```
 /*App-F4-VecText7D-Demo.c
  3D-Environment-C, Version 0.6 Example User Application F4:
  2016.07.10 JMS- Information added in 3DVecText.h                  Version 0.6
  2016.06.28 JMS- Writing vector fonts directly in 3D model space:
  2013.01.17 JMS- Traveler2/Athlon64/WinXPPro/APF9.0:C/OpenGL+CGlut

 Herein: void AppF4(void)
*/
#include <3DEnv.h>   //<- Environment's variables & functions become available.

void AppF4(void) //-----------------------------------------------2016.07.10 JMS
                 //You've entered this application reached by pressing 'F4'
{                //    "F_: Title__________________ yyyy.mm.dd\0";      <-e.g.
        char AppName[]="F4: VecText7D Demo          2016.07.10\0";
  //                     E  ,  N   ,    F  ,  L  ,  R  , T  , B  ,  D
    double FrustCoes[8]={ 1.2,-11.994, 24000.0,-10.4, 10.4,-6.5, 6.5, 24.0 };
  int i,i1,j,j1,k,k1,k2,m,n;
  double Xyzh[4],Rpyh[4];
  double static Timer=0.e0;
  char        buffer[   80],Label[80];
  char static buffer0[1400];
  char static buffer1[1024];
  char static buffer2[ 128];
  char static buffer3[1024];

  int  static CharVecs[257][50][3]; //Each character can have 49 points max.
                                    //[256][*][*] has the edit data.
  int  static nPts;
  int iVmouse[2];
  int  static DrawAllChars=0;
  int         iPass;

 switch(S.ThreePhase) // <- driven by the 3D viewer -------
  {        /*---------------------------------*/
   case(1):Phase One- Transfers, initializations, & non-screen computations:
    if(S.AppInit[S.AppNumber]<1)
     {  S.AppInit[S.AppNumber]=  1;
        S.Scale                = 1.e0; //S.Scale alters depth bounds
        S.FovYZzoom            =  2.e0; //S.FovYZzoom doesn't alter depth bounds
        S.VuMode               =  2;
        S.MousePitch           =-45.e0;
        S.MouseYaw             =  0.e0;
        S.iTeapot               = 7;

        //Unpack the present character set in 'VecText7D' into CharVecs:
        for(n=0;n<256;n++)
         { m=CharLoc[n]; if(m==0) continue; // <- character- undefined.
           i=PointLoc[m];                   // <-          - ASCII code
           j=PointLoc[m+1];                 // <-          - number of points
           for(k=0;k<(j+1);k++) { CharVecs[i][k][0]=PointLoc[m-1+k*3  ];
                                  CharVecs[i][k][1]=PointLoc[m-1+k*3+1];
                                  CharVecs[i][k][2]=PointLoc[m-1+k*3+2];
                                }/*k*/
         }/*n*/

      sprintf(buffer0,
"\n\n\n\n\n"
"A dynamic version of actual physical space can be your filing cabinet!\n"
"\n"
"> Keep the basic instructions, repair-person contact info.,\n"
"    + your additional words of wisdom... with your stationary gear.\n"
"> Present the names of geographic entities and visualize 3D geodesy.\n"
"> Place historic markers wherever you like.\n"
"> Navigate using billboards which mark your turns.\n"
"> ...etc., etc.\n"
"> The uses in simulations and fantasies are unbounded.\n"
"> Share such information with other people almost-effortlessly\n"
"    with reduced miscommunication.\n"
"\nAir vehicle enthusiasts:\n"
"> Space stabilize flight checklists (demo'd @ McDonnell Aircraft~1977)\n"
```

```
"     and the 3D locations of other nearby aircraft & hazards to flight.\n"
"> Use 'future physics' displays     (demo'd @ Honeywell   S&RC  ~1980)\n"
"     to efficiently comprehend the basics of manual vehicle control.\n"
"                                -&-\n"
"> Hope that your VR Headset doesn't go belly-up at a bad time.\n"
"> Beware that enhanced technique without enhanced judgment is perilous\n"
"    ... Mother Nature fully enforces her rules - impartially.\0");


       sprintf(buffer1,
"   \"Four score and seven years ago\n"
"    our forefathers brought fourth\n"
"          on this continent\n"
"               a nation\n"
"        conceived in liberty\n"
"                and\n"
"     dedicated to the proposition\n"
"               that\n"
"      all men are created equal.\"\n\n"
"                 - Abraham Lincoln\n\n"
"                  November 19,1863\n"

"\v\v\v\v\v"
" May both your grasp of the unknown\n"
"               -and- \n"
"your ability to teach what you grasp\n"
"         be enhanced by 3D\n"
"       six degree-of-freedom\n"
"      spatial visualizations\n"
"       of your own creation.\n\n"
"                   - Jeff Setterholm\n\n"
"                       June 24,2016\n\0");
       sprintf(buffer2,

"Setterholm's Law: \x00a9\x0008 1986\n"
                              //^:terminate with a non-hex character
                          //^____ :hex code for backspace
                    //^____ :hex code of the copyright symbol
"\"A doctor's orders are only as good\n"
"  as the nurses carrying them out.\"\0");

       sprintf(buffer3,
"\n\n\n\n\n"
"       A doctor can write all the orders he/she wants to,\n"
"           but if the nurses do not know how to do\n"
"          the treatments, procedures, interventions or \n"
"  administer & manage the medications and utilize the equipment,\n"
"  then the orders will not be carried out in a therapeutic way.\n"
"  We need knowledgeable, skillfully trained professional nurses\n"
"   who have been educated by experienced, knowledgeable nurses.\n"
"      Most doctors are not trained in the art of healing;\n"
"         they are trained in the science of medicine\n"
"         and rely on nurses to carry out the orders.\n\n"
"                                    Donna Setterholm, RN,BSN,MA\n\n"
"                                - may be copied, in its entirety,\n"
"                                     without permission.\n\0");
      S.BrkOn=0;
     }//S.AppInit

    sprintf(S.AppName,"%s\0",AppName);
    // Viewer: Load your frustum coefficients:
    for(i=0;i<8;i++) { S.FrustCoes[i]=FrustCoes[i]; }

           /*---------------------------------*/                          break;
   case(2):// Phase Two- 2D orthographic screen graphics:
    //Author & IP information:
    h4Fill(Xyzh, -.25e0*(S.xyWindowRatio),  .95e0, 0.e0, 1.e0);
    h4Fill(Rpyh,90.00e0                    , 0.00e0, 0.e0, 1.e0);
    sprintf(Label,"\xe0\x08 2016 Jeffrey M. Setterholm\0");
    Xyzh[1]=.95e0; VecText7D(Xyzh,Rpyh, .03e0, 1.,  1  ,Label);
```

```
     sprintf(Label," 8095 230th St. E., Lakeville, MN 55044\0");
     Xyzh[1]=.92e0; VecText7D(Xyzh,Rpyh, .02e0, 1., 11  ,Label);
     sprintf(Label,"        jeff.setterholm@gmail.com  \0");
     Xyzh[1]=.89e0; VecText7D(Xyzh,Rpyh, .02e0, 1.,  6  ,Label);
     sprintf(Label,"This 'App-F4' is freely distributable.\0");
     Xyzh[1]=.86e0; VecText7D(Xyzh,Rpyh, .02e0, 1.,  8  ,Label);

     if(S.SimMode!=2) Timer=Timer+.01e0; if(Timer>20.e0) Timer=0.e0;
     if(S.VuMode==2)
       {PrntOrtho( 4,2, 1,0,"App F4 initializes in Red/Cyan mode.\0");
        PrntOrtho( 5,2, 1,0,"use red/cyan glasses...  or press 'e'.\0");}
     PrntOrtho( 7,2, 1,0,"Hold down the left mouse button & move the mouse.\0");
     PrntOrtho( 8,2, 9,0,"Press the right mouse button for menu access,\0");
     PrntOrtho( 9,2, 9,0,"  ...then select  'Help- keyboard Mouse'.\0");
     PrntOrtho(10,2, 9,0,"  ...& try 'Simulation Ctl' or F11 & F9.\0");

     PrntOrtho(11,2, 1,0,"The characters in function 'VecTest7D' can be\0");
     PrntOrtho(12,2, 1,0,"modified and extended here.\0");
     PrntOrtho(13,2, 1,0,"Menu item 'Breaker Viewer, turns on the editor.\0");

     if(S.Scale<.7e0)
      { h4Fill(Xyzh,  -.85e0, .4e0 , 0.e0, 1.e0);
        h4Fill(Rpyh, 90.00e0, .0e0 , 0.e0, 1.e0);

       sprintf(buffer,
"    6dof Virtual Headsets\n"
"open a vast spatial frontier.\0");
      //VecText7D(    ,     , SizeH, LineWidth, iCol,  Label[] );
        VecText7D(Xyzh,Rpyh, 0.07e0,    3.    ,   8 ,  buffer   );
      //VecText7D(Xyzh,Rpyh, 0.07e0,    6.    ,  15 ,  buffer   );
        VecText7D(Xyzh,Rpyh, 0.03e0,    1.    ,   1 ,  buffer0  );
        VecText7D(Xyzh,Rpyh, 0.03e0,    3.    ,  15 ,  buffer0  );
      }/*(S.Scale<.3e0)*/

     if(S.BrkOn==0)                                                    break;

     //Character Editor:
     PrntOrtho(14,2, 6,0,"Use + & - to select the character of interest.\0");
     PrntOrtho(15,2, 6,0,"Use the mouse to move to points on the grid.\0");
     PrntOrtho(16,2, 6,0," 0 :Clear the character drawing buffer.\0");
     PrntOrtho(17,2, 6,0," 1 :Begin newline.\0");
     PrntOrtho(18,2, 6,0," 2 :Continue line.\0");
     PrntOrtho(19,2, 6,0," 3 :Delete most recent line.\0");
     PrntOrtho(20,2, 6,0," 4 :End revising the character\0");
     PrntOrtho(21,2, 6,0," 5 :Save the new character version.\0");
     PrntOrtho(22,2,11,0," 6 :View the current character set (toggle).\0");
     PrntOrtho(23,2, 2,0," 7 :Export CharLoc[256] & PointLoc[*] to:\0");
     PrntOrtho(24,6, 2,0,S.YourAppOutput);

     i=Brk(0,0,0); //This allows the '+' & '-' keys to change S.BrkLim
                   // ... which will be used as the ASCII character number.
     if(S.BrkLim<  1) S.BrkLim=  1;
     if(S.BrkLim>256) S.BrkLim=256;
     n=S.BrkLim-1;  // [0,255]

     glMatrixMode(GL_MODELVIEW ); glPushMatrix(); glLoadIdentity();
     ProjOrtho(4);  glFlush();

     sprintf(buffer,"n =%4i decimal  %2.2x hex  char= %1c GLUT\0",n,n,n);
     PrntOrtho(26,2, 1, 0,buffer);

     if(S.KbdKey==54){ S.KbdKey=0; DrawAllChars=1-DrawAllChars;}
     if(DrawAllChars==1)
     //6. Draw the current character set:
      {    h4Fill(Rpyh,   90.e0       ,  0.0e0        , 0.e0, 1.e0);
       for(i1=0;i1<16;i1++)
        {for(j1=0;j1<16;j1++)
          {h4Fill(Xyzh,(-9.e0+i1)/10.e0
                      ,( 7.e0-j1)/10.e0
                      , -0.1e0
```

```
                    ,  1.0e0);
        if(i1==0){sprintf(buffer,"%1x\0",j1);
                  Xyzh[0]=Xyzh[0]-.05e0;
                  VecText7D(Xyzh,Rpyh, 0.03e0,1., 5,buffer);
                  Xyzh[0]=Xyzh[0]+.05e0;
                 }
        if(j1==0){sprintf(buffer,"%1x\0",i1);
                  Xyzh[1]=Xyzh[1]+.05e0;
                  VecText7D(Xyzh,Rpyh, 0.03e0,1.,14,buffer);
                  Xyzh[1]=Xyzh[1]-.05e0;
                 }
        i=i1*16+j1; if(CharVecs[i][0][1]==0) continue;
        j=CharVecs[i][0][2];        if(j<=0) continue;
        //Draw each character:
        Colors3D( 1); glLineWidth(1.0f);
        glBegin(GL_LINE_STRIP);
        for(k=1;k<j+1;k++)
         { switch(CharVecs[i][k][0])
            {case(1):    glEnd();
              glBegin(GL_LINE_STRIP);
             case(2):
              glVertex3f( (CharVecs[i][k][1])*.003e0*2.e0/3.e0+Xyzh[0] +.015e0
                         ,(CharVecs[i][k][2])*.003e0          +Xyzh[1]
                         ,-.1);                                          break;
             case(3):                                                    break;
            }/*(S.CharVecs[n][i][0])*/
         }/*k*/
        glEnd();
       }/*j1*/
     }/*i1*/
    glMatrixMode(GL_MODELVIEW ); glPopMatrix();                        return;
   }/*(S.KbdKey)*/

  iVmouse[0]=(floor(S.vMouse[0]*30.e0+50.5e0)-50.e0);
  iVmouse[1]=(floor(S.vMouse[1]*20.e0+50.5e0)-50.e0);
  sprintf(Label,"                        %7i  %3i\0"
                               ,iVmouse[0],iVmouse[1]);
  PrntOrtho(27,2,11, 0,Label);

  //List the point set of the current character, if any:
  if(  CharVecs[n][0][1]>0)
   { j=CharVecs[n][0][2];
     for(k=0;k<(j+1);k++)
      {  sprintf(buffer,"%2i,%5i,%3i\0"
          ,CharVecs[n][k][0],CharVecs[n][k][1],CharVecs[n][k][2]);
         PrntOrtho(28+k,2,11, 0,buffer);
      }/*k*/
   }
  //Draw a plotting grid:
  glShadeModel(GL_FLAT);
  //Horizontal grid lines:
  for(i=-15;i<16;i++)
   {                                 glLineWidth(1.0f);
    if(i==-10 || i== 0 ||  i==10 ) glLineWidth(3.0f);
                                    Colors3D(12);
    if(i==-10          ||  i==10 ) Colors3D( 6);
    glBegin(GL_LINES);
      glVertex3f(-.5,float(i/20.), .5);
      glVertex3f(+.5,float(i/20.), .5);
    glEnd();
    glLineWidth(1.0f);
   }/*i*/

  //Vertical grid lines:
  for(j=-15;j<16;j++)
   {                                          glLineWidth(1.0f);
    if( j==-15 || j==-10 || j== 0 || j== 10 ) glLineWidth(3.0f);
                                              Colors3D(12);
    if( j==-15 || j==-10          || j== 10 ) Colors3D( 6);
    glBegin(GL_LINES);
```

```
        glVertex3f(float(j/20.)*(2./3.),-.75, .5);
        glVertex3f(float(j/20.)*(2./3.),+.75, .5);
       glEnd();
       glLineWidth(1.0f);
      }/*i*/

     for(iPass=1;iPass<3;iPass++)
      {if(iPass==1) m=256;  //Draw the  edited  character:
       if(iPass==2) m=n;    //Draw the existing character:
       j=CharVecs[m][0][2];
       if(j>0)
        {
         glLineWidth(6.0f);
         if(iPass==1) Colors3D( 3);
         if(iPass==2) Colors3D( 8);
         glBegin(GL_LINE_STRIP);
           for(i=1;i<j+1;i++)
            { switch(CharVecs[m][i][0])
               {case(1):    glEnd();
                  glBegin(GL_LINE_STRIP);
                case(2):
                 glVertex3f( (CharVecs[m][i][1]-0.e0)*.05e0*2.e0/3.e0
                            ,(CharVecs[m][i][2]-0.e0)*.05e0
                            ,-.1);                                          break;
                case(3):                                                    break;
               }/*(S.CharVecs[n][i][0])*/
            }/*i*/
          glEnd();
         glLineWidth(1.0f);
        }/*(j>0)*/
      }/*iPass*/

       switch(S.KbdKey)
        { case(48): S.KbdKey=0; //0: Redo the character from scratch:
            for(i=0;i<50;i++) {for(j=0;j<3;j++){CharVecs[256][i][j]=0;}}
            nPts=0;                          CharVecs[256][0][1]=n;         break;
        }
     nPts=CharVecs[256][0][2];
     if( nPts>=0 && nPts<49 && CharVecs[256][0][0]==0 )
      {//Add more points to the character's drawing:
       switch(S.KbdKey)
        { case(49): S.KbdKey=0; //1: Begin another line:
                          nPts      =nPts+1;
                    CharVecs[256][   0][2]=nPts;
                    CharVecs[256][nPts][0]=1;
                    CharVecs[256][nPts][1]=iVmouse[0];
                    CharVecs[256][nPts][2]=iVmouse[1];                  break;
          case(50): S.KbdKey=0; //2: Extend the line:
                         nPts      =nPts+1;
                    CharVecs[256][   0][2]=nPts;
                    CharVecs[256][nPts][0]=2;
                    CharVecs[256][nPts][1]=iVmouse[0];
                    CharVecs[256][nPts][2]=iVmouse[1];                  break;
          case(51): S.KbdKey=0; //3: Erase most recent line:
                    if(nPts<1)                                  break;
                    CharVecs[256][nPts][0]=0;
                    CharVecs[256][nPts][1]=0;
                    CharVecs[256][nPts][2]=0;
                         nPts    =nPts-1;
                    CharVecs[256][   0][2]=nPts;                        break;
          case(52): S.KbdKey=0; //4: End the character
                    CharVecs[256][   0][0]=4;                           break;
        }/*(S.KbdKey)*/
      }/*( nPts>=0 && nPts<49 && S.CharVecs[n][0][2]<=0 )*/

     //List the point set of the revised character, if any:
     if(  CharVecs[256][0][1]>0)
      { j=CharVecs[256][0][2];
        for(k=0;k<(j+1);k++)
         {  sprintf(buffer,"%2i,%5i,%3i\0"
```

```
            ,CharVecs[256][k][0],CharVecs[256][k][1],CharVecs[256][k][2]);
          PrntOrtho(28+k,28, 1, 0,buffer);
        }/*k*/
      }

       switch(S.KbdKey)
        { case(53): S.KbdKey=0; //5: Overwrite CharVecs with edit changes:
           if(CharVecs[256][0][0]!= 4)                                  break;
             CharVecs[256][0][0] = 0;
           if(CharVecs[256][0][1] < 0 || CharVecs[256][0][1]>255)             break;
           if(CharVecs[256][0][2] < 1)   CharVecs[256][0][2]=  0;
           if(CharVecs[256][0][2] >49)   CharVecs[256][0][2]= 49;
           i= CharVecs[256][0][1];      // <-  ASCII code
            for(k=0;k<50;k++)
             { CharVecs[i][k][0]=CharVecs[256][k][0];
               CharVecs[i][k][1]=CharVecs[256][k][1];
               CharVecs[i][k][2]=CharVecs[256][k][2];
             }/*k*/                                              break;
        }/*(S.KbdKey)*/

       switch(S.KbdKey)
        { case(55): S.KbdKey=0; //7: Export the revised character set:
           fprintf(S.fpt," int static CharLoc[256] = {\n\0");
           k=1;
           for(i1=0;i1<256;i1=i1+16)
            {fprintf(S.fpt,"//\0");
              for(j1=0;j1<16;j1++)
               { i=i1+j1;
                 j=CharVecs[i][0][1];
                 if(j==0)  fprintf(S.fpt,"     \0");
                 if(j >0)  fprintf(S.fpt,"  %1c  \0",j);
               }/*j1*/
              fprintf(S.fpt,"%1x\n\0",i1/16);

              for(j1=0;j1<16;j1++)
               { i=i1+j1;
                 j=CharVecs[i][0][1];
                 if(j==0)  fprintf(S.fpt,"   0,\0");
                 if(j >0) {fprintf(S.fpt,"%4i,\0",k);
                           k=k+(1+CharVecs[i][0][2])*3;}
               }/*j1*/
             fprintf(S.fpt,"\n\0");
            }/*i1*/
             fprintf(S.fpt,"};//CharLoc[256]\n\0");

           k2=0;
           fprintf(S.fpt," int static PointLoc[%4i] = {\n\0",k-1);
           k=0;
           for(i1=0;i1<256;i1=i1+16)
            { for(j1=0;j1<16;j1++)
               { i=i1+j1;
                 j=CharVecs[i][0][1];if(j==0) continue;
                 j=CharVecs[i][0][2];
                 for(k1=0;k1<1+j;k1++)
                  {fprintf(S.fpt,"%1i,%3i,%3i,\0"
                     ,CharVecs[i][k1][0],CharVecs[i][k1][1],CharVecs[i][k1][2]);
                   k2=k2+1; if(k2>10000) return;
                   k=k+3; if(k==24) {fprintf(S.fpt,"\n\0"); k=0;}
                  }/*k1*/
               }/*j1*/
            }/*i1*/
           fprintf(S.fpt,"};//PointLoc[]\n\0");
        }/*(S.KbdKey)*/

     glMatrixMode(GL_MODELVIEW ); glPopMatrix();
           /*---------------------------------*/          break;
    case(3):Phase Three- 2D/3D Ee-key-controlled screen graphics:

     if(S.BrkOn==0) //When not character editing...
      { if((S.MouseYaw<80.e0 || S.MouseYaw>100.e0)&& S.Scale>.4e0)
```

```
         { h4Fill(Xyzh,             0.e0, -3.6e0 ,  4.e0-Timer, 1.e0);
           h4Fill(Rpyh,-S.RunTimer*60.e0,  0.e0  , 90.e0      , 1.e0);
           VecText7D(Xyzh,Rpyh, .2e0 ,    3.    ,   5 ,  buffer1  );
         }

        if((S.MouseYaw<-20.e0 || S.MouseYaw>20.e0) && S.Scale>.4e0)
         { h4Fill(Xyzh,-4.6e0,  -4.7e0+S.RunTimer ,  2.6e0, 1.e0);
           h4Fill(Rpyh, 0.e0,  45.e0  ,  0.e0, 1.e0);
           VecText7D(Xyzh,Rpyh, .265e0,   2.    ,   1 ,  buffer2   );
           VecText7D(Xyzh,Rpyh, .150e0,   1.    ,   9 ,  buffer3   );
         }

        if(Timer>16.e0)
        {CubeGrid(11); Teapot(  );}                                          break;
      }/*(S.BrkOn==1)*/

           /*-----------------------------------*/                          break;
   }//S.ThreePhase-------------------------------------------
 }//End AppF4 ------------------------------------------------------------------


 //-------------------------------------------------------------------------------7 9
```