

File created: 7-Oct-88 12:26:27 {INDIGO}<GSLWS>LYRIC>LIBRARY>PLOT OBJECTS2.;6

changes to: (FNS CLIPPED.FILLPOLYGON CLIPPED.POLYGON FINISH-CLIP-POLYGON CLIP-POLYGON-VERTEX CLIP-INSIDEP
CLIP-INTERSECT GETFILLEDPOLYGON DRAWFILLEDPOLYGON CREATEFILLEDPOLYGON DISTANCETOFILLEDPOLYGON
ERASEFILLEDPOLYGON EXTENTOFFILLEDPOLYGON HIGHLIGHTFILLEDPOLYGON PLOTFILLEDPOLYGON
CLIPPED.FINDTO CLIPPED.FINDLINE COPYFILLEDPOLYGON MOVEFILLEDPOLYGON PUTFILLEDPOLYGON)
(VARS PLOT OBJECTS2COMS)
(RECORDS CLIPEDGEINFO FILLEDPOLYCONDATA)

previous date: 5-Oct-88 11:21:10 {INDIGO}<GSLWS>LYRIC>LIBRARY>PLOT OBJECTS2.;1

Read Table: XCL

Package: INTERLISP

Format: XCCS

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

(RPAQQ PLOT OBJECTS2COMS

```
((FNS COPYFILLEDPOLYGON CREATEFILLEDPOLYGON DISTANCETOFILLEDPOLYGON DRAWFILLEDPOLYGON ERASEFILLEDPOLYGON  
EXTENTOFFILLEDPOLYGON GETFILLEDPOLYGON HIGHLIGHTFILLEDPOLYGON MOVEFILLEDPOLYGON PLOTFILLEDPOLYGON  
PUTFILLEDPOLYGON)  
(VARS OBJECT2OPSTABLE)  
(RECORDS FILLEDPOLYCONDATA)  
(P (PLOT.SETUP OBJECT2OPSTABLE))  
(FNS CLIPPED.FILLPOLYGON CLIPPED.POLYGON CLIP-POLYGON-VERTEX FINISH-CLIP-POLYGON CLIP-INSIDEP  
CLIP-INTERSECT)  
(RECORDS CLIPEDGEINFO)))
```

(DEFINEQ

(COPYFILLEDPOLYGON

(LAMBDA (PLOT OBJECT PLOT)

; Edited 5-Oct-88 10:23 by thh:

:: Copyfn for FILLEDPOLYGON objects

```
(LET ((OBJECTDATA (FETCH (PLOT OBJECT OBJECTDATA) OF PLOT OBJECT)))  
(CREATE FILLEDPOLYCONDATA  
POLYGONPOINTS _ (COPYALL (FETCH (FILLEDPOLYCONDATA POLYGONPOINTS) OF OBJECTDATA))  
STYLE _ (COPYALL (FETCH (FILLEDPOLYCONDATA STYLE) OF OBJECTDATA))  
TEXTURE _ (FETCH (FILLEDPOLYCONDATA TEXTURE) OF OBJECTDATA))))
```

(CREATEFILLEDPOLYGON

(LAMBDA (POSITIONS LABEL STYLE TEXTURE MENU)

; Edited 5-Oct-88 12:49 by thh:

```
(CREATEPLOT OBJECT FILLEDPOLYGNFNS 'FILLEDPOLYGON LABEL MENU (|create| FILLEDPOLYCONDATA  
POLYGONPOINTS _ POSITIONS  
STYLE _  
(COND  
( (FIXP STYLE)  
(|create| PLOT.STYLE  
LINewidth _ STYLE))  
( (LISTP STYLE)  
(|create| PLOT.STYLE  
LINewidth _ (CAR STYLE)  
DASHING _ (CADR STYLE)  
COLOR _ (CADDR STYLE)))  
(T (|create| PLOT.STYLE  
LINewidth _ 1)))  
TEXTURE _ TEXTURE))))
```

(DISTANCETOFILLEDPOLYGON

(LAMBDA (FILLEDPOLYGON STREAMPOSITION PLOT)

; Edited 5-Oct-88 10:32 by thh:

```
(LIMETRIC STREAMPOSITION (|for| POINT |in| (|fetch| (FILLEDPOLYCONDATA STREAMPOINTS) |of| (|fetch| OBJECTDATA  
|of| FILLEDPOLYGON))  
|smallest| (LIMETRIC POINT STREAMPOSITION))))
```

(DRAWFILLEDPOLYGON

(LAMBDA (FILLEDPOLYGON VIEWPORT PLOT)

; Edited 5-Oct-88 13:05 by thh:

```
(LET* ((STREAM (|fetch| (VIEWPORT PARENTSTREAM) |of| VIEWPORT))  
(STREAMSUBREGION (|fetch| (VIEWPORT STREAMSUBREGION) |of| VIEWPORT))  
(OBJECTDATA (|fetch| (PLOT OBJECT OBJECTDATA) |of| FILLEDPOLYGON))  
(POINTS (|fetch| (FILLEDPOLYCONDATA POLYGONPOINTS) |of| OBJECTDATA))  
(STREAMPOINTS (|for| PT |in| POINTS |collect| (WORLDTOSTREAM PT VIEWPORT)))  
(STYLE (|fetch| (FILLEDPOLYCONDATA STYLE) |of| OBJECTDATA))  
(LINewidth (TIMES (DSPSCALE NIL STREAM)  
(|fetch| (PLOT.STYLE LINewidth) |of| STYLE)))  
(DASHING (|fetch| (PLOT.STYLE DASHING) |of| STYLE))  
(COLOR (|fetch| (PLOT.STYLE COLOR) |of| STYLE)))  
(CLIPPED.FILLPOLYGON STREAMSUBREGION STREAMPOINTS (|fetch| (FILLEDPOLYCONDATA TEXTURE) |of| OBJECTDATA)  
STREAM  
'REPLACE NIL (< 0 LINewidth)  
LINewidth  
'REPLACE COLOR DASHING)  
(COND  
( (EQ STREAM (WINDOWPROP (|fetch| (PLOT PLOTWINDOW) |of| PLOT)
```

```
'DSP))
(|replace| (FILLED POLYGON DATA STREAM POINTS) |of| OBJECT DATA |with| STREAM POINTS))))))
```

(ERASE FILLED POLYGON

(LAMBDA (FILLED POLYGON VIEWPORT PLOT) ; Edited 5-Oct-88 13:05 by thh:

;; Erase a FILLED POLYGON DATA

```
(LET* ((STREAM (|fetch| (VIEWPORT PARENT STREAM) |of| VIEWPORT))
        (STREAM SUB REGION (|fetch| (VIEWPORT STREAM SUB REGION) |of| VIEWPORT))
        (OBJECT DATA (|fetch| (PLOT OBJECT OBJECT DATA) |of| FILLED POLYGON))
        (STREAM POINTS (|fetch| (FILLED POLYGON DATA STREAM POINTS) |of| OBJECT DATA))
        (STYLE (|fetch| (FILLED POLYGON DATA STYLE) |of| OBJECT DATA))
        (LINE WIDTH (IPLUS 2 (|fetch| (PLOT STYLE LINE WIDTH) |of| STYLE)))
        (COLOR (|fetch| (PLOT STYLE COLOR) |of| STYLE)))
        (CLIPPED.FILL POLYGON STREAM SUB REGION STREAM POINTS (|fetch| (FILLED POLYGON DATA TEXTURE) |of| OBJECT DATA)
         STREAM
         'ERASE NIL (< 0 (|fetch| (PLOT STYLE LINE WIDTH) |of| STYLE))
         LINE WIDTH
         'ERASE COLOR))))
```

(EXTENT OF FILLED POLYGON

(LAMBDA (FILLED POLYGON) ; Edited 5-Oct-88 10:50 by thh:

```
(|bind| (MINX _ MAX.FLOAT)
        (MAXX _ MIN.FLOAT)
        (MINY _ MAX.FLOAT)
        (MAXY _ MIN.FLOAT)
        X Y |for| POSITION |in| (|fetch| (FILLED POLYGON DATA POLYGON POINTS) |of| (|fetch| OBJECT DATA |of| FILLED POLYGON))
|declare| (TYPE FLOATING MINX MAXX MINY MAXY X Y) |do| (SETQ X (|fetch| XCOORD |of| POSITION))
          (SETQ Y (|fetch| YCOORD |of| POSITION))
          (COND
            ((FLESSP X MINX)
             (SETQ MINX X)))
          (COND
            ((FGREATERP X MAXX)
             (SETQ MAXX X)))
          (COND
            ((FLESSP Y MINY)
             (SETQ MINY Y)))
          (COND
            ((FGREATERP Y MAXY)
             (SETQ MAXY Y)))
|finally| (RETURN (|create| EXTENT
                      MINX _ MINX
                      MAXX _ MAXX
                      MINY _ MINY
                      MAXY _ MAXY))))
```

(GET FILLED POLYGON

(LAMBDA (PROPLST) ; Edited 5-Oct-88 13:22 by thh:

```
(LET ((STYLE LST (LISTGET PROPLST 'STYLE)))
        (|create| FILLED POLYGON DATA
        POLYGON POINTS _ (LISTGET PROPLST 'POLYGON POINTS)
        STYLE _ (|create| PLOT STYLE
        LINE WIDTH _ (CAR STYLE LST)
        DASHING _ (CADR STYLE LST)
        COLOR _ (CADDR STYLE LST))
        TEXTURE _ (LISTGET PROPLST 'TEXTURE))))
```

(HIGHLIGHT FILLED POLYGON

(LAMBDA (FILLED POLYGON VIEWPORT PLOT) ; Edited 5-Oct-88 13:12 by thh:

```
(LET* ((STREAM (|fetch| (VIEWPORT PARENT STREAM) |of| VIEWPORT))
        (STREAM SUB REGION (|fetch| (VIEWPORT STREAM SUB REGION) |of| VIEWPORT))
        (OBJECT DATA (|fetch| (PLOT OBJECT OBJECT DATA) |of| FILLED POLYGON))
        (STREAM POINTS (|fetch| (FILLED POLYGON DATA STREAM POINTS) |of| OBJECT DATA))
        (STYLE (|fetch| (FILLED POLYGON DATA STYLE) |of| OBJECT DATA))
        (LINE WIDTH (IPLUS 2 (|fetch| (PLOT STYLE LINE WIDTH) |of| STYLE)))
        (COLOR (|fetch| (PLOT STYLE COLOR) |of| STYLE)))
        (CLIPPED.FILL POLYGON STREAM SUB REGION STREAM POINTS BLACKSHADE STREAM 'INVERT NIL
         (< 0 (|fetch| (PLOT STYLE LINE WIDTH) |of| STYLE))
         LINE WIDTH
         'INVERT COLOR))))
```

(MOVE FILLED POLYGON

(LAMBDA (FILLED POLYGON DX DY PLOT) ; Edited 5-Oct-88 11:09 by thh:

```
(LET ((POINTS (FETCH (FILLED POLYGON DATA POLYGON POINTS) OF (FETCH OBJECT DATA OF FILLED POLYGON)))
        (FOR POINT IN POINTS DO (REPLACE XCOORD OF POINT WITH (PLUS DX (FETCH XCOORD OF POINT)))
        (REPLACE YCOORD OF POINT WITH (PLUS DY (FETCH YCOORD OF POINT)))))
```

(PLOT FILLED POLYGON

(LAMBDA (PLOT POSITIONS LABEL STYLE TEXTURE MENU NODRAW FLG) ; Edited 5-Oct-88 11:11 by thh:

```
(COND
```

```
((NOT (|type?| PLOT PLOT))
 (HELP "NOT a PLOT" PLOT))
(ADD PLOT OBJECT (CREATE FILLED POLYGON POSITIONS LABEL STYLE TEXTURE MENU)
 PLOT NODRAW FLG))
```

(PUT FILLED POLYGON

```
(LAMBDA (PLOT OBJECT PLOT STREAM) ; Edited 5-Oct-88 11:13 by thh:
 (PROG ((OBJECT DATA (|fetch| (PLOT OBJECT OBJECT DATA) |of| PLOT OBJECT)
 STYLE)
 (SETQ STYLE (|fetch| (FILLED POLYGON DATA STYLE) |of| OBJECT DATA))
 (PRINTOUT STREAM (" \, "POLYGON POINTS" \, .P2 (|fetch| (FILLED POLYGON DATA POLYGON POINTS) |of| OBJECT DATA)
 \, "TEXTURE" \, .P2 (|fetch| (FILLED POLYGON DATA TEXTURE) |of| OBJECT DATA)
 \, "STYLE" \, .P2 (LIST (|fetch| (PLOT STYLE LINEWIDTH) |of| STYLE)
 (|fetch| (PLOT STYLE DASHING) |of| STYLE)
 (|fetch| (PLOT STYLE COLOR) |of| STYLE)
 \, ")"))))
)
```

(RPAQQ OBJECT 2OPSTABLE

```
((FILLED POLYGON (DRAW FN DRAW FILLED POLYGON)
 (ERASE FN ERASE FILLED POLYGON)
 (HIGHLIGHT FN HIGHLIGHT FILLED POLYGON)
 (MOVE FN MOVE FILLED POLYGON)
 (LABEL FN LABEL GENERIC)
 (EXTENT FN EXTENT OF FILLED POLYGON)
 (DISTANCE FN DISTANCE TO FILLED POLYGON)
 (COPY FN COPY FILLED POLYGON)
 (PUT FN PUT FILLED POLYGON)
 (GET FN GET FILLED POLYGON)))
```

```
(DECLARE \: EVAL@COMPILE
```

```
(DATATYPE FILLED POLYGON DATA (POLYGON POINTS STREAM POINTS STYLE TEXTURE)
 STYLE _ 1)
)
```

```
(/DECLARE DATATYPE 'FILLED POLYGON DATA ' (POINTER POINTER POINTER POINTER)
 ;; ---field descriptor list elided by lister---
 ' 8)
```

```
(PLOT.SETUP OBJECT 2OPSTABLE)
```

```
(DEFINEQ
```

(CLIPPED.FILL POLYGON

```
(LAMBDA (CLIPPING REGION POINTS TEXTURE STREAM OPERATION WIND NUMBER DRAW? WIDTH DRAW OPERATION COLOR DASHING)
 ; Edited 7-Oct-88 09:03 by thh:
 ;; Clip filled polygon against CLIPPING REGION. If DRAW? is non-NIL, the clipped perimeter of the polygon is drawn as well using the remaining
 ;; parameters.
 (LET ((CLIPPED POINTS (CLIPPED.POLYGON CLIPPING REGION POINTS)))
 ; CLIPPED POINTS is NIL if polygon doesn't intersect
 ; CLIPPING REGION
 (COND
 (CLIPPED POINTS
 ;; fill clipped polygon
 (FILL POLYGON CLIPPED POINTS TEXTURE STREAM OPERATION WIND NUMBER)
 ;; draw if requested
 (AND DRAW? (|bind| (START _ (CAR POINTS)) |first| (MOVETO (|fetch| XCOORD |of| START)
 (|fetch| YCOORD |of| START)
 STREAM)
 |for| PT |in| (CDR POINTS) |do| (CLIPPED.DRAW TO CLIPPING REGION
 (|fetch| XCOORD |of| PT)
 (|fetch| YCOORD |of| PT)
 WIDTH DRAW OPERATION STREAM COLOR DASHING)
 |finally| (CLIPPED.DRAW TO CLIPPING REGION (|fetch| XCOORD |of| START)
 (|fetch| YCOORD |of| START)
 WIDTH DRAW OPERATION STREAM COLOR DASHING))))))))))
```

(CLIPPED.POLYGON

```
(LAMBDA (CLIPPING REGION POINTS) ; Edited 6-Oct-88 17:10 by thh:
 ;; clips polygon whose vertices are given in POINTS to CLIPPING REGION using Sutherland-Hodgman algorithm. cf. p.450 of Foley and Van Dam
 (LET* ((LEFT (|fetch| LEFT |of| CLIPPING REGION))
 (RIGHT (|fetch| RIGHT |of| CLIPPING REGION))
 (TOP (|fetch| TOP |of| CLIPPING REGION))
 (BOTTOM (|fetch| BOTTOM |of| CLIPPING REGION))
 (EDGES (LIST (|create| CLIP EDGE INFO
 X _ LEFT
 Y _ BOTTOM
 END _ TOP
```

```

VERTICAL? _ T)
(|create| CLIPEDGEINFO
X _ LEFT
Y _ TOP
END _ RIGHT
VERTICAL? _ NIL)
(|create| CLIPEDGEINFO
X _ RIGHT
Y _ TOP
END _ BOTTOM
VERTICAL? _ T)
(|create| CLIPEDGEINFO
X _ RIGHT
Y _ BOTTOM
END _ LEFT
VERTICAL? _ NIL)))

```

CLIPPEDPOINTS)

;; each edge in EDGES is a pair of points such that on moving from first to second, inside of CLIPPINGREGION is on the right. THESE
;; ARE LEFT, TOP, RIGHT AND BOTTOM EDGES RESPECTIVELY.

```

(FOR PT IN POINTS DO (SETQ CLIPPEDPOINTS (CLIP-POLYGON-VERTEX PT EDGES CLIPPEDPOINTS)))
(FINISH-CLIP-POLYGON EDGES CLIPPEDPOINTS)))

```

(CLIP-POLYGON-VERTEX

(LAMBDA (POINT EDGES CLIPPEDPOINTS)

; Edited 6-Oct-88 16:02 by thh:

;;; implements single step of Sutherland-Hodgman algorithm

```

(COND
(EDGES (LET* ((EDGE (CAR EDGES))
(PREVPOINT (|fetch| (CLIPEDGEINFO PREVPT) |of| EDGE))
(PREVINSIDE? (|fetch| (CLIPEDGEINFO PREVINSIDE?) |of| EDGE))
(INSIDE? (CLIP-INSIDEP POINT EDGE))))
;; update points and check for intersection
(COND
((|fetch| (CLIPEDGEINFO FIRSTPT) |of| EDGE)
;; this is not first point of polygon to be clipped with this edge
(COND
((NEQ PREVINSIDE? INSIDE?) ; polygon side crosses edge
(SETQ CLIPPEDPOINTS (CLIP-POLYGON-VERTEX (CLIP-INTERSECT PREVPOINT POINT EDGE)
(CDR EDGES)
CLIPPEDPOINTS))))))
(T ;; this is first point of the polygon for this edge
(|replace| (CLIPEDGEINFO FIRSTPT) |of| EDGE |with| POINT)
(|replace| (CLIPEDGEINFO FIRSTINSIDE?) |of| EDGE |with| INSIDE?)))
(|replace| (CLIPEDGEINFO PREVPT) |of| EDGE |with| POINT)
(|replace| (CLIPEDGEINFO PREVINSIDE?) |of| EDGE |with| INSIDE?))
;;
;; check if new point should be included
(COND
(INSIDE? (SETQ CLIPPEDPOINTS (CLIP-POLYGON-VERTEX POINT (CDR EDGES)
CLIPPEDPOINTS))))))
(T
(PUSH CLIPPEDPOINTS POINT)))
CLIPPEDPOINTS))

```

(FINISH-CLIP-POLYGON

(LAMBDA (EDGES CLIPPEDPOINTS)

; Edited 6-Oct-88 16:10 by thh:

```

(COND
(EDGES (LET ((EDGE (CAR EDGES)))
(COND
((AND CLIPPEDPOINTS (NEQ (|fetch| (CLIPEDGEINFO FIRSTINSIDE?) |of| EDGE)
(|fetch| (CLIPEDGEINFO PREVINSIDE?) |of| EDGE)))
; last side of polygon crosses edge
(SETQ CLIPPEDPOINTS (CLIP-POLYGON-VERTEX (CLIP-INTERSECT (|fetch| (CLIPEDGEINFO FIRSTPT)
|of| EDGE)
(|fetch| (CLIPEDGEINFO PREVPT)
|of| EDGE)
EDGE)
(CDR EDGES)
CLIPPEDPOINTS))))))
(|replace| (CLIPEDGEINFO FIRSTPT) |of| EDGE |with| NIL)
(FINISH-CLIP-POLYGON (CDR EDGES)
CLIPPEDPOINTS)))
(T CLIPPEDPOINTS)))

```

(CLIP-INSIDEP

(LAMBDA (PT EDGE)

; Edited 6-Oct-88 16:32 by thh:

;; T if PT is on or to the right of the directed EDGE (which is the inside of the region of which it is a part)

```
(COND
  ((|fetch| (CLIPEDGEINFO VERTICAL?) |of| EDGE) ; vertical edge
   (COND
    ((GREATERP (|fetch| (CLIPEDGEINFO END) |of| EDGE)
              (|fetch| (CLIPEDGEINFO Y) |of| EDGE)) ; edge is going up, right is positive x-axis
     (GEQ (|fetch| XCOORD |of| PT)
          (|fetch| (CLIPEDGEINFO X) |of| EDGE)))
    (T (LEQ (|fetch| XCOORD |of| PT)
            (|fetch| (CLIPEDGEINFO X) |of| EDGE))))
   )
  (T ; horizontal edge
   (COND
    ((GREATERP (|fetch| (CLIPEDGEINFO END) |of| EDGE)
              (|fetch| (CLIPEDGEINFO X) |of| EDGE)) ; edge is going right, right is negative y-axis
     (LEQ (|fetch| YCOORD |of| PT)
          (|fetch| (CLIPEDGEINFO Y) |of| EDGE)))
    (T (GEQ (|fetch| YCOORD |of| PT)
            (|fetch| (CLIPEDGEINFO Y) |of| EDGE))))))
  )
```

(CLIP-INTERSECT

(LAMBDA (P1 P2 EDGE) ; Edited 6-Oct-88 16:42 by thh:

;; returns point where segment between P1 and P2 intersect EDGE (the two points are on opposite sides of the edge)

```
(COND
  ((|fetch| (CLIPEDGEINFO VERTICAL?) |of| EDGE) ; vertical edge
   (LET ((X (|fetch| (CLIPEDGEINFO X) |of| EDGE)))
    (|create| POSITION
              XCOORD _ X
              YCOORD _ (PLUS (|fetch| YCOORD |of| P1)
                             (QUOTIENT (TIMES (DIFFERENCE X (|fetch| XCOORD |of| P1))
                                           (DIFFERENCE (|fetch| YCOORD |of| P2)
                                                       (|fetch| YCOORD |of| P1)))
                             (DIFFERENCE (|fetch| XCOORD |of| P2)
                                           (|fetch| XCOORD |of| P1))))))
   )
  (T ; horizontal edge
   (LET ((Y (|fetch| (CLIPEDGEINFO Y) |of| EDGE)))
    (|create| POSITION
              XCOORD _ (PLUS (|fetch| XCOORD |of| P1)
                             (QUOTIENT (TIMES (DIFFERENCE Y (|fetch| YCOORD |of| P1))
                                           (DIFFERENCE (|fetch| XCOORD |of| P2)
                                                       (|fetch| XCOORD |of| P1)))
                             (DIFFERENCE (|fetch| YCOORD |of| P2)
                                           (|fetch| YCOORD |of| P1))))))
    YCOORD _ Y))))))
```

```
)
(DECLARE\ : EVAL@COMPILE
(RECORD CLIPEDGEINFO (X Y END VERTICAL? FIRSTPT FIRSTINSIDE? PREVPT PREVINSIDE?))
)
(PUTPROPS PLOTOBJECTS2 COPYRIGHT ("Xerox Corporation" 1988))
```

FUNCTION INDEX

CLIP-INSIDEP	4	COPYFILLEDPOLYGON	1	EXTENTOFFILLEDPOLYGON ...	2	PLOTFILLEDPOLYGON	2
CLIP-INTERSECT	5	CREATEFILLEDPOLYGON	1	FINISH-CLIP-POLYGON	4	PUTFILLEDPOLYGON	3
CLIP-POLYGON-VERTEX	4	DISTANCETOFFILLEDPOLYGON .	1	GETFILLEDPOLYGON	2		
CLIPPED.FILLPOLYGON	3	DRAWFILLEDPOLYGON	1	HIGHLIGHTFILLEDPOLYGON ..	2		
CLIPPED.POLYGON	3	ERASEFILLEDPOLYGON	2	MOVEFILLEDPOLYGON	2		

RECORD INDEX

CLIPEDGEINFO	5	FILLEDPOLYGONDATA	3
--------------------	---	-------------------------	---

VARIABLE INDEX

OBJECT2OPSTABLE	3
-----------------------	---
