILLRECTANGLE.IP strm scany ix 1 (IDIFFERENCE
                ex ix)
              )
        )
      )
    (IF (EQ FILL.WRULE 0)
      THEN (for scan in scanlist
        do (SETQ ix (fetch (SCAN LX) of scan))
          (SETQ tx (fetch (SCAN RX) of scan))
          (if (EQ count 0)
            then (SETQ ex (IPLUS tx 1)))
          (SETQ count (IPLUS count (fetch (SCAN WC) of scan)))
          (if (EQ count 0)
            then ;; Unreflect coordinates back
                (FILLRECTANGLE.IP strm scany ix 1
                  (IDIFFERENCE ex ix)
                )
            )
          )
        )
      )
    )
  )
]
)
)

```

(DEFINEQ

**(MAKESCAN**

[LAMBDA (EDGE LIST)

; Edited 24-Aug-87 22:35 by FS

;; Returns a scan object given a edge list

```

(LET [(SELF (create SCAN
  REST _ (SORT EDGE LIST (FUNCTION LINE.LESSP)
    (SHEDSCAN SELF)
    SELF)])

```

**(SHEDSCAN**

[LAMBDA (SELF)

; Edited 13-Jun-2021 14:41 by rmk:

(\* \* remove current edge and replace with next)

```

(LET (X0 Y0 DX DY YDIR EDGE TAIL BRES)
  (SETQ TAIL (fetch REST of SELF))
  (COND
    (TAIL (SETQ EDGE (CAR TAIL))
      (GETLINEORIG EDGE X0 Y0)
      (GETLINEDIFF EDGE DX DY)
      (SETQ BRES (MAKEBRES X0 Y0 DX DY))
      (SETQ YDIR (fetch (XXLINE WIND) of EDGE))
      (replace (SCAN GEOM) of SELF with BRES)
      (replace (SCAN REST) of SELF with (CDR TAIL))
      (replace (SCAN BY) of SELF with Y0)
      (replace (SCAN TY) of SELF with (IPLUS Y0 DY))
      (replace (SCAN WC) of SELF with YDIR])
  )
)

```

**(NORMSECT**

```
[LAMBDA (SELF Y) ; Edited 24-Aug-87 19:29 by FS
;; Cause the scan object to update its scan segment based on scany. Assumes that dy is positive, which is true from MakeLine
(\NORMSECT SELF Y)]
```

**(CRITSECT**

```
[LAMBDA (SELF SCANY) (* FS "10-Feb-86 16:47")
```

(\* \* Fetch more segments while on critical pt)

```
(LET (IX1 TX1)
  (if \FILL.DEBUG
    then (printout T "SCANY: " SCANY T))
  (NORMSECT SELF SCANY)
  (if (fetch (SCAN REST) of SELF)
    then (SETQ IX1 (fetch LX of SELF))
         (SETQ TX1 (fetch RX of SELF))
    [while (AND (EQ SCANY (fetch (SCAN TY) of SELF))
                (fetch (SCAN REST) of SELF))
      do (SHEDSCAN SELF)
         (NORMSECT SELF SCANY)
         (SETQ IX1 (IMIN IX1 (fetch (SCAN LX) of SELF)))
         (SETQ TX1 (IMAX TX1 (fetch (SCAN RX) of SELF))
         (replace (SCAN LX) of SELF with IX1)
         (replace (SCAN RX) of SELF with TX1])
    ]
  )
```

(DEFINEQ

**(POLYSHADE.BLT**

```
[LAMBDA (STRM ALIST FILL.SHADE FILL.FRULE FILL.WRULE) ; Edited 2-May-2023 15:36 by Imm
  (DECLARE (SPECVARS FILL.WRULE)) ; Edited 1-Feb-89 18:28 by FS
```

;; Generic version of polygon code, works for any stream which can do BLTSHADE. Expects integer line lists, for Bltshade destinations, works in  
;; dev. coords, should limit to clip region if possible

```
(PROG (fulllist currlist clist nlist scan currcrit fullcrit crity count scany ix tx ex by ty)
  (SETQ fulllist (MAPCAR ALIST 'MAKESCAN))
  (SETQ fulllist (SORT fulllist 'CRIT.LESSP))
  (SETQ currlist NIL)
  (SETQ currcrit MIN.INTEGER)
  (SETQ scany (fetch (SCAN BY) of (CAR fulllist)))
  [while (OR currlist fulllist) do ;; merge new critical edges
    (while [AND (LISTP fulllist)
                (IEQP scany (fetch (SCAN BY) of (CAR fulllist))
              do (SETQ scan (CAR fulllist))
                 (SETQ currlist (CONS scan currlist))
                 (SETQ fulllist (CDR fulllist))]
    (if (LISTP fulllist)
      then (SETQ fullcrit (fetch (SCAN BY) of (CAR fulllist)))
      else (SETQ fullcrit MAX.INTEGER))
    ;; paint critical scan line
    (for scan in currlist do (CRITSECT scan scany))
    (SORT currlist 'SCAN.LESSP)
    (\DrawScanList.Blt STRM currlist scany)
    ;; cull out exhausted edges
    (SETQ scany (IPLUS scany 1))
    ;(ILEQ currcrit scany)
    (SETQ currcrit MAX.INTEGER)
    (SETQ clist NIL)
    [for scan in currlist do (SETQ ty (fetch (SCAN TY) of scan))
                          (SETQ currcrit (IMIN currcrit ty))
                          (if (ILEQ scany ty)
                            then (SETQ clist (CONS scan clist))
                                (SETQ currcrit (IMIN currcrit ty))
                          )
    (SETQ currlist clist)
    ;; paint normal scan lines
    (SETQ crity (IMIN currcrit fullcrit))
    (while (ILESSP scany crity) do (for scan in currlist
                                  do (NORMSECT scan scany))
          (SORT currlist 'SCAN.LESSP)
          (\DrawScanList.Blt STRM currlist scany)
          (SETQ scany (IPLUS scany 1])
  (RETURN NIL])
```

**(POLYSHADE.DISPLAY**

```
[LAMBDA (STRM ALIST FILL.SHADE FILL.FRULE FILL.WRULE) ; Edited 2-May-2023 15:37 by Imm
  (DECLARE (SPECVARS FILL.WRULE)) ; Edited 24-Aug-87 19:47 by FS
```

;; Generic version of polygon code, works for any device which can do pilot bbt. Expects integer line lists, for Bltshade destinations, works in dev.

;; coords, should limit to clip region if possible

```

(PROG (fulllist currlist clist nlist scan currcrit fullcrit crity count scany ex ty table)
  (SETQ table (FILL.INITBLE STRM FILL.SHADE FILL.FRULE))
  (SETQ fulllist (for I in ALIST collect (MAKESCAN I)))
  (SETQ fulllist (SORT fulllist 'CRIT.LESSP))
  (SETQ currlist NIL)
  (SETQ currcrit MIN.SMALLP)
  (SETQ scany (fetch (SCAN BY) of (CAR fulllist)))
  [while (OR currlist fulllist) do ;; merge new critical edges
    (while [AND (LISTP fulllist)
              (IEQP scany (fetch (SCAN BY) of (CAR fulllist))]
      do (SETQ scan (CAR fulllist))
        (SETQ currlist (CONS scan currlist))
        (SETQ fulllist (CDR fulllist)))
    (COND
      [(LISTP fulllist)
       (SETQ fullcrit (fetch (SCAN BY) of (CAR fulllist))
         (T (SETQ fullcrit MAX.SMALLP)))]
    ;; paint critical scan line
    (for scan in currlist do (CRITSECT scan scany))
    (SORT currlist 'SCAN.LESSP)
    (\DrawScanList.Display STRM currlist scany)
    ;; cull out exhausted edges
    (SETQ scany (IPLUS scany 1))
    ;(ILEQ currcrit scany)
    (SETQ currcrit MAX.SMALLP)
    (SETQ clist NIL)
    [for scan in currlist do (SETQ ty (fetch (SCAN TY) of scan))
      (SETQ currcrit (IMIN currcrit ty))
      (COND
        ((ILEQ scany ty)
         (SETQ clist (CONS scan clist))
         (SETQ currcrit (IMIN currcrit ty)))]
    (SETQ currlist clist)
    ;; paint normal scan lines
    (SETQ crity (IMIN currcrit fullcrit))
    (while (ILESSP scany crity) do (for scan in currlist
      do (\NORMSECT scan scany))
      (SORT currlist 'SCAN.LESSP)
      (\DrawScanList.Display STRM currlist
        scany)
      (SETQ scany (IPLUS scany 1]

(RETURN NIL])

```

(\POLYSHADE.XSCAN.IP

[LAMBDA (STRM ALIST FILL.SHADE FILL.FRULE FILL.WRULE) ; Edited 1-Feb-89 18:40 by FS

;; Current version of polygon code, Expects integer line lists, must be used in device coordinates (transposed 300 dpi)

```

(PROG (fulllist currlist clist nlist scan currcrit fullcrit crity count scany ix tx ex by ty)
  (SETCOLOR.IP STRM FILL.SHADE FILL.FRULE) ; cache texture as ip sampled black
  (SETQ fulllist (for I in ALIST collect (MAKESCAN I)))
  (SETQ fulllist (SORT fulllist 'CRIT.LESSP))
  (SETQ currlist NIL)
  (SETQ currcrit MIN.INTEGER)
  (SETQ scany (fetch (SCAN BY) of (CAR fulllist)))
  [while (OR currlist fulllist) do ;; merge new critical edges
    (while [AND (LISTP fulllist)
              (IEQP scany (fetch (SCAN BY) of (CAR fulllist))]
      do (SETQ scan (CAR fulllist))
        (SETQ currlist (CONS scan currlist))
        (SETQ fulllist (CDR fulllist)))
    (COND
      [(LISTP fulllist)
       (SETQ fullcrit (fetch (SCAN BY) of (CAR fulllist))
         (T (SETQ fullcrit MAX.INTEGER)))]
    ;; paint critical scan line
    (for scan in currlist do (CRITSECT scan scany))
    (SORT currlist 'SCAN.LESSP)
    (\DrawScanList.XScan.IP STRM currlist scany)
    ;; cull out exhausted edges
    (SETQ scany (IPLUS scany 1))
    ;(ILEQ currcrit scany)
    (SETQ currcrit MAX.INTEGER)
    (SETQ clist NIL)
    [for scan in currlist do (SETQ ty (fetch (SCAN TY) of scan))
      (SETQ currcrit (IMIN currcrit ty))
      (COND
        ((ILEQ scany ty)
         (SETQ clist (CONS scan clist)))]

```

```

                                (SETQ currcrit (IMIN currcrit ty)
                                (SETQ currlist clist)
;; paint normal scan lines
                                (SETQ crity (IMIN currcrit fullcrit))
                                (while (ILESSP scany crity) do (for scan in currlist
                                                                do (NORMSECT scan scany))
                                    (SORT currlist 'SCAN.LESSP)
                                    (\DrawScanList.XScan.IP STRM currlist
                                     scany)
                                    (SETQ scany (IPLUS scany 1]
                                (RETURN NIL])

```

(POLYSHADE.YSCAN.IP

```

[LAMBDA (STRM ALIST FILL.SHADE FILL.FRULE FILL.WRULE) ; Edited 6-Feb-89 18:32 by FS
;; Current version of polygon code, Expects integer line lists, must be used in device coordinates (transposed 300 dpi)
(PROG (fulllist currlist clist nlist scan currcrit fullcrit crity count scany ix tx ex by ty)
      (SETCOLOR.IP STRM FILL.SHADE FILL.FRULE) ; cache texture as ip sampled black
      (SETQ fulllist (MAPCAR ALIST 'MAKESCAN))
      (SETQ fulllist (SORT fulllist 'CRIT.LESSP))
      (SETQ currlist NIL)
      (SETQ currcrit MIN.INTEGER)
      (SETQ scany (fetch (SCAN BY) of (CAR fulllist)))
      [while (OR currlist fulllist) do ;; merge new critical edges
          (while [AND (LISTP fulllist)
                    (IEQP scany (fetch (SCAN BY) of (CAR fulllist))]
                do (SETQ scan (CAR fulllist))
                    (SETQ currlist (CONS scan currlist))
                    (SETQ fulllist (CDR fulllist)))
          (COND
            [(LISTP fulllist)
             (SETQ fullcrit (fetch (SCAN BY) of (CAR fulllist))]
             (T (SETQ fullcrit MAX.INTEGER)))
          ;; paint critical scan line
          (for scan in currlist do (CRITSECT scan scany))
          (SORT currlist 'SCAN.LESSP)
          (\DrawScanList.YScan.IP STRM currlist scany)
          ;; cull out exhausted edges
          (SETQ scany (IPLUS scany 1))
          ;(IEQP currcrit scany)
          (SETQ currcrit MAX.INTEGER)
          (SETQ clist NIL)
          [for scan in currlist do (SETQ ty (fetch (SCAN TY) of scan))
                                  (SETQ currcrit (IMIN currcrit ty))
                                  (COND
                                    ((ILEQ scany ty)
                                     (SETQ clist (CONS scan clist))
                                     (SETQ currcrit (IMIN currcrit ty))
                                  (SETQ currlist clist)
          ;; paint normal scan lines
          (SETQ crity (IMIN currcrit fullcrit))
          (while (ILESSP scany crity) do (for scan in currlist
                                                                do (NORMSECT scan scany))
                                    (SORT currlist 'SCAN.LESSP)
                                    (\DrawScanList.YScan.IP STRM currlist
                                     scany)
                                    (SETQ scany (IPLUS scany 1]
          (RETURN NIL])
)

```

(DEFINEQ

(POLYSHADE.BLT

```

[LAMBDA (STREAM POINTS TEXTURE OPERATION WINDNUMBER) (* FS "30-Oct-85 17:32")
(* * Convert knot list into internal data structures)
(LET (ILIST LLIST)
  (SETQ ILIST (PREPLOOP POINTS))
  (SETQ LLIST (MAPCAR ILIST 'KNOTLINE))
  (if (AND (NEQ WINDNUMBER 0)
           (NEQ WINDNUMBER 1))
      then (SETQ WINDNUMBER FILL.WRULE))
  (\POLYSHADE.BLT STREAM LLIST TEXTURE OPERATION WINDNUMBER])

```

(POLYSHADE.DISPLAY

```

[LAMBDA (STREAM POINTS TEXTURE OPERATION WINDNUMBER) ; Edited 24-Aug-87 19:44 by FS
;; Convert knot list into internal data structures, for now call generic routine

```

```
(LET (ILIST LLIST)
      (SETQ ILIST (PREPLOOP POINTS))
      (SETQ LLIST (MAPCAR ILIST 'KNOTLINE))
      (if (AND (NEQ WINDNUMBER 0)
               (NEQ WINDNUMBER 1))
          then (SETQ WINDNUMBER FILL.WRULE))
      (POLYSHADE.DISPLAY STREAM LLIST TEXTURE OPERATION WINDNUMBER}))
```

**(POLYSHADE.IP**

```
[LAMBDA (STREAM POINTS TEXTURE OPERATION WINDNUMBER) ; Edited 14-Jun-90 09:29 by takeshi
  ;; Default is to scan convert in Y direction.
  (DECLARE (GLOBALVARS PRINTER.DEFAULT.SCAN.DIRECTION))
  (POLYSHADE.SCAN.IP STREAM POINTS TEXTURE OPERATION WINDNUMBER (COND
    [(EQ (LISTGET (fetch (STREAM OTHERPROPS)
                        of STREAM)
                  'PRINTERMODE)
          'LANDSCAPE)
     ; This stream is treated as Landscape
     (COND
      ((EQ (LISTGET (fetch (STREAM OTHERPROPS)
                          of STREAM)
                    'P.SCAN.DIRECTION)
            'X)
          (PROGN 'Y))
       (T 'X))
      (T (LISTGET (fetch (STREAM OTHERPROPS)
                        of STREAM)
                  'P.SCAN.DIRECTION))
         ; Portrait
         ))
    ]))
```

**(POLYSHADE.SCAN.IP**

```
[LAMBDA (STREAM POINTS TEXTURE OPERATION WINDNUMBER SCANDIRECTION)
  ; Edited 25-May-2023 21:48 by lmm
  ; Edited 2-May-2023 09:12 by lmm
  ; Edited 1-Feb-89 18:53 by FS
  ;; Convert micras to device units, and transpose, tell Interpress to take dev units back to micras, convert knot list into internal data structures
  (LET (ILIST LLIST XTOX XTOY)
        (if (EQ SCANDIRECTION 'X)
            then (SETQ XTOX (\IPC MicrasToDev)) ; just scale
                (SETQ XTOY 0)
            else (SETQ XTOX 0) ; transpose & scale
                (SETQ XTOY (\IPC MicrasToDev)))
        [if (NUMBERP (CAAR POINTS))
            then (SETQ POINTS (IMLTLIST POINTS XTOX XTOY 0 XTOY XTOX 0))
            else (SETQ POINTS (for I in POINTS collect (IMLTLIST I XTOX XTOY 0 XTOY XTOX 0))
        (APPENDOP.IP STREAM (\IPC DOSAVESIMPLEBODY))
        (APPENDOP.IP STREAM (\IPC {}))
        (SCALE.IP STREAM 8.466666) ; 2540micraspi / 300dpi
        (CONCATT.IP STREAM) ; Convert to integer
        (SETQ ILIST (PREPLOOP POINTS))
        (SETQ LLIST (MAPCAR ILIST 'KNOTLINE))
        (if (AND (NEQ WINDNUMBER 0)
                 (NEQ WINDNUMBER 1))
            then (SETQ WINDNUMBER FILL.WRULE))
        (if (EQ SCANDIRECTION 'X)
            then (\POLYSHADE.XSCAN.IP STREAM LLIST TEXTURE OPERATION WINDNUMBER)
            else (\POLYSHADE.YSCAN.IP STREAM LLIST TEXTURE OPERATION WINDNUMBER))
        (APPENDOP.IP STREAM (\IPC {}))
  )
```

(DEFINEQ

**(FILLTRIANGLE**

```
[LAMBDA (X1 Y1 X2 Y2 X3 Y3 TEXTURE STREAM OPERATION) ; Edited 24-Aug-87 19:43 by FS
  (LET* ([TEMPLIST '( (10 . 10)
                      (100 . 100)
                      (200 . 30)
                    (PT1 (CAR TEMPLIST))
                    (PT2 (CADR TEMPLIST))
                    (PT3 (CADDR TEMPLIST)))
  ;; Smash temporary list here
  (RPLACA PT1 X1)
  (RPLACD PT1 Y1)
  (RPLACA PT2 X2)
  (RPLACD PT2 Y2)
  (RPLACA PT3 X3)
  (RPLACD PT3 Y3)
  (IF (EQ (IMAGESTREAMTYPE STREAM)
```

```

      ' INTERPRESS)
    THEN (\FILLPOLYGON.IP STREAM TEMPLIST TEXTURE OPERATION)
    ELSE (FILLPOLYGON TEMPLIST TEXTURE STREAM OPERATION)
)

```

;; Filled Circles

(DEFINEQ

(CIRC SHADE.BLT

[LAMBDA (STREAM CX CY R TEXTURE OPERATION)

; Edited 31-Mar-88 22:08 by FS

;; Bresenham's circle drawing routine, x and y are reversed

(LET (Y X E U V X0 Y0)

;; Avoid calling DSPOPERATION every call to bitblt

```

  (IF (NOT OPERATION)
    THEN (SETQ OPERATION (DSPOPERATION NIL STREAM)))

```

```

  (SETQ Y 0)
  (SETQ X R)
  (SETQ U 1)
  (SETQ V (IDIFFERENCE 1 (ITIMES 2 R)))
  (SETQ E (IDIFFERENCE 1 R))
  (BLT SHADE TEXTURE STREAM (IDIFFERENCE CX X)
    CY
    (IPLUS X X 1)
    1 OPERATION)

```

```

  (while (ILESSP Y X) do (SETQ X0 X)
    (SETQ Y0 Y)
    (if (MINUSP E)
      then (SETQ Y (ADD1 Y))
            (SETQ U (IPLUS U 2))
            (SETQ V (IPLUS V 2))
            (SETQ E (IPLUS E U))
      else (SETQ Y (ADD1 Y))
            (SETQ X (SUB1 X))
            (SETQ U (IPLUS U 2))
            (SETQ V (IPLUS V 4))
            (SETQ E (IPLUS E V)))
    (BLT SHADE TEXTURE STREAM (IDIFFERENCE CX X)
      (IDIFFERENCE CY Y)
      (IPLUS X X 1)
      1 OPERATION)
    (BLT SHADE TEXTURE STREAM (IDIFFERENCE CX X)
      (IPLUS CY Y)
      (IPLUS X X 1)
      1 OPERATION)
    (if (AND (NOT (EQUAL X0 X))
              (IGREATERP (IDIFFERENCE X0 Y0)
                1))
      then (BLT SHADE TEXTURE STREAM (IDIFFERENCE CX Y0)
        (IDIFFERENCE CY X0)
        (IPLUS Y0 Y0 1)
        1 OPERATION)
        (BLT SHADE TEXTURE STREAM (IDIFFERENCE CX Y0)
          (IPLUS CY X0)
          (IPLUS Y0 Y0 1)
          1 OPERATION]))

```

(CIRC SHADE.DISPLAY

[LAMBDA (STREAM CX CY R TEXTURE OPERATION)

; Edited 31-Mar-88 18:01 by FS

;; Bresenham's circle drawing routine, x and y are reversed

(LET (Y X E U V X0 Y0 table)

;; Initialize microcode table support

```

  (SETQ table (FILL.INITTBLE STREAM TEXTURE OPERATION))

```

;; Do Bresenham circle

```

  (SETQ Y 0)
  (SETQ X R)
  (SETQ U 1)
  (SETQ V (IDIFFERENCE 1 (ITIMES 2 R)))
  (SETQ E (IDIFFERENCE 1 R))
  (FILL.LINE table CY (IDIFFERENCE CX X)
    (IPLUS CX X))

```

```

  (while (ILESSP Y X) do (SETQ X0 X)
    (SETQ Y0 Y)
    (if (MINUSP E)
      then (SETQ Y (ADD1 Y))
            (SETQ U (IPLUS U 2))
            (SETQ V (IPLUS V 2))
            (SETQ E (IPLUS E U))
      else (SETQ Y (ADD1 Y))

```



```

        (SETQ X (SUB1 X))
        (SETQ U (IPLUS U 2))
        (SETQ V (IPLUS V 4))
        (SETQ E (IPLUS E V))
(FILL.LINE table (IDIFFERENCE CY Y)
  (IDIFFERENCE CX X)
  (IPLUS CX X))
(FILL.LINE table (IPLUS CY Y)
  (IDIFFERENCE CX X)
  (IPLUS CX X))
(if (AND (NOT (EQUAL X0 X))
  (IGREATERP (IDIFFERENCE X0 Y0)
    1))
  then (FILL.LINE table (IDIFFERENCE CY X0)
    (IDIFFERENCE CX Y0)
    (IPLUS CX Y0))
    (FILL.LINE table (IPLUS CY X0)
      (IDIFFERENCE CX Y0)
      (IPLUS CX Y0]))

```

(\CIRCSHADE.IP

[LAMBDA (STREAM CX CY R TEXTURE OPERATION) ; Edited 31-Mar-88 18:05 by FS

;; Bresenham's circle drawing routine, x and y are reversed; Used in device coordinates 300dpi transposed

```

(LET (Y X E U V X0 Y0)
  (SETCOLOR.IP STREAM TEXTURE OPERATION) ; cache texture as ip sampled black
  (SETQ Y 0)
  (SETQ X R)
  (SETQ U 1)
  (SETQ V (IDIFFERENCE 1 (ITIMES 2 R)))
  (SETQ E (IDIFFERENCE 1 R))
  (FILLRECTANGLE.IP STREAM CX (IDIFFERENCE CY X)
    1
    (IPLUS X X 1))
  (while (ILESSP Y X) do (SETQ X0 X)
    (SETQ Y0 Y)
    [COND
      ((MINUSP E)
        (SETQ Y (ADD1 Y))
        (SETQ U (IPLUS U 2))
        (SETQ V (IPLUS V 2))
        (SETQ E (IPLUS E U)))
      (T (SETQ Y (ADD1 Y))
        (SETQ X (SUB1 X))
        (SETQ U (IPLUS U 2))
        (SETQ V (IPLUS V 4))
        (SETQ E (IPLUS E V))
        (FILLRECTANGLE.IP STREAM (IDIFFERENCE CX Y)
          (IDIFFERENCE CY X)
          1
          (IPLUS X X 1))
        (FILLRECTANGLE.IP STREAM (IPLUS CX Y)
          (IDIFFERENCE CY X)
          1
          (IPLUS X X 1))
        (COND
          ((AND (NOT (EQUAL X0 X))
            (IGREATERP (IDIFFERENCE X0 Y0)
              1))
            (FILLRECTANGLE.IP STREAM (IDIFFERENCE CX X0)
              (IDIFFERENCE CY Y0)
              1
              (IPLUS Y0 Y0 1))
            (FILLRECTANGLE.IP STREAM (IPLUS CX X0)
              (IDIFFERENCE CY Y0)
              1
              (IPLUS Y0 Y0 1]))

```

(\CIRCSHADE.XSCAN.IP

[LAMBDA (STREAM CX CY R TEXTURE OPERATION) ; Edited 1-Feb-89 17:38 by FS

;; Bresenham's circle drawing routine; Used in device coordinates 300dpi, x and y are reversed. Some printers prefer to scan in the X direction, for  
 ;; those printers use this routine instead.

```

(LET (Y X E U V X0 Y0)
  (SETCOLOR.IP STREAM TEXTURE OPERATION) ; cache texture as ip sampled black
  (SETQ Y 0)
  (SETQ X R)
  (SETQ U 1)
  (SETQ V (IDIFFERENCE 1 (ITIMES 2 R)))
  (SETQ E (IDIFFERENCE 1 R))
  (FILLRECTANGLE.IP STREAM (IDIFFERENCE CX X)
    CY
    (IPLUS X X 1)
    1)
  (while (ILESSP Y X) do (SETQ X0 X)

```

```
(SETQ YO Y)
(if (MINUSP E)
  then (SETQ Y (ADD1 Y))
        (SETQ U (IPLUS U 2))
        (SETQ V (IPLUS V 2))
        (SETQ E (IPLUS E U))
  else (SETQ Y (ADD1 Y))
        (SETQ X (SUB1 X))
        (SETQ U (IPLUS U 2))
        (SETQ V (IPLUS V 4))
        (SETQ E (IPLUS E V)))
(FILLRECTANGLE.IP STREAM (IDIFFERENCE CX X)
 (IDIFFERENCE CY Y)
 (IPLUS X X 1)
 1)
(FILLRECTANGLE.IP STREAM (IDIFFERENCE CX X)
 (IPLUS CY Y)
 (IPLUS X X 1)
 1)
(if (AND (NOT (EQUAL X0 X))
        (IGREATERP (IDIFFERENCE X0 Y0)
 1)))
  then (FILLRECTANGLE.IP STREAM (IDIFFERENCE CX Y0)
 (IDIFFERENCE CY X0)
 (IPLUS Y0 Y0 1)
 1)
        (FILLRECTANGLE.IP STREAM (IDIFFERENCE CX Y0)
 (IPLUS CY X0)
 (IPLUS Y0 Y0 1)
 1]))
```

)

(DEFINEQ

**(CIRCSHADE.BLT**

[LAMBDA (STREAM CENTERX CENTERY RADIUS TEXTURE OPERATION) ; Edited 31-Mar-88 21:56 by FS

;; GenericBlit circle shader, check args

```
(if (AND (NUMBERP CENTERX)
        (NUMBERP CENTERY)
        (NUMBERP RADIUS)
        (STREAMP STREAM)
        (TEXTUREP TEXTURE))
```

```
  then (\CIRCSHADE.BLT STREAM CENTERX CENTERY RADIUS TEXTURE OPERATION)
  else (ERROR "Bad argument(s)"])
```

**(CIRCSHADE.DISPLAY**

[LAMBDA (STREAM CENTERX CENTERY RADIUS TEXTURE OPERATION) ; Edited 31-Mar-88 18:01 by FS

;; Check args; This routine not currently used \*\*

```
(if (AND (NUMBERP CENTERX)
        (NUMBERP CENTERY)
        (NUMBERP RADIUS)
        (STREAMP STREAM)
        (TEXTUREP TEXTURE))
```

```
  then (\CIRCSHADE.DISPLAY STREAM CENTERX CENTERY RADIUS TEXTURE OPERATION)
  else (ERROR "Bad argument(s)"])
```

**(CIRCSHADE.IP**

[LAMBDA (STREAM CENTERX CENTERY RADIUS TEXTURE OPERATION) ; Edited 1-Feb-89 17:12 by FS

;; Interpress2.1 doesn't support ARCTO, so must either approximate a circle (as here), or scan convert it (e.g. CIRCSHADE.IP)  
 ;; This code does not generate as nicely "round" circles as circshade.ip (the difference is visible to the naked eye). However, this code should be  
 ;; better for landscape printing, for code which uses pushstate/popstate, and for printers which scan in the X direction (e.g. Fuji Xerox XP-9),  
 ;; because it generates a simpler master.  
 ;; Wimp out and display regular N-gon. For smaller circles, can use fewer points? Could also render two half circles (thus allowing twice the  
 ;; number of points since there are two trajectories), but what the heck.  
 ;; Note also the clipping code isn't integrated with this (nor TRAJECTORY.IP, or others).  
 (FILLNGON.IP STREAM 90 RADIUS CENTERX CENTERY TEXTURE OPERATION])

)

(DEFINEQ

**(FILLCIRCLE.IP**

[LAMBDA (STREAM CENTERX CENTERY RADIUS TEXTURE OPERATION) ; Edited 1-Feb-89 17:12 by FS

;; Interpress2.1 doesn't support ARCTO, so must either approximate a circle (as here), or scan convert it (e.g. CIRCSHADE.IP)  
 ;; This code does not generate as nicely "round" circles as circshade.ip (the difference is visible to the naked eye). However, this code should be  
 ;; better for landscape printing, for code which uses pushstate/popstate, and for printers which scan in the X direction (e.g. Fuji Xerox XP-9),  
 ;; because it generates a simpler master.  
 ;; Wimp out and display regular N-gon. For smaller circles, can use fewer points? Could also render two half circles (thus allowing twice the  
 ;; number of points since there are two trajectories), but what the heck.

:: Note also the clipping code isn't integrated with this (nor TRAJECTORY.IP, or others).  
(FILLNGON.IP STREAM 90 RADIUS CENTERX CENTERY TEXTURE OPERATION])

(FILLNGON.IP

[LAMBDA (IPSTREAM NPOINTS RADIUS CENTERX CENTERY TEXTURE OPERATION)  
; Edited 2-May-2023 08:46 by Imm  
; Edited 1-Feb-89 17:19 by FS

:: Create and fill a regular polygon (standing on its tip). Since its convex, we can use the primitive IP operator to do the job. Note there is no  
:: clipping in this routine.  
:: Could have used FILLTRAJECTORY.IP, but this function CONSeS less. Could have walked 1/8 of circle and used symmetry, but what the  
:: heck.....

(LET (BASEANGLE ANGLE X Y)

:: Try to avoid limitations of printers. Anything more than 64 or so looks for all intents and purposes like a circle anyway.

(if (IGREATERP NPOINTS MAXSEGSPERTRAJECTORY)  
then (SETQ NPOINTS MAXSEGSPERTRAJECTORY))  
(SETQ BASEANGLE (FQUOTIENT 360 NPOINTS))  
(APPENDOP.IP IPSTREAM (\IPC DOSAVESIMPLEBODY)) ; Save state (to undo SETCOLOR)  
(APPENDOP.IP IPSTREAM (\IPC {}))  
(SETCOLOR.IP IPSTREAM TEXTURE OPERATION)  
(MOVETO.IP IPSTREAM CENTERX (IPLUS CENTERY RADIUS)) ; handle 0 point specially

:: Note that the trajectory is not closed, IP spec says outlines get closed anyway.

(for I from 1 to (SUB1 NPOINTS) do (SETQ ANGLE (TIMES I BASEANGLE))  
; Since these are micas, we can avoid some floating point by  
; forcing values to be integer  
[SETQ X (IPLUS CENTERX (TIMES RADIUS (SIN ANGLE)]  
[SETQ Y (IPLUS CENTERY (TIMES RADIUS (COS ANGLE)]  
(LINETO.IP IPSTREAM X Y))  
(APPENDINTEGER.IP IPSTREAM 1) ; number of trajectories  
(APPENDOP.IP IPSTREAM (\IPC MAKEOUTLINE))  
(APPENDOP.IP IPSTREAM (\IPC MASKFILL))  
(APPENDOP.IP IPSTREAM (\IPC {})) ; restore state  
NIL])

)  
(MOVD 'FILLCIRCLE.IP 'CIRCSHADE.IP)

::: Considering scan direction of the printer  
::: You must set these vars before opening IMAGESTREAM

(RPAQQ PRINTER.DEFAULT.SCAN.DIRECTION Y)  
(RPAQQ PRINTER.SCAN.DIRECTIONS.LIST NIL)

::: PBBT Optimized routines

(RPAQQ PBBT.PANEL  
( (NULL (SETQ MYDSTRM (\OUTSTREAMARG MYWIN))  
(TIMEALL (FILL.CONT MYDSTRM (RAND)  
'REPLACE 10 300)  
20)  
(TIMEALL (FILL.TEST MYDSTRM (RAND)  
'REPLACE 10 300)  
20)  
(TIMEALL (FILL.XPER MYDSTRM (RAND)  
'REPLACE 10 300)  
20)  
(TIMEALL (FILLPOLYGON KLIST (RAND)  
MYDSTRM)  
20)  
(TIMEALL (BLTSHADE (RAND)  
MYDSTRM 10 10 290 290 'REPLACE)  
20))))

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

(DATATYPE FILL.TBLE (FBBT TX TY ADDR LLEN TXTW TXTH TXTA BITS LFT RGT TOP BOT TEXT STRM))  
)

(/DECLAREDATATYPE 'FILL.TBLE '(POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER  
POINTER POINTER POINTER POINTER POINTER)

:: ---field descriptor list elided by lister---

'30)

(/DECLAREDATATYPE 'FILL.TBLE '(POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER



```

      (SETQ TXTA GRAYBASE)
      (SETQ TEXT TEXTUREBM)
      (SETQ STRM DISPLAYSTREAM)
      (SETQ BITS NBITS)
      (SETQ LFT LEFT)
      (SETQ RGT RIGHT)
      (SETQ TOP TOPP)
      (SETQ BOT BOTTOM)
      (RETURN FILLTBLE])

```

**(FILL.INCY**

```

[LAMBDA (TBLE) (* FS "29-Oct-85 19:46")
  (with FILL.TBLE TBLE (*)
    (SETQ ADDR (IPLUS ADDR LLEN))
    (with PILOTBBT FBFT (*)
      (SETQ PBTDEST ADDR])

```

**(FILL.LINE**

```

[LAMBDA (TBLE Y XL XR) (* FS "29-Oct-85 20:19")
  (with FILL.TBLE TBLE (*)
    (SETQ Y (IDIFFERENCE TY Y))
    (SETQ XL (IPLUS TX XL))
    (SETQ XR (IPLUS TX XR))
    (\LINEBLT FBFT XL Y XR ADDR LLEN LFT RGT BOT TOP TXTW TXTH TXTA BITS])

```

)

(DEFINEQ

**(FILL.TEST**

```

[LAMBDA (STRM TEXT OPER YMIN YMAX) (* FS "29-Oct-85 19:59")
  (LET (TBLE)
    (SETQ OPER 'REPLACE)
    (SETQ TBLE (FILL.INITBLE STRM TEXT OPER))
    (for I from YMIN to YMAX do (FILL.LINE TBLE I I 300])

```

**(FILL.XPER**

```

[LAMBDA (STRM TEXT OPER YMIN YMAX) (* FS "30-Oct-85 19:30")
  (LET (TBLE)
    (SETQ OPER 'REPLACE)
    (SETQ TBLE (FILL.INITBLE STRM TEXT OPER))
    (for I from YMIN to YMAX do (.WHILE.TOP.DS. STRM (FILL.LINE TBLE I I 300])

```

**(FILL.CONT**

```

[LAMBDA (STRM TEXT OPER YMIN YMAX) (* FS "29-Oct-85 20:11")
  (LET (TBLE)
    (SETQ OPER 'REPLACE)
    (SETQ TBLE (FILL.INITBLE STRM TEXT OPER))
    (for I from YMIN to YMAX do (BLTSHADE TEXT STRM I I (IDIFFERENCE 300 I)
      1 OPER])

```

)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

```

(FILESLoad (LOADCOMP)
  INTERPRESS XXGEOM)

```

)

---

**FUNCTION INDEX**

CIRCSHADE.BLT .....	10	FILL.LINE .....	13	POLYSHADE.BLT .....	6	\CIRCSHADE.IP .....	9
CIRCSHADE.DISPLAY .....	10	FILL.TEST .....	13	POLYSHADE.DISPLAY .....	6	\CIRCSHADE.XSCAN.IP .....	9
CIRCSHADE.IP .....	10	FILL.XPER .....	13	POLYSHADE.IP .....	7	\POLYSHADE.BLT .....	4
CRIT.LESSP .....	1	FILLCIRCLE.IP .....	10	POLYSHADE.SCAN.IP .....	7	\POLYSHADE.DISPLAY .....	4
CRITSECT .....	4	FILLNGON.IP .....	11	SCAN.LESSP .....	1	\POLYSHADE.XSCAN.IP .....	5
FILL.CONT .....	13	FILLTRIANGLE .....	7	SHEDSCAN .....	3	\POLYSHADE.YSCAN.IP .....	6
FILL.INCY .....	13	MAKESCAN .....	3	\CIRCSHADE.BLT .....	8		
FILL.INITTBLE .....	12	NORMSECT .....	3	\CIRCSHADE.DISPLAY .....	8		

---

**VARIABLE INDEX**

FILL.WRULE .....	2	PRINTER.DEFAULT.SCAN.DIRECTION ..	11	\FILL.DEBUG .....	2
PBBT.PANEL .....	11	PRINTER.SCAN.DIRECTIONS.LIST .....	11		

---

**MACRO INDEX**

\DrawScanList.Blt .....	2	\DrawScanList.XScan.IP .....	2	\NORMSECT .....	2
\DrawScanList.Display .....	2	\DrawScanList.YScan.IP .....	3		

---

**RECORD INDEX**

FILL.TBLE .....	11	SCAN .....	2
-----------------	----	------------	---

---