```
!S0-AppF2.f95  Group ID: #12    Dodecahedron - the 12-face Platonic Solid.
!                               SA2-hVecMath16Mod.f95 computes the geometry.
!2025.05.24.1840cdt- Application F2

!            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 S12AppF2
!Subroutine PLabelView(A3,P3L,iCol,WidthOfLine)
!-------------------------------------------------------------------------------7 9

Subroutine S12AppF2
!2025.04.20.1520cdt JMS- Dodecahedron - the 12-face Platonic Solid.
 use TaskDef   ,only: Analyst,Phone,Street,CityStateZip,IPSummary
 use ioDef     ,only: Up,Ut,iTeapot
 use ScreenDef ,only: yWindowFull,xyWindowRatio,PixelsPerCM,MagFactor
 use MouseDef  ,only: MChan
 use AppsDef   ,only: AppBanner,AppInFile
 use ViewDef   ,only: ThreePhase,V,jVG,Key
 use ModelDef  ,only: M7Init,M7,nM7L,Nnutate
 use SimDef    ,only: RunTimer,IterTotal
 use S2Callback,only: CheckGL
 use OpenGLRec,only:    glPolygonMode,GL_FRONT,GL_LINE,GL_BACK        &
                       ,glLineWidth,glShadeModel,GL_SMOOTH            &
                       ,glBegin,GL_Quads,GL_LINES,glVertex3fv,glEnd   &
                       ,glGetIntegerv,glMatrixMode,GL_MATRIX_MODE     &
                       ,GL_MODELVIEW,glPushMatrix,glPopMatrix,glFlush &
                       ,glMultMatrixd,glLoadIdentity                  &
                       ,glFloat,GL_LINE_STRIP,GL_POINTS,GL_LIGHTING   &
                       ,glVertex3f,glVertex3d,glVertex3dv             &
                       ,glPointSize,glDisable,GL_FLAT

 use TubeDef

 use DodecIO
     !^ Note: all the reals in DodecIO are quad precision! Use dble() here.
 implicit none
 !-----                                                                Arguments
  integer(4)::iP
 !-----                                                                Internals
  integer(4)::Init,i

  real(8)   ::Angle     !Yaw angle- real(8)
  real(8)   ::Angledv(3)!                   - vector form

  real(4)   ::Anglef    !         - real(4)
```

```
63     real(4)   ::Anglefv(3)!                   - vector form
64
65     real(8)   ::sY,cY     !         - cosine, sine
66     real(8)   ::vTemp(4)  !         - cosine, sine
67     integer(4)::iDone,nPoint
68     real(8)   ::P1(4)
69
70     !-- Graphics-use
71     character::chn*80
72     real(8)   ::vTemph(4)  !,vMag,vMag1
73     integer(4)::MtxMode(1)
74     integer(4)::nRow,nCol
75
76     real(8)   ::XYZRPYwC(7) !(X,Y,Z,Roll,Pitch,Yaw,height of Character)
77     real(4)   ::LineWidth = 2
78     real(4)   ::LHeight   = 2
79     character(len= 3)::A3
80     character(len=80)::PText
81
82    !--EndDefs-------------------------------------------------
83   !  AppNumber    = 2
84      iP = Ut
85      if(Init.eq.0) then
86       !--Establish screen default 7DoF, if desired:
87        AppBanner(   2) ='S12AppF2: Dodecahedron Visualization'
88        AppInFile(   2) = char(0)
89        Analyst(     2) = '    Jeff Setterholm'
90        Street(      2) = '    8095 230th St.E.'
91        CityStateZip(2) = 'Lakeville, MN 55044 USA'
92        Phone(       2) = '     (nnn)-nnn-nnn'
93        IPsummary(   2) = '         Free'
94                                                    !a35
95     !----- Dodecahedron initilization - Jeff Setterholm's algorithm:
96        Dd%iFaceSee(0:12)=(/0,1,0,0,0,0,1,0,0,0,1,1,1/) !Face#1,#6,#10,#11,& #12
97        Dd%iFaceSee(0:12)=(/0,1,1,0,0,1,1,0,0,0,1,1,1/)
98        Dd%iFaceSee(0)   =1 !=0 !=-1
99        Dd%iEdgeSee(0)   =0 !=-1
100       Dd%iVertexSee(0) =0 !=-1
101
102       if(iP>5)write(iP,"(/'Subroutine S12AppF2 calling DodecModel16() @L103:')")
103       call DodecModel16(iP)   !(This will compute the dodecahedron geometry
104       call SaveOutFile
105
106       iTeapot        =0  !Turns off the teapot
107      !Setting up the default modelview matrix:
108       M7(0)%DoF      = (/ 0.d0, 0.d0, 0.d0,   0.d0, 0.d0,  0.d0, 5.d0 /)
109      !M7(0)%ioe(1)   = 7 !Immediate control- mag   by  vertical  mouse motion.
110       M7(0)%ioe(1)   = 5 !                  - pitch by  vertical  mouse motion.
111       M7(0)%ioe(2)   = 6 !                  - yaw   by horizontal mouse motion.
112       M7Init           = 0
113       call Motion7(0,0,6)  !X  Y  Z  R  P  Y MAG
114      !M7(0)%Locked   = (/0, 1, 1, 1, 1, 1, 1, 0  /) !On screen "L"'s Mag moves
115       M7(0)%Locked   = (/0, 0, 0, 0, 0, 0, 0, 0  /) !On screen "L"'s Unlocked
116       M7(0)%DoFReset = M7(0)%DoF
117
118       Key%VuMode      = 1  !Perspective
119       Key%SplitScreen = 0  !Screen not split
120       nNutate = 1          !Nutate the modelview matrix
121       Init=1
122     endif !Init=0
123
124     select case(ThreePhase)
125      case(1)  !Background number crunching, if any:
```

```
125     case(1)  !Background number crunching, if any.
126                                                          call CheckGL(+120108)
127      !--Establish screen default 7DoF, if desired:
128
129      if(Up>5) then !This section of code only runs when you request a printout.
130        !--As a novice, call your numeric subroutine in case(1).
131        !   & press "p" for a live DOS screen printout
132        !        or "P" for printout to a file (e.g. when you set Up = 13).
133        write(Up,"('F2:TP',i1,'Up',i2)") ThreePhase,Up
134        write(uP,"(/'S12AppF2:')")
135        write(uP,"(  f20.12,' = RunTimer')") RunTimer
136
137      endif!uP>5
138                                                          call CheckGL(-120119)
139
140    !~~~~~~~~~~~~~~~
141     case(2)  !Update variables & 2D screen info: Superceded by split screen(s)
142
143     case(3)  !Draw/redraw app. 2D & 3D graphics:   (ThreePhase==3)
144                                                          call CheckGL(+120125)
145 !!!if(1>0) return
146    !--- Recurring visualization
147    call glGetIntegerv(GL_MATRIX_MODE,MtxMode)
148
149    !--- Insert your 2-D code here:
150    call glMatrixMode(GL_MODELVIEW)  ;call glPushMatrix;  call glLoadIdentity
151                                    nRow = 3;                    nCol = 2
152        write(Ptext, "('SA2-hVecMath16Mod.f95/DodecModel16()')")
153                                 nRow=nRow+1; call PrntOrtho(nRow,nCol,1, 0,PText)
154        write(Ptext, "('   computes the geometry.')")
155                                 nRow=nRow+1; call PrntOrtho(nRow,nCol,1, 0,PText)
156
157    call glMatrixMode(GL_MODELVIEW)  ;call glPopMatrix
158
159    !--- Insert your 3-D code here:
160    call glMatrixMode(GL_MODELVIEW)  ;call glPushMatrix
161
162    !-- Draw arcs:
163    call glShadeModel(GL_SMOOTH)
164   ! call glLineWidth(3._glfloat)
165    call glLineWidth(LineWidth)
166    call Colors3D(2)
167
168    !call glBegin(GL_LINE_LOOP)
169    !Draw a unit circle in the X-Y plane:
170    call glBegin(GL_LINE_strip) !;call Colors3D(8)
171             call glvertex3d( 0.d0,0.d0,0.d0 )
172      do i=-90,90 ;Angle=i
173
174        call glvertex3d( dcosd(Angle) ,dsind(Angle) ,0.d0 )
175
176    enddo!i
177    call glend ;call glFlush                             ;call CheckGL(-120148)
178
179    !Draw an arc in the X-Z plane:
180    call Colors3D(2)
181    call glBegin(GL_LINE_strip)
182      do i=-90,90 ;Angle=i
183        call glvertex3d( dcosd(Angle),0.d0,-dsind(Angle) )
184      enddo!i
185        call glvertex3d( 0.d0,0.d0,-0.79465447229d0 )
186    call glend ;call glFlush
```

```
187
188    !Draw an arc inscribing the five vertices of face#11:
189     vTemph=(/  0.60706199821d0, 0.d0, -0.79465447229d0, 1.d0 /)
190     call Colors3D(2)
191     call glBegin(GL_LINE_strip)
192       do i=-108,108 ;Angle=i ;cY=vTemph(1)*dcosd(Angle)
193                               sY=vTemph(1)*dsind(Angle)
194            call glvertex3d(cY,sY,vTemph(3))
195       enddo!i
196     call glend ;call glFlush
197
198     !--- Visualize vertex & face-center points:
199     call Colors3D( 1)
200     call glShadeModel(GL_SMOOTH)
201    !call glLineWidth(4._glfloat)
202     call glPointSize(6._glfloat)
203    !call Colors3D(15)
204    !call Colors3D(15)
205     call glBegin(GL_POINTS)
206       do i=1,20 ;if(Dd%iVertexSee(i).eq.0) cycle
207         call glvertex3dv(dble(Dd%Vertex(1:3,i)))
208       enddo!i
209     call glend ;call glFlush
210
211    !call Colors3D(1)                                  !13->1
212     call glBegin(GL_POINTS)
213       do i=1,12 ;if(Dd%iFaceSee(i).eq.0) cycle
214         call glvertex3dv(dble(Dd%vNormal(1:3,i)))
215       enddo!i
216     call glend ;call glFlush
217
218     !--- For each selected face:
219        !if(iP.gt.5) write(iP,"('Face drawing:')")                    !Diagnostics
220     do i=1,12 ;if(Dd%iFaceSee(i).eq.0) cycle
221       !Number the face:
222       write(chn(1:3),"(i2,1x)") i
223       if(Dd%iFaceSee(0).gt.-1)  &
224       call PLabelView(chn(1:3),dble(Dd%vNormal(1:3,i)), 1,3.)
225
226       !Draw the outward normal:
227       if(Dd%iFaceSee(0).le.-2)  exit
228       call glShadeModel(GL_SMOOTH) ;call glLineWidth(2._glfloat)
229       call Colors3D(15)
230       call glBegin(GL_LINES)
231                                  vTemp(1:3  )=Dd%vNormal(1:3,i)*.75d0
232         call glvertex3dv(        vTemp(1:3  ) ) !0.d0,0.d0,0.d0)
233         call glvertex3dv(dble(Dd%vNormal(1:3,i)))
234        !if(iP.gt.5) write(iP,"(i2,3f20.10)") i,Dd%vNormal(1:3,i)       !Diagnostics
235       call glend ;call glFlush
236     enddo!i
237     !--- For each selected vertex:
238     do i=1,20 ;if(Dd%iVertexSee(i).eq.0) cycle
239       !Number the vertices:
240       if(Dd%iVertexSee(0).le.-1) exit
241        write(chn(1:3),"(i2,1x)") i
242         if(Dd%iVertexSee(0).ge.0) &
243         call PLabelView(chn(1:3),dble(Dd%Vertex(1:3,i)),12,3.)
244     enddo!j
245     !--- For each selected edge:
246     do i=1,30 ;if(Dd%iEdgeSee(i).eq.0) cycle
247       !Draw the edges:
248       if(Dd%iEdgeSee(0).le.-2) exit
```

```
249       call glShadeModel(GL_SMOOTH) ;call glLineWidth(2._glfloat)
250       call Colors3D(13) !(mod(i,10)+6) !
251       call glBegin(GL_LINES)
252         call glvertex3dv(dble(Dd%Vertex(1:3,Dd%iEdge(1,i))))
253         call glvertex3dv(dble(Dd%Vertex(1:3,Dd%iEdge(2,i))))
254       call glend ;call glFlush
255
256      !The solid edge tubes are drawn here.
257       call Colors3D(1)
258       iDone=-2
259       P1=(Dd%Vertex(1:4,Dd%iEdge(1,i)))
260       call Sphere(.02d0,P1)
261       call Tubes(iDone,nPoint,P1,.02d0,0)
262       P1=(Dd%Vertex(1:4,Dd%iEdge(2,i)))
263       call Sphere(.02d0,P1)
264       call Tubes(iDone,nPoint,P1,.02d0,0)
265
266       !Number the edges:
267       call glDisable(GL_LIGHTING)
268       if(Dd%iEdgeSee(0).le.-1) cycle
269       vTemph=( Dd%Vertex(1:4,Dd%iEdge(1,i)) + Dd%Vertex(1:4,Dd%iEdge(2,i)) )/2.d0
270       write(chn(1:3),"(i2,1x)") i
271       call PLabelView(chn(1:3),vTemph(1:3),13,2.)
272     enddo!i
273
274  !--Draw colored axes:
275     call glLineWidth(2.)
276    !--
277     call Colors3D(11) !Green
278     call glBegin(GL_LINES)
279       call glVertex3f(0.,0.,0.); call glVertex3f(1.,0.,0.)
280     call glEnd
281    !--
282     call Colors3D( 9) !Yellow
283     call glBegin(GL_LINES)
284       call glVertex3f(0.,0.,0.); call glVertex3f(0.,1.,0.)
285     call glEnd
286    !--
287     call Colors3D( 7) !Red
288     call glBegin(GL_LINES)
289       call glVertex3f(0.,0.,0.); call glVertex3f(0.,0.,1.)
290     call glEnd; call glFlush
291    !--
292
293    !if(nCol>0) then
294      !--Axis labels:
295      !              XYZRPYwC=(/   X ,   Y  , Z   , Roll, Pitch, Yaw,width Ch/)
296                     XYZRPYwC=(/ .8d0, .0d0,-.02d0, 0.d0,  0.d0,  0.d0 ,.1d0 /)
297      !call VecFont7(XYZRPYwC,LineWidth,iCol,Label)
298       PText="+X" ; call VecFont7(XYZRPYwC,LineWidth, 11 ,PText) !Green
299
300                     XYZRPYwC=(/ .0d0, .8d0,-.02d0, 0.d0,  0.d0, 90.d0 ,.1d0 /)
301       PText="+Y" ; call VecFont7(XYZRPYwC,LineWidth,  9 ,PText) !Yellow
302
303                     XYZRPYwC=(/ .0d0, .02d0, .8d0, 0.d0,-90.d0,  0.d0 ,.1d0 /)
304       PText="+Z" ; call VecFont7(XYZRPYwC,LineWidth,  7 ,PText) !Red
305                                                          call CheckGL(-120266)
306      !---
307      call glMatrixMode(GL_MODELVIEW ) ;call glPopMatrix
308      call glMatrixMode(MtxMode(1))
309      call glFlush                                       ;call CheckGL(-120280)
310      end select!ThreePhase
```

```
311
312     return
313 !End Subroutine DodecViz
314 End Subroutine S12AppF2
315 !-------------------------------------------------------------------------------7 9
316
317 Subroutine PLabelView(A3,P3L,iCol,WidthOfLine)
318 !2025.04,17.1220cdt JMS
319 !2013.10.15.0640cdt JMS- PLabelView(+WidthOfLine)->public
320  !-----
321  implicit none                                                          !arguments
322   real(8)    ::P3L(3)
323   integer(4)::iCol
324   real(4)    ::WidthOfLine
325  !---                                                                   !internals
326   character ::chn*80
327   integer(4)::j
328   real(8)    ::rA(3),xA(3)
329
330   real(8)    ::XYZRPYwC(7) !(X,Y,Z,Roll,Pitch,Yaw,height of Character)
331   character(len= 3)::A3
332   character(len=80)::PText
333
334   !----------
335    xA(1)=P3L(1) ;rA(1)=  0.d0 != 90.d0
336    xA(2)=P3L(2) ;rA(2)=  0.d0 !=-90.d0
337    xA(3)=P3L(3) ;rA(3)= 90.d0 !=  0.d0
338 !! call glLineWidth(WidthOfLine)       !-- This doesn't appear to work yet
339    do j=1,3
340      select case(j)
341       case(1);write(chn,"(     4x     ,f5.2    ,a1)")   P3L(1),char(0)
342               xA(2)=P3L(2)+.03d0+.20d0
343       case(2);write(chn,"(a3,a1)") A3,char(0)
344               xA(2)=P3L(2)+.03d0
345        XYZRPYwC=(/ xA(1),xA(2),xA(3),rA(1),rA(2),rA(3),.04d0 /)     !2025.04.15
346                        ! VecFont7(XYZRPYwC,  LineWidth,iCol,Label)
347        PText=chn ; call VecFont7(XYZRPYwC,WidthOfLine,iCol,PText)
348
349       case(3);write(chn,"(     4x     ,f5.2    ,a1)")   P3L(3),char(0)
350               xA(2)=P3L(2)+.03d0-.20d0
351      end select!j
352    enddo!j
353    return
354 End Subroutine PLabelView
355 !-------------------------------------------------------------------------------7 9
356
```