

```

/*App-F5-3dv-Viewer.c
C Opengl (GLUT) Example User Application F5:
2016.06.28.1400cdt JMS- .3dv file viewer
2013.01.17.0900cst JMS- Traveler2/Athlon64/Wi nXPPro/APF9.0: C/OpenGL+CGlut

Herein: void AppF5(void)
*/
#include <3DEnv.h> //<- Environment's variables & functions become available.

void AppF5(void) //-----2016.05.30.1300cdt JMS
{
    //You've entered this application reached by pressing 'F4'
    // "F.: Title_____ yyyy. mm. dd\0"; <- e. g.
    char AppName[] = "F5: 3dv-Viewer 2016.06.28\0";
    // E , N , F , L , R , T , B , D
    double FrustCoes[8]={ 1.2, -4.0 , 6.0, -10.4, 10.4, -6.5, 6.5, 24.0 };
    int i, iColor, nC, nP;
    char Label[80];
    double Xyzh[4], Rpyh[4];
    char static Buffer[1000]=
{"Oscar Garcia copyrighted '3dv.exe' in 1988/1991 - a very useful.\n"
" color, mouse-rotatable, perspective/orthographic vector viewer.\n\n"

"The ASCII text datafile format is remarkably simple:\n"
" Line ( 1) : N ... the number of points.\n"
" Lines ( 2)-(N+1) ; One point per line, a left-handed (-X, Y, Z) triple.\n"
" Line (N+2) : M ... the number of connects.\n"
" Lines (N+3)-(N+M+2) ; one connects per line, which has two integers\n"
" a DOS color [0,15], and a point number.\n\n"

"Color = 0 = 'Black' acts as a 'move' against the black background.\n"
"Color > 0 draws a colored vector to the current connect's point number\n"
" from the preceding connect's point number.\n"
" ... almost nothing to it! & Highly functional! ... Great job, Oscar!\n\n"

"The points of my version are Flight Simulation (X, Y, Z) triples.\n"

"Step through all the connect points of the .3dv file using 'b' & '-', '+'.\n"
" & you can 'scale' in at will using this viewer.\n"
};

switch(S.ThreePhase) // <- driven by the 3D viewer -----
{
    /*-----*/
    case(1): //Phase One- Transfers, initializations, & non-screen computations:
        if(S.AppInit[S.AppNumber]<1)
        {
            S.iTeapot = 1;
            S.AppInit[S.AppNumber] = 1;
            S.Scale = 1.e0; //S.Zoom alters depth bounds
            S.FovYzzoom = 2.e0; //S.FovYzzoom doesn't alter depth bounds
            S.VuMode = 1;
            S.BrkLim = 1; //80
        } //S.AppInit

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

        if(S.Init3dv==0) Read3dvFile();

        /*-----*/
        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);
            break;
}

```

```

sprintf(Label, "      jeff.setterholm@gmail.com \0");
Xyzh[1]=.89e0; VecText7D(Xyzh, Rpyh, .02e0, 1., 6, Label);
sprintf(Label, "This 'App-F5' is freely distributable.\0");
Xyzh[1]=.86e0; VecText7D(Xyzh, Rpyh, .02e0, 1., 8, Label);

if(S.Init3dv<0)
{sprintf(Label, "File read was unsuccessful.\0");
 h4Fill(Xyzh, -.85e0, .4e0, 0.e0, 1.e0);
 h4Fill(Rpyh, 90.00e0, .0e0, 0.e0, 1.e0);
 VecText7D(Xyzh, Rpyh, 0.07e0, 3., 2, Label); return;}

if(S.Brk0n==0)
{sprintf(Label, "Enable 'Breaker' viewer & use + & - keys.\0");
 PrntOrtho(4, 2, 1, 0, Label);

 PrntOrtho(5, 2, 6, 0, "Press 'v' for .3dv file format information.\0");
 if(S.NowView>0)
 { h4Fill(Xyzh, -1.3e0, .5e0, 0.e0, 1.e0);
   h4Fill(Rpyh, 90.00e0, .0e0, 0.e0, 1.e0);
   VecText7D(Xyzh, Rpyh, 0.03e0, 1., 1, Buffer); return;}
 return;}

if(S.Brk0n==1)
{
 S.BrkCur= 0;
 i=Brk(84, 0, 0);
 if(S.BrkLim< 1) S.BrkLim=1;
 if(S.BrkLim>S.nConnects) S.BrkLim=S.nConnects;
 nP=S.iComms[2*(S.BrkLim-1)]-1; S.PoIX= S.Points[nP*3];
 S.PoIY= S.Points[nP*3+1];
 S.PoIZ= S.Points[nP*3+2];

 sprintf(Label, "Breaker -- index --> Point#%8i:\0", nP+1);
 PrntOrtho(4, 2, 1, 0, Label);

 sprintf(Label, "X=%10.4f Y=%10.4f Z=%10.4f\0", S.PoIX, S.PoIY, S.PoIZ);
 PrntOrtho(5, 2, 1, 0, Label);
}
/*-----*/ break;
case(3): //Phase Three- 2D/3D Ee-key-controlled screen graphics:
 if(S.Init3dv<0) {CubeGrid(11); return;}
 if(S.NowView>0) return;

//Mark the PoI (Point of Interest) with a 3D cross:
Colors3D(1);
glLineWidth(2.0f);
glShadeModel(GL_FLAT);
glBegin(GL_LINES);
//Mark the PoI:
glVertex3d(S.PoIX-.01e0, S.PoIY, S.PoIZ);
glVertex3d(S.PoIX+.01e0, S.PoIY, S.PoIZ);
glVertex3d(S.PoIX, S.PoIY-.01e0, S.PoIZ);
glVertex3d(S.PoIX, S.PoIY+.01e0, S.PoIZ);
glVertex3d(S.PoIX, S.PoIY, S.PoIZ-.01e0);
glVertex3d(S.PoIX, S.PoIY, S.PoIZ+.01e0);
glEnd();

//Draw the .3dv graphics object:
glLineWidth(1.0f);
glBegin(GL_LINES);
for(nC=1; nC<S.nConnects; nC++)
{ iColor= S.iComms[2*nC+1]; //<-- These are DOS colors.
 if(iColor<=0) continue; //<-- DOS 'Black' is a move.
 Colors3D(ColorsD0Sto3D[iColor]); //<- ... converted to 3D colors.
 nP=(S.iComms[2*nC-2]-1)*3;
 glVertex3d(S.Points[nP], S.Points[nP+1], S.Points[nP+2]);
 nP=(S.iComms[2*nC]-1)*3;
 glVertex3d(S.Points[nP], S.Points[nP+1], S.Points[nP+2]);
} /*nC*/
glEnd();

```

```

    }//S.ThreePhase-----*/
} //End AppF5 -----

void Read3dvFile(void) //-----2016.06.27.1010cdt JMS
//This is NOT a robust datafile reader... but does support this demonstration.
{char string[80];
 int nC, nP;

 if(S.Init3dv<0) return;

 S.Init3dv=-1; //... assume failure.

 //Read the .3dv file name from App-F5-3dvFileName.ini:
 fprintf(S.fp, "Looking for the 3dv file in 'App-F5-3dvFileName.ini': \n");
 S.fp3dv= fopen("App-F5-3dvFileName.ini\0", "r");
 if(fscanf(S.fp3dv, "%s", S.File3dv) <=0) {return;}
 fclose(S.fp3dv);

 //Open the .3dv file:
 //sprintf(S.File3dv, "DrawingFS.3dv\0");
 fprintf(S.fp, "Opening 3dv file: %-35s\n", S.File3dv);
 S.fp3dv= fopen(S.File3dv, "r");
 if(S.fp3dv==NULL) {fprintf(S.fp, "File not found\n"); return;}
 //Read the points data:
 if(fscanf(S.fp3dv, "%i ", &S.nPoints) <=0) {return;}
 fprintf(S.fp, "Number of points=%10i\n", S.nPoints);
 S.Points=(double*) calloc((S.nPoints)*3, sizeof(double));
 for(nP=0; nP<S.nPoints; nP++)
 {if(fgets(string, 80, S.fp3dv) >0)
 { //fprintf(S.fp, "%-60s\0", string);
 if(sscanf(string, "%lf %lf %lf \n"
 , &S.Points[nP*3+0], &S.Points[nP*3+1], &S.Points[nP*3+2]) >0)
 ; //{fprintf(S.fp, "%12.6f %12.6f %12.6f\n"
 // , S.Points[nP*3+0], S.Points[nP*3+1], S.Points[nP*3+2]);}
 }
 //else S.Init3dv=-2; return; //not all the points were found.
 } /*nP*/
 //Read the point connects & colors data:
 if(fscanf(S.fp3dv, "%i ", &S.nConnects) <=0) {return;}
 fprintf(S.fp, "Number of lines =%10i\n", S.nConnects);
 S.iComms=(int*) calloc((S.nConnects)*2, sizeof(int));
 for(nC=0; nC<S.nConnects; nC++)
 {if(fgets(string, 80, S.fp3dv) >0)
 { //fprintf(S.fp, "%-60s\0", string);
 if(sscanf(string, "%li %li \n"
 , &S.iComms[nC*2+0], &S.iComms[nC*2+1]) >0)
 ; //{fprintf(S.fp, "%12i %4i\n"
 // , S.iComms[nC*2+0], S.iComms[nC*2+1]);}
 }
 //else S.Init3dv=-3; return; //not all the connects were found.
 } /*nC*/
 fclose(S.fp3dv);
 S.Init3dv= 1;

 //free(S.Points);
 //free(S.iComms);
} //End Read3dvFile-----

//-----7 9

```