```
!S0-AppF1-stub.f95  Group ID: #11    Stereo-3D Simulation Environment Vsn:1.00
!2025.05.24.1840cdt- Application F1

!            Author- Jeffrey M. Setterholm, Lakeville,MN 55044 USA
!         IP Status- Free source code (e.g.: post copyright)
!
!          Computer- "T3"/Dell Precision T3500/Intel i5 E5520/Win10Pro-21H2
!                     ^name ^Mfgr.Id           ^chipset       ^OS
!                    /Absoft Pro Fortran 21.0.2/GeForce GTX 1050/f90gl~Glut3.7
!                     ^compiler ~Fortran 95     ^graphics card   ^graphics

!    f90gl bindings- public domain; see "https://math.nist.gov/f90gl/"

!Disclaimer:
! **********************************************************************
! **********   Individual cognition is always flawed,  **********
! **********         including yours and mine.         **********
! **********                  - So: -                  **********
! **********       Use this code at your own risk.     **********
! **********************************************************************

!Table of Contents: ...use to search...
! Subroutine S11AppF1
!-------------------------------------------------------------------------------7 9

Subroutine S11AppF1
!2025.02.09.1325cst JMS- An application stub.
 use TaskDef   ,only: Analyst,Phone,Street,CityStateZip,IPSummary
 use ioDef     ,only: Up,Ut,iTeapot
 use ScreenDef ,only: yWindowFull,xyWindowRatio,PixelsPerCM,MagFactor,nCharMaxY
 use MouseDef  ,only: MChan
 use AppsDef   ,only: AppBanner,AppInFile
 use ViewDef   ,only: ThreePhase,V,jVG
 use ModelDef  ,only: M7Init,M7,nM7L
 use SimDef    ,only: RunTimer,IterTotal
 use S2Callback,only: CheckGL
 use OpenGLRec,only:  &                                !Ref: OpenGL GL/GLU/GLUT docs
      glVertex3f   , glShadeModel, GL_SMOOTH   , GL_FLAT, glBegin  &
     ,GL_LINE_STRIP, GL_LINES    , glLineWidth ,glFlush , glEnd    &
     ,GL_POINTS    , glPointSize                       &
     ,glGetIntegerv,glMatrixMode,GL_MATRIX_MODE    &!used by vector-font labels
     ,GL_MODELVIEW,glPushMatrix,glPopMatrix,glFlush &
     ,glMultMatrixd,glLoadIdentity

!--End Globals
 implicit none
 !--Arguments
 !--Internals
  integer(4)::ModeFlag=0
  integer(4)::iCol=0
  integer(4)::nRow,nCol
  character(len=80)::PText
  integer(4)::Init,i

! Font draw:
  character:: cLabelL*80 !Label- text
  real(8)   :: PosLLCq(3) !     - position of the lower left corner (not homog.)
  real(8)   :: RpyDq(3)   !     - Roll,pitch, & yaw of the label    (not homog.)
  real(8)   :: SizeHq     !     - size
  integer(4):: iColor     !     - color
  real(4)   :: fLineWidth !     - line width

```

```
63    integer(4)::MtxMode(1)
64    real(8)   ::XYZRPYwC(7) !(X,Y,Z,Roll,Pitch,Yaw,height of Character)
65
66    real(8)   ::Var(9)      !Mouse-controlled real variables
67
68  ! integer(4):: iP       !Print flag. Prints when iP>5 to unit# IP.
69
70    !--EndDefs-----------------------------------------------
71  !  AppNumber    = 1
72     AppBanner(   1) ='S11AppF1: Application Stub - for user use.'
73     AppInFile(   1) = char(0)
74     Analyst(     1) = '    Jeff Setterholm'
75     Street(      1) = '    8095 230th St.E.'
76     CityStateZip(1) = 'Lakeville, MN 55044 USA'
77     Phone(       1) = '     (nnn)-nnn-nnn'
78     IPsummary(   1) = '         Free'
79
80                                                 !a35
81     if((ThreePhase==1).and.(Up>5)) &
82       write(Up,"('F1:TP',i1,'Up',i2)") ThreePhase,Up
83     select case(ThreePhase)
84      case(1)  !Background number crunching, if any:
85                                                        call CheckGL(+110055)
86       !--Establish screen default 7DoF, if desired:
87       if(Init==0) then
88         Init=1
89       endif!Init=0
90
91       if(Up>5) then !This section of code only runs when you request a printout.
92         !--In simple first use, call your numeric subroutine from here.
93         !   & press "p" for a live DOS screen printout
94         !       or "P" for printout to a file (e.g. when you set Up = 13).
95         write(uP,"(/'S11AppF1 - ...TBD:')")
96         write(uP,"(  f20.12,' = RunTimer'               )") RunTimer
97
98       endif!uP>5
99
100      if(RunTimer==0.d0) then
101       if(nM7L==2) then
102       !Mouse-controlled `live` variables example:
103        Var(1) = +Mchan(13)%rm11*xyWindowRatio*MagFactor/M7(0)%DoF(7)!LMB-Up  -X
104        Var(2) = -Mchan(14)%rm11              *MagFactor/M7(0)%DoF(7)!        -Y
105        Var(3) = +Mchan(15)%rm11*xyWindowRatio*MagFactor/M7(0)%DoF(7)!   -down-X
106        Var(4) = -Mchan(16)%rm11              *MagFactor/M7(0)%DoF(7)!        -Y
107        Var(5) = +Mchan(21)%rm11*xyWindowRatio*MagFactor/M7(0)%DoF(7)!RMB-Up  -X
108        Var(6) = -Mchan(22)%rm11              *MagFactor/M7(0)%DoF(7)!        -Y
109        Var(7) = +Mchan(23)%rm11*xyWindowRatio*MagFactor/M7(0)%DoF(7)!   -down-X
110        Var(8) = -Mchan(24)%rm11              *MagFactor/M7(0)%DoF(7)!        -Y
111       endif!nM7L=2
112      else
113       !Clock-controlled variables example:
114        Var(1) = +1.0d0*dsin(RunTimer)
115        Var(2) = +1.0d0*dcos(RunTimer)
116      endif!RunTimer==0
117
118     case(2)  !Update variables & 2D screen info: Superceded by split screen(s)
119
120     case(3)  !Draw/redraw app. 2D & 3D graphics:   (ThreePhase==3)
121
122     !Draw stationary vector font information:
123      call glGetIntegerv(GL_MATRIX_MODE,MtxMode)
124      call glMatrixMode(GL_MODELVIEW ); call glPushMatrix; call glLoadIdentity
125
```

```
123
126         cLabelL    = "Press Left-Mouse-Button LMB & move the mouse."
127         XYZRPYwC   = (/ 0.d0, -5.d0, -5.0d0, 0.d0, 0.d0,90.d0, .2d0/)
128         call VecFont7(XYZRPYwC,1.,1,cLabelL)
129
130         cLabelL    = "         Press F2 to see the Dodecaheron app."
131         XYZRPYwC   = (/ 0.d0, -5.d0, -4.5d0, 0.d0, 0.d0,90.d0, .2d0/)
132         call VecFont7(XYZRPYwC,1.,1,cLabelL)
133
134       call glMatrixMode(GL_MODELVIEW ); call glPopMatrix   ;call glFlush
135       call glMatrixMode(  MtxMode(1))                      ;call glFlush
136
137      !Show Mouse/Clock variable values
138                                  nRow = 3;                    nCol = 2
139         write(Ptext,"('Mouse & timer variables:  (set using ""M"")')")
140                                 nRow=nRow+1; call PrntOrtho(nRow,nCol, 2, 0,PText)
141         write(Ptext,"('Var(1) = ',f10.6,' LMB-Up  -X' )") Var(1)
142                                 nRow=nRow+1; call PrntOrtho(nRow,nCol,13, 0,PText)
143         write(Ptext,"('Var(2) = ',f10.6,'    -    -y' )") Var(2)
144                                 nRow=nRow+1; call PrntOrtho(nRow,nCol,13, 0,PText)
145         write(Ptext,"('Var(3) = ',f10.6,'    -down-X' )") Var(3)
146                                 nRow=nRow+1; call PrntOrtho(nRow,nCol,13, 0,PText)
147         write(Ptext,"('Var(4) = ',f10.6,'    -    -y' )") Var(4)
148                                 nRow=nRow+1; call PrntOrtho(nRow,nCol,13, 0,PText)
149         write(Ptext,"('Var(5) = ',f10.6,' RMB-Up  -X' )") Var(5)
150                                 nRow=nRow+1; call PrntOrtho(nRow,nCol,13, 0,PText)
151         write(Ptext,"('Var(6) = ',f10.6,'    -    -y' )") Var(6)
152                                 nRow=nRow+1; call PrntOrtho(nRow,nCol,13, 0,PText)
153         write(Ptext,"('Var(7) = ',f10.6,'    -down-X' )") Var(7)
154                                 nRow=nRow+1; call PrntOrtho(nRow,nCol,13, 0,PText)
155         write(Ptext,"('Var(8) = ',f10.6,'    -    -y' )") Var(8)
156                                 nRow=nRow+1; call PrntOrtho(nRow,nCol,13, 0,PText)
157         write(Ptext,"('Var( ) values: proportional to zoomed screen')")
158                                 nRow=nRow+1; call PrntOrtho(nRow,nCol,13, 0,PText)
159         write(Ptext,"('.rm11  values: [-1.0,1.0]')")
160                                 nRow=nRow+1; call PrntOrtho(nRow,nCol,13, 0,PText)
161
162
163      !stationary font information example:
164                                 nRow = nCharMaxY-23;             nCol = 2
165       Ptext = "Type `Q` or `q` to quit."
166                                 nRow=nRow+2; call PrntOrtho(nRow,nCol,7, 0,PText)
167       if(mod(IterTotal,100)==10)                                    goto 20
168       Ptext = "Press Left-Mouse-Button LMB & move the mouse."
169                                 nRow=nRow+2; call PrntOrtho(nRow,nCol,1, 0,PText)
170       Ptext = "  ...default is Pitch & Yaw control."
171                                 nRow=nRow+1; call PrntOrtho(nRow,nCol,1, 0,PText)
172       Ptext = "Toggle H for help; it scrolls."
173                                nRow=nRow+2; call PrntOrtho(nRow,nCol,1, 0,PText)
174       Ptext = "Lower-right-screen controls `Point of Interest' "
175                                 nRow=nRow+2; call PrntOrtho(nRow,nCol,1, 0,PText)
176       Ptext = "  ...LMB clicks -> mouse-control of six axes & Mag."
177                                 nRow=nRow+1; call PrntOrtho(nRow,nCol,1, 0,PText)
178       Ptext = "~ captures full resolution (<=4k) .bmp screen selfies;"
179                                 nRow=nRow+2; call PrntOrtho(nRow,nCol,1, 0,PText)
180       Ptext = "   ...to save ink on hardcopies - press C."
181                                 nRow=nRow+1; call PrntOrtho(nRow,nCol,1, 0,PText)
182       Ptext = "T changes the Teapot. N nutates, e.g.: for monocular people."
183                                nRow=nRow+2; call PrntOrtho(nRow,nCol,1, 0,PText)
184       Ptext = "Press e twice for red|cyan 3D; then:"
185                                 nRow=nRow+2; call PrntOrtho(nRow,nCol,1, 0,PText)
186       Ptext = "  ...press E once for split-screen 3D for Head-Mounted Displays."
```

```
                                nRow=nRow+1; call PrntOrtho(nRow,nCol,1, 0,PText)
      Ptext = "  ...press E again for crossed-eye 3D."
                                nRow=nRow+1; call PrntOrtho(nRow,nCol,1, 0,PText)
      Ptext = "Function keys F1 - f8 change apps."
                                nRow=nRow+2; call PrntOrtho(nRow,nCol,1, 0,PText)
20    continue

     !Dynamic vector font information:
      cLabelL    = "Labelling with AlphaJS() & Jeff`s font."
      PosLLCq    = (/-1.d0, 1.d0, 0.d0/)
      RpyDq      = (/ 0.d0, 0.d0, 0.d0/)
      SizeHq     =     .1d0
      iColor     =    6
      fLineWidth =    1.
      call AlphaJS(cLabelL,PosLLCq,RpyDq,SizeHq,iColor,fLineWidth, 0) !,Up)

      cLabelL    = "Labelling with VecFont7() & GLUT_STROKE_MONO_ROMAN."
      XYZRPYwC   = (/-1.d0,-1.d0, 0.d0, 0.d0, 0.d0, 0.d0, .1d0/)
      call VecFont7(XYZRPYwC,1.,8,cLabelL)

     !Draw the "Free" vector character char(224) in 3D
      cLabelL    = char(224)//"=Free"
      PosLLCq    = (/ 0.d0,-1.d0,-2.1d0/)
      RpyDq      = (/ 0.d0, 0.d0, 90.d0/)
      SizeHq     =     .33d0
      iColor     =    1
      fLineWidth =    1.
     call AlphaJS(cLabelL,PosLLCq,RpyDq,SizeHq,iColor,fLineWidth, 0) !,Up)
      cLabelL    = "My Fortran 95 3D-simulation source code is Free."
      PosLLCq    = (/ 0.d0,-3.3d0,-1.8d0/)
      RpyDq      = (/ 0.d0, 0.d0, 90.d0/)
      SizeHq     =     .15d0
      iColor     =    1
      fLineWidth =    1.
     call AlphaJS(cLabelL,PosLLCq,RpyDq,SizeHq,iColor,fLineWidth, 0) !,Up)
      cLabelL    = "f90gl graphics IP: see https://math.nist.gov/f90gl/"
      PosLLCq    = (/ 0.d0,-3.4d0,-1.5d0/)
      RpyDq      = (/ 0.d0, 0.d0, 90.d0/)
      SizeHq     =     .15d0
      iColor     =    1
      fLineWidth =    1.
     call AlphaJS(cLabelL,PosLLCq,RpyDq,SizeHq,iColor,fLineWidth, 0) !,Up)
      cLabelL    = "No warranties are expressed or implied."
      PosLLCq    = (/ 0.d0,-2.6d0,-1.2d0/)
      RpyDq      = (/ 0.d0, 0.d0, 90.d0/)
      SizeHq     =     .15d0
      iColor     =    7
      fLineWidth =    1.
     call AlphaJS(cLabelL,PosLLCq,RpyDq,SizeHq,iColor,fLineWidth, 0) !,Up)

                                                          call CheckGL(-110211)
    end select!ThreePhase
                                                                        return
End Subroutine S11AppF1
!-------------------------------------------------------------------------------7 9


```