File created: 2-Feb-87 21:13:01 **{ERIS}<IRIS>NEXT>IRISDEMOFNS.;10**

changes to: (VARS IRISDEMOFNSCOMS)
(FNS TETRA TETRA.DRAW.FACE TETRA.OBJ)

previous date: 4-Mar-86 10:57:38 {ERIS}<IRIS>NEXT>IRISDEMOFNS.;8

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
(RPAQQ IRISDEMOFNSCOMS
       [(FNS IRIS.DEGREES SNOW SPHERE TETRA TETRA.COLOR TETRA.DRAW.FACE TETRA.OBJ TETRA.TILT.AND.RECURSE)
        (VARS IRIS.TILT TETRA.COLOR TETRA.EDGE.COLOR TETRA.SHRINK TETRA.TILT IV.DEFAULT.STYLE)
        ;; minimal 3-d support for the tetra demo
        (RECORDS 3POINT)
        (FNS 3DOT 3DRAWTO 3MOVETO 3NORMALIZE 3PLUS 3POINT 3UNITCROSSPRODUCT 3DIFFERENCE 3CROSSPRODUCT 3LENGTH
             3LINE 3TIMES DRAW.FACE? IRIS.XLATE)
        (VARS \IRIS.DUMMYBUFFER \IRIS.FEEDBACKBUFFER)
        (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA)
                                                                                (NLAML)
                                                                                (LAMA TETRA])

(DEFINEQ

(IRIS.DEGREES
  [LAMBDA (DEGREES)                                          (* edited%: "13-Dec-85 18:32")

          (* Takes an angle in degrees and returns an angle as the iris likes it
          (tenths))

    (FIX (TIMES DEGREES 10])

(SNOW
  [LAMBDA (N)                                                (* edited%: "11-Dec-85 23:12")
    (for I to (OR N (RAND 5 20)) do (IRIS.PUSHMATRIX)
                                    (IRIS.TRANSLATE (RAND 0 SCREENWIDTH)
                                           (RAND 0 SCREENHEIGHT)
                                           0 \IRISSTREAM)
                                    (IRIS.ROTATE (RAND 0 1800)
                                           88)
                                    (IRIS.ROTATE (RAND 0 1800)
                                           89)
                                    (IRIS.ROTATE (RAND 0 1800)
                                           90)
                                    (IRIS.SCALE (RAND 0.1 1)
                                           (RAND 0.1 1)
                                           (RAND 0.1 1))
                                    (SPHERE " Noel" (RAND 5 90)
                                           (RAND 1 3))
                                    (IRIS.POPMATRIX])

(SPHERE
  [LAMBDA (MSG THETA COUNT)                                  (* edited%: "11-Dec-85 15:24")
    (IRIS.PUSHMATRIX)
    (OR THETA (SETQ THETA 30))
    (OR COUNT (SETQ COUNT 3))
    (DSPCOLOR 'RED \IRISSTREAM)
    (IRIS.PUSHMATRIX)
    (for I from 0 to (IQUOTIENT 360 THETA) do (MOVETO 0 0 \IRISSTREAM)
                                              (DSPCOLOR (IMOD I 7)
                                                     \IRISSTREAM)
                                              (IRIS.ROTATE (ITIMES 10 THETA)
                                                     IRIS.ZAXIS)
                                              (PRINTOUT \IRISSTREAM MSG))
    (IRIS.POPMATRIX)
    (IRIS.PUSHMATRIX)
    (IRIS.ROTATE 900 IRIS.YAXIS)
    (DSPCOLOR 'BLACK \IRISSTREAM)
    (SELECTQ COUNT
         (1 NIL)
         (for I from 0 to (IQUOTIENT 360 THETA) do (MOVETO 0 0 \IRISSTREAM)
                                                   (DSPCOLOR (IMOD I 7)
                                                          \IRISSTREAM)
                                                   (IRIS.ROTATE (ITIMES 10 THETA)
                                                          IRIS.ZAXIS)
                                                   (PRINTOUT \IRISSTREAM MSG)))
    (IRIS.POPMATRIX)
    (DSPCOLOR 'CYAN \IRISSTREAM)
    (IRIS.ROTATE 900 IRIS.XAXIS)
    (SELECTQ COUNT
         ((1 2)
              NIL)
```

```
        (for I from 0 to (IQUOTIENT 360 THETA) do (MOVETO 0 0 \IRISSTREAM)
                                                  (DSPCOLOR (IMOD I 7)
                                                         \IRISSTREAM)
                                                  (IRIS.ROTATE (ITIMES 10 THETA)
                                                         IRIS.ZAXIS)
                                                  (PRINTOUT \IRISSTREAM MSG)))
    (IRIS.POPMATRIX])
```

(TETRA

```
  [CL:LAMBDA (&OPTIONAL (SIDE-LENGTH 200)
                    (RECURSIVE-DEPTH 3)
                    (SHRINK-FACTOR TETRA.SHRINK)
                    (STYLE 'WIREFRAME)
                    (DONTBASERECURSE NIL))                    ; Edited 31-Jan-87 17:29 by gbn
```

;;; Draws a recursive tetrahedron. shrinkfactor is the ratio of side length of parent and child. style is one of 'wireframe, polygon or normal.

```
          (LET ((RECURSIVE-DEPTH (OR RECURSIVE-DEPTH 5))
                (SHRINK-FACTOR (OR SHRINK-FACTOR TETRA.SHRINK))
                (STYLE (OR STYLE IV.DEFAULT.STYLE)))
               (if (EQ 0 RECURSIVE-DEPTH)
                   then                                       ; done
                        NIL
                 else (TETRA.OBJ SIDE-LENGTH (TETRA.COLOR RECURSIVE-DEPTH)
                                 STYLE DONTBASERECURSE)
                      (if (NOT DONTBASERECURSE)
                          then (IRIS.PUSHMATRIX)
                               (IRIS.ROTATE (IRIS.DEGREES 180)
                                      IRIS.YAXIS)
                               (IRIS.ROTATE (IRIS.DEGREES (MINUS TETRA.TILT))
                                      IRIS.XAXIS)
                               (IRIS.TRANSLATE 0 (QUOTIENT SIDE-LENGTH (SQRT 3))
                                      0)
                               (TETRA.TILT.AND.RECURSE SIDE-LENGTH RECURSIVE-DEPTH SHRINK-FACTOR STYLE)
                               (IRIS.POPMATRIX))
                      (IRIS.PUSHMATRIX)
                      (IRIS.TRANSLATE 0 (QUOTIENT SIDE-LENGTH (SQRT 3))
                             0)                                ; move the origin to the middle of the base of the tetrahedron
                      (TETRA.TILT.AND.RECURSE SIDE-LENGTH RECURSIVE-DEPTH SHRINK-FACTOR STYLE)
                      (IRIS.ROTATE (IRIS.DEGREES 120)
                             IRIS.ZAXIS)
                      (TETRA.TILT.AND.RECURSE SIDE-LENGTH RECURSIVE-DEPTH SHRINK-FACTOR STYLE)
                      (IRIS.ROTATE (IRIS.DEGREES 120)
                             IRIS.ZAXIS)
                      (TETRA.TILT.AND.RECURSE SIDE-LENGTH RECURSIVE-DEPTH SHRINK-FACTOR STYLE)
                      ;; (IRIS.TRANSLATE 0 (MINUS (QUOTIENT X (SQRT 3))) 0) (IRIS.ROTATE (IRIS.DEGREES 180) IRIS.XAXIS)
                      ;; (IRIS.ROTATE (IRIS.DEGREES 180) IRIS.ZAXIS) (TETRA.TILT.AND.RECURSE X RECDEPTH)
                      (IRIS.POPMATRIX])
```

(TETRA.COLOR

```
  [LAMBDA (COLOR)                                             (* gbn "21-Feb-86 17:11")
    (IMOD COLOR 8])
```

(TETRA.DRAW.FACE

```
  [LAMBDA (STYLE COLOR LEFT RIGHT TOP)                        ; Edited 31-Jan-87 18:44 by gbn
```

;; handles drawing a single face. Left right and top are just logical names for the points of the triangle. They need not correspond to Tetra's ;; interpretation of those names.

```
    (SELECTQ STYLE
         (WIREFRAME)
         ((POLYGON NORMALS BACKFACES)
              (if (NOT DONTBASERECURSE)
                  then (IRIS.POLF 3 (LIST LEFT RIGHT FRONT)))
              (DSPCOLOR COLOR \IRISSTREAM)
              (IRIS.POLF 3 (LIST LEFT RIGHT TOP))             ; (IRIS.POLF 3 (LIST FRONT RIGHT TOP)) (IRIS.POLF 3 (LIST
                                                              ; FRONT LEFT TOP))
              (DSPCOLOR TETRA.EDGE.COLOR \IRISSTREAM)
              (3MOVETO \IRISSTREAM LEFT)
              (3DRAWTO \IRISSTREAM RIGHT)
              (3DRAWTO \IRISSTREAM TOP)
              (3DRAWTO \IRISSTREAM LEFT)
              (SELECTQ STYLE
                   (POLYGON)
                   (NORMALS                                   ; compute and draw a normal to the face
                            [LET* ((LEFTTOP (3DIFFERENCE TOP LEFT))
                                   (LEFTRIGHT (3DIFFERENCE RIGHT LEFT))
                                   (NORMALENDPT (3CROSSPRODUCT LEFTTOP LEFTRIGHT))
                                   (NORMAL (3DIFFERENCE NORMALENDPT LEFT)))
                                  (3LINE LEFT (3PLUS LEFT (3TIMES (3NORMALIZE NORMAL)
                                                                  50])
                   (BACKFACES                                 ; compute and draw a normal to the face
                              (LET* ((LEFTTOP (3DIFFERENCE TOP LEFT))
                                     (LEFTRIGHT (3DIFFERENCE RIGHT LEFT))
```

```
                                    (NORMALENDPT (3CROSSPRODUCT LEFTTOP LEFTRIGHT))
                                    (NORMAL (3DIFFERENCE NORMALENDPT LEFT))
                                    (EYEVECTOR (3DIFFERENCE (IRIS.XLATE IV.VIEWPT)
                                                            LEFT)))
                                   (if (LESSP (3DOT EYEVECTOR NORMAL)
                                              0.0)
                                       then                          ; this is not a backface so drawit
                                            (DSPCOLOR COLOR \IRISSTREAM)
                                            (IRIS.POLF 3 (LIST LEFT RIGHT TOP))
                                            (DSPCOLOR TETRA.EDGE.COLOR \IRISSTREAM)
                                            (3MOVETO \IRISSTREAM LEFT)
                                            (3DRAWTO \IRISSTREAM RIGHT)
                                            (3DRAWTO \IRISSTREAM TOP)
                                            (3DRAWTO \IRISSTREAM LEFT))))
                    (ERROR "Unknown drawing style: " STYLE)))
            (ERROR "Unknown drawing style: " STYLE])
```

(TETRA.OBJ

```
  [LAMBDA (X COLOR STYLE DONTBASERECURSE)                    ; Edited 31-Jan-87 17:35 by gbn
```

;;; The function that draws a single tetrahedron (and optionally, it's faces.)

```
    (LET ([TOP (3POINT 0 (QUOTIENT X (SQRT 3))
                       (SQRT (TIMES (TIMES X X)
                                    (QUOTIENT 8 3.0]
          (LEFT (3POINT (MINUS X)
                        0 0))
          (RIGHT (3POINT X 0 0))
          (FRONT (3POINT 0 (TIMES (SQRT 3)
                                  X)
                         0)))
          (IRIS.PUSHATTRIBUTES)
          (SELECTQ STYLE
               (WIREFRAME (DSPCOLOR COLOR \IRISSTREAM)
                          (3MOVETO \IRISSTREAM LEFT)
                          (3DRAWTO \IRISSTREAM RIGHT)
                          (3DRAWTO \IRISSTREAM FRONT)
                          (3DRAWTO \IRISSTREAM LEFT)
                          (3DRAWTO \IRISSTREAM TOP)
                          (3DRAWTO \IRISSTREAM RIGHT)
                          (3MOVETO \IRISSTREAM FRONT)
                          (3DRAWTO \IRISSTREAM TOP))
               ((POLYGON NORMALS BACKFACES)
                    (DSPCOLOR COLOR \IRISSTREAM)
                    (if (NOT DONTBASERECURSE)
                        then (TETRA.DRAW.FACE STYLE COLOR LEFT RIGHT FRONT))
                    (TETRA.DRAW.FACE STYLE COLOR LEFT RIGHT TOP)
                    (TETRA.DRAW.FACE STYLE COLOR RIGHT FRONT TOP)
                    (TETRA.DRAW.FACE STYLE COLOR FRONT LEFT TOP))
               (ERROR "Unknown drawing style: " STYLE))
          (IRIS.POPATTRIBUTES])
```

(TETRA.TILT.AND.RECURSE

```
  [LAMBDA (X RECDEPTH SHRINKFACTOR STYLE)                    (* edited%: "16-Dec-85 17:41")
```

(* * sets up the transformations to recurse, and calls tetra)

(* * called with 0 0 0 already placed at the "bottom edge" on the face of the larger tetra)

(* BOTTOMY is the y component of the point BOTTOM, which is not explicitly calculated)

```
    (LET [(BOTTOMY (QUOTIENT X (SQRT 3]
          (IRIS.PUSHMATRIX)
          (IRIS.TRANSLATE 0 (MINUS (QUOTIENT X (SQRT 3)))
                 0)
          (IRIS.ROTATE (IRIS.DEGREES TETRA.TILT)
                 IRIS.XAXIS)
          (IRIS.TRANSLATE 0 (DIFFERENCE BOTTOMY (TIMES BOTTOMY SHRINKFACTOR))
                 0)
          (IRIS.SCALE SHRINKFACTOR SHRINKFACTOR SHRINKFACTOR)
          (TETRA (TIMES SHRINKFACTOR X)
                 (SUB1 RECDEPTH)
                 SHRINKFACTOR STYLE T)                       (* IRIS.TRANSLATE 0 (MINUS BOTTOMY) 0)
                                                             (* put 0 0 0 back on the edge of the larger tetra)
          (IRIS.POPMATRIX])
```

)

```
(RPAQQ IRIS.TILT 70.52878)

(RPAQQ TETRA.COLOR BLUE)

(RPAQQ TETRA.EDGE.COLOR BLACK)

(RPAQQ TETRA.SHRINK 0.7)
```

```
(RPAQQ TETRA.TILT 70.52878)

(RPAQQ IV.DEFAULT.STYLE WIREFRAME)
```

;; minimal 3-d support for the tetra demo

```
(DECLARE%: EVAL@COMPILE

(RECORD 3POINT (3X 3Y 3Z))
)

(DEFINEQ
```

(3DOT

```
  [LAMBDA (A B)                                          (* gbn " 3-Mar-86 17:54")
    (PLUS (TIMES (fetch 3X of A)
                 (fetch 3X of B))
          (TIMES (fetch 3Y of A)
                 (fetch 3Y of B))
          (TIMES (fetch 3Z of A)
                 (fetch 3Z of B])
```

(3DRAWTO

```
  [LAMBDA (STREAM XOR3PT Y Z)                            (* edited%: "13-Dec-85 16:16")
    (if (NUMBERP XOR3PT)
        then (IRIS.DRAW XOR3PT Y Z STREAM)
      else (IRIS.DRAW (fetch 3X of XOR3PT)
                      (fetch 3Y of XOR3PT)
                      (fetch 3Z of XOR3PT)
                      STREAM])
```

(3MOVETO

```
  [LAMBDA (STREAM XOR3PT Y Z)                            (* edited%: "13-Dec-85 16:16")
    (if (NUMBERP XOR3PT)
        then (IRIS.MOVE XOR3PT Y Z STREAM)
      else (IRIS.MOVE (fetch 3X of XOR3PT)
                      (fetch 3Y of XOR3PT)
                      (fetch 3Z of XOR3PT)
                      STREAM])
```

(3NORMALIZE

```
  [LAMBDA (3VECTOR)                                      (* gbn " 3-Mar-86 15:51")
```

(* * Produces a vector with the same direction but unit magnitude as 3VECTOR)

```
    (LET ((LENGTH (3LENGTH 3VECTOR)))
         (3POINT (QUOTIENT (fetch 3X of 3VECTOR)
                           LENGTH)
                 (QUOTIENT (fetch 3Y of 3VECTOR)
                           LENGTH)
                 (QUOTIENT (fetch 3Z of 3VECTOR)
                           LENGTH])
```

(3PLUS

```
  [LAMBDA (A B)                                          (* gbn " 3-Mar-86 14:46")
                                                         (* vector sum of a and b)
    (3POINT (PLUS (fetch 3X of A)
                  (fetch 3X of B))
            (PLUS (fetch 3Y of A)
                  (fetch 3Y of B))
            (PLUS (fetch 3Z of A)
                  (fetch 3Z of B])
```

(3POINT

```
  [LAMBDA (X Y Z)                                        (* edited%: "13-Dec-85 16:02")
                                                         (* creates a 3-d point)
    (create 3POINT
            3X _ X
            3Y _ Y
            3Z _ Z])
```

(3UNITCROSSPRODUCT

```
  [LAMBDA (A B)                                          (* gbn " 3-Mar-86 15:51")
    (LET* ((NORMAL (3CROSSPRODUCT A B))
           (LENGTH (3LENGTH NORMAL)))
          (replace 3X of NORMAL with (QUOTIENT (fetch 3X of NORMAL)
                                               LENGTH))
          (replace 3Y of NORMAL with (QUOTIENT (fetch 3Y of NORMAL)
                                               LENGTH))
          (replace 3Z of NORMAL with (QUOTIENT (fetch 3Z of NORMAL)
                                               LENGTH))
```

```
          NORMAL])
```

(3DIFFERENCE

```
  [LAMBDA (DEST SOURCE)                                    (* gbn "28-Feb-86 17:13")
                                                           (* vector difference from source to dest)
    (3POINT (DIFFERENCE (fetch 3X of DEST)
                        (fetch 3X of SOURCE))
            (DIFFERENCE (fetch 3Y of DEST)
                        (fetch 3Z of SOURCE))
            (DIFFERENCE (fetch 3Z of DEST)
                        (fetch 3Z of SOURCE])
```

(3CROSSPRODUCT

```
  [LAMBDA (A B)                                            (* gbn "28-Feb-86 17:17")
    (3POINT (DIFFERENCE (TIMES (fetch 3Y of A)
                               (fetch 3Z of B))
                        (TIMES (fetch 3Z of A)
                               (fetch 3Y of B)))
            (DIFFERENCE (TIMES (fetch 3Z of A)
                               (fetch 3X of B))
                        (TIMES (fetch 3X of A)
                               (fetch 3Z of B)))
            (DIFFERENCE (TIMES (fetch 3X of A)
                               (fetch 3Y of B))
                        (TIMES (fetch 3Y of A)
                               (fetch 3X of B])
```

(3LENGTH

```
  [LAMBDA (A)                                              (* gbn " 3-Mar-86 15:36")

          (* * returns the euclidean norm of the |3d| vector)

    (SQRT (PLUS (TIMES (fetch 3X of A)
                       (fetch 3X of A))
                (TIMES (fetch 3Y of A)
                       (fetch 3Y of A))
                (TIMES (fetch 3Z of A)
                       (fetch 3Z of A])
```

(3LINE

```
  [LAMBDA (A B)                                            (* gbn "28-Feb-86 17:22")
    (3MOVETO \IRISSTREAM A)
    (3DRAWTO \IRISSTREAM B])
```

(3TIMES

```
  [LAMBDA (VECTOR SCALAR)                                  (* gbn " 3-Mar-86 14:47")
    (3POINT (TIMES (fetch 3X of VECTOR)
                   SCALAR)
            (TIMES (fetch 3Y of VECTOR)
                   SCALAR)
            (TIMES (fetch 3Z of VECTOR)
                   SCALAR])
```

(DRAW.FACE?

```
  [LAMBDA (LEFT RIGHT TOP COLOR)                           (* gbn " 3-Mar-86 18:45")

          (* handles drawing a single face. Left right and top are just logical names for the points of the triangle.
          They need not correspond to Tetra's interpretation of those names.)

    (LET* ((LEFTTOP (3DIFFERENCE TOP LEFT))
           (LEFTRIGHT (3DIFFERENCE RIGHT LEFT))
           (NORMALENDPT (3CROSSPRODUCT LEFTTOP LEFTRIGHT))
           (NORMAL (3DIFFERENCE NORMALENDPT LEFT))
           (EYEVECTOR (3DIFFERENCE (IRIS.XLATE IV.VIEWPT)
                                   LEFT)))
          (if (GREATERP (3DOT EYEVECTOR NORMAL)
                        0.0)
              then                                         (* this is not a backface so drawit)
                    (DSPCOLOR (OR COLOR 'CYAN)
                              \IRISSTREAM)
                    (IRIS.POLF 3 (LIST LEFT RIGHT TOP))
                    (DSPCOLOR TETRA.EDGE.COLOR \IRISSTREAM)
                    (3MOVETO \IRISSTREAM LEFT)
                    (3DRAWTO \IRISSTREAM RIGHT)
                    (3DRAWTO \IRISSTREAM TOP)
                    (3DRAWTO \IRISSTREAM LEFT])
```

(IRIS.XLATE

```
  [LAMBDA (3VECTOR)                                        (* gbn " 3-Mar-86 17:18")
    (IRIS.FEEDBACK \IRIS.DUMMYBUFFER 9)
```

```
    (IRIS.XFPT (fetch 3X of 3VECTOR)
           (fetch 3Y of 3VECTOR)
           (fetch 3Z of 3VECTOR))
    (if (NOT (EQUAL (IRIS.ENDFEEDBACK \IRIS.FEEDBACKBUFFER)
                    9))
        then (HELP "NINE ITEMS NOT RETURNED"))
    (3POINT (create FLOATP
                    HIWORD _ (ELT \IRIS.FEEDBACKBUFFER 2)
                    LOWORD _ (ELT \IRIS.FEEDBACKBUFFER 3))
           (create FLOATP
                   HIWORD _ (ELT \IRIS.FEEDBACKBUFFER 4)
                   LOWORD _ (ELT \IRIS.FEEDBACKBUFFER 5))
           (create FLOATP
                   HIWORD _ (ELT \IRIS.FEEDBACKBUFFER 6)
                   LOWORD _ (ELT \IRIS.FEEDBACKBUFFER 7])

)

(RPAQ \IRIS.DUMMYBUFFER (READARRAY 9 'FIXP 1))


(1 1 1 1 1 1 1 1 1 NIL)

(RPAQ \IRIS.FEEDBACKBUFFER (READARRAY 9 'FIXP 1))


(56 17275 9800 17288 8544 17585 41814 17585 41814 NIL)

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

(ADDTOVAR NLAMA )

(ADDTOVAR NLAML )

(ADDTOVAR LAMA TETRA)
)
```

FUNCTION INDEX

VARIABLE INDEX

RECORD INDEX