File created: 11-Dec-87 13:48:32 **{ERINYES}<LISPUSERS>LYRIC>AIREGIONS.;1**

changes to: (VARS AIREGIONSCOMS)

previous date: 9-Mar-87 16:37:52 {DANTE}<LISPUSERS>LYRIC>AIREGIONS.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;
;; Copyright (c) 1985, 1986, 1987 by XEROX Corporation. All rights reserved.

```
(RPAQQ AIREGIONSCOMS
       [(FILES FILLREGION)
        (FNS ADD.IREGION ALL.IREGIONS CREATEIR REMOVE.IREGION DOSELECTED.IREGION EDIT.MASK IREGIONP IREGIONPROP
             SHOW.ALL.IREGIONS INTERSECTING.IREGIONS? INVERT.IREGION WHICH.IREGIONS SURROUNDIR)
        (FNS IN.CURSOR.REGION \IR.CLIP.REGION \IR.SHOW.REGION \VALID.POSITION.LIST \SAME.IREGIONS.LIST
             \WITH.INTERSECTION \IREGION.ON.WINDOWP)
        (RECORDS IREGION)
        (PROP ARGNAMES IREGIONPROP)
        (VARS (DEFAULT.IREGION.SHADE 65535))
        (GLOBALVARS DEFAULT.IREGION.SHADE)
        (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA)
                                                                              (NLAML)
                                                                              (LAMA IREGIONPROP])

(FILESLOAD FILLREGION)

(DEFINEQ
```

(ADD.IREGION

```
  [LAMBDA (WINDOW IREGION)                                  (* GWexler " 7-Feb-86 10:22")
    (if (AND (NOT (\IREGION.ON.WINDOWP IREGION WINDOW))
             (IREGIONP IREGION)
             (WINDOWP WINDOW))
        then (WINDOWPROP WINDOW 'BUTTONEVENTFN 'IN.CURSOR.REGION)
             (WINDOWADDPROP WINDOW 'IREGIONSLIST IREGION T)
             IREGION])
```

(ALL.IREGIONS

```
  [LAMBDA (WINDOW)                                          (* GWexler " 7-Feb-86 10:23")

          (* * This function returns all the IREGIONs of WINDOW.)

    (if (WINDOWP WINDOW)
        then (WINDOWPROP WINDOW 'IREGIONSLIST)
      else NIL])
```

(CREATEIR

```
  [LAMBDA (WINDOW SHADE BUTTONEVENTFN HELPSTRING REGION POSLIST) (* GWexler " 7-Feb-86 10:30")

          (* * This function is the top-level interface to creating IREGIONS and associating them with a given window.)

    (PROG (TEMP-REGION TEMP-BITMAP POS MASK POSITIONS)

          (* If WINDOW is not a window, then return. If the user wants just an IREGION record, he/she should use
          (create IREGION BUTTONEVENTFN _ ??? USERDATA _ ??? REGION _ ??? MASK _ ??? SHADE _ ??? HELPSTRING _
          ???))

          (OR (WINDOWP WINDOW)
              (RETURN NIL))
          (WINDOWPROP WINDOW 'BUTTONEVENTFN 'IN.CURSOR.REGION)
          (SETQ TEMP-REGION (if (REGIONP REGION)
                                then

          (* Here, the user passes us a REGION relative to WINDOW and we translate it to screen coordinates.
          (We need this in screen coordinates since that is what GETREGION returns.))

                                     (LIST [IPLUS (CAR REGION)
                                                  (CAR (WINDOWPROP WINDOW 'REGION]
                                           [IPLUS (CADR REGION)
                                                  (CADR (WINDOWPROP WINDOW 'REGION]
                                           (CADDR REGION)
                                           (CADDDR REGION))
                                else                                (* If REGION was not a region, then we prompt the user for one.)
                                     (PROMPTPRINT "Please closely surround the irregular region " "without cutting
                                            off the region's borders")
                                     (TOTOPW WINDOW)
                                     (GETREGION)))

          (* Here, we clip TEMP-REGION so that it fits inside the window so that we don't waste any space with too large bitmaps.
          If the region did not fall inside the window at all, then \IR.CLIP.REGION returns NIL and we return from CREATEIR.)
```

```
(SETQ TEMP-REGION (OR (\IR.CLIP.REGION TEMP-REGION WINDOW)
                      (RETURN NIL)))

(* We create two bitmaps the size of TEMP-REGION that will contain the mask for the IREGION.
We need one extra just for temporary storage in CREATEIR.)

(SETQ TEMP-BITMAP (BITMAPCREATE (CADDR TEMP-REGION)
                        (CADDDR TEMP-REGION)
                        1))                          (* We save the window image in TEMP-REGION in
                                                     TEMP-BITMAP.)
(TOTOPW WINDOW)
(BITBLT WINDOW (CAR TEMP-REGION)
       (CADR TEMP-REGION)
       TEMP-BITMAP 0 0)                              (* MASK is just a copy of TEMP-BITMAP)
(SETQ MASK (BITMAPCOPY TEMP-BITMAP))

(* The user can pass either NIL or a list of positions to fill from in the argument POSLIST.
\VALID.POSITION.LIST returns either NIL %, ERROR, or a list of positions.
ERROR means to abort the call to CREATEIR since POSLIST was not in any valid form.)

(SETQ POSITIONS (\VALID.POSITION.LIST POSLIST))
(OR (NOT (EQ 'ERROR POSITIONS))
    (RETURN NIL))

(* If the user did not specify POSLIST %, then we box out TEMP-REGION to let him/her know where they are working.)

(OR POSITIONS (\IR.SHOW.REGION WINDOW TEMP-REGION))
[if (NOT POSITIONS)
    then

(* This part prompts for positions to area fill from until the user buttons outside of TEMP-REGION.)

        (PROMPTPRINT "Point your mouse to the inside of the region and " "left-button the irregular
              shapes in that region. " "To stop selection(s), left-button anywhere away " "from the
              squared-off region.")
        (TOTOPW WINDOW)
        (until (NOT (INSIDEP TEMP-REGION (CAR (SETQ POS (GETPOSITION WINDOW)))
                             (CDR POS)))
           do (FILL.REGION MASK (CONS (IDIFFERENCE (CAR POS)
                                                   (CAR TEMP-REGION))
                                      (IDIFFERENCE (CDR POS)
                                                   (CADR TEMP-REGION)))
                           65535)                    (* Show in WINDOW, what was just filled.)
              (BITBLT MASK 0 0 WINDOW (CAR TEMP-REGION)
                     (CADR TEMP-REGION))
              (TOTOPW WINDOW))
  else

(* Here, the user has given us a list of positions, and we just go through each one and fill.)

      (for P in POSITIONS do (if (INSIDEP TEMP-REGION P)
                                 then (FILL.REGION MASK (CONS (IDIFFERENCE (CAR P)
                                                                           (CAR TEMP-REGION))
                                                              (IDIFFERENCE (CDR P)
                                                                           (CADR TEMP-REGION)))
                                                   65535]
(PROMPTPRINT)                                        (* Unbox the region if need be.)
(OR POSITIONS (\IR.SHOW.REGION WINDOW TEMP-REGION))  (* Set up MASK to have only black where the area fill filled.)
(BITBLT TEMP-BITMAP 0 0 MASK 0 0 NIL NIL 'INPUT 'ERASE)
                                                     (* Restore the window to its original image.)
(BITBLT TEMP-BITMAP 0 0 WINDOW (CAR TEMP-REGION)
       (CADR TEMP-REGION))                           (* Add the appropriate IREGION to the windows
                                                     IREGIONSLIST...)
(RETURN (ADD.IREGION WINDOW
                (create IREGION
                        BUTTONEVENTFN _ BUTTONEVENTFN
                        USERDATA _ NIL
                        REGION _ TEMP-REGION
                        MASK _ MASK
                        SHADE _ (if (NOT (OR (NUMBERP SHADE)
                                             (BITMAPP SHADE)))
                                    then DEFAULT.IREGION.SHADE
                                  else SHADE)
                        HELPSTRING _ HELPSTRING])
```

(REMOVE.IREGION

```
  [LAMBDA (WINDOW IREGION)                           (* GWexler " 7-Feb-86 10:24")
                                                     (* Given a window and an IREGION, remove it from that
                                                     windows IREGIONSLIST)

    (if (\IREGION.ON.WINDOWP IREGION WINDOW)
        then (WINDOWDELPROP WINDOW 'IREGIONSLIST IREGION)
             IREGION])
```

(DOSELECTED.IREGION

```
  [LAMBDA (WINDOW IREGION BUTTON)                                    (* GWexler " 7-Feb-86 10:24")
    (if (AND (\IREGION.ON.WINDOWP IREGION WINDOW)
             (IREGIONPROP IREGION 'BUTTONEVENTFN))
        then (APPLY* (IREGIONPROP IREGION 'BUTTONEVENTFN)
                     WINDOW IREGION BUTTON])
```

(EDIT.MASK

```
  [LAMBDA (IREGION)                                                  (* GWexler " 7-Feb-86 10:24")
                                                                     (* A way of simply editing the MASK of an IREGION)
    (if (IREGIONP IREGION)
        then (EDITBM (fetch (IREGION MASK) of IREGION])
```

(IREGIONP

```
  [LAMBDA (IREGION)                                                  (* GWexler " 7-Feb-86 10:24")

          (* Tests to see if IREGION is a valid IREGION. Returns IREGION if true NIL otherwise.)

    (if (type? IREGION IREGION)
        then IREGION
      else NIL])
```

(IREGIONPROP

```
  [LAMBDA X                                                          (* GWexler " 7-Feb-86 10:24")

          (* Modeled after WINDOWPROP. This is a convenient way of associating information with an IREGION.
          If the property of the IREGION is not one of the fields, then it is stored on the USERDATA field in property list format.)

    (PROG (IREGION PROP NEWVALUE)
          (SETQ IREGION (ARG X 1))
          (SETQ PROP (ARG X 2))                                      (* Abort if we do not have an IREGION.)
          (OR (IREGIONP IREGION)
              (RETURN NIL))
          (if (EQ X 3)
              then                                                   (* In this case, we have a value to set, so do replaces or
                                                                     LISTPUTs.)
                   (SETQ NEWVALUE (ARG X 3))
                   [if [NOT (MEMB PROP (RECORDFIELDNAMES 'IREGION]
                       then                                          (* In this case, we store the property on the USERDATA field.)
                            (if (NOT (fetch (IREGION USERDATA) of IREGION))
                                then                                 (* If USERDATA is NIL, we need to set it to
                                                                     (PROPNAME VALUE))
                                     (replace (IREGION USERDATA) of IREGION with (LIST PROP NEWVALUE))
                              else                                   (* USERDATA should already be in proplist format so just do a
                                                                     LISTPUT)
                                   (LISTPUT (fetch (IREGION USERDATA) of IREGION)
                                            PROP NEWVALUE))
                            [if (NOT NEWVALUE)
                                then (replace (IREGION USERDATA) of IREGION
                                        with (for X on (fetch (IREGION USERDATA) of IREGION)
                                                by (CDDR X) when (NOT (EQ (CAR X)
                                                                          PROP))
                                                join (LIST (CAR X)
                                                           (CADR X]
                            (RETURN NEWVALUE)
                     else                                            (* Here, PROP is one of the IREGIONS fields, so we just
                                                                     replace it.)
                          (RETURN (SELECTQ PROP
                                      (BUTTONEVENTFN
                                          (replace (IREGION BUTTONEVENTFN) of IREGION with NEWVALUE))
                                      (USERDATA (replace (IREGION USERDATA) of IREGION with NEWVALUE))
                                      (REGION (replace (IREGION REGION) of IREGION with NEWVALUE))
                                      (MASK (replace (IREGION MASK) of IREGION with NEWVALUE))
                                      (SHADE (replace (IREGION SHADE) of IREGION with NEWVALUE))
                                      (HELPSTRING (replace (IREGION HELPSTRING) of IREGION with NEWVALUE))
                                      (ERROR "Not a valid IREGION prop: " PROP]
                else                                                 (* No NEWVALUE was specified so we just want to fetch the
                                                                     right info.)
                     (if [NOT (MEMB PROP (RECORDFIELDNAMES 'IREGION]
                         then

          (* PROP is not one of IREGIONs fields, so we need to get it off of USERDATA)

                              (RETURN (LISTGET (fetch (IREGION USERDATA) of IREGION)
                                               PROP))
                       else                                          (* Simply fetch the right field of IREGION.)
                            (RETURN (SELECTQ PROP
                                        (BUTTONEVENTFN
                                            (fetch (IREGION BUTTONEVENTFN) of IREGION))
                                        (USERDATA (fetch (IREGION USERDATA) of IREGION))
                                        (REGION (fetch (IREGION REGION) of IREGION))
                                        (MASK (fetch (IREGION MASK) of IREGION))
                                        (SHADE (fetch (IREGION SHADE) of IREGION))
                                        (HELPSTRING (fetch (IREGION HELPSTRING) of IREGION))
                                        (ERROR "Not a valid IREGION prop: " PROP])
```

(SHOW.ALL.IREGIONS

```
[LAMBDA (WINDOW SHADE DELAY)                                    (* GWexler " 7-Feb-86 10:24")
```

(* This function cycles through each IREGION on WINDOW and flashes it in black and waits DELAY milliseconds. There is code here to do the right thing when the user aborts.)

```
(LET ((IRS (ALL.IREGIONS WINDOW)))
     (for X in IRS do (RESETLST
                          [RESETSAVE NIL (LIST 'IREGIONPROP X 'SHADE (IREGIONPROP X 'SHADE]
                          (IREGIONPROP X 'SHADE (OR SHADE 65535))
                          (INVERT.IREGION WINDOW X)
                          (RESETSAVE NIL (LIST 'INVERT.IREGION WINDOW X))
                          (BLOCK (OR (NUMBERP DELAY)
                                     500))])
```

(INTERSECTING.IREGIONS?

```
[LAMBDA (WINDOW FLG)                                            (* GWexler " 7-Feb-86 10:24")
```

(* This sets up WINDOW to specify whether or not its overlapping active regions all get called when buttoned in the overlapping area %. NIL means no overlapping and T means yes.)

```
(WINDOWPROP WINDOW 'IR.INTERSECTIONFLG FLG])
```

(INVERT.IREGION

```
[LAMBDA (WINDOW IREGION)                                        (* GWexler " 7-Feb-86 10:25")
```

(* Simply inverts IREGION associated with WINDOW %. Doesn't do anything if IREGION is not on the window's IREGIONLIST.)

```
(if (\IREGION.ON.WINDOWP IREGION WINDOW)
    then (TOTOPW WINDOW)
         (BITBLT (fetch (IREGION MASK) of IREGION)
                 0 0 WINDOW (CAR (fetch (IREGION REGION) of IREGION))
                 (CADR (fetch (IREGION REGION) of IREGION))
                 NIL NIL 'MERGE 'INVERT (fetch (IREGION SHADE) of IREGION])
```

(WHICH.IREGIONS

```
[LAMBDA (WINDOW POSORX Y)                                       (* GWexler " 7-Feb-86 10:25")
                                                                (* Returns all the IREGIONS of WINDOW in a list.
                                                                NIL if there were none.)
```

(* * IF CURSOR IN REGION'S WINDOW AND REGION'S MASK, IDENTIFY REGION)

(* * X AND Y are optional arguments. Same with WINDOW. The user could just say (WHICHIR) and it would notice where it is, or Programmatically, the user could pass a window and an X and Y to get which Iregion.)

```
(LET* [(W (OR (WINDOWP WINDOW)
              (WHICHW)))
       (POSITION (if (POSITIONP POSORX)
                     then POSORX
                   else (if (OR (NOT (NUMBERP POSORX))
                                (NOT (NUMBERP Y)))
                            then (CONS (LASTMOUSEX W)
                                       (LASTMOUSEY W))
                          else (CONS POSORX Y]
       (for SOME-IREGION in (WINDOWPROP W 'IREGIONSLIST)
          when [NOT (ZEROP (BITMAPBIT (fetch (IREGION MASK) of SOME-IREGION)
                                      (IDIFFERENCE (CAR POSITION)
                                                   (CAR (fetch (IREGION REGION) of SOME-IREGION)))
                                      (IDIFFERENCE (CDR POSITION)
                                                   (CADR (fetch (IREGION REGION) of SOME-IREGION]
          collect SOME-IREGION])
```

(SURROUNDIR

```
[LAMBDA (WINDOW SHADE BUTTONEVENTFN HELPSTRING POSLIST INSIDE.POS)
                                                                (* GWexler " 7-Feb-86 10:46")
```

(* * This function is the top-level interface to creating IREGIONS inwhich the innereds specified are ignored and the entire surround regions becomes the AIREGION which is associated with the given window)

```
(PROG (TEMPW TEMP-REGION TEMP-BITMAP POS MASK POSITIONS POINTS.LST TEMPPOS)
```

(* If WINDOW is not a window, then return. If the user wants just an IREGION record, he/she should use (create IREGION BUTTONEVENTFN _ ??? USERDATA _ ??? REGION _ ??? MASK _ ??? SHADE _ ??? HELPSTRING _ ???))

```
(OR (WINDOWP WINDOW)
    (RETURN NIL))
(WINDOWPROP WINDOW 'BUTTONEVENTFN 'IN.CURSOR.REGION)
(PROMPTPRINT "Please button the area around the particular region.  To end, hold the SHIFT key when
```

```
            hitting the last point with the mouse.")
        (FLASHWINDOW PROMPTWINDOW 2)
        (TOTOPW WINDOW)
        [OR (LISTP POSLIST)
            (while [AND (NOT (KEYDOWNP 'LSHIFT))
                        (NOT (KEYDOWNP 'RSHIFT]
               do [SETQ POSLIST (APPEND POSLIST (LIST (SETQ TEMPPOS (GETPOSITION WINDOW]
                  (BITBLT (CAR CROSSHAIRS)
                         0 0 WINDOW (DIFFERENCE (CAR TEMPPOS)
                                                (CADR CROSSHAIRS))
                         (DIFFERENCE (CDR TEMPPOS)
                                     (CDDR CROSSHAIRS))
                         NIL NIL 'INPUT 'INVERT]
        [OR [NLSETQ (SETQ TEMP-REGION (WINDOWPROP (SETQ TEMPW
                                                  (CREATEW (LIST 0 0 (PLUS (APPLY 'MAX
                                                                                  (for I in POSLIST
                                                                                     collect (CAR I)))
                                                                           50)
                                                                   (PLUS (APPLY 'MAX (for I in POSLIST
                                                                                        collect (CDR I)))
                                                                         50))
                                                           NIL NIL T))
                                                'REGION]
            (SETQ TEMP-REGION (WINDOWPROP (SETQ TEMPW (CREATEW '(0 0 10 10)
                                                               NIL NIL T))
                                          'REGION]
        (PROMPTPRINT "Please button once INSIDE the region")
        (FLASHWINDOW PROMPTWINDOW 2)
        [OR (POSITIONP INSIDE.POS)
            (AND (SETQ INSIDE.POS (GETPOSITION WINDOW))
                 (for POSITION in POSLIST do (BITBLT (CAR CROSSHAIRS)
                                                     0 0 WINDOW (DIFFERENCE (CAR POSITION)
                                                                            (CADR CROSSHAIRS))
                                                     (DIFFERENCE (CDR POSITION)
                                                                 (CDDR CROSSHAIRS))
                                                     NIL NIL 'INPUT 'INVERT]

        (* * MAYBE TEMPORARILY XOR IT ON...)

        (DRAWCURVE POSLIST T 1 NIL TEMPW)

        (* * Note that having the window open when drawing the curve is kludgy.
        In the KOTO release, DIG will be implemented in the system so rather than having the TEMPW, use an IMAGESTREAM via
        OPENIMAGESTREAM and do a IMDRAWCURVE and IMBITBLT.
        In this mannor, all drawing is hidden from the user and it makes the package a lot cleaner.)

        (SETQ TEMP-BITMAP (BITMAPCREATE (CADDR TEMP-REGION)
                                        (CADDDR TEMP-REGION)
                                        1))                    (* We save the window image in TEMP-REGION in
                                                               TEMP-BITMAP.)
        (TOTOPW TEMPW)
        (BITBLT TEMPW (CAR TEMP-REGION)
               (CADR TEMP-REGION)
               TEMP-BITMAP 0 0)
        (CLOSEW TEMPW)                                         (* MASK is just a copy of TEMP-BITMAP)
        (SETQ MASK (BITMAPCOPY TEMP-BITMAP))

        (* The user can pass either NIL or a list of positions to fill from in the argument POSLIST.
        \VALID.POSITION.LIST returns either NIL %, ERROR, or a list of positions.
        ERROR means to abort the call to CREATEIR since POSLIST was not in any valid form.)

        (FILL.REGION MASK (CONS (IDIFFERENCE (CAR INSIDE.POS)
                                             (CAR TEMP-REGION))
                                (IDIFFERENCE (CDR INSIDE.POS)
                                             (CADR TEMP-REGION)))
               65535)                                          (* (BITBLT MASK 0 0 WINDOW (CAR TEMP-REGION)
                                                               (CADR TEMP-REGION)))

        (* * Removing the border)

        (BITBLT TEMP-BITMAP 0 0 MASK 0 0 NIL NIL 'INPUT 'ERASE)
                                                               (* (BITBLT TEMP-BITMAP 0 0 WINDOW
                                                               (CAR TEMP-REGION) (CADR TEMP-REGION)))
        (RETURN (ADD.IREGION WINDOW
                        (create IREGION
                                BUTTONEVENTFN _ BUTTONEVENTFN
                                USERDATA _ NIL
                                REGION _ TEMP-REGION
                                MASK _ MASK
                                SHADE _ (if (NOT (OR (NUMBERP SHADE)
                                                     (BITMAPP SHADE)))
                                            then DEFAULT.IREGION.SHADE
                                          else SHADE)
                                HELPSTRING _ HELPSTRING])

)
```

```
(DEFINEQ
```

(IN.CURSOR.REGION

```
  [LAMBDA (WINDOW)                                          (* GWexler " 7-Feb-86 10:27")
                                                            (* This is WINDOWs BUTTONEVENTFN that gets called for
                                                            windows with IREGIONS.)

          (* * IF CURSOR IN REGION'S WINDOW AND REGION'S MASK, SHADE REGION)

    (PROG [OLD-REGIONS TEMPX TEMPY BUTTON ALLREADY.PROMPT (TIMEOUT.WAIT 1500)
           (TIMEOUT (SETUPTIMER 1500 NIL 'TICKS 'MILLISECONDS))
           (INTERSECTIONFLG (WINDOWPROP WINDOW 'IR.INTERSECTIONFLG]
          (TOTOPW WINDOW)

          (* OLD-REGIONS starts out to be all the IREGIONS currently pointed to.
          \WITH.INTERSECTION is used to determine whether this is a list of all the IREGIONS or just one.)

          (SETQ OLD-REGIONS (\WITH.INTERSECTION (WHICH.IREGIONS WINDOW
                                    INTERSECTIONFLG))
          (SETQ TEMPX (LASTMOUSEX WINDOW))
          (SETQ TEMPY (LASTMOUSEY WINDOW))                  (* Start out and invert all the IREGIONS being pointed to.)
          (for IR in OLD-REGIONS do (INVERT.IREGION WINDOW IR))   (* BUTTON is set so that the user can use this value.)
          (SETQ BUTTON (SELECTQ LASTMOUSEBUTTONS
                               (4 'LEFT)
                               (2 'RIGHT)
                               (1 'MIDDLE)
                               (NULL)))                     (* Now we loop until the mouse comes back up.)
          [while (NOT (MOUSESTATE UP))
             do (PROG NIL
                      (if (OR (if (EQ TEMPX (SETQ TEMPX (LASTMOUSEX WINDOW)))
                                  then (EQ TEMPY (SETQ TEMPY (LASTMOUSEY WINDOW)))
                                else (SETQ TEMPY (LASTMOUSEY WINDOW))
                                     NIL)
                              (\SAME.IREGIONS.LIST OLD-REGIONS (\WITH.INTERSECTION (WHICH.IREGIONS WINDOW
                                                                                          (CONS TEMPX TEMPY))
                                                                                   INTERSECTIONFLG)))
                          then (COND
                                  ((AND (NOT ALLREADY.PROMPT)
                                        (TIMEREXPIRED? TIMEOUT 'TICKS)
                                        (NOT (NULL OLD-REGIONS)))
                                   (PROMPTPRINT)
                                   (for IR in OLD-REGIONS do (printout PROMPTWINDOW (OR (fetch (IREGION HELPSTRING)
                                                                                           of IR)
                                                                                        "Will select this
                                                                                        IRREGULAR region when you
                                                                                        release the button.")
                                                                                     T))
                                   (SETQ ALLREADY.PROMPT T)))
                        else

          (* * WE have moved AND we are at a new Iregion)

                             (if ALLREADY.PROMPT
                                 then (PROMPTPRINT))
                             (for IR in OLD-REGIONS do (INVERT.IREGION WINDOW IR))
                             (SETQ OLD-REGIONS (\WITH.INTERSECTION (WHICH.IREGIONS WINDOW (CONS TEMPX TEMPY))
                                                                   INTERSECTIONFLG))
                             (for IR in OLD-REGIONS do (INVERT.IREGION WINDOW IR))
                             (SETQ TIMEOUT (SETUPTIMER 1500 TIMEOUT 'TICKS 'MILLISECONDS))
                             (SETQ ALLREADY.PROMPT NIL]
          (if ALLREADY.PROMPT
              then (PROMPTPRINT))
          (for IR in OLD-REGIONS do (INVERT.IREGION WINDOW IR))
          (for IR in OLD-REGIONS do (DOSELECTED.IREGION WINDOW IR BUTTON])
```

(\IR.CLIP.REGION

```
  [LAMBDA (REG WINDOW)                                      (* GWexler " 7-Feb-86 10:27")

          (* This function takes a regions and a window and returns a new region that fits inside the window.
          If region starts out completely outside of window, then NIL is returned.)

    (PROG [LEFT RIGHT TOP BOTTOM (W.REG (WINDOWPROP WINDOW 'REGION]
          [SETQ W.REG (LIST (CAR W.REG)
                            (CADR W.REG)
                            [IDIFFERENCE (CADDR W.REG)
                                   (ITIMES 2 (WINDOWPROP WINDOW 'BORDER]
                            (IDIFFERENCE (CADDDR W.REG)
                                   (ITIMES 2 (WINDOWPROP WINDOW 'BORDER]
          (if (OR (GREATERP (fetch (REGION LEFT) of REG)
                            (fetch (REGION PRIGHT) of W.REG))
                  (GREATERP (fetch (REGION BOTTOM) of REG)
                            (fetch (REGION PTOP) of W.REG))
                  (ILESSP (fetch (REGION PRIGHT) of REG)
                          (fetch (REGION LEFT) of W.REG))
                  (ILESSP (fetch (REGION PTOP) of REG)
                          (fetch (REGION BOTTOM) of W.REG)))
```

```
            then (printout T "None of the region was inside the window." T)
                 (RETURN NIL))
         (SETQ LEFT (IMAX (fetch (REGION LEFT) of REG)
                          (fetch (REGION LEFT) of W.REG)))
         (SETQ RIGHT (IMIN (fetch (REGION PRIGHT) of REG)
                           (fetch (REGION PRIGHT) of W.REG)))
         (SETQ TOP (IMIN (fetch (REGION PTOP) of REG)
                         (fetch (REGION PTOP) of W.REG)))
         (SETQ BOTTOM (IMAX (fetch (REGION BOTTOM) of REG)
                            (fetch (REGION BOTTOM) of W.REG)))
         (RETURN (LIST (IDIFFERENCE LEFT (fetch (REGION LEFT) of W.REG))
                       (IDIFFERENCE BOTTOM (fetch (REGION BOTTOM) of W.REG))
                       (IDIFFERENCE RIGHT LEFT)
                       (IDIFFERENCE TOP BOTTOM])
```

(\IR.SHOW.REGION

```
[LAMBDA (WINDOW REGION)                                    (* GWexler " 7-Feb-86 10:27")
                                                           (* This function draws a box specified by REGION on WINDOW
    using invert mode.)
    (LET ((REG (COPY REGION))
          (W (IPLUS (CADDR REGION)
                    (CAR REGION)
                    1))
          (H (IPLUS (CADDDR REGION)
                    (CADR REGION)
                    1)))
         (RPLACA REG (SUB1 (CAR REG)))
         (RPLACA (CDR REG)
                 (SUB1 (CADR REG)))
         (DRAWLINE (CAR REG)
                   (CADR REG)
                   W
                   (CADR REG)
                   1
                   'INVERT WINDOW)
         (DRAWLINE (CAR REG)
                   H W H 1 'INVERT WINDOW)
         (DRAWLINE (CAR REG)
                   (CADR REG)
                   (CAR REG)
                   H 1 'INVERT WINDOW)
         (DRAWLINE W (CADR REG)
                   W H 1 'INVERT WINDOW])
```

(\VALID.POSITION.LIST

```
[LAMBDA (POSITIONLIST)                                     (* GWexler " 7-Feb-86 10:34")

          (* Checks out to see if POSITIONLIST is either NIL a valid position, or a list of valid positions.
          Returns either NIL ERROR or a list of positions (maybe only one element list.))

    (if (NOT POSITIONLIST)
        then NIL
      else (if (POSITIONP POSITIONLIST)
               then (LIST POSITIONLIST)
             else (if (LISTP POSITIONLIST)
                      then (if (for P in POSITIONLIST always (POSITIONP P))
                               then POSITIONLIST
                             else (printout T "Not all elements in this list are positions." T)
                                  'ERROR)
                    else (printout T "POSITIONLIST must be NIL, a position, or a list of positions." T)
                         'ERROR])
```

(\SAME.IREGIONS.LIST

```
[LAMBDA (LIST1 LIST2)                                      (* GWexler " 7-Feb-86 10:27")
                                                           (* Tests to see if two lists of regions are lists of the same
                                                           regions.)
    (if (NEQ (LENGTH LIST1)
             (LENGTH LIST2))
        then NIL
      else (for ELT in LIST1 always (FMEMB ELT LIST2])
```

(\WITH.INTERSECTION

```
[LAMBDA (IRLIST FLG)                                       (* GWexler " 7-Feb-86 10:34")

          (* If FLG is T then return IRLIST, otherwise return NIL or a list with just one element.)

    (if FLG
        then IRLIST
      else (if (NOT IRLIST)
               then NIL
             else (LIST (CAR IRLIST])
```

```
(\IREGION.ON.WINDOWP
  [LAMBDA (IREGION WINDOW)                                    (* GWexler " 7-Feb-86 10:27")
    (if [AND (IREGIONP IREGION)
             (WINDOWP WINDOW)
             (FMEMB IREGION (WINDOWPROP WINDOW 'IREGIONSLIST]
        then IREGION
      else NIL])

)

(DECLARE%: EVAL@COMPILE

(DATATYPE IREGION (BUTTONEVENTFN                              (* The users function to be called.)
                  USERDATA                                    (* This is where the users data is kept in proplist format.)
                  REGION                                      (* The region relative to the window where the IREGION is
                                                              located.)
                  MASK

          (* The mask is the same size as region and contains black where ever the IREGION is to be active.)

                  SHADE                                       (* The shade which IREGION inverts to %.)
                  HELPSTRING                                  (* The helpstring to be printed when the button is held down in
                                                              this IREGION)
                  )                                           (* there used to be a TYPE? here but it was useless and
                                                              removed)
       )
)

(/DECLAREDATATYPE 'IREGION '(POINTER POINTER POINTER POINTER POINTER POINTER)
       ;; ---field descriptor list elided by lister---
       '12)

(PUTPROPS IREGIONPROP ARGNAMES (NIL (IREGION PROP {NEWVALUE}) . U))

(RPAQQ DEFAULT.IREGION.SHADE 65535)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS DEFAULT.IREGION.SHADE)
)

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

(ADDTOVAR NLAMA )

(ADDTOVAR NLAML )

(ADDTOVAR LAMA IREGIONPROP)
)

(PUTPROPS AIREGIONS COPYRIGHT ("XEROX Corporation" 1985 1986 1987))
```

FUNCTION INDEX

VARIABLE INDEX

PROPERTY INDEX

RECORD INDEX