*File created:* 6-May-87 14:09:27 **{QV}<PEDERSEN>LISP>TWODGRAPHICS.;3**

*changes to:* (VARS TWODGRAPHICSCOMS)
(FNS CLIPPED.BITBLT CREATEVIEWPORT SETSTREAMSUBREGION SETWORLDREGION TWODGRAPHICS.BITBLT
CLIPPED.BLTSHADE COMPUTETRANSFORM COMPUTEWORLDREGION STREAMREGIONTOWORLDREGION STREAMTOWORLD
TWODGRAPHICS.CLOSEFN TWODGRAPHICS.DRAWTO TWODGRAPHICS.DRAWLINE TWODGRAPHICS.DRAWTOPT
TWODGRAPHICS.DSPFILL TWODGRAPHICS.DSPRESET TWODGRAPHICS.MOVETO TWODGRAPHICS.MOVETOPT
TWODGRAPHICS.PLOTAT TWODGRAPHICS.RELDRAWTO TWODGRAPHICS.RELDRAWTOPT TWODGRAPHICS.RELMOVETO
TWODGRAPHICS.RESHAPEFN WORLDREGIONTOSTREAMREGION WORLDTOSTREAM CLIPCODE CLIPPED.DESTREGION
CLIPPED.DRAWBETWEEN CLIPPED.DRAWLINE CLIPPED.DRAWTO CLIPPED.PLOTAT CLIPPED.PRIN1
CLIPPED.RELDRAWTO CLIPPED.SOURCEREGION REPLACE.REGION)

*previous date:* 6-May-87 12:19:11 {QV}<PEDERSEN>LISP>TWODGRAPHICS.;2

*Read Table:* INTERLISP

*Package:* INTERLISP

*Format:* XCCS

;;


```
(RPAQQ TWODGRAPHICSCOMS
       (;; World to window transforms
        (FNS CREATEVIEWPORT COMPUTETRANSFORM COMPUTEWORLDREGION SETSTREAMSUBREGION SETWORLDREGION
             STREAMREGIONTOWORLDREGION STREAMTOWORLD TWODGRAPHICS.BITBLT TWODGRAPHICS.CLOSEFN
             TWODGRAPHICS.DRAWBETWEEN TWODGRAPHICS.DRAWLINE TWODGRAPHICS.DRAWTO TWODGRAPHICS.DRAWTOPT
             TWODGRAPHICS.DSPFILL TWODGRAPHICS.DSPRESET TWODGRAPHICS.INIT TWODGRAPHICS.MOVETO
             TWODGRAPHICS.MOVETOPT TWODGRAPHICS.PLOTAT TWODGRAPHICS.RELDRAWTO TWODGRAPHICS.RELDRAWTOPT
             TWODGRAPHICS.RELMOVETO TWODGRAPHICS.RELMOVETOPT TWODGRAPHICS.RESHAPEFN WORLDREGIONTOSTREAMREGION
             WORLDTOSTREAM)
        (MACROS STREAMTOWORLDX STREAMTOWORLDXLENGTH STREAMTOWORLDY STREAMTOWORLDYLENGTH WORLDTOSTREAMX
                WORLDTOSTREAMXLENGTH WORLDTOSTREAMY WORLDTOSTREAMYLENGTH)
        (RECORDS VIEWPORT)
        ;; Primitive clipping FNS
        (FNS CLIPCODE CLIPPED.BITBLT CLIPPED.BLTSHADE CLIPPED.DRAWBETWEEN CLIPPED.DRAWLINE CLIPPED.DRAWTO
             CLIPPED.PLOTAT CLIPPED.PRIN1 CLIPPED.RELDRAWTO)
        (MACROS SWAPARGS)
        ;; For unboxed floating point games
        (DECLARE%: DONTCOPY DOEVAL@COMPILE (FILES UNBOXEDOPS))
        (DECLARE%: DONTEVAL@LOAD DONTCOPY DOEVAL@COMPILE (LOCALVARS . T))))

;; World to window transforms

(DEFINEQ

(CREATEVIEWPORT
  [LAMBDA (STREAM STREAMSUBREGION SOURCE)                    ; Edited 6-May-87 10:37 by jop
    ;; Create a viewport. If source is a region , then treat it as a region in world coorinates and set up the transformation to stream coorindates. If
    ;; source is a Viewport, inherit the transformation and set up the world coordinates. If Source is NIL then supply a default WORLDREGION. In
    ;; either case if STREAM is a STREAM then inter the viewport in the VIEWPORTS property of the window.
    (PROG ((STREAMCLIPPINGREGION (DSPCLIPPINGREGION NIL STREAM))
           VIEWPORT)
          [COND
             ((NULL STREAMSUBREGION)
              (SETQ STREAMSUBREGION (with REGION STREAMCLIPPINGREGION (CREATEREGION LEFT BOTTOM WIDTH HEIGHT]
          [COND
             ((NULL SOURCE)
              (SETQ SOURCE (CREATEREGION 0.0 0.0 1.0 1.0]
          (COND
             ((NOT (SUBREGIONP STREAMCLIPPINGREGION STREAMSUBREGION))
              (CL:ERROR "~s not a subregion of ~s" STREAMSUBREGION STREAMCLIPPINGREGION)))
          [SETQ VIEWPORT (COND
                            ((type? REGION SOURCE)
                             (COMPUTETRANSFORM (create VIEWPORT
                                                       PARENTSTREAM _ STREAM
                                                       STREAMSUBREGION _ STREAMSUBREGION
                                                       WORLDREGION _ SOURCE)))
                            ((type? VIEWPORT SOURCE)
                             (COMPUTEWORLDREGION (create VIEWPORT
                                                         PARENTSTREAM _ STREAM
                                                         STREAMSUBREGION _ STREAMSUBREGION using SOURCE)))
                            (T (ERROR "Not region or viewort: ~S" SOURCE]
          (COND
             ((WINDOWP STREAM)
              (TWODGRAPHICS.INIT STREAM)
              (WINDOWADDPROP STREAM 'VIEWPORTS VIEWPORT)))
          (RETURN VIEWPORT])
```

(COMPUTETRANSFORM

```
[LAMBDA (VIEWPORT)                                          ; Edited 5-May-87 16:32 by jop
  ;; Computes the world to window transformation given a viewport's windowsubregion and worldregion
  (PROG ((STREAMSUBREGION (fetch (VIEWPORT STREAMSUBREGION) of VIEWPORT))
         (WORLDREGION (fetch (VIEWPORT WORLDREGION) of VIEWPORT)))
                                                            ; SUB1 since we are dealing width an integer grid
        (replace WORLDTOSTREAMMX of VIEWPORT with (FQUOTIENT (SUB1 (fetch WIDTH of STREAMSUBREGION))
                                                             (fetch WIDTH of WORLDREGION)))
        [replace WORLDTOSTREAMAX of VIEWPORT with (FDIFFERENCE (fetch LEFT of STREAMSUBREGION)
                                                               (FTIMES (fetch WORLDTOSTREAMMX of VIEWPORT)
                                                                       (fetch LEFT of WORLDREGION]
                                                            ; Ditto
        (replace WORLDTOSTREAMMY of VIEWPORT with (FQUOTIENT (SUB1 (fetch HEIGHT of STREAMSUBREGION))
                                                             (fetch HEIGHT of WORLDREGION)))
        [replace WORLDTOSTREAMAY of VIEWPORT with (FDIFFERENCE (fetch BOTTOM of STREAMSUBREGION)
                                                               (FTIMES (fetch WORLDTOSTREAMMY of VIEWPORT)
                                                                       (fetch BOTTOM of WORLDREGION]
        (replace STREAMTOWORLDMX of VIEWPORT with (FQUOTIENT 1.0 (fetch WORLDTOSTREAMMX of VIEWPORT)))
        [replace STREAMTOWORLDAX of VIEWPORT with (UFMINUS (FQUOTIENT (fetch WORLDTOSTREAMAX of VIEWPORT)
                                                                      (fetch WORLDTOSTREAMMX of VIEWPORT]
        (replace STREAMTOWORLDMY of VIEWPORT with (FQUOTIENT 1.0 (fetch WORLDTOSTREAMMY of VIEWPORT)))
        [replace STREAMTOWORLDAY of VIEWPORT with (UFMINUS (FQUOTIENT (fetch WORLDTOSTREAMAY of VIEWPORT)
                                                                      (fetch WORLDTOSTREAMMY of VIEWPORT]
        (RETURN VIEWPORT])
```

## (COMPUTEWORLDREGION

```
[LAMBDA (VIEWPORT)                                          ; Edited 5-May-87 16:32 by jop
  ;; Given a Viewport's World to Stream transformation computes the corresponding World region
  (PROG ((STREAMSUBREGION (fetch (VIEWPORT STREAMSUBREGION) of VIEWPORT))
         (MX (fetch (VIEWPORT WORLDTOSTREAMMX) of VIEWPORT))
         (AX (fetch (VIEWPORT WORLDTOSTREAMAX) of VIEWPORT))
         (MY (fetch (VIEWPORT WORLDTOSTREAMMY) of VIEWPORT))
         (AY (fetch (VIEWPORT WORLDTOSTREAMAY) of VIEWPORT))
         WORREGION)
        [SETQ WORREGION (with REGION STREAMSUBREGION (CREATEREGION (FQUOTIENT (FDIFFERENCE LEFT AX)
                                                                              MX)
                                                                  (FQUOTIENT (FDIFFERENCE BOTTOM AY)
                                                                             MY)
                                                                  (FQUOTIENT WIDTH MX)
                                                                  (FQUOTIENT HEIGHT MY]
        (replace (VIEWPORT WORLDREGION) of VIEWPORT with WORREGION)
        (RETURN VIEWPORT])
```

## (SETSTREAMSUBREGION

```
[LAMBDA (REGION VIEWPORT)                                   ; Edited 6-May-87 10:38 by jop
  ;; Set the STREAMSUBREGION of a VIEWPORT
  (if (NOT (type? VIEWPORT VIEWPORT))
      then (CL:ERROR "Not a VIEWPORT: ~s" VIEWPORT))
  (if (NOT (SUBREGIONP (WINDOWPROP (fetch PARENTSTREAM of VIEWPORT)
                                   'WINCLIPPINGREGION)
                       REGION))
      then (CL:ERROR "Not a subregion of stream: ~s" REGION))
  (replace (VIEWPORT STREAMSUBREGION) of VIEWPORT with REGION)
  (COMPUTETRANSFORM VIEWPORT])
```

## (SETWORLDREGION

```
[LAMBDA (REGION VIEWPORT)                                   ; Edited 6-May-87 10:38 by jop
  ;; Set the WORLDREGION of a VIEWPORT
  (if (NOT (type? VIEWPORT VIEWPORT))
      then (CL:ERROR "Not a viewport: ~s" VIEWPORT))
  (replace (VIEWPORT WORLDREGION) of VIEWPORT with REGION)
  (COMPUTETRANSFORM VIEWPORT])
```

## (STREAMREGIONTOWORLDREGION

```
[LAMBDA (REGION VIEWPORT)                                   ; Edited 5-May-87 16:33 by jop
  (CREATEREGION (STREAMTOWORLDX (fetch (REGION LEFT) of REGION)
                                VIEWPORT)
                (STREAMTOWORLDY (fetch (REGION BOTTOM) of REGION)
                                VIEWPORT)
                (STREAMTOWORLDXLENGTH (fetch (REGION WIDTH) of REGION)
                                      VIEWPORT)
                (STREAMTOWORLDYLENGTH (fetch (REGION HEIGHT) of REGION)
                                      VIEWPORT])
```

## (STREAMTOWORLD

```
[LAMBDA (PT VIEWPORT OLDPT)                                 ; Edited 5-May-87 16:33 by jop
  ;; smashes OLDPT if provided
  (COND
```

```
    (OLDPT (create POSITION
                  XCOORD _ (STREAMTOWORLDX (fetch (POSITION XCOORD) of PT)
                                           VIEWPORT)
                  YCOORD _ (STREAMTOWORLDY (fetch (POSITION YCOORD) of PT)
                                           VIEWPORT)
           smashing OLDPT))
    (T (create POSITION
           XCOORD _ (STREAMTOWORLDX (fetch (POSITION XCOORD) of PT)
                                    VIEWPORT)
           YCOORD _ (STREAMTOWORLDY (fetch (POSITION YCOORD) of PT)
                                    VIEWPORT])
```

## (TWODGRAPHICS.BITBLT

```
[LAMBDA (SOURCE SOURCELEFT SOURCEBOTTOM DESTINATIONVIEWPORT DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT
           SOURCETYPE OPERATION TEXTURE CLIPPINGREGION)    ; Edited 6-May-87 10:38 by jop
```

;; Destination MUST be a VIEWPORT. Source can be either a VIEWPORT or some other form of BITMAP (in which case no transformations are performed) or NIL

```
  (if (NULL DESTINATIONVIEWPORT)
      then (SETQ DESTINATIONVIEWPORT TWODGRAPHICS.CURRENTVIEWPORT))
  (if (NOT (type? VIEWPORT DESTINATIONVIEWPORT))
      then (CL:ERROR "Destination not a viewport: ~s" DESTINATIONVIEWPORT))
  (LET* ((STREAM (fetch (VIEWPORT PARENTSTREAM) of DESTINATIONVIEWPORT))
         (STREAMSUBREGION (fetch (VIEWPORT STREAMSUBREGION) of DESTINATIONVIEWPORT))
         (STREAMLEFT (if (NULL DESTINATIONLEFT)
                         then (fetch (REGION LEFT) of STREAMSUBREGION)
                       else (WORLDTOSTREAMX DESTINATIONLEFT DESTINATIONVIEWPORT)))
         (STREAMBOTTOM (if (NULL DESTINATIONBOTTOM)
                           then (fetch (REGION BOTTOM) of STREAMSUBREGION)
                         else (WORLDTOSTREAMY DESTINATIONBOTTOM DESTINATIONVIEWPORT)))
         [STREAMCLIPPINGREGION (if (NULL CLIPPINGREGION)
                                   then STREAMSUBREGION
                                 else (INTERSECTREGIONS STREAMSUBREGION (WORLDREGIONTOSTREAMREGION
                                                                           CLIPPINGREGION
                                                                           DESTINATIONVIEWPORT]
         (SOURCEBITMAP SOURCE)
         (SOURCEBITMAPLEFT SOURCELEFT)
         (SOURCEBITMAPBOTTOM SOURCEBOTTOM)
         (SOURCEWIDTH WIDTH)
         (SOURCEHEIGHT HEIGHT))
        [if (type? VIEWPORT SOURCE)
            then (SETQ SOURCEBITMAP (fetch (VIEWPORT PARENTSTREAM) of SOURCE))
                 (LET ((SOURCESUBREGION (fetch (VIEWPORT STREAMSUBREGION) of SOURCE)))
                      (SETQ SOURCEBITMAPLEFT (if (NULL SOURCELEFT)
                                                 then (fetch (REGION LEFT) of SOURCESUBREGION)
                                               else (WORLDTOSTREAMX SOURCELEFT SOURCE)))
                      (SETQ SOURCEBITMAPBOTTOM (if (NULL SOURCEBOTTOM)
                                                   then (fetch (REGION BOTTOM) of SOURCESUBREGION)
                                                 else (WORLDTOSTREAMY SOURCEBOTTOM SOURCE)))
                      (SETQ SOURCEWIDTH (if (NULL WIDTH)
                                            then (fetch (REGION WIDTH) of SOURCESUBREGION)
                                          else (WORLDTOSTREAMXLENGTH WIDTH SOURCE)))
                      (SETQ SOURCEHEIGHT (if (NULL HEIGHT)
                                             then (fetch (REGION HEIGHT) of SOURCESUBREGION)
                                           else (WORLDTOSTREAMYLENGTH HEIGHT SOURCE)))
                      (SETQ STREAMCLIPPINGREGION (INTERSECTREGIONS STREAMCLIPPINGREGION SOURCESUBREGION]
        [if (EQ SOURCETYPE 'TEXTURE)
            then (SETQ SOURCEWIDTH (if (NULL WIDTH)
                                       then (fetch (REGION WIDTH) of STREAMSUBREGION)
                                     else (WORLDTOSTREAMXLENGTH WIDTH DESTINATIONVIEWPORT)))
                 (SETQ SOURCEHEIGHT (if (NULL HEIGHT)
                                        then (fetch (REGION HEIGHT) of STREAMSUBREGION)
                                      else (WORLDTOSTREAMYLENGTH HEIGHT DESTINATIONVIEWPORT]
        (CLIPPED.BITBLT STREAMCLIPPINGREGION SOURCEBITMAP SOURCEBITMAPLEFT SOURCEBITMAPBOTTOM STREAM
               STREAMLEFT STREAMBOTTOM SOURCEWIDTH SOURCEHEIGHT SOURCETYPE OPERATION TEXTURE])
```

## (TWODGRAPHICS.CLOSEFN

```
[LAMBDA (W)                                                ; Edited 5-May-87 16:34 by jop
```

;; Break circularities

```
  (WINDOWPROP W 'TWODPROPS? NIL)
  (WINDOWPROP W 'VIEWPORTS NIL)
  (WINDOWPROP W 'WINCLIPPINGREGION NIL)
  (WINDOWDELPROP W 'CLOSEFN (FUNCTION TWODGRAPHICS.CLOSEFN))
  (WINDOWDELPROP W 'RESHAPEFN (FUNCTION TWODGRAPHICS.RESHAPEFN])
```

## (TWODGRAPHICS.DRAWBETWEEN

```
[LAMBDA (PT1 PT2 WIDTH OPERATION VIEWPORT COLOR DASHING)        (* jop%: " 4-Dec-85 15:38")
  (TWODGRAPHICS.DRAWLINE (fetch XCOORD of PT1)
         (fetch YCOORD of PT1)
         (fetch XCOORD of PT2)
         (fetch YCOORD of PT2)
         WIDTH OPERATION VIEWPORT COLOR DASHING])
```

## (TWODGRAPHICS.DRAWLINE

```
[LAMBDA (X1 Y1 X2 Y2 WIDTH OPERATION VIEWPORT COLOR DASHING)   ; Edited 5-May-87 17:12 by jop
  (LET ((STREAM (fetch (VIEWPORT PARENTSTREAM) of VIEWPORT))
        (CLIPPINGREGION (fetch (VIEWPORT PARENTSTREAM) of VIEWPORT))
        (STREAMX1 (WORLDTOSTREAMX X1 VIEWPORT))
        (STREAMY1 (WORLDTOSTREAMY Y1 VIEWPORT))
        (STREAMX2 (WORLDTOSTREAMX X2 VIEWPORT))
        (STREAMY2 (WORLDTOSTREAMY Y2 VIEWPORT)))
       (CLIPPED.DRAWLINE CLIPPINGREGION STREAMX1 STREAMY1 STREAMX2 STREAMY2 WIDTH OPERATION STREAM COLOR
              DASHING])
```

## (TWODGRAPHICS.DRAWTO

```
[LAMBDA (X Y WIDTH OPERATION VIEWPORT COLOR DASHING)           ; Edited 5-May-87 16:34 by jop
  (LET ((STREAM (fetch (VIEWPORT PARENTSTREAM) of VIEWPORT))
        (CLIPPINGREGION (fetch (VIEWPORT STREAMSUBREGION) of VIEWPORT))
        (STREAMX (WORLDTOSTREAMX X VIEWPORT))
        (STREAMY (WORLDTOSTREAMY Y VIEWPORT)))
       (CLIPPED.DRAWTO CLIPPINGREGION STREAMX STREAMY WIDTH OPERATION STREAM COLOR DASHING])
```

## (TWODGRAPHICS.DRAWTOPT

```
[LAMBDA (PT WIDTH OPERATION VIEWPORT COLOR DASHING)            ; Edited 5-May-87 17:13 by jop
  (TWODGRAPHICS.DRAWTO (fetch XCOORD of PT)
         (fetch YCOORD of PT)
         WIDTH OPERATION VIEWPORT COLOR DASHING])
```

## (TWODGRAPHICS.DSPFILL

```
[LAMBDA (REGION TEXTURE OPERATION VIEWPORT)                    ; Edited 5-May-87 17:14 by jop
  (LET ((STREAM (fetch (VIEWPORT PARENTSTREAM) of VIEWPORT)))
       (TWODGRAPHICS.BITBLT NIL NIL NIL VIEWPORT NIL NIL NIL NIL 'TEXTURE (OR OPERATION (DSPOPERATION NIL
                                                                                          STREAM))
             (OR TEXTURE (DSPTEXTURE NIL STREAM))
             REGION])
```

## (TWODGRAPHICS.DSPRESET

```
[LAMBDA (VIEWPORT)                                             ; Edited 5-May-87 17:14 by jop
  ;; RESET a VIEWPORT
  (LET ((STREAM (fetch (VIEWPORT PARENTSTREAM) of VIEWPORT))
        (STREAMSUBREGION (fetch (VIEWPORT STREAMSUBREGION) of VIEWPORT)))
       (DSPXPOSITION (DSPLEFTMARGIN NIL STREAM)
             STREAM)
       [DSPYPOSITION (DIFFERENCE (fetch (REGION TOP) of STREAMSUBREGION)
                           (FONTPROP STREAM 'ASCENT]
       (TWODGRAPHICS.DSPFILL NIL NIL 'REPLACE VIEWPORT])
```

## (TWODGRAPHICS.INIT

```
[LAMBDA (W)                                                    (* jop%: "23-Feb-86 19:55")
  (COND
     ((NULL (WINDOWPROP W 'TWODPROPS?))
      (WINDOWPROP W 'TWODPROPS? T)
      (WINDOWPROP W 'VIEWPORTS NIL)
      (WINDOWPROP W 'WINCLIPPINGREGION (DSPCLIPPINGREGION NIL W))
      (WINDOWADDPROP W 'CLOSEFN (FUNCTION TWODGRAPHICS.CLOSEFN))
      (WINDOWADDPROP W 'RESHAPEFN (FUNCTION TWODGRAPHICS.RESHAPEFN)
             T])
```

## (TWODGRAPHICS.MOVETO

```
[LAMBDA (X Y VIEWPORT)                                         ; Edited 5-May-87 17:16 by jop
  (MOVETO (WORLDTOSTREAMX X VIEWPORT)
          (WORLDTOSTREAMY Y VIEWPORT)
          (fetch PARENTSTREAM of VIEWPORT])
```

## (TWODGRAPHICS.MOVETOPT

```
[LAMBDA (PT VIEWPORT)                                          ; Edited 5-May-87 17:16 by jop
  (TWODGRAPHICS.MOVETO (fetch XCOORD of PT)
         (fetch YCOORD of PT)
         VIEWPORT])
```

## (TWODGRAPHICS.PLOTAT

```
[LAMBDA (PT GLYPH VIEWPORT OPERATION)                          ; Edited 5-May-87 17:16 by jop
  (PROG ((STREAM (fetch (VIEWPORT PARENTSTREAM) of VIEWPORT))
         (STREAMSUBREGION (fetch (VIEWPORT STREAMSUBREGION) of VIEWPORT)))
        (CLIPPED.PLOTAT STREAMSUBREGION (WORLDTOSTREAM PT VIEWPORT)
               GLYPH STREAM OPERATION])
```

## (TWODGRAPHICS.RELDRAWTO

```
[LAMBDA (DELTAX DELTAY WIDTH OPERATION VIEWPORT COLOR DASHING) ; Edited 5-May-87 17:16 by jop
```

```
    (LET ((STREAM (fetch (VIEWPORT PARENTSTREAM) of VIEWPORT))
          (CLIPPINGREGION (fetch (VIEWPORT PARENTSTREAM) of VIEWPORT))
          (STREAMDX (WORLDTOSTREAMXLENGTH DELTAX VIEWPORT))
          (STREAMDY (WORLDTOSTREAMYLENGTH DELTAY VIEWPORT)))
         (CLIPPED.DRAWTO CLIPPINGREGION STREAMDX STREAMDY WIDTH OPERATION VIEWPORT COLOR DASHING])
```

## (TWODGRAPHICS.RELDRAWTOPT

```
  [LAMBDA (DPT WIDTH OPERATION VIEWPORT COLOR DASHING)          ; Edited 5-May-87 17:16 by jop
    (TWODGRAPHICS.RELDRAWTO (fetch XCOORD of DPT)
           (fetch YCOORD DPT)
           WIDTH OPERATION VIEWPORT COLOR DASHING])
```

## (TWODGRAPHICS.RELMOVETO

```
  [LAMBDA (DX DY VIEWPORT)                                      ; Edited 5-May-87 17:17 by jop
    (LET ((STREAM (fetch (VIEWPORT PARENTSTREAM) of VIEWPORT)))
         (RELMOVETO (WORLDTOSTREAMXLENGTH DX VIEWPORT)
                (WORLDTOSTREAMYLENGTH DY VIEWPORT)
                STREAM])
```

## (TWODGRAPHICS.RELMOVETOPT

```
  [LAMBDA (DPT VIEWPORT)                                        (* jop%: "23-Feb-86 19:29")

          (* *)

    (TWODGRAPHICS.RELMOVETO (fetch XCOORD of DPT)
           (fetch YCOORD of DPT)
           VIEWPORT])
```

## (TWODGRAPHICS.RESHAPEFN

```
  [LAMBDA (WINDOW)                                              ; Edited 5-May-87 17:17 by jop
    ;; updates all viewports associated with window
    (PROG ((OLDCLIPPINGREGION (WINDOWPROP WINDOW 'WINCLIPPINGREGION))
           (NEWCLIPPINGREGION (DSPCLIPPINGREGION NIL WINDOW))
           WIDTHRATIO HEIGHTRATIO)
          (SETQ WIDTHRATIO (FQUOTIENT (fetch (REGION WIDTH) of NEWCLIPPINGREGION)
                                      (fetch (REGION WIDTH) of OLDCLIPPINGREGION)))
          (SETQ HEIGHTRATIO (FQUOTIENT (fetch (REGION HEIGHT) of NEWCLIPPINGREGION)
                                       (fetch (REGION HEIGHT) of OLDCLIPPINGREGION)))
          (bind REGION for V in (WINDOWPROP WINDOW 'VIEWPORTS)
             do (SETQ REGION (fetch (VIEWPORT STREAMSUBREGION) of V))
                [replace (VIEWPORT STREAMSUBREGION) of V with (with REGION REGION (CREATEREGION (FIXR (FTIMES
                                                                                                       WIDTHRATIO
                                                                                                          LEFT))
                                                                                                  (FIXR (FTIMES HEIGHTRATIO
                                                                                                                BOTTOM))
                                                                                                  (FIXR (FTIMES WIDTHRATIO
                                                                                                                WIDTH))
                                                                                                  (FIXR (FTIMES HEIGHTRATIO
                                                                                                                HEIGHT]
                      (COMPUTETRANSFORM V))
          (WINDOWPROP WINDOW 'WINCLIPPINGREGION NEWCLIPPINGREGION)
          (RETURN WINDOW])
```

## (WORLDREGIONTOSTREAMREGION

```
  [LAMBDA (REGION VIEWPORT)                                     ; Edited 5-May-87 17:17 by jop
    (CREATEREGION (WORLDTOSTREAMX (fetch (REGION LEFT) of REGION)
                         VIEWPORT)
           (WORLDTOSTREAMY (fetch (REGION BOTTOM) of REGION)
                  VIEWPORT)
           (WORLDTOSTREAMXLENGTH (fetch (REGION WIDTH) of REGION)
                  VIEWPORT)
           (WORLDTOSTREAMYLENGTH (fetch (REGION HEIGHT) of REGION)
                  VIEWPORT])
```

## (WORLDTOSTREAM

```
  [LAMBDA (PT VIEWPORT OLDPT)                                   ; Edited 5-May-87 17:17 by jop
    (COND
       (OLDPT (create POSITION
                      XCOORD _ (WORLDTOSTREAMX (fetch (POSITION XCOORD) of PT)
                                               VIEWPORT)
                      YCOORD _ (WORLDTOSTREAMY (fetch (POSITION YCOORD) of PT)
                                               VIEWPORT)
              smashing OLDPT))
       (T (create POSITION
                  XCOORD _ (WORLDTOSTREAMX (fetch (POSITION XCOORD) of PT)
                                           VIEWPORT)
                  YCOORD _ (WORLDTOSTREAMY (fetch (POSITION YCOORD) of PT)
                                           VIEWPORT])

)
```

```
(DECLARE%: EVAL@COMPILE

(PUTPROPS STREAMTOWORLDX MACRO [OPENLAMBDA (X VIEWPORT)
                                   (PROG (RESULT)
                                         (DECLARE (TYPE FLOATP RESULT))
                                         (RETURN (SETQ RESULT (FPLUS (fetch STREAMTOWORLDAX of VIEWPORT)
                                                                     (FTIMES (fetch STREAMTOWORLDMX
                                                                                of VIEWPORT)
                                                                             (FLOAT X])

(PUTPROPS STREAMTOWORLDXLENGTH MACRO [OPENLAMBDA (DX VIEWPORT)
                                         (PROG (RESULT)
                                               (DECLARE (TYPE FLOATP RESULT))
                                               (RETURN (SETQ RESULT (FTIMES (fetch STREAMTOWORLDMX
                                                                               of VIEWPORT)
                                                                            (FLOAT DX])

(PUTPROPS STREAMTOWORLDY MACRO [OPENLAMBDA (Y VIEWPORT)
                                   (PROG (RESULT)
                                         (DECLARE (TYPE FLOATP RESULT))
                                         (RETURN (SETQ RESULT (FPLUS (fetch STREAMTOWORLDAY of VIEWPORT)
                                                                     (FTIMES (fetch STREAMTOWORLDMY
                                                                                of VIEWPORT)
                                                                             (FLOAT Y])

(PUTPROPS STREAMTOWORLDYLENGTH MACRO [OPENLAMBDA (DY VIEWPORT)
                                         (PROG (RESULT)
                                               (DECLARE (TYPE FLOATP RESULT))
                                               (RETURN (SETQ RESULT (FTIMES (fetch STREAMTOWORLDMY
                                                                               of VIEWPORT)
                                                                            (FLOAT DY])

(PUTPROPS WORLDTOSTREAMX MACRO [OPENLAMBDA (X VIEWPORT)
                                   (UFIX (FPLUS (fetch WORLDTOSTREAMAX of VIEWPORT)
                                                (FTIMES (fetch WORLDTOSTREAMMX of VIEWPORT)
                                                        (FLOAT X])

(PUTPROPS WORLDTOSTREAMXLENGTH MACRO [OPENLAMBDA (DX VIEWPORT)
                                         (UFIX (FTIMES (fetch WORLDTOSTREAMMX of VIEWPORT)
                                                       (FLOAT DX])

(PUTPROPS WORLDTOSTREAMY MACRO [OPENLAMBDA (Y VIEWPORT)
                                   (UFIX (FPLUS (fetch WORLDTOSTREAMAY of VIEWPORT)
                                                (FTIMES (fetch WORLDTOSTREAMMY of VIEWPORT)
                                                        (FLOAT Y])

(PUTPROPS WORLDTOSTREAMYLENGTH MACRO [OPENLAMBDA (DY VIEWPORT)
                                         (UFIX (FTIMES (fetch WORLDTOSTREAMMY of VIEWPORT)
                                                       (FLOAT DY])
)

(DECLARE%: EVAL@COMPILE

(DATATYPE VIEWPORT (PARENTSTREAM STREAMSUBREGION WORLDREGION (WORLDTOSTREAMMX FLOATP)
                           (WORLDTOSTREAMAX FLOATP)
                           (WORLDTOSTREAMMY FLOATP)
                           (WORLDTOSTREAMAY FLOATP)
                           (STREAMTOWORLDMX FLOATP)
                           (STREAMTOWORLDAX FLOATP)
                           (STREAMTOWORLDMY FLOATP)
                           (STREAMTOWORLDAY FLOATP)))
)

(/DECLAREDATATYPE 'VIEWPORT '(POINTER POINTER POINTER FLOATP FLOATP FLOATP FLOATP FLOATP FLOATP FLOATP FLOATP)
       ;; ---field descriptor list elided by lister---
       '22)

;; Primitive clipping FNS

(DEFINEQ

(CLIPCODE
  [LAMBDA (X Y LEFT RIGHT TOP BOTTOM)                    ; Edited  5-May-87 17:18 by jop
    ;; Cohen-Sutherland clip codes.  Assumes integer args
    ;; RIGHT and TOP are one past the region.
    ;; RIGHT and TOP are one past the region.
    (LET [(ABOVEBIT (COND
                        ((GREATERP Y TOP)
                         8)
                        (T 0)))
          (BELOWBIT (COND
                        ((GREATERP BOTTOM Y)
                         4)
```

```
                    (T 0)))
          (RIGHTBIT (COND
                       ((GREATERP X RIGHT)
                        2)
                       (T 0)))
          (LEFTBIT (COND
                      ((GREATERP LEFT X)
                       1)
                      (T 0]
         (LOGOR ABOVEBIT BELOWBIT RIGHTBIT LEFTBIT])
```

## (CLIPPED.BITBLT

```
[LAMBDA (CLIPPINGREGION SOURCEBITMAP SOURCELEFT SOURCEBOTTOM DESTINATION DESTINATIONLEFT DESTINATIONBOTTOM
               WIDTH HEIGHT SOURCETYPE OPERATION TEXTURE)      ; Edited 6-May-87 12:40 by jop
                                                               ; Process defaults

   ;; It assumed that source must be a window or a bitmap -- and hence has scale 1

   (if (EQ SOURCETYPE 'TEXTURE)
       then (CLIPPED.BLTSHADE CLIPPINGREGION TEXTURE DESTINATION DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT
                   OPERATION)
     else (COND
             ((NULL SOURCELEFT)
              (SETQ SOURCELEFT 0)))
          (COND
             ((NULL SOURCEBOTTOM)
              (SETQ SOURCEBOTTOM 0)))
          (COND
             ((NULL DESTINATIONLEFT)
              (SETQ DESTINATIONLEFT 0)))
          (COND
             ((NULL DESTINATIONBOTTOM)
              (SETQ DESTINATIONBOTTOM 0)))
          [COND
             ((NULL WIDTH)
              (SETQ WIDTH (COND
                            ((WINDOWP SOURCEBITMAP)
                             (WINDOWPROP SOURCEBITMAP 'WIDTH))
                            (T (BITMAPWIDTH SOURCEBITMAP]
          [COND
             ((NULL HEIGHT)
              (SETQ HEIGHT (COND
                             ((WINDOWP SOURCEBITMAP)
                              (WINDOWPROP SOURCEBITMAP 'HEIGHT))
                             (T (BITMAPHEIGHT SOURCEBITMAP]
          (LET* ((CLIP-LEFT (fetch (REGION LEFT) of CLIPPINGREGION))
                 (CLIP-BOTTOM (fetch (REGION BOTTOM) of CLIPPINGREGION))
                 (CLIP-WIDTH (fetch (REGION WIDTH) of CLIPPINGREGION))
                 (CLIP-HEIGHT (fetch (REGION HEIGHT) of CLIPPINGREGION))
                 (SCALE (DSPSCALE NIL DESTINATION))
                 (NEW-LEFT DESTINATIONLEFT)
                 (NEW-BOTTOM DESTINATIONBOTTOM)
                 (NEW-WIDTH (TIMES SCALE WIDTH))
                 (NEW-HEIGHT (TIMES SCALE HEIGHT))
                 CLIPPED?)
                (COND
                   ((GREATERP CLIP-LEFT NEW-LEFT)
                    (SETQ CLIPPED? T)
                    (SETQ NEW-WIDTH (DIFFERENCE NEW-WIDTH (DIFFERENCE CLIP-LEFT NEW-LEFT)))
                    (SETQ NEW-LEFT CLIP-LEFT)))
                (COND
                   ((GREATERP CLIP-BOTTOM NEW-BOTTOM)
                    (SETQ CLIPPED? T)
                    (SETQ NEW-HEIGHT (DIFFERENCE NEW-HEIGHT (DIFFERENCE CLIP-BOTTOM NEW-BOTTOM)))
                    (SETQ NEW-BOTTOM CLIP-BOTTOM)))
                [COND
                   ((GREATERP (PLUS NEW-LEFT NEW-WIDTH)
                              (PLUS CLIP-LEFT CLIP-WIDTH))
                    (SETQ CLIPPED? T)
                    (SETQ NEW-WIDTH (DIFFERENCE (PLUS CLIP-LEFT CLIP-WIDTH)
                                                NEW-LEFT]
                [COND
                   ((GREATERP (PLUS NEW-BOTTOM NEW-HEIGHT)
                              (PLUS CLIP-BOTTOM CLIP-HEIGHT))
                    (SETQ CLIPPED? T)
                    (SETQ NEW-HEIGHT (DIFFERENCE (PLUS CLIP-BOTTOM CLIP-HEIGHT)
                                                 NEW-BOTTOM]
                (COND
                   ((NULL CLIPPED?)                            ; No clipping
                    (BITBLT SOURCEBITMAP SOURCELEFT SOURCEBOTTOM DESTINATION DESTINATIONLEFT DESTINATIONBOTTOM
                           WIDTH HEIGHT SOURCETYPE OPERATION))
                   ((OR (GEQ 0 NEW-WIDTH)
                        (GEQ 0 NEW-HEIGHT))                    ; Gross clipping
                    NIL)
                   (T                                          ; Adjusted bitblt
                      (BITBLT SOURCEBITMAP (PLUS SOURCELEFT (IQUOTIENT (DIFFERENCE NEW-LEFT DESTINATIONLEFT)
                                                                       SCALE))
```

```
                              (PLUS SOURCEBOTTOM (IQUOTIENT (DIFFERENCE NEW-BOTTOM DESTINATIONBOTTOM)
                                                            SCALE))
                              DESTINATION NEW-LEFT NEW-BOTTOM (IQUOTIENT NEW-WIDTH SCALE)
                              (IQUOTIENT NEW-HEIGHT SCALE)
                              SOURCETYPE OPERATION])
```

## (CLIPPED.BLTSHADE

```
  [LAMBDA (CLIPPINGREGION TEXTURE DESTINATION DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT OPERATION)
                                                                          ; Edited  6-May-87 12:14 by jop
                                                                          ; Process defaults
    (COND
       ((NULL DESTINATIONLEFT)
        (SETQ DESTINATIONLEFT 0)))
    (COND
       ((NULL DESTINATIONBOTTOM)
        (SETQ DESTINATIONBOTTOM 0)))
    [COND
       ((NULL WIDTH)
        (SETQ WIDTH (COND
                       ((WINDOWP DESTINATION)
                        (WINDOWPROP DESTINATION 'WIDTH))
                       (T (BITMAPWIDTH DESTINATION]
    [COND
       ((NULL HEIGHT)
        (SETQ HEIGHT (COND
                        ((WINDOWP DESTINATION)
                         (WINDOWPROP DESTINATION 'HEIGHT))
                        (T (BITMAPHEIGHT DESTINATION]
    (LET ((CLIP-LEFT (fetch (REGION LEFT) of CLIPPINGREGION))
          (CLIP-BOTTOM (fetch (REGION BOTTOM) of CLIPPINGREGION))
          (CLIP-WIDTH (fetch (REGION WIDTH) of CLIPPINGREGION))
          (CLIP-HEIGHT (fetch (REGION HEIGHT) of CLIPPINGREGION))
          (NEW-LEFT DESTINATIONLEFT)
          (NEW-BOTTOM DESTINATIONBOTTOM)
          (NEW-WIDTH WIDTH)
          (NEW-HEIGHT HEIGHT))
         (COND
            ((GREATERP CLIP-LEFT NEW-LEFT)
             (SETQ NEW-WIDTH (DIFFERENCE NEW-WIDTH (DIFFERENCE CLIP-LEFT NEW-LEFT)))
             (SETQ NEW-LEFT CLIP-LEFT)))
         (COND
            ((GREATERP CLIP-BOTTOM NEW-BOTTOM)
             (SETQ NEW-HEIGHT (DIFFERENCE NEW-HEIGHT (DIFFERENCE CLIP-BOTTOM NEW-BOTTOM)))
             (SETQ NEW-BOTTOM CLIP-BOTTOM)))
         [COND
            ((GREATERP (PLUS NEW-LEFT NEW-WIDTH)
                       (PLUS CLIP-LEFT CLIP-WIDTH))
             (SETQ NEW-WIDTH (DIFFERENCE (PLUS CLIP-LEFT CLIP-WIDTH)
                                         NEW-LEFT]
         [COND
            ((GREATERP (PLUS NEW-BOTTOM NEW-HEIGHT)
                       (PLUS CLIP-BOTTOM CLIP-HEIGHT))
             (SETQ NEW-HEIGHT (DIFFERENCE (PLUS CLIP-BOTTOM CLIP-HEIGHT)
                                          NEW-BOTTOM]
         (COND
            ((OR (GEQ 0 NEW-WIDTH)
                 (GEQ 0 NEW-HEIGHT))                                      ; Gross clipping
             NIL)
            (T                                                            ; Adjusted bitblt
               (BLTSHADE TEXTURE DESTINATION NEW-LEFT NEW-BOTTOM NEW-WIDTH NEW-HEIGHT OPERATION])
```

## (CLIPPED.DRAWBETWEEN

```
  [LAMBDA (CLIPPINGREGION FIRSTPOSITION SECONDPOSITION WIDTH OPERATION STREAM COLOR DASHING)
                                                                          ; Edited  5-May-87 17:19 by jop
    (CLIPPED.DRAWLINE CLIPPINGREGION (fetch (POSITION XCOORD) of FIRSTPOSITION)
           (fetch (POSITION YCOORD) of FIRSTPOSITION)
           (fetch (POSITION XCOORD) of SECONDPOSITION)
           (fetch (POSITION YCOORD) of SECONDPOSITION)
           WIDTH OPERATION STREAM COLOR DASHING])
```

## (CLIPPED.DRAWLINE

```
  [LAMBDA (CLIPPINGREGION X1 Y1 X2 Y2 WIDTH OPERATION STREAM COLOR DASHING)
                                                                          ; Edited  5-May-87 17:19 by jop
    ;; Clip against CLIPPINGREGION and draw in STREAM.  Implements Cohen-Sutherland clipping.  From Foley and Van Dam, pg.  146
    (PROG ((CLIPLEFT (fetch LEFT of CLIPPINGREGION))
           (CLIPRIGHT (fetch RIGHT of CLIPPINGREGION))
           (CLIPTOP (fetch TOP of CLIPPINGREGION))
           (CLIPBOTTOM (fetch BOTTOM of CLIPPINGREGION))
           (OLDX2 X2)
           (OLDY2 Y2)
           OUTCODE1 OUTCODE2 ACCEPT DONE)
          [repeatuntil DONE do (SETQ OUTCODE1 (CLIPCODE X1 Y1 CLIPLEFT CLIPRIGHT CLIPTOP CLIPBOTTOM))
                               (SETQ OUTCODE2 (CLIPCODE X2 Y2 CLIPLEFT CLIPRIGHT CLIPTOP CLIPBOTTOM))
```

```
                        (COND
                           [(EQ 0 (LOGAND OUTCODE1 OUTCODE2))
                                                        ; Possible accept
                            (COND
                               ((SETQ ACCEPT (EQ 0 (LOGOR OUTCODE1 OUTCODE2)))
                                                        ; accept
                                (SETQ DONE T))
                               (T                       ; Find intersections
                                  [COND
                                     ((EQ 0 OUTCODE1)   ; Swap points so (X1 . Y1) is guaranteed to be outside
                                      (LET (TEMP)
                                           (SWAPARGS TEMP X1 X2)
                                           (SWAPARGS TEMP Y1 Y2)
                                           (SWAPARGS TEMP OUTCODE1 OUTCODE2]
                                  (COND
                                     ((NEQ 0 (LOGAND OUTCODE1 8))
                                                        ; divide line at top
                                      [SETQ X1 (PLUS X1 (QUOTIENT (TIMES (DIFFERENCE X2 X1)
                                                                         (DIFFERENCE CLIPTOP Y1))
                                                                  (DIFFERENCE Y2 Y1]
                                      (SETQ Y1 CLIPTOP))
                                     ((NEQ 0 (LOGAND OUTCODE1 4))
                                                        ; divide line at bottom
                                      [SETQ X1 (PLUS X1 (QUOTIENT (TIMES (DIFFERENCE X2 X1)
                                                                         (DIFFERENCE CLIPBOTTOM Y1))
                                                                  (DIFFERENCE Y2 Y1]
                                      (SETQ Y1 CLIPBOTTOM))
                                     ((NEQ 0 (LOGAND OUTCODE1 2))
                                                        ; divide line at right
                                      [SETQ Y1 (PLUS Y1 (QUOTIENT (TIMES (DIFFERENCE Y2 Y1)
                                                                         (DIFFERENCE CLIPRIGHT X1))
                                                                  (DIFFERENCE X2 X1]
                                      (SETQ X1 CLIPRIGHT))
                                     (T                 ; divide line at left
                                        [SETQ Y1 (PLUS Y1 (QUOTIENT (TIMES (DIFFERENCE Y2 Y1)
                                                                           (DIFFERENCE CLIPLEFT X1))
                                                                    (DIFFERENCE X2 X1]
                                        (SETQ X1 CLIPLEFT]
                           (T                           ; Reject
                              (SETQ DONE T]             ; actually draw a line if one accepted
                  (COND
                     (ACCEPT (DRAWLINE X1 Y1 X2 Y2 WIDTH OPERATION STREAM COLOR DASHING)))
                                                        ; Correctly Update posistion in stream
                  (MOVETO OLDX2 OLDY2 STREAM])
```

## (CLIPPED.DRAWTO

```
  [LAMBDA (CLIPPINGREGION X Y WIDTH OPERATION STREAM COLOR DASHING)
                                                        ; Edited 5-May-87 17:19 by jop
    (CLIPPED.DRAWLINE CLIPPINGREGION (DSPXPOSITION NIL STREAM)
           (DSPYPOSITION NIL STREAM)
           X Y WIDTH OPERATION STREAM COLOR DASHING])
```

## (CLIPPED.PLOTAT

```
  [LAMBDA (CLIPPINGREGION PT GLYPH STREAM OPERATION)    ; Edited 5-May-87 17:19 by jop
    (PROG ((WIDTHGLYPH (BITMAPWIDTH GLYPH))
           (HEIGHTGLYPH (BITMAPHEIGHT GLYPH))
           NEWX NEWY)
          [SETQ NEWX (DIFFERENCE (fetch XCOORD of PT)
                                 (TIMES (DSPSCALE NIL STREAM)
                                        (IQUOTIENT WIDTHGLYPH 2]
          [SETQ NEWY (DIFFERENCE (fetch YCOORD of PT)
                                 (TIMES (DSPSCALE NIL STREAM)
                                        (IQUOTIENT HEIGHTGLYPH 2]
          (CLIPPED.BITBLT CLIPPINGREGION GLYPH 0 0 STREAM NEWX NEWY WIDTHGLYPH HEIGHTGLYPH 'INPUT OPERATION])
```

## (CLIPPED.PRIN1

```
  [LAMBDA (CLIPPINGREGION EXPR STREAM)                  ; Edited 5-May-87 17:19 by jop
    (PROG ((STRINGREGION (STRINGREGION EXPR STREAM))
           IREGION)
          (COND
             ((SUBREGIONP CLIPPINGREGION STRINGREGION)  ; No clipping
              (PRIN1 EXPR STREAM))
             (T (SETQ IREGION (INTERSECTREGIONS STRINGREGION CLIPPINGREGION))
                (COND
                   ((AND IREGION (IEQP (fetch (REGION HEIGHT) of IREGION)
                                       (fetch (REGION HEIGHT) of STRINGREGION)))
                                                        ; Some chars visible
                    (bind (MINX _ (fetch (REGION LEFT) of CLIPPINGREGION))
                          (MAXX _ (fetch (REGION RIGHT) of CLIPPINGREGION))
                          (X _ (DSPXPOSITION NIL STREAM))
                          (Y _ (DSPYPOSITION NIL STREAM))
                          NEXTX CHARWIDTH for I from 1 to (NCHARS EXPR)
                       do (SETQ CHARWIDTH (CHARWIDTH (NTHCHARCODE EXPR I)
                                                     STREAM))
```

```
                        (SETQ NEXTX (IPLUS X CHARWIDTH))
                        (COND
                           ((NOT (OR (ILESSP X MINX)
                                     (IGREATERP NEXTX MAXX)))
                            (PRIN1 (NTHCHAR EXPR I)
                                   STREAM))
                           (T (MOVETO NEXTX Y STREAM)))
                        (SETQ X NEXTX])
```

## (CLIPPED.RELDRAWTO

```
  [LAMBDA (CLIPPINGREGION DX DY WIDTH OPERATION STREAM COLOR DASHING)
                                                                    ; Edited  5-May-87 17:19 by jop
    (PROG ((X (DSPXPOSITION NIL STREAM))
           (Y (DSPYPOSITION NIL STREAM)))
          (CLIPPED.DRAWLINE CLIPPINGREGION X Y (PLUS X DX)
                 (PLUS Y DY)
                 WIDTH OPERATION STREAM COLOR DASHING])

)

(DECLARE%: EVAL@COMPILE

(PUTPROPS SWAPARGS MACRO ((TEMP FIRST SECOND)
                          (SETQ TEMP FIRST)
                          (SETQ FIRST SECOND)
                          (SETQ SECOND TEMP)))
)

;; For unboxed floating point games

(DECLARE%: DONTCOPY DOEVAL@COMPILE

(FILESLOAD UNBOXEDOPS)
)

(DECLARE%: DONTEVAL@LOAD DONTCOPY DOEVAL@COMPILE

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(LOCALVARS . T)
)
)

(PUTPROPS TWODGRAPHICS COPYRIGHT ("Xerox Corporation" 1984 1985 1986 1987))
```

## FUNCTION INDEX



## MACRO INDEX



## RECORD INDEX