

```

1  !S5Screen.f95      Group ID: #5      Stereo-3D Simulation Environment Vsn:1.00
2  !2025.05.24.1840cdt- Screen viewing utilities.
3
4  !      Author- Jeffrey M. Setterholm, Lakeville,MN 55044 USA
5  !      IP Status- Free source code (e.g.: post copyright)
6  !
7  !      Computer- "T3"/Dell Precision T3500/Intel i5 E5520/win10Pro-21H2
8  !                  ^name ^mfr.Id          ^chipset      ^OS
9  !                  /Absoft Pro Fortran 21.0.2/GeForce GTX 1050/f90gl~Glut3.7
10 !                  ^compiler ~Fortran 95      ^graphics card ^graphics
11
12 !      f90gl bindings- public domain; see "https://math.nist.gov/f90gl/"
13
14 !Disclaimer:
15 !*****
16 !      ***** Individual cognition is always flawed, *****
17 !      ***** including yours and mine. *****
18 !      ***** - So: - *****
19 !      ***** Use this code at your own risk. *****
20 !      *****
21
22 !Table of Contents: ...use to search...
23 ! Subroutine PrntOrtho(nRow,mColumn,iColorFG,iColorBg,PText)
24 ! Subroutine PrintableIchar(iCharIn,iCharOut)
25 ! Subroutine Seeh4d(Vin,nRowIn,nColIn,iColor,Label,iP)
26 ! Subroutine Seeh44d(Hin,nRowIn,nColIn,iColor,Label,iP)
27 ! Subroutine Teapot
28 ! Subroutine Colors3D(nCol)
29 ! Subroutine ScreenSelfie
30 ! Subroutine ShowProjectAndModel(nRowIn,nColIn,iColor,Label20)
31 ! Subroutine HfsToHgl(Hfs,Hgl)
32 ! Subroutine HglToHfs(Hgl,Hfs)
33 ! Subroutine GLv16toHgl(GLv,Hgl)
34 ! Subroutine BbFog
35 !-----7 9
36
37 Subroutine PrntOrtho(nRowIn,mColumnIn,iColorFG,iColorBg,PText)
38 !2023.09.21.0520cdt JMS- nRowIn<0: map to right side of present screen &
39 !                        nColIn<0: map to bottom side of present screen
40 !2020.04.04.1100cdt JMS- Printing to the orthographic screen. v:0.5
41 ! Print text on a raw OpenGL screen (Proj. & Modelview matrices are identities)
42 ! using an OpenGL bitmap font.
43 ! Row & column sizing is established in GlutHandoff (font choice dependent).
44 !--Globals in S1ModDef.f95
45 use OpenGLRec,only: & !Ref: OpenGL GL/GLU/GLUT docs
46     glMatrixMode,glGetIntegerv,glPushMatrix,glPopMatrix,glRasterPos3d &
47     ,GL_MATRIX_MODE,GL_PROJECTION,GL_MODELVIEW,glLoadIdentity,glFlush &
48     ,glutBitmapCharacter,GLUT_BITMAP_8_BY_13,GLUT_BITMAP_9_BY_15
49 use ScreenDef,only: & !screen & colors
50     PixelsPerCM,nCharCenY,cwidth,cHeight,xPrint,yPrint,zPrint &
51     ,nCharMaxX,nCharMaxY &
52     ,SideInit,Side,nCharOffsets,nCharMaxXS,nCharCenXS
53 !--End Globals
54 implicit none
55 !--Arguments
56 integer(4):: nRowIn,mColumnIn,iColorFG,iColorBg
57 character(len=80):: PText
58 !--Internals
59 integer(4):: nRow,mColumn
60 integer(4):: iCharL,n,nChar,MtxMode(1)
61 !--EndDefs-----
62

```

```

63   if(SideInit==0) then
64     nCharOffsets=(/      0      ,      0      , nCharMaxX/2 /)
65     nCharMaxXS  =(/ nCharMaxX/2,nCharMaxX, nCharMaxX/2 /)
66     nCharCenXS  =nCharMaxXS/2
67     SideInit    =1
68   endif!SideInit==0
69
70   nRow  =nRowIn;  if(nRow <0) nRow  =nCharMaxY      +nRow  !2023.09.21
71   mColumn=mColumnIn;if(mColumn<0) mColumn=nCharMaxXS(Side)+mColumn!2023.09.21
72
73   !--Save MatrixMode,PROJECTION,MODELVIEW & load Identities:
74   call Colors3D(iColorFG)
75   call glGetIntegerv(GL_MATRIX_MODE,MtxMode)  !GL_MATRIX_MODE=ba0=2976 dec.
76   call glMatrixMode(GL_PROJECTION); call glPushMatrix; call glLoadIdentity
77   call glMatrixMode(GL_MODELVIEW ); call glPushMatrix; call glLoadIdentity
78   call glFlush
79   !--Position the first character:
80   xPrint= -1.00d0 +(nCharOffsets(Side)+mColumn-1)*cwidth
81   yPrint=  1.00d0 -              nRow      *cHeight
82   zPrint= -0.10d0
83   call glRasterPos3d( xPrint,yPrint,zPrint)
84   ! Write the string as individual character cwidth*cHeight bitmaps.
85     nChar=len_trim(PText)
86   do n=1,nChar; if(n+mColumn>nCharMaxXS(Side)) exit
87     call PrintableIchar(iChar(PText(n:n)),iCharL)          ;if(iCharL==0) exit
88     if(PixelsPerCm<40.d0) then
89       call glutBitmapCharacter(GLUT_BITMAP_8_BY_13,iCharL)
90     else
91       call glutBitmapCharacter(GLUT_BITMAP_9_BY_15,iCharL)
92     endif!PixelsPerCm<40.d0
93   enddo!n
94   call glFlush
95   !--Restore PROJECTION, MODELVIEW, & MatrixMode:
96   call glMatrixMode(GL_PROJECTION); call glPopMatrix
97   call glMatrixMode(GL_MODELVIEW ); call glPopMatrix
98   call glMatrixMode(MtxMode(1))          ;call glFlush;return
99 End Subroutine PrntOrtho
100 !-----7 9
101
102 Subroutine PrintableIchar(iCharIn,iCharOut)
103 !2020.04.22.1755cdt JMS- Returns a printable character or 0
104 implicit none
105 !--Arguments
106 integer(4):: iCharIn,iCharOut
107 !--Internals
108 integer(4)::iCharL
109 !--EndDefs-----
110     iCharL = iCharIn
111   select case(iCharL)
112     case( :31 ); iCharOut=0 ;return
113     case( 32:126); iCharOut=iCharL ;return
114     case( 127); iCharOut=0 ;return
115     case(128:255); iCharOut=iCharL ;return
116     case(256: ); iCharOut=0 ;return
117   end select!iCharL
118 End Subroutine PrintableIchar
119 !-----7 9
120
121 Subroutine Seeh4d(Vin,nRowIn,nColIn,iColor,Label,iP)
122 !2020.04.19.17205cdt JMS- Displays Vin(4) at screen location nRowIn,nColIn.
123 !--Globals S1ModDef.f95: 2024.03.18
124 use ScreenDef ,only: nCharMaxX,nCharMaxY !screen & colors !2020.04.10
125 !--End Globals

```

```

125 !--END GLOBALS
126 implicit none
127 !--Arguments
128 real(8) ::Vin(4) !Input matrix to be displayed
129 integer(4)::nRowIn,nColIn !Text- screen upper-left character position
130 integer(4)::iColor ! - color on screen
131 character::Label*80
132 integer(4)::iP ! >5: Printing enable, use: write(iP,...)
133 !--Internals
134 integer(4)::i
135 integer(4)::nRow,nCol !Internal row/column counter
136 !character::PText*80
137 character(len=80)::PText
138 !--EndDefs-----
139 if(iColor.gt.0) then !Print Label & Vin(4) to screen
140 !Offset down/right if>0 up/left if<0 Text block below has
141 nRow=NrowIn; if(nRow<0) nRow=nCharMaxY+nRow+1 ! 5 rows &
142 nCol=nColIn; if(nCol<0) nCol=nCharMaxx+nCol+1 !56 columns
143 call PrntOrtho(nRow,nCol, 1 ,0,Label)
144 write(PText,"(4f14.6)") Vin
145 nRow=nRow+1; call PrntOrtho(nRow,nCol,iColor,0,PText)
146 endif!iColor>0
147 if(iP.gt.5) then !Print Label & Vin(4) to unit iP.
148 write(iP,"(80a1:)" ) (Label(i:i),i=1,len_trim(Label))
149 write(iP,"(4f20.9)") Vin
150 endif
151 End Subroutine Seeh4d ;return
152 !-----7-9
153
154 Subroutine Seeh44d(Hin,nRowIn,nColIn,iColor,Label,iP)
155 !2020.04.19.1720cdt JMS- Displays Hin(4,4) at screen location nRowIn,nColIn.
156 !--Globals S1ModDef.f95: 2024.03.18
157 use ScreenDef ,only: nCharMaxX,nCharMaxY !screen & colors !2020.04.10
158 !--End Globals
159 implicit none
160 !--Arguments
161 real(8) ::Hin(4,4) !Input matrix to be displayed
162 integer(4)::nRowIn,nColIn !Text- screen upper-left character position
163 integer(4)::iColor ! - color on screen
164 character::Label*80
165 integer(4)::iP ! >5: Printing enable, use: write(iP,...)
166 !--Internals
167 integer(4)::i,j
168 integer(4)::nRow,nCol !Internal row/column counter
169 !character::PText*80
170 character(len=80)::PText
171 !--EndDefs-----
172 if(iColor.gt.0) then !Print Label & Hin(4,4) to screen
173 !Offset down/right if>0 up/left if<0 Text block below has
174 nRow=NrowIn; if(nRow<0) nRow=nCharMaxY+nRow+1 ! 5 rows &
175 nCol=nColIn; if(nCol<0) nCol=nCharMaxx+nCol+1 !56 columns
176 call PrntOrtho(nRow,nCol, 1 ,0,Label)
177 do i=1,4
178 write(PText,"(4f14.6)") (Hin(i,j),j=1,4)
179 nRow=nRow+1; call PrntOrtho(nRow,nCol,iColor,0,PText)
180 enddo!i
181 endif!iColor>0
182 if(iP.gt.5) then !Print Label & Hin(4,4) to unit iP.
183 write(iP,"(80a1:)" ) (Label(i:i),i=1,len_trim(Label))
184 do i=1,4
185 write(iP,"(4f20.9)") (Hin(i,j),j=1,4)
186 enddo!i

```

```

187     endif ;return
188 End Subroutine Seeh44d
189 !-----7-9
190
191 Subroutine Teapot
192 !2020.06.01.0405cdt JMS- The GLUT Teapot
193 !--Globals S1ModDef.f95: 2024.03.18
194 use OpenGLRec,only: & !Ref: OpenGL GL/GLU/GLUT Documentation
195     glShadeModel ,GL_SMOOTH ,GL_FLAT,glLineWidth &
196     ,glLightfv ,GL_LIGHT0 ,GL_LIGHT1 ,GL_POSITION ,GL_DIFFUSE &
197     ,glLightModelfv,GL_LIGHT_MODEL_AMBIENT ,GL_SPECULAR &
198     ,glMaterialfv ,GL_FRONT ,GL_EMISSION ,GL_AMBIENT &
199     ,glEnable ,GL_LIGHTING,GL_SHININESS,glDisable,glFlush &
200     ,glTranslatef,glutWireSphere,glutSolidTeapot,glutWireTeapot &
201     ,glMaterialf,GL_MATRIX_MODE,GL_MODELVIEW,glRotated &
202     ,glGetInterv,glMatrixMode,glPushMatrix,glPopMatrix! &
203     !,glColorMaterial,glColor4fv,GL_COLOR_MATERIAL
204 use ioDef ,only:iTeapot !Input/Output & Flags
205 use ScreenDef,only:ColorRGBA,iScrnColor!,ForceRGBA !screen & colors
206 use ViewDef ,only:Key
207 !use VuDef ,only:VuMode !Viewer 2D/3D Projection
208 !--End Globals
209 implicit none
210 !--Internals
211 integer(4)::ForceRGBA=0
212 integer(4)::MtxMode(1)
213 integer(4)::iT
214 !Redbook teapot's surface optical properties for bronze:
215 ! Material surface optical coefficients:
216 real(4):: ambient(4) = (/ 0.2125 , 0.1275 , 0.054 , .02/)
217 real(4):: diffuse(4) = (/ 0.714 , 0.4284 , 0.181 , 1.0 /)
218 real(4):: specular(4) = (/ 0.393548, 0.271906, 0.166721, 1.0 /)
219 ! Light source location & color:
220 real(4):: pos0(4) = (/ 1.0 , 1.0 , 1.0 , 1.0 /)
221 real(4):: pos1(4) = (/ -1.0 , 0.0 , -1.0 , 1.0 /)
222 real(4):: white(4) = (/ 1.0 , 1.0 , 1.0 , .9 /)
223 real(4):: gray(4) = (/ 0.5 , 0.5 , 0.5 , .9 /)
224 !real(4):: L0ambient(4) = (/ 0.2 , 0.2 , 0.2 , 1.0 /)
225 real(4):: L0ambient(4) = (/ 0.6 , 0.6 , 0.6 , 1.0 /)
226 real(4):: noemission(4) = (/ 0.0 , 0.0 , 0.0 , 0.0 /)
227 !--EndDefs-----
228
229 !Roll ModelView by -90. degrees so the teapot draws top-up (toward -Z):
230 call glGetInterv(GL_MATRIX_MODE,MtxMode)
231 call glMatrixMode(GL_MODELVIEW ); call glPushMatrix
232 if(Key%VuMode>-1) call glRotated(-90.0d0, 1.0d0, 0.0d0, 0.0d0)
233
234 if(iTeapot<1) return
235 iT=iTeapot-1
236 !if( iT .gt.2) call cubeGrid(12)
237 if(mod(iT,3).eq.0) call glShadeModel(GL_SMOOTH) !Smooth facets
238 if(mod(iT,3).ne.0) call glShadeModel(GL_FLAT) ! Flat facets
239 call glLineWidth(1.)
240 !--Define Light0's position, color, & ambient contribution:
241
242 ! Display LIGHT0 as an emitting 400-facet wire sphere .5 units in diameter:
243 call glLightfv( GL_LIGHT0,GL_POSITION , pos0 )
244 call glLightfv( GL_LIGHT1,GL_POSITION , pos1 )
245 call glLightModelfv(GL_LIGHT_MODEL_AMBIENT, noemission )
246 call glMaterialfv( GL_FRONT ,GL_EMISSION , white )
247 !if(iScrnColor== 1) &
248 !call glMaterialfv( GL_FRONT ,GL_EMISSION , gray )
249 call glEnable(GL_LIGHTING); call glEnable(GL_LIGHT0)

```

```

249 call glEnable(GL_LIGHTING); call glEnable(GL_LIGHT0)
250 call glTranslatef( pos0(1), pos0(2) , pos0(3) )
251 call glutWireSphere( .05d0, 20, 20 ) !View the light
252 call glTranslatef(-pos0(1),-pos0(2) , -pos0(3) )
253 call glDisable(GL_LIGHT0); call glDisable(GL_LIGHTING)
254
255 call glEnable(GL_LIGHTING); call glEnable(GL_LIGHT1)
256 call glTranslatef( pos1(1), pos1(2) , pos1(3) )
257 call glutWireSphere( .05d0, 20, 20 ) !View the light
258 call glTranslatef(-pos1(1),-pos1(2) , -pos1(3) )
259 call glDisable(GL_LIGHT1); call glDisable(GL_LIGHTING)
260
261 call glMaterialfv(GL_FRONT, GL_EMISSION , noemission)
262 call glLightfv( GL_LIGHT0 , GL_DIFFUSE , white )
263 call glLightfv( GL_LIGHT1 , GL_DIFFUSE , white )
264
265 !--Enable material surface optical properties:
266 !call glEnable(GL_COLOR_MATERIAL); <- isn't working here
267 ambient = (/ 0.2125 , 0.1275 , 0.054 , .02/)
268 diffuse = (/ 0.714 , 0.4284 , 0.181 , 1.0 /)
269 specular = (/ 0.393548, 0.271906, 0.166721, 1.0 /)
270
271 call glMaterialfv(GL_FRONT, GL_AMBIENT , ambient ) !Bronze
272 call glMaterialfv(GL_FRONT, GL_DIFFUSE , diffuse )
273 call glMaterialfv(GL_FRONT, GL_SPECULAR , specular )
274 call glMaterialf( GL_FRONT, GL_SHININESS, 25.6 ) ;call glFlush
275
276 call glEnable( GL_LIGHTING); call glEnable( GL_LIGHT0 ) !Enable lighting
277 call glEnable( GL_LIGHT1 )
278
279 if((mod(it,3)<2).and.(it<6)) call glutSolidTeapot(0.6d0)!draw the Teapot:
280 if( mod(it,3)==2) call glutWireTeapot( 0.6d0)
281 call glDisable(GL_LIGHT0)
282 call glDisable(GL_LIGHT1); call glDisable(GL_LIGHTING) !Disable lighting
283 call glShadeModel(GL_FLAT) !Revert to the unsmoothed glShadeModel
284
285 !--Restore PROJECTION, MODELVIEW, & MatrixMode:
286 call glMatrixMode(GL_MODELVIEW ); call glPopMatrix
287 call glMatrixMode(MtxMode(1)) ;call glFlush;return
288 End Subroutine Teapot
289 !-----7 9
290
291 Subroutine Colors3D(nCol)
292 !2021.03.05.0750cst JMS- Major re-arrangement of colors - Grayscale leading.
293 !2020.04.05.1125cdt JMS- The 14 color palette presently in use in 3Denv.exe
294 !--Globals S1ModDef.f95: 2024.03.18
295 use OpenGLRec ,only:glColor4fv,glFlush !Ref: OpenGL docs
296 use ScreenDef ,only:iScrncolor,ForceRGBA,colorRGBA!screen & colors
297 use ViewDef ,only: ThreePhase,Key !Viewer 2D/3D Proj.
298 !--End Globals
299 implicit none
300 !--Arguments
301 integer(4)::nCol
302 !--Internals
303 !Setup your color choices for a Black background screen.
304 !colors are modified here for viewing on a white background, or for Red/Cyan 3D
305 !A baseline of a few robust, vibrant drawing colors.
306
307 ! Cyan = 'not Red' = !Red
308 ! Magenta = 'not Green' = !Green
309 ! Yellow = 'not Blue' = !Blue
310

```

```

311 !Custom character usages:
312 !   My colors      (c,0) 'move without draw' - a no-op in this context
313 !                  (clarifies & simplifies the content of .3dv files.)
314
315 !                  IndCol[16][3]      IndColOnWhite[16][3]
316 !Background:      Define on Black    Modified for White      Numeric
317 !                  Red Green Blue      Red Green Blue      Brightnesses
318 !   white      (c,1) (255, 255, 255)    ( 0, 0, 0)      0: none
319 !   light Red   ,2) (255, 159, 159)    (159, 0, 0)      32: 1/8
320 !   Red         ,3) (255, 0, 0)         no change      63: 1/4
321 ! !Green= Magenta ,4) (255, 0, 255)    (159, 0, 159)    95: 1/3
322 ! !Blue = Yellow  ,5) (255, 255, 0)    (159, 159, 0)    127: 1/2
323 !   light Green ,6) (159, 255, 159)    ( 0, 159, 0)    159: 2/3
324 !   Green       ,7) ( 0, 255, 0)         no change      191: 3/4
325 ! !Red = Cyan    ,8) ( 0, 255, 255)    ( 0, 159, 159)   223: 7/8
326 !   light Blue   ,9) (159, 159, 255)    ( 0, 0, 159)    255: full
327 !   Blue         ,a) ( 0, 0, 255)         no change
328 !   light Gray   ,b) (191, 191, 191)    ( 95, 95, 95)
329 !   Gray         ,c) (127, 127, 127)    no change
330 !   Brown        ,d) (255, 127, 0)      (159, 127, 0)
331 !   Purple        ,e) (255, 127, 255)    (159, 95, 159)
332 ! !White Black   ,f) ( 0, 0, 0)         (255, 255, 255)
333 !!="not"
334 !   0 <begin line> 1 white      2~LtRed      3 Red
335 !   4 Magenta      5 Yellow     6 LtGreen    7 Green
336 !   8 Cyan         9 LtBlue    10 Blue      11 LtGray
337 !   12 Gray       13 Brown     14 Purple    15 Black
338 !--                                     !Internals
339 ! integer(4)::i,j
340 ! integer(4)::IndCol(3,0:15)
341 ! data ((IndCol(i,j),i=1,3),j=0,15) /
342 !   0, 0, 0, 255,255,255, 255, 95, 95, 255, 0, 0 &
343 !   ,255, 0,255, 255,255, 0, 159,255,159, 0,255, 0 &
344 !   , 0,255,255, 95, 95,255, 0, 0,255, 159,159,159 &
345 !   , 95, 95, 95, 255,127, 0, 255, 95,255, 0, 0, 0 /
346 !   Red Green Blue, Red Green Blue , Red Green Blue , Red Green Blue
347 ! integer(4)::IndColOnWhite(3,0:15)
348 ! data ((IndColOnWhite(i,j),i=1,3),j=0,15)/
349 !   0, 0, 0, 0, 0, 0, 191, 0, 0, 255, 0, 0 &
350 !   ,191, 0,191, 159,159, 0, 0,159, 0, 0,255, 0 &
351 !   , 0,255,255, 127,127,255, 0, 0,255, 95, 95, 95 &
352 !   ,159,159,159, 159,127, 0, 159, 95,159, 255,255,255 /
353
354 !Color Comparisons:
355 !DOS# 'Colors3D' Red Grn Blu 'DOS'colors Red Grn Blu 3D# approx. DXF#
356 ! 0<- 0 BeginLn( na, na ,na) 0 gxBLACK ( 0, 0, 0) =15~0 = 0
357 ! 15= 1 white (255,255,255) 1 gxBLUE ( 0, 0,255) =10 = 5
358 ! 12= 2 LtRed (255,159,159) 2 gxGREEN ( 0,255, 0) = 7 = 3
359 ! 4= 3 Red (255, 0, 0) 3 gxCYAN ( 0,255,255) = 8 = 4
360 ! 5= 4 Magenta(255, 0,255) 4 gxRED (255, 0, 0) = 3 = 1
361 ! 14= 5 Yellow (255,255, 0) 5 gxMAGENTA (255, 0,255) = 4 = 6
362 ! 10= 6 LtGreen(159,255,159) 6 gxBROWN (170, 85, 0) 13 = 62
363 ! 2= 7 Green ( 0,255, 0) 7 gxGRAY (170,170,170) 11 = 9
364 ! 3= 8 Cyan ( 0,255,255) 8 gxDARKGRAY ( 85, 85, 85) 12 = 8
365 ! 9= 9 LtBlue (159,159,255) 9 gxLIGHTBLUE (127,127,255) 9 =161
366 ! 1= 10 Blue ( 0, 0,255) 10 gxLIGHTGREEN (127,255,127) 6 = 81
367 ! 7= 11 LtGray (191,191,191) 11 gxLIGHTCYAN (127,255,255) x->8 =131
368 ! 8= 12 Gray (127,127,127) 12 gxLIGHTRED (255,127,127) = 2 = 11
369 ! 6= 13 Brown (255,127, 0) 13 gxLIGHTMAGENTA(255,127,255) =14 =211
370 ! 13= 14 Purple (255,127,255) 14 gxYELLOW (255,255, 0) = 5 = 2
371 ! 0= 15 Black ( 0, 0, 0) 15 gxWHITE (255,255,255) = 1 = 7
372 !                                     No color passed = -1
373 !-----

```

```

373 !-----
374 !The following color index conversions are located in VecText.h:
375
376 !Converts DOS colors to Colors3D colors: (...interprets black as 'move')
377 !!integer(4),static::ColorsDSto3D(16)=(/15,10,7,8,3,4,13,11,12,9,6,8,2,14,5,1/)
378 !...for importing DOS colors in .3dv files.
379
380 !Converts Colors3D colors to DOS colors: (...interprets 'move' as black)
381 !!integer(4),static::Colors3DtoDOS(16)=(/0,15,12,4,5,14,10,2,3,9,1,7,8,6,13,0/)
382 !...for exporting DOS colors to .3dv files.
383
384 !DOS -to- ACAD R12 Color conversion table (my wag-to-wag):
385 !   Reference http://sub-atomic.com/~moses/acadcolors.html
386 !Converts DOS colors to ACAD R12 colors:
387 !!integer(4),static::ColorsDStoDXFR12(16)= &
388 !!                                     (/0,5,3,4,1,6,62,9,8,161,81,131,11,211,2,7/)
389 !...for exporting ACADR12 colors to .dxf files.
390 !
391 !-----
392 !
393 !-----
394 !Custom character usages:
395 !   My colors      (c,0) 'move without draw' - a no-op in this context
396 !                                     (clarifies & simplifies the content of .3dv files.)
397 !
398 !
399 !Background:      IndCol[16][3]   IndColOnWhite[16][3]   Numeric
400 !                  Define on Black Modified for White   Brightnesses
401 !                  Red Grn Blu     Red Grn Blu           0: none
402 !   light white    (c,1) (255,255,255)   ( 0, 0, 0)
403 !   light Gray     ,2) (159,159,159)     (127,127,127)
404 !   dark Gray      ,3) ( 95, 95, 95)      (191,191,191)
405 !   white Black    ,4) ( 51, 51, 51)      (223,223,223)
406 !   !white Black   ,5) ( 0, 0, 0)         (255,255,255)
407 !   Grayscale ^
408 !   Colors below:
409 !   light Red      ,6) (255,159,159)      (159, 0, 0)   |   Brightness
410 !   Red            ,7) (255, 0, 0)         no change   |   32: 1/8
411 !   !Green= Magenta ,8) (255, 0,255)       (159, 0,159)  |   51: 1/5
412 !   !Blue = Yellow  ,9) (255,255, 0)       (159,159, 0)  |   63: 1/4
413 !   light Green     ,a) (159,255,159)      ( 0,159, 0)   |   95: 1/3
414 !   Green           ,s) ( 0,255, 0)         no change     |   127: 1/2
415 !   !Red = Cyan     ,c) ( 0,255,255)       ( 0,159,159)  |   159: 2/3
416 !   light Blue      ,d) (159,159,255)      ( 0, 0,159)   |   191: 3/4
417 !   Blue            ,e) ( 0, 0,255)         no change     |   204: 4/5
418 !   Brown           ,f) (255,127, 0)       (159,127, 0)  |   223: 7/8
419 !   != "not"
420 !   0 <begin line>  1 white                2 LtGray                3 Gray
421 !   4 DkGray        5 Black                 6~LtRed                 7 Red
422 !   8 Magenta       9 Yellow                10 LtGreen               11 Green
423 !   12 Cyan         13 LtBlue               14 Blue                 15 Brown
424 !--
425 integer(4)::i,j
426 integer(4)::IndCol(3,0:15)
427 data ((IndCol(i,j),i=1,3),j=0,15) /
428   0, 0, 0, 255,255,255, 159,159,159, 95, 95, 95 &
429   , 51, 51, 51, 0, 0, 0, 255, 95, 95, 255, 0, 0 &
430   , 255, 0,255, 255,255, 0, 159,255,159, 0,255, 0 &
431   , 0,255,255, 95, 95,255, 0, 0,255, 255,127, 0 /
432 !   Red Grn Blu | Red Grn Blu | Red Grn Blu | Red Grn Blu
433 integer(4)::IndColOnWhite(3,0:15)
434 data ((IndColOnWhite(i,j),i=1,3),j=0,15)/
435   0, 0, 0, 0, 0, 0, 127,127,127, 191,191,191 &

```

```

435 ! , 223,223,223, 255,255,255, 159, 0, 0, 255, 0, 0 &
436 ! , 159, 0,159, 159,159, 0, 0,159, 0, 0,255, 0 &
437 ! , 0,255,255, 127,127,255, 0, 0,255, 159,127, 0 /
438 ! Red Grn Blu | Red Grn Blu | Red Grn Blu | Red Grn Blu
439
440 data ((IndColonWhite(i,j),i=1,3),j=0,15)/ &
441 0, 0, 0, 0, 0, 0, 127,127,127, 191,191,191 &
442 , 223,223,223, 255,255,255, 159, 0, 0, 255, 0, 0 &
443 , 159, 0,159, 223,223, 0, 0,159, 0, 0,255, 0 & !2023.09.21
444 , 0,255,255, 127,127,255, 0, 0,255, 255,127, 0 / !2023.09.21
445 ! Red Grn Blu | Red Grn Blu | Red Grn Blu | Red Grn Blu
446
447 !Color Comparisons:
448 !DOS# 'Colors3D' Red Grn Blu 'DOS'colors Red Grn Blu 3D# approx. DXF#
449 ! 0<- 0 BeginLn( na, na,na) 0 gxBLACK ( 0, 0, 0) =5~0 = 0
450 ! 15= 1 white (255,255,255) 1 gxBLUE ( 0, 0,255) =10 = 5
451 ! 7 2 LtGray (159,159,159) 2 gxGREEN ( 0,255, 0) = 7 = 3
452 ! 8 3 Gray ( 95, 95, 95) 3 gxCYAN ( 0,255,255) = 8 = 4
453 ! 13= 4 DkGray ( 32, 32, 32) 4 gxRED (255, 0, 0) = 3 = 1
454 ! ^mapping to Purple for now
455 ! 0= 5 Black ( 0, 0, 0) 5 gxMAGENTA (255, 0,255) = 4 = 6
456 ! 12= 6 LtRed (255,159,159) 6 gxBROWN (170, 85, 0) 13 = 62
457 ! 4= 7 Red (255, 0, 0) 7 gxGRAY (170,170,170) 11 = 9
458 ! 5= 8 Magenta(255, 0,255) 8 gxDARKGRAY ( 85, 85, 85) 12 = 8
459 ! 14= 9 Yellow (255,255, 0) 9 gxLIGHTBLUE (127,127,255) 9 =161
460 ! 10 10 LtGreen(159,255,159) 10 gxLIGHTGREEN (127,255,127) 6 = 81
461 ! 2= 11 Green ( 0,255, 0) 11 gxLIGHTCYAN (127,255,255) x->8 =131
462 ! 3= 12 Cyan ( 0,255,255) 12 gxLIGHTRED (255,127,127) = 2 = 11
463 ! 9 13 LtBlue (159,159,255) 13 gxLIGHTMAGENTA(255,127,255) =14 =211
464 ! 1= 14 Blue ( 0, 0,255) 14 gxYELLOW (255,255, 0) = 5 = 2
465 ! 6 15 Brown (255,127, 0) 15 gxWHITE (255,255,255) = 1 = 7
466 ! 11= 12 also
467 !
468 ! No color passed = -1
469 ! ACAD decides =255
470 !real(4)::ColorRGBA(4) !=(Red,Green,Blue,Alpha);
471 !-----
472 !The following color index conversions are located in VecText.h:
473
474 !Converts DOS colors to Colors3D colors: (...interprets black as 'move')
475 !0 1 2 3 4 5 6 7 8 9 a b c d e f
476 integer(4),static::ColorsDOSTo3D(16)=(/ 0,14,11,12,7,8,15,2,3,13,10,12,6,4,9,1/)
477 !...for importing DOS colors in .3dv files.
478
479 !Converts Colors3D colors to DOS colors: (...interprets 'move' as black)
480 !0 1 2 3 4? 5 6 7 8 9 a b c d e f
481 integer(4),static::Colors3DToDOS(16)=(/0,15,7,8,13,0,12,4,5,14,10, 2,3,9,1,6/) !11
482 !...for exporting DOS colors to .3dv files.
483
484 !DOS -to- ACAD R12 Color conversion table (my wag-to-wag):
485 ! Reference http://sub-atomic.com/~moses/acadcolors.html
486 !Converts DOS colors to ACAD R12 colors:
487 integer(4),static::ColorsDOSToDXFR12(16)= &
488 (/0,5,3,4,1,6,62,9,8,161,81,131,11,211,2,7/)
489 !...for exporting ACADR12 colors to .dxf files.
490
491 integer(4)::n
492 !---EndDefs-----
493
494 ! ForceRGBA<0, causes your color 'ColorRGBA[4]' to be used:
495 if(ForceRGBA< 0)then;call glColor4fv(ColorRGBA); call glFlush
496
497 endif

```



```

497      return
498      n=nCol
499      ! ForceRGBA>0 is used by Red/Cyan 3D for color override:
500      if(ForceRGBA> 0)      n=ForceRGBA
501      if((n< 1).or.(n> 15)) n=13      !Default to Brown.
502
503      !Red/Cyan color swap for white screen in VuMode=2:
504      if((Key%VuMode==2).and.(Key%SplitScreen==0)) then
505          if((ForceRGBA> 0).and.(iScrnColor==5)) then !was (iScrnColor==15)) then
506              select case(n) !-----
507                  case( 7); n=12 !Red  -> Cyan
508                  case(12); n= 7 !Cyan -> Red
509              end select !n -----
510          endif !ForceRGBA>0 & iScrnColor=15
511      endif !VuMode=2
512
513      if(iScrnColor==15) then
514          ColorRGBA(1)= IndCol(1,n)      /255.0 !Red
515          ColorRGBA(2)= IndCol(2,n)      /255.0 !Green
516          ColorRGBA(3)= IndCol(3,n)      /255.0 !Blue
517          ColorRGBA(4)= 1.0              !Alpha
518      endif !iScrnColor=15
519
520      if(iScrnColor== 1) then
521          ColorRGBA(1)= IndColOnWhite(1,n)/255.0 !Red
522          ColorRGBA(2)= IndColOnWhite(2,n)/255.0 !Green
523          ColorRGBA(3)= IndColOnWhite(3,n)/255.0 !Blue
524          ColorRGBA(4)= 1.0              !Alpha
525      endif !iScrnColor=1
526
527      call glColor4fv(ColorRGBA)          ;call glFlush
528      return
529 End Subroutine Colors3D
530 !-----7 9
531
532 Subroutine ScreenSelfie
533 !2023.09.08.0955cdt JMS- "SelfieN.bmp"[0,9] -> "SelfieNN.bmp"[1,20]
534 !2020.04.05.1250cdt JMS- This routine dumps the OpenGL "console application"
535 !                          screen as "Selfie_.Bmp" which is formatted as a
536 !                          24 bit-per-pixel full color image.
537 !                          The "~" tilda key triggers the dump.
538 !--Globals                                S1ModDef.f95: 2024.03.18
539 use OpenGLRec,only: & !Ref: OpenGL GL/GLU/GLUT doc
540 glFloat,GLsizei,GLint,GL_FRONT_LEFT,glFlush,GLUT_SCREEN_WIDTH      &
541 ,GL_RGB,GL_FLOAT,glReadBuffer,glReadPixels,glutGet,GLUT_SCREEN_HEIGHT
542 use ioDef ,only: Us,iAlloc,uSelfie !Input/Output & Flags
543 use ScreenDef ,only: & !screen & colors !--End Globals
544 xwindowFull,ywindowFull,xwindowMM,ywindowMM
545
546 implicit none
547 !--Internals
548 type :: BmpHdrL ! Bitmap Header (54 bytes)
549 character:: BM*2 ! "BM" mandatory
550 integer(4):: nSizeTot ! 54+nClrUsed*4
551 !+(((nwidth*(nBitsPP/8)+3)/4)*4)*nHeight *
552 integer(2):: nReserv1,nReserv2 ! 0, 0 not used
553 integer(4):: nOffBit,nSizeH ! 54,40
554 integer(4):: nwidth,nHeight ! __,__ (pixels,pixels)
555 integer(2):: nPlanes,nBitsPP ! 1,__ (24:TC, 8:Grey scales)
556 integer(4):: nCompres,nSizeC ! 0,__ Orthos use in error
557 real(4) :: XPPM,YPPM ! __,__ scaling (USGS float)
558 integer(4):: nClrUsed,nClrImpo ! __, 0 ( 0:TC,256:Grey scales)

```

```

559 end type      BmpHdrL; type(BmpHdrL) :: BH
560
561 character                               ::Selfie*12="SelfieNN.bmp"
562 real(kind=gfFloat),allocatable::fImage(:)
563 character                               ,allocatable::Image2(:)*1
564 integer(kind=GLsizei)::iwidth,jHeight
565 integer(kind=GLint)  :: i3w
566 integer(4)           ::i,iHr3,iHr4,j,k
567 integer(kind=GLint)  ::iw0=0, jh0=0
568 integer(4)           ::nDump= 0
569 !Character*1          ::DumpFile(20)
570 Character             ::PText*80
571 integer(4)           ::PixelRowPadBytes
572 character*1          ::PadByte=char(0)
573 !--EndDefs-----
574
575 nDump=nDump+1
576 if(nDump>20) then
577     write(PText, "('20 screen dumps max.');" );      call PrntOrtho(7,2, 3,0,PText)
578                                                         return
579 endif!(nDump>9)
580 write(Selfie, "('Selfie',i2.2,'.bmp');" ) nDump  !Selfie1.bmp,Selfie2.bmp,etc.
581
582 !--Size the output image
583 iwidth =glutGet(GLUT_SCREEN_WIDTH )
584 jHeight=glutGet(GLUT_SCREEN_HEIGHT)
585
586 !--Define the .Bmp header:
587 BH%BM      = "BM"
588 BH%nBitsPP = 24
589 BH%nSizeH   = 40
590 BH%nReserv1 = 0
591 BH%nReserv2 = 0
592 BH%nOffBit  = 14 + BH%nSizeH
593                                     iHr3 = iwidth*3
594                                     iHr4  = ((iHr3+3)/4)*4
595     PixelRowPadBytes = iHr4-iHr3
596 BH%nSizeTot = BH%nOffBit+iHr4*jHeight
597 BH%nWidth   = iwidth
598 BH%nHeight  = jHeight
599 BH%nCompres = 0                      !added; unspecified but wasn't zero      2018.07.03
600 BH%nPlanes  = 1
601 BH%nSizeC   = 0
602 BH%XpPM     = 1000.*xwindowFull/xwindowMM      != .5296908e-41  2020.05.01
603 BH%YpPM     = 1000.*ywindowFull/ywindowMM      !
604 BH%nCtUsed  = 0                      ! added                                2018.07.03
605 BH%nCtImpo  = 0                      !
606
607 !--Bitmap header prints to the DOS screen. Toggle the 'escape' key.2018.07.03
608 write(us, "(/'Screen Selfie: bitmap header Specifics:')" )
609 write(us, "(54 Bytes Name      Value' )" )
610 write(us, "( 1: 2      BM      = 'a2                      )" )BH%BM
611 write(us, "( 3: 6      nSizeTot = ',i10,' bytes'          )" )BH%nSizeTot
612 write(us, "( 7: 8      nReserv1 = ',i10,' = 0 :not used'    )" )BH%nReserv1
613 write(us, "( 9:10     nReserv2 = ',i10,' = 0 :not used'    )" )BH%nReserv2
614 write(us, "(11:14     nOffBit  = ',i10,' =54 bytes'        )" )BH%nOffBit
615 write(us, "(15:18     nSizeH   = ',i10,' =40 bytes'        )" )BH%nSizeH
616 write(us, "(19:22     nWidth   = ',i10,' pixels'          )" )BH%nWidth
617 write(us, "(23:26     nHeight  = ',i10,' pixels'          )" )BH%nHeight
618 write(us, "(27:28     nPlanes  = ',i10,'                  )" )BH%nPlanes
619 write(us, "(29:30     nBitsPP  = ',i10,' bits-per-pixel'   )" )BH%nBitsPP
620 write(us, "(31:34     nCompres = ',i10,' = 0 :no compression'" )BH%nCompres
621 write(us, "(35:38     nSizeC   = ',i10,' = 0 :no color table'" )BH%nSizeC

```

```

621 write(Us,('33:30      isizeC = ',i10,' = 0 :no color table') )BH%isizeC
622 write(Us,('39:42      XpPM    = ',f12.1,' using =nwidth'   )")BH%XpPM
623 write(Us,('43:46      YpPM    = ',f12.1,' using =nHeight'  )")BH%YpPM
624 write(Us,('47:50      nClrUsed = ',i10,' = 0 :24bit color'  )")BH%nClrUsed
625 write(Us,('51:54      nClrImpo = ',i10,' = 0 :24bit color'  )")BH%nClrImpo
626
627 write(Us,("/'Writing:',a12,' size:',i9,' x ',i9)") Selfie,iwidth,jHeight
628
629 !--Allocate the one-row read & write image arrays:
630         i3W = 3*iwidth
631 allocate(fImage(i3W ),stat=iAlloc) !Read as floats
632         fImage = 0.
633 allocate(Image2(iHr4),stat=iAlloc) !Write as characters(+pads,if needed.)
634         Image2 = char(0)
635 !--Export a fairly-standard .bmp file:
636 open(unit=UsSelfie,File=Selfie,form='binary',action='write') !<works!
637 !open(unit=UsSelfie,File=Selfie,form='unformatted',action='write')
638 !         !!!!!!!!!!!!!!! doesn't work!
639         write(UsSelfie) BH
640
641         call glReadBuffer(GL_FRONT_LEFT) !Dump the left-front buffer.
642         do j=0,jHeight-1 ! A row at a time
643             jh0=j; iw0=0 ! Read the unpadded row.
644             call glReadPixels( iw0,jh0,iwidth,1,GL_RGB,GL_FLOAT,fImage)
645             call glFlush
646             do i=1,iwidth !RGB -> BGR, the .bmp's format
647                 k=i*3-2
648                 Image2(k )=char(int(fImage(k+2)*255.)) !Blue <- first
649                 Image2(k+1)=char(int(fImage(k+1)*255.))
650                 Image2(k+2)=char(int(fImage(k )*255.)) !Red -> last
651             enddo!i
652             write(UsSelfie) Image2 !Write the row (+pads).
653         enddo!j
654         close(UsSelfie)
655
656         deallocate(fImage)
657         deallocate(Image2)
658         write(Us,("a11,' - has been exported to your working directory.',a1)") &
659                                     char(7) ;return
660 End Subroutine ScreenSelfie
661 !-----7 9
662
663 Subroutine ShowProjectAndModel(nRowIn,nColIn,iColor,Label20)
664 !2021.10.04.1530cdt JMS- Column ordering is being used here!
665 !2021.10.04.1425cdt JMS- Added FrustNom(0:8)
666 !2020.06.29.0805cdt JMS- OpenGL's Projection & Modelview Matrices - Live.
667 ! - Traveler2/Athlon64/winXPro/APF9.0Fortran/OpenGL+Cglut
668 ! Note: Expected Depth = Returned Depth*2.-1.
669 !--Globals S1ModDef.f95: 2024.03.18
670 use OpenGLRec,only: & !Ref: OpenGL GL/GLU/GLUT docs
671         glGetDoublev,GL_PROJECTION_MATRIX,GL_MODELVIEW_MATRIX,glFlush
672 use ioDef ,only: NowView !Input/Output 2020.04.11
673 use ScreenDef ,only: nCharMaxX,nCharMaxY,Side !screen & colors
674 use ViewDef ,only: FrustNom & !(0:8) = Z,D,E,N,F,L,R,T,B <-Nominal values
675                                     ,UnitCube0 & ! <-actual values
676                                     ,Key
677 !--End Globals
678 implicit none
679 !--Arguments
680 integer(4)::nRowIn,nColIn
681 integer(4)::iColor ! Text color on screen
682 character::Label20*20

```

```

683 ! integer(4)::iP                                ! >5: Printing enable, use: write(iP,...)
684 !--Internals
685 integer(4)::nRow,nCol
686 real(8)    ::ModelView(16)                      !Modelview Matrix
687 real(8)    ::Projection(16)                     !Projection Matrix
688 real(8)    ::H(4,4)
689 character::PText*80
690 integer(4)::i
691 !--EndDefs-----
692 !--Get the matrices from OpenGL:
693 call glGetDoublev(GL_PROJECTION_MATRIX,Projection); call glFlush
694 call glGetDoublev(GL_MODELVIEW_MATRIX ,ModelView ); call glFlush
695                                     if(NowView<1) return
696 !Offset down/right if>0      Up/left if<0  Text block below has
697 nRow=NrowIn; if(nRow<0) nRow=nCharMaxY+nRow+1 !12 rows &
698 nCol=nColIn; if(nCol<0) nCol=nCharMaxx+nCol+1 !48 columns
699 write(Ptext,"('ShowProjectModel - from:',1x,a20)") Label20
700                                     nRow=nRow+1; call PrntOrtho(nRow,nCol, 1, 0,PText)
701 write(Ptext,"('PROJECTION Matrix (Live):      iEye=',i2)") Key%iEye
702                                     nRow=nRow+1; call PrntOrtho(nRow,nCol,iColor, 0,PText)
703 call GLv16toHgl(Projection,H)
704 !do i=1,4; write(Ptext,"(4f12.6)") H(i,1:4)
705 do i=1,4; write(Ptext,"(4f12.6)") H(1:4,i)          !Column ordering 2021.10.04
706                                     nRow=nRow+1; call PrntOrtho(nRow,nCol, 1, 0,PText)
707 enddo!i
708 write(Ptext,"('MODELVIEW Matrix (Live):')")
709                                     nRow=nRow+1; call PrntOrtho(nRow,nCol,iColor, 0,PText)
710 call GLv16toHgl(Modelview,H)
711 !do i=1,4; write(Ptext,"(4f12.6)") H(i,1:4)
712 do i=1,4; write(Ptext,"(4f12.6)") H(1:4,i)          !Column ordering 2021.10.04
713                                     nRow=nRow+1; call PrntOrtho(nRow,nCol, 1, 0,PText)
714 enddo
715 write(Ptext,"('UnitCube0(0:8) = Z,D,E,N,F,L,R,T,B:')")
716                                     nRow=nRow+1; call PrntOrtho(nRow,nCol,iColor, 0,PText)
717 write(Ptext,"(      5f12.6,' :Z,D,N,L,T')") &
718 UnitCube0(0:1),UnitCube0(3),UnitCube0(5),UnitCube0(7)
719                                     nRow=nRow+1; call PrntOrtho(nRow,nCol, 1, 0,PText)
720 write(Ptext,"(12x,4f12.6,' : E,F,R,B')") &
721 UnitCube0(2),UnitCube0(4),UnitCube0(6),UnitCube0(8)
722                                     nRow=nRow+1; call PrntOrtho(nRow,nCol, 1, 0,PText)
723 write(Ptext,"('""P"" reports far more details.')")
724                                     nRow=nRow+1; call PrntOrtho(nRow,nCol,iColor, 0,PText)
725                                     return
726 End Subroutine ShowProjectAndModel
727 !Example use: Printing at lower right corner of screen.
728 ! call ShowProjectAndModel(-12,-48, 6,'S_Basics UserV L167 ')          !Diagnostic
729 !-----7 9
730
731 Subroutine HfsToHgl(Hfs,Hgl)
732 !2020.05.09.1250cdt JMS- Projection transform- encode to OpenGL's screen.
733 implicit none          ! Hgl(4,4) <-- Hfs(4,4) transform.
734 !                      ^^                      ^^standard flight simulation coordinates
735 !--Arguments
736 real(8)    ::Hfs(4,4) !A homogeneous projection matrix in: Flight Sim. coords.
737 real(8)    ::Hgl(4,4) !                      : OpenGL      coords.
738 !--Internal
739 real(8)    ::H(4,4)
740 !--EndDefs-----
741 H=Hfs          ! GL      FS
742 Hgl(3,1:4)= H(1,1:4) ! +Z <-- +X
743 Hgl(1,1:4)= H(2,1:4) ! +X <-- +Y
744 Hgl(2,1:4)=-H(3,1:4) ! +Y <-- -Z
745 Hgl(4,1:4)=-H(4,1:4) ! w <-- w :the homogeneous 4th row

```

```

743     Hgl(4,1:4) = H(4,1:4) ! +w <-- +w : the homogeneous 4th row
746     return
747 End Subroutine HfsToHgl
748 !-----7-9
749
750 Subroutine HglToHfs(Hgl,Hfs)
751 !2020.05.09.1250cdt JMS- Projection transform- decode from OpenGL's screen.
752 implicit none ! Hfs(4,4) <-- Hgl(4,4) transform.
753 !--Arguments
754 real(8) :: Hgl(4,4) ! A homogeneous projection transform in: OpenGL coords.
755 real(8) :: Hfs(4,4) ! : Flt. Sim. coords.
756 !--Internal
757 real(8) :: H(4,4)
758 !--EndDefs-----
759 H=Hgl ! FS GL
760 Hfs(2,1:4) = H(1,1:4) ! +Y <-- +X
761 Hfs(3,1:4) = -H(2,1:4) ! -Z <-- +Y
762 Hfs(1,1:4) = H(3,1:4) ! +X <-- +Z
763 Hfs(4,1:4) = H(4,1:4) ! +w <-- +w : the homogeneous 4th row
764 return
765 End Subroutine HglToHfs
766 !-----7-9
767
768 Subroutine GLv16toHgl(GLv,Hgl)
769 !2020.06.29.0810cdt JMS- From: OpenGL's serial 1D transform reporting.
770 implicit none ! To: H(4,4), still in OpenGL coordinates.
771 !--Argument & Result
772 real(8) :: GLv(16) ! (4,4) transform exported in 1D in row order.
773 real(8) :: Hgl(4,4) ! Corresponding matrix.
774 !--EndDefs-----
775 Hgl(1,1:4) = GLv( 1: 4)
776 Hgl(2,1:4) = GLv( 5: 8)
777 Hgl(3,1:4) = GLv( 9:12)
778 Hgl(4,1:4) = GLv(13:16) ;return
779 End Subroutine GLv16toHgl
780 !-----7-9
781
782 Subroutine BbFog
783 !2023.09.05.0700cdt JMS- Translucent sheets of fog
784 !--Globals
785 use ioDef ,only: Up,Ut !Files,Units,TimeStamp,Selfies,Flags
786 use ScreenDef ,only: ForceRGBA,ColorRGBA !<0 & r(4)
787 use KeyboardDef ,only: KbdKey
788 use MouseDef ,only: MChan,MChanX,MChanY !Mouse
789 use S2Callback ,only: CheckGL
790 !Use TubeDef
791
792 use OpenGLRec ,only: & !Ref: OpenGL GL/GLU/GLUT docs
793 glBegin ,glEnd ,glFlush &
794 glMatrixMode ,GL_MATRIX_MODE &
795 glPushMatrix ,glPopMatrix &
796 GL_MODELVIEW ,glLoadIdentity &
797 glGetIntegerv ,glPushMatrix ,glPopMatrix &
798 glPolygonMode ,GL_LINE ,GL_FILL &
799 ,GL_Front ,GL_BACK &
800 ,GL_QUADS ,glVertex3fv &
801 ,glBlendFunc ,GL_BLEND &
802 ,glEnable ,glDisable &
803 ,GL_SRC_ALPHA ,GL_ONE_MINUS_SRC_ALPHA &
804 ,GL_DST_ALPHA
805 !--End Globals
806 implicit none

```

```

807  !--Arguments
808  !--Internals
809  integer(4)::InitL,MtxMode(1),n
810  real(4)    ::Pc(3,4),S1=6.
811  !--EndDefs-----
812  if(InitL==0) then
813      Pc(1:3,1) = (/0.,-1., -1./)
814      Pc(1:3,2) = (/0.,+1., -1./)
815      Pc(1:3,3) = (/0.,-1., +1./)
816      Pc(1:3,4) = (/0.,+1., +1./)
817      InitL=1
818  endif!InitL=0
819
820  call glGetIntegerv(GL_MATRIX_MODE,MtxMode)
821  call glMatrixMode(GL_MODELVIEW ); call glPushMatrix; call glLoadIdentity
822
823  !Enable blending:
824  call glPolygonMode(GL_FRONT,GL_Fill)
825  call glPolygonMode(GL_BACK ,GL_Fill)
826  call glEnable(GL_BLEND)
827  call glBlendFunc(GL_SRC_ALPHA,GL_ONE_MINUS_SRC_ALPHA)
828  !call glBlendFunc(GL_SRC_ALPHA,GL_DST_ALPHA)
829
830  !Draw "planes of obscuration" i.e. Pseudo-Fog:
831  ForceRGBA = -1
832  ColorRGBA = (/ .0,.0,.0,.1 /)
833  ColorRGBA = (/ .0,.0,.0,.05 /)
834  call Colors3D(0)
835
836  call glBegin(GL_Quads)
837      do n = +8,-8,-2; Pc(1,1:4) = n/20.
838          call glVertex3fv(Pc(1:3,1)*S1)
839          call glVertex3fv(Pc(1:3,2)*S1)
840          call glVertex3fv(Pc(1:3,4)*S1)
841          call glVertex3fv(Pc(1:3,3)*S1)
842      enddo!n
843  call glEnd
844
845  call glDisable(GL_BLEND)
846  ForceRGBA = 0 ;call checkGL(-50813)
847
848  call glPolygonMode(GL_FRONT,GL_LINE)
849  call glPolygonMode(GL_BACK ,GL_LINE)
850  call glMatrixMode(GL_MODELVIEW ); call glPopMatrix
851  call glMatrixMode( MtxMode(1)) ;call glFlush
852                                     call checkGL(-50813)
853                                     return
854 End Subroutine BbFog
855 !-----7 9
856
857

```