

```

1  !S8Help.F95      Group ID: #8      Stereo-3D Simulation Environment Vsn:1.00
2  !2025.05.24.1840cdt- S8Help.txt file -> Screen
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 ^Mfgr.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 Boilerplate(iDetails)
24 ! Subroutine TextToScreen(HelpFileIn,nRowIn,nColIn,iColor,iP)
25
26 !-----7-9
27
28 Subroutine Boilerplate(iDetails)
29 !2025.04.18.1315cdt JMS- Displays key variables on the acreen
30 ! OpenGL Screen: +X:right,+Y:up,+Z:into screen (left-handed coordinates)
31 ! The screen has only proportions, with vert. & depth ranges [-1.,+1.]
32 !--Globals in S1ModDef.f95
33 use ioDef ,only: DaTimeLabel,NowView,iTeapot,Up,UpVerb !Input/Output
34 use TaskDef ,only: ExeBanner & !Project/Context
35 ,Analyst,Phone,Street,CityStateZip,IPSummary
36 use ScreenDef ,only: & !screen & colors
37 nCharMaxX,ctlKbd,xwindowFull,ywindowFull,lCharX,lCharY,nCharMaxXS,Side &
38 ,nCharMaxY,nCharCenXS
39 use MouseDef ,only: MouseScreenh & !Mouse
40 ,iMouseState,iMouseButton,MouseDirect,Mouse,MouseU,MChan,MChanX,MChanY &
41 ,iMouse7DoF,MouseDirectPrev,MXY,MXYdel
42 use KeyboardDef,only: & !Keyboard
43 KbdKey,KbdKeyIn,ArrowKey,ArrowKeyIn,iKeyMods
44 use ModelDef ,only: M7,Nnutate !Model7DoF
45 use ViewDef ,only: Key
46 use SimDef ,only: SimMode,RunTimer
47 use HelpDef ,only: HhHelpView
48 use AppsDef ,only: AppNumber,AppBanner
49 use BreakPtDef ,only: BrkOn !BreakPoint & scroll tracker
50
51 !--End Globals
52 implicit none
53 !--Arguments
54 integer(4)::iDetails
55 !--Internals
56 integer(4)::iP
57 integer(4)::nRow,nCol,iColor !,Pass
58 character(len=80)::PText
59 !--EndDefs-----
60 iP=Up !; if(UpVerb==0) iP=0
61
62 !--Introductory on-screen information begins here:

```

```

63  !--At upper left of screen:
64  if(iP>5) write(iP,"('Boilerplate: nCharMaxXS=',3i5,' side=',i2))" &
65  nCharMaxXS , Side
66  if(NowView/=1) return
67  nRow = 0; nCol = 1
68  write(PText,"(a58)") ExeBanner !White
69  nRow=nRow+1; call PrntOrtho(nRow, 5, 1, 0,PText)
70  write(PText,"('F',i1,': ',a58)") AppNumber,AppBanner(AppNumber)!White
71  nRow=nRow+1; call PrntOrtho(nRow, 1, 1, 0,PText)
72  !--At upper right of screen:
73  nRow = 0; nCol=nCharMaxXS(Side)-22
74  call Datime18
75  write(PText,"(a18)") DaTimeLabel !White
76  nRow=nRow+1; call PrntOrtho(nRow,nCol, 1, 0,PText)
77  !write(PText,"('AppNumber = F',i1.1)") S.AppNumber !Yellow
78  ! nRow=nRow+1; call PrntOrtho(nRow,nCol, 9, 0,PText)
79  iColor=1 !White
80  if(NowView<1) return !2021.10.04
81  select case(SimMode)
82  case(0); write(PText,"('Sim Timer uses F9-F12')");iColor=2 !LWhite
83  case(1); write(PText,"('Reset Timer=',f8.2)") RunTimer
84  case(2); write(PText,"('Hold Timer=',f8.2)") RunTimer
85  case(3); write(PText,"('Run Timer=',f8.2)") RunTimer
86  case(4); write(PText,"('Stop Timer=',f8.2)") RunTimer
87  case(5); write(PText,"('Drop Timer=',f8.2)") RunTimer
88  end select!(S.SimMode)
89  nRow=nRow+1; call PrntOrtho(nRow,nCol,iColor, 0,PText)
90  write(PText,"('Qq^Q:->QUIT Hh:Help')")
91  write(PText,"('Hh:Help=', ' Qq^Q:->QUIT')") !Cyan
92  nRow=nRow+1; call PrntOrtho(nRow,nCol,12, 0,PText)
93  select case(Key%SplitScreen)
94  case(-1); write(PText,"('E:-1 Right|L')") !-1
95  case( 0); write(PText,"('E: 0 FullScr')") ! 0
96  case(+1); write(PText,"('E:+1 Left|Rt ')")!+1
97  end select!(Key.Split) !White
98  nRow=nRow+1; call PrntOrtho(nRow,nCol, 1, 0,PText)
99  select case(Key%VuMode)
100 case(-1); write(PText,"('e:-1 raw OpenGL')")!-1
101 case( 0); write(PText,"('e:0 Ortho')") !0
102 case( 1); write(PText,"('e:2 Persp')") !1
103 case( 2); write(PText,"('e:3 3D')") !2
104 end select!(Key.VuMode) !White
105 call PrntOrtho(nRow,nCol+13, 1, 0,PText)
106 write(PText,"('~ :Screen Selfie->.bmp')") !Cyan
107 nRow=nRow+1; call PrntOrtho(nRow,nCol,12, 0,PText)
108 write(PText,"('` :Depth Selfie')") !Cyan
109 nRow=nRow+1; call PrntOrtho(nRow,nCol,12, 0,PText)
110 !write(PText,"('on')")
111 ! if(S.DepthSelfie>0) call PrntOrtho(nRow,nCol+17, 2, 0,PText)
112
113 write(PText,"('Escape:toggles screen')") !Cyan
114 nRow=nRow+1; call PrntOrtho(nRow,nCol,12, 0,PText)
115 if(iDetails<2) return
116
117 if(UpVerb==2) return !2021.10.04:
118 !--Author info:
119 !!write(PText,"('Email ServerName')") !LRed
120 !! nRow=nRow+1; call PrntOrtho(nRow,nCol, 6, 0,PText)
121 !!write(PText,"('@')"); call PrntOrtho(nRow,nCol+ 5,10, 0,PText)
122 !!write(PText,"('.com')"); call PrntOrtho(nRow,nCol+16,10, 0,PText)
123 !!write(PText,"(a14)") S.Phone !White
124 !! nRow=nRow+1; call PrntOrtho(nRow,nCol+ 2, 1, 0,PText)
125 if(iDetails<2) return

```

```

126  !--KeyPress info:
127      nRow=nRow+1
128      write(Ptext, "('ctl-K redirection: ')" ) !LWhite
129      if(ctlKbd==1) call PrntOrtho(nRow,nCol, 2, 0,PText)
130      write(Ptext, "('keyChar = ',i5,1x,a1)" ) kbdKey,kbdKeyIn,char(kbdKeyIn)
131      nRow=nRow+1; call PrntOrtho(nRow,nCol, 2, 0,PText)
132      write(Ptext, "('ArrowKey= ',i5,1x,a1)" ) ArrowKey,ArrowKeyIn
133      nRow=nRow+1; call PrntOrtho(nRow,nCol, 2, 0,PText)
134      write(Ptext, "('ikeyMods= ',i5, ' s, ^, Alt' )" ) iKeyMods
135      nRow=nRow+1; call PrntOrtho(nRow,nCol, 2, 0,PText)
136      if(iDetails<4)return
137      write(Ptext, "('Tt: iTeapot = ',i5)" ) iTeapot !Magenta
138      nRow=nRow+1; call PrntOrtho(nRow,nCol, 8, 0,PText)
139      write(Ptext, "('Bb: BrkOn = on' )" ); iColor= 4; if(BrkOn>0) iColor=1
140      nRow=nRow+1; call PrntOrtho(nRow,nCol,iColor, 0,PText)
141      write(Ptext, "('Nn: Nutation= on' )" ); iColor= 4; if(Nnutate>0)iColor=1
142      nRow=nRow+1; call PrntOrtho(nRow,nCol,iColor, 0,PText)
143      write(Ptext, "(' (X,Y,Depth) GL-screen: ')" ) !LWhite
144      nRow=nRow+1; call PrntOrtho(nRow,nCol, 2, 0,PText)
145      write(Ptext, "(3f6.3)" )
146      ,MouseScreenh(1) &
147      ,MouseScreenh(2) &
148      ,MouseScreenh(3) !LWhite
149      nRow=nRow+1; call PrntOrtho(nRow,nCol, 2, 0,PText)
150      nRow=nRow+1
151      if(iDetails<5)return
152  !--Current mouse location info:
153  !      nRow =4; nCol=1
154      write(Ptext, "(' State= ',i2)" ) iMouseState !LGreen
155      nRow=nRow+1; call PrntOrtho(nRow,nCol,10, 0,PText)
156      write(Ptext, "(' Button= ',i2)" ) iMouseButton !LGreen
157      nRow=nRow+1; call PrntOrtho(nRow,nCol,10, 0,PText)
158      write(Ptext, "(' iMouse7DoF= ',i2)" ) iMouse7DoF !LGreen
159      nRow=nRow+1; call PrntOrtho(nRow,nCol,10, 0,PText)
160      write(Ptext, "('MouseDirect= ',i2,1x,a6)" ) MouseDirect,MouseU%ID !LGreen
161      nRow=nRow+1; call PrntOrtho(nRow,nCol,10, 0,PText)
162      write(Ptext, "(' ',i2, ' ^prev' )" ) MouseDirectPrev !LWhite
163      nRow=nRow+1; call PrntOrtho(nRow,nCol, 2, 0,PText)
164      write(Ptext, "('DoF`s ',a6,3x,a6)" ) MchanX%ID,MchanY%ID !LWhite
165      nRow=nRow+1; call PrntOrtho(nRow,nCol, 1, 0,PText)
166      write(Ptext, "('Mchan #',i2.2, ' & #',i2.2)" ) MchanX%mc,MchanY%mc
167      nRow=nRow+1; call PrntOrtho(nRow,nCol, 2, 0,PText)
168      write(Ptext, "('Pix Hor & Ver' )" ) !LGreen
169      nRow=nRow+1; call PrntOrtho(nRow,nCol,10, 0,PText)
170      write(Ptext, "(' max ',i5,2x,i7)" ) xwindowFull,ywindowFull
171      nRow=nRow+1; call PrntOrtho(nRow,nCol, 2, 0,PText)
172      write(Ptext, "('Mouse ',i5,4x,i5)" ) MchanX%Pp(1)%iPix,MchanY%Pp(1)%iPix
173      nRow=nRow+1; call PrntOrtho(nRow,nCol,10, 0,PText)
174      write(Ptext, "(' .r01 ',f6.3,f9.3)" ) MchanX%r01,MchanY%r01 !LWhite
175      nRow=nRow+1; call PrntOrtho(nRow,nCol, 2, 0,PText)
176      write(Ptext, "(' .rm11 ',f6.3,f9.3)" ) MchanX%rm11,MchanY%rm11
177      nRow=nRow+1; call PrntOrtho(nRow,nCol, 2, 0,PText)
178      write(Ptext, "(' .v ',f6.3,f9.3)" ) MchanX%v,MchanY%v !LWhite
179      nRow=nRow+1; call PrntOrtho(nRow,nCol, 2, 0,PText)
180      write(Ptext, "('Txt Col & Row' )" ) !LGreen
181      nRow=nRow+1; call PrntOrtho(nRow,nCol,10, 0,PText)
182      write(Ptext, "(' max ',i5,2x,i7)" ) nCharMaxX,nCharMaxY !Green
183      nRow=nRow+1; call PrntOrtho(nRow,nCol, 2, 0,PText)
184      write(Ptext, "(' -> ',i5,4x,i5)" ) MchanX%Pp(1)%nChar,MchanY%Pp(1)%nChar
185      nRow=nRow+1; call PrntOrtho(nRow,nCol,10, 0,PText)
186      write(Ptext, "(' <- ',i5,4x,i5)" ) MouseU%ix/lcharX -nCharMaxX &

```

```

187                                     ,MouseU%iY/lCharY-1-nCharMaxY
188                                     nRow=nRow+1; call PrntOrtho(nRow,nCol,10, 0,PText)
189 ! write(Ptext,"(4x,'>',a6,2x,'^',a6)") MChan(3).ID,MChan(4).ID !White
190 !                                     nRow=nRow+2; call PrntOrtho(nRow,nCol, 1, 0,PText)
191 write(Ptext,"('Mouse history:      Iter#')") !LWhite
192                                     nRow=nRow+2; call PrntOrtho(nRow,nCol-4, 2, 0,PText)
193 write(Ptext,"('MXYdel=',2i5,i3,i6)") MXYdel !LLWhite
194                                     nRow=nRow+1; call PrntOrtho(nRow,nCol-4, 3, 0,PText)
195 write(Ptext,"('MXY(1)=' ,2i5,i3,i6)") MXY(1:4,1)
196                                     nRow=nRow+1; call PrntOrtho(nRow,nCol-4, 3, 0,PText)
197 write(Ptext,"('MXY(2)=' ,2i5,i3,i6)") MXY(1:4,2)
198                                     nRow=nRow+1; call PrntOrtho(nRow,nCol-4, 3, 0,PText)
199 write(Ptext,"('MXY(3)=' ,2i5,i3,i6)") MXY(1:4,3)
200                                     nRow=nRow+1; call PrntOrtho(nRow,nCol-4, 3, 0,PText)
201 !--Auxiliary mouse inputs for manual control of eight app. variables:
202 if(iMouse7DoF==2) then
203 write(Ptext,"('M` manual app. controls:')") !White
204                                     nRow=nRow+1; call PrntOrtho(nRow,nCol-10, 1, 0,PText)
205 write(Ptext,"('Mchan #',i2.2,' & #',i2.2)") MChanX%mc,MChanY%mc
206 write(Ptext,"('Mchan(13).rm11=',f7.4,' LMB-up-X')") MChan(13)%rm11
207                                     nRow=nRow+1; call PrntOrtho(nRow,nCol-10, 13, 0,PText)
208 write(Ptext,"('Mchan(14).rm11=',f7.4,' -Y')") MChan(14)%rm11
209                                     nRow=nRow+1; call PrntOrtho(nRow,nCol-10, 13, 0,PText)
210 write(Ptext,"('Mchan(15).rm11=',f7.4,' -dn-X')") MChan(15)%rm11
211                                     nRow=nRow+1; call PrntOrtho(nRow,nCol-10, 13, 0,PText)
212 write(Ptext,"('Mchan(16).rm11=',f7.4,' -Y')") MChan(16)%rm11
213                                     nRow=nRow+1; call PrntOrtho(nRow,nCol-10, 13, 0,PText)
214 write(Ptext,"('Mchan(21).rm11=',f7.4,' RMB-up-X')") MChan(21)%rm11 !LWhite
215                                     nRow=nRow+1; call PrntOrtho(nRow,nCol-10, 13, 0,PText)
216 write(Ptext,"('Mchan(22).rm11=',f7.4,' -Y')") MChan(22)%rm11
217                                     nRow=nRow+1; call PrntOrtho(nRow,nCol-10, 13, 0,PText)
218 write(Ptext,"('Mchan(23).rm11=',f7.4,' -dn-X')") MChan(23)%rm11
219                                     nRow=nRow+1; call PrntOrtho(nRow,nCol-10, 13, 0,PText)
220 write(Ptext,"('Mchan(24).rm11=',f7.4,' -Y')") MChan(24)%rm11
221                                     nRow=nRow+1; call PrntOrtho(nRow,nCol-10, 13, 0,PText)
222 endif!iMouse7DoF==2
223 !                                     if(iDetails<6)return
224 !--At upper center of screen - IP information:
225 !write(Ptext,"(a1,' 2024 ',a35)") char(169) & !<-Copyright symbol
226 !                                     ,Analyst; nCol =nCharCenXS(Side)
227 !write(Ptext,"(a1,' 2024 ',a35)") char(224) & !<-Free symbol (wrong font)
228 !                                     ,Analyst; nCol =nCharCenXS(Side)
229 write(Ptext,"(1x, 6x ,a35)") Analyst(AppNumber) !<-Name only
230                                     nCol =nCharCenXS(Side)
231                                     nRow=1 ; call PrntOrtho(nRow,nCol-17, 1, 0,PText)
232 write(Ptext,"(a35)") Street(AppNumber) !LWhite
233                                     nRow=nRow+1; call PrntOrtho(nRow,nCol-10, 2, 0,PText)
234 write(Ptext,"(a35)") CityStateZip(AppNumber)
235                                     nRow=nRow+1; call PrntOrtho(nRow,nCol-10, 2, 0,PText)
236 write(Ptext,"(a35)") Phone(AppNumber)
237                                     nRow=nRow+1; call PrntOrtho(nRow,nCol-10, 2, 0,PText)
238 write(Ptext,"(a35)") IPSummary(AppNumber) !Red
239                                     nRow=nRow+1; call PrntOrtho(nRow,nCol-10, 7, 0,PText)
240                                     return
241 End Subroutine Boilerplate
242 !-----7 9
243
244 Subroutine TextToScreen(nRowIn,nColIn,iColor,iP)
245 !2024.04.02.1340cdt JMS- Draws screen text, scrollable by '+' & '-':
246 !--Globals
247 use S2Callback,only:CheckGL !OpenGL CB's, GlutHandoff, & glGetError
248 use ioDef ,only: & !Files,Units,TimeStamp,Selfies,Flags
249 use Ut,only:Uread,Uwrite,Uselfie,UTimeLabel

```

```

249      or,os,ot,oread,owrite,osetitle,datimelabel
250  use ScreenDef ,only: nCharMaxXS,nCharMaxY,Side      !screen & colors
251  use KeyboardDef,only: kbdkey
252  use AppsDef ,only: AppNumber
253  use HelpDef ,only: &      !Help (a text block)
254      HhHelpView,nHelpLines,BufferHhHelpView
255  use BreakPtDef ,only: BrkOn !BreakPoint & scrolling tracker
256  !--End Globals
257  implicit none
258  !--Arguments
259  character::HName*80
260  character::HNamePrev*80
261  integer(4)::nRowIn,nColIn      !Text- screen upper-left character position
262  integer(4)::iColor            ! - color on screen
263  integer(4)::iP                ! >5: Printing enable, use: write(iP,...)
264  !--Internals
265  integer(4)::Init=0,i,j,Nfirst=1
266  logical ::LExists             !File existence flag
267  character::HelpText(0:200)*80
268  ! integer(4)::iAlloc           !Array allocation flag =0:success
269  integer(4)::nRow,nCol         !Internal row/column counter
270  character(len=80)::PText
271  integer(4)::HhPrev = 0
272  integer(4)::AppPrev = 0
273  integer(4)::iFound
274  !--EndDefs-----
275  !call CheckGL(46)
276  if(iColor<=0) return
277      HName = char(0)
278  select case(HhHelpView)
279  case(1);      HName = 'S8Help.txt' !'h'
280  case(2);      HName = 'S8Help.txt' !'h'
281  select case(AppNumber)
282  case( :1);      HName = 'S8Help.txt'
283  case(2:8); write(HName, "('S0-AppF',i1,'-Help.txt')") AppNumber
284  case(9: );
285  end select!AppNumber
286  end select!HhHelpView
287  if(HName==HNamePrev) then
288  if(iFound==0) return
289  !Use the current Helpfile:
290  goto 30
291  endif!(HName==HNamePrev)
292  !Attempt to import file HName...
293  Nfirst=1
294  HelpText(0)="`Bb` toggles `+` & `` scrolling of this text."
295  inquire(file=HName,exist=LExists)
296  if(LExists)then
297  open(unit=Ur,file=HName,action='read',err=27)
298  do i=1,200
299  read(Ur,"(a80)",err=27,end=28) HelpText(i)
300  HelpText(i)(80:80)=char(0)
301  write(Us,*) HelpText(i)
302  enddo
303  HhPrev = HhHelpView
304  close(Ur); iFound = 1; i = 200; HNamePrev = HName ;goto 30
305  27 close(Ur); iFound = 0 ;return
306  28 close(Ur); iFound = 1; i=i-1 ; HNamePrev = HName ;goto 30
307  else; iFound = 0 ;return
308  endif!LEExists
309  !--S8Help.txt is imported into HelpText()
310  30 continue

```

```

311  if(BrkOn>0) then
312    select case(kbdkey) !-----
313    !--Adjust Nfirst using the +- keys:
314    case(43,61); KbdKey=0; Nfirst=Nfirst+1      !: '+= '
315    if(Nfirst>i-20) Nfirst= i-20
316    case(45 ); KbdKey=0; Nfirst=Nfirst-1      !: '- '
317    if(Nfirst<1) Nfirst= 1
318    end select !kbdkey -----^
319  endif!BrkOn>0
320
321  !Offset down/right if>0      Up/left if<0      Text block below has
322  nRow=NrowIn !; if(nRow<0) nRow=nCharMaxY+nRow+1      !<=100 rows      -&-
323  nCol=nColIn !; if(nCol<0) nCol=nCharMaxX+nCol+1      !<= 80 columns
324  call CheckGL(+80283)
325  write(PText,"('Hh:Help=',i1)") HhHelpview      !Cyan
326  call PrntOrtho(nRow+1,nCol,12, 0,PText)
327  do j=0,i; if(j>200) exit
328  if((j>0).and.(j<Nfirst)) cycle
329  if((len_trim(HelpText(j))>0).and.(len_trim(HelpText(j))<=80)) &
330  call PrntOrtho(nRow,nCol,iColor,0,HelpText(j))
331  nRow=nRow+1
332  if(j==0) nRow=nRow+1
333  enddo
334  call CheckGL(-80291)      ;return
335  End Subroutine TextToScreen
336  !-----7 9
337
338

```