

```

1  !S2ModCallbacks.f95   Group ID: #2   Stereo-3D Simulation Environment Vsn:1.00
2  !2025.05.24.1840cdt- OpenGL Callback subroutines.  Why this way? On my computer
3  !                      GlutHandoff finds its callbacks more reliably.
4  !                      - S2ModCallbacks is a pass-through to S4Callbacks.
5
6  !                      Author- Jeffrey M. Setterholm, Lakeville,MN 55044 USA
7  !                      IP Status- Free source code (e.g.: post copyright)
8  !
9  !                      Computer- "T3"/Dell Precision T3500/Intel i5 E5520/win10Pro-21H2
10 !                      ^name ^Mfgr.Id          ^chipset          ^OS
11 !                      /Absoft Pro Fortran 21.0.2/GeForce GTX 1050/f90gl~Glut3.7
12 !                      ^compiler ~Fortran 95      ^graphics card  ^graphics
13
14 !                      f90gl bindings- public domain; see "https://math.nist.gov/f90gl/"
15
16 !Disclaimer:
17 !*****
18 ! *****      Individual cognition is always flawed,      *****
19 ! *****      including yours and mine.                    *****
20 ! *****      - So: -                                       *****
21 ! *****      Use this code at your own risk.              *****
22 ! *****
23
24 !Table of Contents: ...use to search...
25 !Module S2Callback
26 ! Subroutine cbUserView                                !Called by: OpenGL
27 ! Subroutine cbKeyboard(Key,xCursor,yCursor)           !           :
28 ! Subroutine cbSpecialFunctionKeys(Key,xCursor,yCursor) !           :
29 ! Subroutine cbMouseMotion(ix,iY)  Active & Passive    !           :
30 ! Subroutine cbMouseButtons(Button,State,ix,iY)        !           :
31 ! Subroutine GlutHandoff                                !           : S3Main.f95
32
33 ! Subroutine CheckGL(Line)                             !           ...once
34 !End Module S2Callback
35
36 !use S2Callback
37 !-----7 9
38
39 Module S2Callback
40
41   implicit none
42   !--
43   public ::      & !Passthroughs to S4Callbacks.f95 subroutines:
44     cbUserView   & ! UserView
45     ,cbKeyboard  & ! Keyboard
46     ,cbSpecialFunctionKeys & ! SpecialFunctionKeys
47     ,cbMouseMotion & ! MouseMotion & MousePassiveMotion
48     ,cbMouseButtons & ! MouseButtons
49     , GlutHandoff & !<-- Not a Passthrough, the actual code.
50     ,CheckGL     & !For use in isolating|debugging OpenGL calls.
51
52   !use S2Callback      !OpenGL Callback subroutines
53   !--
54   contains
55   !-----7 9
56
57   Subroutine cbUserView
58   !2020.04.02.1300cdt JMS- callBack- User(i.e.your) View generator:
59   implicit none
60   !--
61   call CheckGL(+20039)
62   call UserView

```

```

63  call CheckGL(-20041)                                     ;return
64  End Subroutine cbUserView
65  !-----7 9
66
67  Subroutine cbKeyboard(aKey,xCursor,yCursor)
68  !2020.04.02.1300cdt JMS- Callback- keyboard:
69  use OpenGLRec,only: & !Ref: OpenGL GL/GLU/GLUT docs
70      glubyte,glcint
71  !--
72  implicit none                                           !Arguments
73  character(kind=glubyte),intent(inout)::aKey
74  integer(kind=glcint) ,intent(inout)::xCursor
75  integer(kind=glcint) ,intent(inout)::yCursor
76  !--                                                     !EndArgs
77  call CheckGL(+20055)
78      call Keyboard(aKey,xCursor,yCursor)
79  call CheckGL(-20057)                                     ;return
80  End Subroutine cbKeyboard
81  !-----7 9
82
83  Subroutine cbSpecialFunctionKeys(Key,xCursor,yCursor)
84  !2020.04.02.1300cdt JMS- Callback- special function keys (F1-F12, arrows, etc.)
85  ! - F9...F12 -> Reset,Hold,Operate,& Stop the timer.
86  use OpenGLRec,only: & !Ref: OpenGL GL/GLU/GLUT docs    2020.04.05
87      glubyte,glcint
88  !--
89  implicit none                                           !Arguments
90  integer(kind=glcint),intent(inout)::Key,xCursor,yCursor
91  !--                                                     !EndArgs
92  call CheckGL(+20070)
93      call SpecialFunctionKeys(Key,xCursor,yCursor)
94  call CheckGL(-20072)                                     ;return
95  End Subroutine cbSpecialFunctionKeys
96  !-----7 9
97
98  Subroutine cbMouseMotion(iX,iY)
99  !2020.04.02.1300cdt JMS- Callback- mouse motion:
100 use OpenGLRec,only: glcint!Ref: OpenGL GL/GLU/GLUT docs
101 !--
102 implicit none                                           !Arguments
103 integer(kind=glcint),intent(inout)::iX,iY
104 !--                                                     !EndArgs
105 call CheckGL(+20083)
106      call MouseMotion(iX,iY)
107 call CheckGL(-20085)                                     ;return
108 End Subroutine cbMouseMotion
109 !-----7 9
110
111 Subroutine cbMouseButtons(Button,State,iX,iY)
112 !2020.04.02.1300cdt JMS- Callback- mouse "button press"
113 use OpenGLRec,only: glcint!Ref: OpenGL GL/GLU/GLUT docs
114 !--
115 implicit none                                           !Arguments
116 integer(kind=glcint),intent(inOut)::Button,State,iX,iY
117 !--                                                     !EndArgs
118 call CheckGL(+20096)
119      call MouseButtons(Button,State,iX,iY)
120 call CheckGL(-20098)                                     ;return
121 End Subroutine cbMouseButtons
122 !-----7 9
123
124 Subroutine GlutHandoff
125 !2021.10.04.1630cdt JMS- Checking for screen width undefined 1142

```

```

125 !2021.10.04.1030cdt JMS- checking for screen width undefined L142
126 !2020.06.19.1125cdt JMS- Initialization of the GLUT callbacks & handoff:
127 use OpenGLRec,only: & !Ref: OpenGL GL/GLU/GLUT Documentation
128     glutInitDisplayMode ,glutIdleFunc &
129     ,GLUT_SCREEN_WIDTH ,GLUT_SCREEN_HEIGHT ,GLUT_DOUBLE &
130     ,GLUT_SCREEN_WIDTH_MM ,GLUT_SCREEN_HEIGHT_MM ,GLUT_RGBA &
131     ,glutInitWindowSize ,glutCreateWindow ,GLUT_DEPTH &
132     ,glutKeyboardFunc ,glutMouseFunc ,glutInit &
133     ,glutSpecialFunc ,glutMotionFunc ,glcint &
134     ,glutDisplayFunc ,glutPassiveMotionFunc ,glutGet &
135     ,glutFullScreen ,glutMainLoop ,glFlush
136 use ScreenDef ,only: & !screen & colors
137     xwindowFull,ywindowFull,xwindowMM,ywindowMM,PixelsPerCM,xywindowRatio &
138     ,iFullScreen,iScrnColor,winId,lCharX,lCharY,nCharMaxX,nCharMaxY &
139     ,nCharCenX,nCharCenY,cwidth,cHeight,MagFactor
140 use KeyboardDef,only:Dtor !Keyboard
141 use ViewDef ,only:FrustNom,UnitCube0 !View bounds
142
143 !--
144 implicit none !Internals
145 integer(kind=glcint)::ix,iy
146 !-- !EndInts
147 iScrnColor=15; Dtor = dasin(1.d0)/90.d0
148 call glutInit
149 !call CheckGL(+20126) (...OpenGL is initially unhappy.)
150 xwindowFull =glutGet(GLUT_SCREEN_WIDTH )
151 ywindowFull =glutGet(GLUT_SCREEN_HEIGHT )
152 xwindowMM =glutGet(GLUT_SCREEN_WIDTH_MM )
153 ywindowMM =glutGet(GLUT_SCREEN_HEIGHT_MM)
154 if(xwindowMM>0) then
155     PixelsPerCM=10.d0*xwindowFull/dbl(xwindowMM)
156     MagFactor = ywindowFull/(PixelsPerCM*5.08)
157     if(FrustNom(0)>0.d0) then
158         FrustNom(6) = xwindowMM/50.8d0; FrustNom(5)=-FrustNom(6) !R & L- inches
159         FrustNom(8) = ywindowMM/50.8d0; FrustNom(7)=-FrustNom(8) !B & T-
160     endif!FrustNom(0)>0
161     UnitCube0 = FrustNom
162
163     else; pause "OpenGL - screen width undefined. Press enter to continue."
164
165 endif !xwindowMM>0
166 !-- Font selection for use with Projection=I, Modelview=I (Native OpenGL):
167 if(PixelsPerCm<40.d0) then
168     lCharX = 8 !GLUT_BITMAP_8_BY_13
169     lCharY =14 !... slightly increases line spacing
170     !*** Font is chosen in PrntOrtho ***
171 else
172     lCharX = 9 !GLUT_BITMAP_9_BY_15 !2022.07.09
173     lCharY =16 !The font is called within PrntOrtho
174 endif!PixelsPerCm<40.
175 !-- Text->screen character placement parameters:
176 nCharMaxX =(xwindowFull+lCharX/2)/lCharX !2020.06.18
177 nCharMaxY =(ywindowFull+lCharY/2)/lCharY !2020.06.18
178 nCharCenX =nCharMaxX/2+1
179 nCharCenY =nCharMaxY/2+1
180 xywindowRatio=dbl(xwindowFull)/dbl(ywindowFull)
181 cwidth =2.d0/nCharMaxX
182 cHeight =2.d0/nCharMaxY
183 !-- Firing up OpenGL:
184 call glutInitDisplayMode(ior(GLUT_DOUBLE,ior(GLUT_RGBA,GLUT_DEPTH)))
185 call glutInitWindowSize(xwindowFull,ywindowFull)
186 winid=glutCreateWindow("Sn3D")

```

```

187  !-- GLUT Callbacks:
188  call glutKeyboardFunc(cbKeyboard)
189  call glutSpecialFunc(cbSpecialFunctionKeys)
190  call glutDisplayFunc(cbUserView)
191  call glutIdleFunc(  cbUserView)
192  call glutMouseFunc(      cbMouseButtons)
193  call glutMotionFunc(      cbMouseMotion )
194  ix=xwindowFull/2; iY=ywindowFull/2
195  call      cbMouseMotion(ix,iY) !Does initialization
196  call glutPassiveMotionFunc(cbMouseMotion )
197  !! call OpenGLInfo(Ut)
198  !Initially select full screen mode:
199  call glutFullScreen; iFullScreen=1; call glFlush
200  call CheckGL(-20173)
201  call glutMainLoop
202  !OpenGL takes over for good... & calls all the shots.
203  return
204  End Subroutine GlutHandoff
205  !-----7 9
206
207  Subroutine CheckGL(Line)
208  !2025.02.03.0920cdt JMS- current generic version
209  !2020.05.09.0855cdt JMS- OpenGL Error Checking
210  !Code Blocks/Group ID#'s:
211  !#1  S1ModDef      .f95
212  !#2  S2ModCallback.f95
213  !#3  S3Main        .f95
214  !#4  S4Callbacks   .f95
215  !#5  S5Screen       .f95
216  !#6  S6View8        .f95
217  !#7  S7Motion7      .f95
218  !#8  S8Help         .f95
219  !#9  S9Font         .f95
220  !#11 S-11AppF1      .f95
221  !#14 S-14AppF4      .f95
222  !#15 S-15AppF5      .f95
223  !
224  !                               Line#'s: Entry   Return
225  !                               b:code block;  n:line number +bbnnnn -bbnnnn
226  !                               bb=02 for S2CallbackModule.f95: +20000 -20000
227  !                               >0:entry <0:exit
228  !Subroutine cbUserView                !+20039 -20041
229  !Subroutine cbKeyboard(Key,xCursor,yCursor) !+20055 -20057
230  !Subroutine cbSpecialFunctionKeys(Key,xCursor,yCursor) !+20070 -20072
231  !Subroutine cbMouseMotion(ix,iY)       !+20083 -20085
232  !Subroutine cbMouseButtons(Button,State,ix,iY) !+20096 -20098
233  !Subroutine GlutHandoff                !         -20173
234  ! Here in CheckGL(Line)- pause before halt:                -20229
235  !Line#'s      ^^^^^      ^^^^^
236  !Assign other block numbers to other code blocks.
237  !See also: S3Main.f95\BreakHere- for debug isolation of non-OpenGL errors.
238  !use S2Callback,only:CheckGL !OpenGL CB's, GlutHandoff, & glGetError 2024.03.18
239
240  !--Globals                                S1ModDef.f95: 2024.03.18
241  use OpenGLRec,only: & !Ref: OpenGL GL/GLU/GLUT docs
242  glenum,glubyte,glGetError,gluErrorString
243  use ioDef ,only: us !Files,Units,TimeStamp,Selfies,Flags
244  !--End Globals
245  implicit none
246  !--Internals
247  integer(4)::Line
248  integer(kind=glenum)::iErrCode
249  character(kind=glubyte)::iErrStr(17)*1

```

```

249 character(kind=gubbyte), parameter :: cErrStr(17) = ' '
250 integer(4) :: nErrors=0, nBadCheckGLCalls=0, iGroup=0, iLine=0
251 !-----
252 iErrCode=glGetError() ; if(iErrCode==0) return
253 nBadCheckGLCalls=nBadCheckGLCalls+1 !; nErrors=0
254 goto 15
255 10 iErrCode=glGetError()
256 if(iErrCode==0) then !; write(Us,"(a1)") char(7) !Beep
257 !For debugging purposes- flag the return line.
258 ! At least one error just occurred, & you'll return
259 ! to the source of that error:
260 return
261 endif !iErrCode==0
262 15 nErrors=nErrors+1 ; cErrStr=char(0)
263 cErrStr=gluErrorString(iErrCode)
264 if(Line.ne.0)
265 write(6,"(2i6,' CheckGL:@Line',SP,i8,' GLErr=',z8,' hex :',17a1)") &
266 nBadCheckGLCalls,nErrors,Line,iErrCode,cErrStr
267 iGroup=abs(Line)/10000; iLine=abs(Line)-iGroup*10000
268 select case(iGroup)
269 case(1); write(6,"(13x,'S1ModDef.f95 ',i4)") iLine
270 case(2); write(6,"(13x,'S2ModCallbacks.f95 ',i4)") iLine
271 case(3); write(6,"(13x,'S3Main.f95 ',i4)") iLine
272 case(4); write(6,"(13x,'S4Callbacks.f95 ',i4)") iLine
273 case(5); write(6,"(13x,'S5Screen.f95 ',i4)") iLine
274 case(6); write(6,"(13x,'S6View8.f95 ',i4)") iLine
275 case(7); write(6,"(13x,'S7Motion7.f95 ',i4)") iLine
276 case(8); write(6,"(13x,'S8Help.f95 ',i4)") iLine
277 case(9); write(6,"(13x,'S9Font.f95 ',i4)") iLine
278 case(10); write(6,"(13x,'-unspecified-',i4)") iLine
279 case(11); write(6,"(13x,'S-11AppF1.f95 ',i4)") iLine
280 case(12); write(6,"(13x,'S-12AppF2.f95 ',i4)") iLine
281 case(13); write(6,"(13x,'S-13AppF3.f95 ',i4)") iLine
282 case(14); write(6,"(13x,'S-14AppF4.f95 ',i4)") iLine
283 case(15); write(6,"(13x,'S-15AppF5.f95 ',i4)") iLine
284 case(16); write(6,"(13x,'S-16AppF6.f95 ',i4)") iLine
285 case(17); write(6,"(13x,'S-17AppF7.f95 ',i4)") iLine
286 case(18); write(6,"(13x,'S-18AppF8.f95 ',i4)") iLine
287 case(19); write(6,"(13x,'-unlisted.f95 ',i4)") iLine
288 !User-added Groups examples:
289 case(21); write(6,"(13x,'S0-RubikVis.f95-1',i4)") iLine
290 case(22); write(6,"(13x,'S0-RubikVis.f95-2',i4)") iLine
291 case(23); write(6,"(13x,'S0-RubikFaceId.f_',i4)") iLine
292 case default
293 write(6,"(13x,'-uncharted-.f95 ',i4)") iLine
294 end select !iGroup
295 if(Line.eq.0)
296 write(6,"(2i6,' :@Line 0', ' GLErr=',z8,' hex :',17a1)") &
297 nBadCheckGLCalls,nErrors,iErrCode,cErrStr
298 call jPause('S2ModCallbacks@L276- ^ CheckGL: OpenGL Error detected.')
299 if(nErrors>=500) then
300 write(6,*) &
301 'CheckGL @L-20229: 500 accumulated errors. Stopping.'
302 !pause'CheckGL @L-20229: 200 accumulated errors. Press enter to halt.'
303 stop 'S2Callback{CheckGL: Halt!'
304 endif !nErrors>=200
305 goto 10
306 End Subroutine CheckGL
307 !-----7 9
308
309 End Module S2Callback
310 !-----7 9

```

```
311
312 !-----7 9
313 !use S2Callback      !OpenGL Callback subroutines (pass-throughs)
314 !      cbUserView,cbKeyboard,cbSpecialFunctionKeys      &!Exec & kbd callbacks:
315 !      ,cbMouseMotion,cbMouseButtons,GlutHandoff,CheckGL      !Mouse callbacks:
316 !-----7 9
317
318
```