```
/*3DenvC.h
  3D-Environment, Version c0.60 by  Jeffrey M. Setterholm
  Written in C and realized using Silicon Graphics' OpenGL & GLUT.

  2016.07.10 JMS-                                                Version 0.6
  2016.07.08 JMS- Updated BufferMmMenuView.
  2016.06.28 JMS- Added MmMenuView & BufferMmMenuView.           Version 0.52
  2016.06.24 JMS- Primary header file.                           Version 0.5
  2013.01.17 JMS- Traveler2/Athlon64/WinXPPro/APF9.0:C/OpenGL+CGlut

  C is the native language of OpenGL, and most of the OpenGL documentation.

  I wrote this software using Absoft's Pro Fortran 9.0's 'C' compiler
  & debugger. Per the company's request:
              "Later version of Absoft Pro Fortran do not include
                   our C compiler and do not read .gui files."

  `The OpenGL Utility Toolkit (GLUT) Programming Interface`
                           Mark J. Kilgard, Silicon Graphics, November 13, 1996
  `OpenGL Programming Guide, Second Edition`  ISBN 0-201-46138-2            1997
  `OpenGL Reference Manual`                   ISBN 0-201-46140-4            1997
  `OpenGL SuperBible`                         ISBN 1-57169-164-2            1999
  `C in a Nutshell, A Desktop Quick Reference`ISBN-13: 978-0-596-00697-6    2006

  Function 'HindSight' calls the application programs;
      when S.ThreePhase=case(2) you have an:
  OpenGL ortho screen:+X:right,+Y:up   ,+Z:into screen (left  hand coordinates)
    & when S.ThreePhase=case(3) you have a:
  Flight  Sim  screen:+X:fwd  ,+Y:right,+Z:down        (right hand coordinates)

  --------------- Libraries Used: ------------
  Absoft Pro Fortran 9.0
  USER_LIBS=  "C:\Absoft90\LIB\OPENGL32.LIB"
              "C:\Absoft90\LIB\glut32.lib"
  ------------------------------------------                             */

#include <stdio.h>
#include <math.h>
#include <string.h>
#include <stdlib.h>
#include <stdbool.h>
#include <string.h>
#include <time.h>
#include <gl/glut.h>
#include <limits.h>
//-----------------------------------------------

  /* S. : Simulation Generic Variables */
struct
{
 char   Banner[21];        //Displays- at the upper right of the screen
 char   DateTimeExe[19];   //        - the date of the .exe at upper left
 char   DaTimeLabel[19];   //        - Runtime
 char   Author[35];        //
 char   YourAppOutput[35]; //Output filename. Default:"YourApp.txt"
 FILE   *fpt;              //file pointer to ASCII output file- "YourApp.txt"

 FILE   *fp;               //Switchable file pointer to an ASCII output file
                           //   usually                    *fp=*fpt
 int    NowView;           //Toggles the screen-viewing of details- via 'vV'
 int    NowPrint;          //=0 usually; during one iteration...
                           //=1 by pressing key: 'p' :*fp=*fpt   ->.txt
                           //=2                : 'P' :*fp=stdout ->screen

 int    DepthSelfie;    //Toggle the "`" key to display a screen depth map.

 //Overall program Driver at each Callback:
 int    ThreePhase;     /* = 1 Raw initialization
                           = 2 Update variables display 2D information
                           = 3 Display the modelview graphics of the app. */
```

```
  int    ThreePhaseEnder; //>0 terminates the present callback from OpenGL

  //Application Selection:

  int    AppNumber,      //# of app. you are viewing.              "IpStudy"
         AppNumberNew,   //Use: F_ =[1,...,8] ... the Function keys
         AppInit[10];
  char   AppName[80];

  //Simulation Mode:
  int    SimMode,    //=-1:ReStartup  'Delete" key          Sets Init=0;
         SimModeNew; /*= 0:Startup nAppInUseChanged>0
                       = 1:Reset   F 9   <-Function keys  RunTimer=        0.00;
                       = 2:Hold    F10
                       = 3:Run     F11                    RunTimer=RunTimer+.01;
                       = 4:Stop    F12
                       = 5:Quit    'Q'or'q', pressed twice                    */
  double RunTimer;   //A clock that increments during "Run" +.01 for each iter.
  int    IterTotal;  //Iteration counter- total since launching CEnv.exe
                     //                    e.g.: used to close opening messages.
  int    IterRun;    //                 - total in the current 'Run' (F11->Run)

  // Projection Mode:
  int    VuMode;     /*Viewing mode
                       =-1 2D LeftHand rawGL Left handed   OpenGL screen coords.
                                              +X:Right  , +Y:Up   , +Z:Forward
                       = 0 2D Orthographic   Right handed   Flight Sim. Coords.
                                             ^ +X:Forward, +Y:Right, +Z:Down
                       = 1 2D Perspective    ^       one view         E=0.
                       = 2 3D   Red&Cyan     ^       left&right overlaid
                       = 3 3D Right|Left     ^   right  |   left
                       = 4 3D  Left|Right    ^   left   |  right             */

  int    xWindowFull,yWindowFull;
  double xyWindowRatio;

  double vScrnHVD[2][3]; /* ...defined in ProjOrtho()

    ColorStd() incoming color override:                                      */
  int    ForceRGBA;      /*>0: use ColorStd(ForceRGBA)
                           =0: use ColorStd(   nCol  )
                           <0: force ColorRGBA[4]=S.ColorRGBA[4]             */
  float  ColorRGBA[4];   /*Forced color components, each in: [0.,1.]
                           e.g. "Red" is {1.,0.,0.,1.}; "Cyan"is {0.,1.,1.,1.}*/

  int    iTeapot;        //Use: Tt: ...draws the GLUT Teapot eight ways.

       /*Keyboard primitives:*/
  unsigned char KbdKey;         /* <- K's are upper case */
  int           ArrowKey;
       /*Menu & Mode:*/
  int    Menu,MenuPrev,MenuNew;
  double Timer;

       /*Mouse:*/
  int    MouseLmbWasPressed;
  struct
   {int    iX;
    int    iY;
    double XyzS[4];
   } Mouse[2];

  int    MmMenuView;                                    //Added: 2016.06.26 JMS

  int    MouseInit;             // =0
  int    MouseUse;              // =1
  int    MouseXc   ,MouseXact;     //0. @left
  int    MouseYc   ,MouseYact;     //0. @top

  float  ScreenDepthMap[1200][1920];
```

```
141
142  double Bmouse[4][4];          //Mouse- 6dof
143  double rMouse[2];             //     - position- in(0.,1.)
144  double vMouse[4];             //               - in ortho volume
145  double pMouse[4];             //               - in persp volume
146  float  MouseScreenh[4];                                          //2015.09.15
147
148       /*Screen variables:*/
149  int    iFullScreen;
150  int    iScrnColor;
151       /* ...used by PrntOrtho:*/
152  int    lCharX     ,lCharY;     //  =8->=9, =13 extra lateral space
153  int    nCharMaxX  ,nCharMaxY;
154  int    nCharCenX;
155  double cWidth     ,cHeight;
156  double xPrint, yPrint, zPrint;
157
158  double FovYZzoom;
159
160  //Projection Parameters:  Eight degrees of freedom ("8dof")
161            // ^
162  double E ;// 0 Eye +Y (right) offset  (3D:>0., 2D:=0.)
163            //
164  double N ;// 1    Near  <0.,>-D
165  double F ;// 2    Far   >0.
166            //
167  double L ;// 3 Y: Left         + is  right
168  double R ;// 4    Right >Left
169  double T ;// 5 Z: Top          + is   down
170  double B ;// 6    Bottom >Top
171  double D ;// 7 X: Depth >0.    + is forward
172            //   Your viewpoint will be moved backward, along the -X axis,
173            //                   by distance D; hence the view is "HindSight",
174            //                    ...like watching from the rear of an airplane.
175  double FrustCoes[8]; // =ENFLRTBD
176
177  //Projection:
178  int   iSide;  // =-1:Left eye & monouclar,=+1:Right eye
179  int   e[3];   // Eye offset in use.
180  //          - homgeneous arrays:
181  double Frustum3h[4][4][3];
182  //          - lateral clipping planes:
183  double ClipPlane1[4];
184  double ClipPlane2[4];
185  double ClipPlane3[4];
186  double ClipPlane4[4];
187  //ModelView-  homgeneous arrays:
188  double      Attitudeh[ 4]; //Model- Rotation  3dof
189  double KbdRoll   ;  //^0                                          "<",">"
190  double MousePitch;  //^1
191  double MouseYaw  ;  //^2
192  double W1        ;  //^3
193  double Attitudeh44[4][4];   //Homogeneous 4x4- as a matrix
194  double Attititudeh16[16];   //               - as a linear string- OpenGL i/o)
195  double Attitudeh44inv[4][4];//Inverse of Attitudeh44[4][4]
196  //Translation- Point of interest='PoI'        3dof
197  double PoIX      ;
198  double PoIY      ;
199  double PoIZ      ;
200  double W2        ;
201  //Scale expansion/contraction around the PoI  1dof
202  double Scale     ;
203  double PoIScaleh44[4][4];  //Homogeneous 4x4- as a matrix
204  double PoIScaleh16[16];    //               - as a linear string- OpenGL i/o)
205
206  //Model nutation... aids one-eyed depth perception.
207  int    Nutate;        //Nutation- toggle on/off- controlled by 'n'or'N'
208  double NutateAng;     //        - angle- (pitch=*2, yaw=*5)
209  double Nutate4[4];    //        -attitude (roll,pitch,yaw,1.)
210  double Nutate44[4][4];//        - homogeneous matrix
```

```
211
212  //double DtoR; = (double)asin(1.0)/((double) 90.e0);
213
214  //Break point control: "BrkVal=Brk(BrkRefNum,BrkN,BrkM)":-----------
215  int  BrkOn;           //=0:off,=1:on. When on: +- keys move BrkLim
216  int  BrkLim;          //Progress- counting- Reference
217  int  BrkCur;          //                  - incremental count1<=P<=Pref
218                        //                    return when P=Pref
219  int  BrkVal;
220  int  BrkLineNum;
221  int  BrkN;
222  int  BrkM;
223  int  BrkDone;         //        - completion flag when >0
224  char BrkLab[80];      //        - label- incremental
225  char BrkLabelOut[80]; //               - Output when P=PRef
226
227  //.3dv 3D Data Access (App F5 - not included in v:c0,5):------------
228  int    Init3dv;
229  char   File3dv[35];        //3dv- filename. Default:""DrawingFS.3dv""
230  FILE*  fp3dv;              //   - file pointer
231  size_t nConnects,nPoints;  //   - number of points and connects
232  double *Points;            //   -  points  vector =  nPoints *3
233  int    *iConns;            //   - connects vector = nConnects*2
234
235 } S; //<-Simulation Generic Variables
236
237  char static BufferMmMenuView[1760]={
238 "Holding down the Left-Mouse-Button & moving the mouse rotates ModelSpace.\n\n"
239
240 "The 'm' or 'M' key has toggled this text 'on'.\n"
241 "  Use this if/when the Right-Mouse-Button won't bring up the menu,\n"
242 "    as a reminder that keyboard keys will access the menu functions\n"
243 "    as follows:\n\n"
244 "Main menu:               Apps Access:            Simulation Ctls:\n"
245 "   `:Depth Selfies         F1:Ortho Projection      F9:Reset\n"
246 "   ~:Selfies->.bmp         F2:Mandelbrot Set       F10:Hold\n"
247 "  cC:Screen Color          F3:                     F11:Run\n"
248 "  p :Print to-.txt file    F4:VecText7D Demo       F12:Stop\n"
249 "   P:         -DOS screen  F5:3dv-Viewer            ...runs the clock.\n"
250 "  vV:Viewing info toggle   F6:Invert44-Demo\n"
251 "  bB:'Breaker Viewer'      F7:\n"
252 "  qQ:Quit                  F8:Hindsight Demo\n\n"
253 " ESC:Full/Partial screen toggle(press while in this window)\n\n"
254 "Via Keyboard only:     7dof Model Control:                Left Mouse Button\n"
255 "  Screen affects:        Translation & Scale     Rotation\n"
256 "    eE:Eye Mode            fF: +X: aft/Forward     Yaw & Pitch:    LMB-Down\n"
257 "    nN:model Nutation      rR: +Y:left/Right       Cursor control:    -Up\n"
258 "    -+:'Breaker'index ctl. dD: +Z:  up/Down      <>:Roll\n"
259 "    tT:Teapot              sS: Scale decrease/increase\n"
260 "                       'Home': resets model translation,Scale,& Roll\n\n"
261
262 "3DEnv.exe  V0.6  2016.07.10\0"};
263
264 #include <3DVecText.h>
265
266 //-----------------------------------------------
267
268 ////////////////////////////////////////////////////////////////////////////
269
270 /* ---Subroutines in CEnv.c-----------------------------------------2016.06.24
271 int main(void)
272 int GlutHandoff()     <-- Initializes & launches OpenGL's GLUT interface.
273 void cbUserView(void) <-- Provides overall simulation control; calls HindSight.
274 void DaTime(void)
275 void Colors3D(int nCol)
276 void ColorStd(int nCol)
277
278 //Manual controls:
279 void cbKeyboard(unsigned char Key,int xCursor,int yCursor)
280 void cbSpecialFunctionKeys(int Key,int xCursor,int yCursor)
```

```
281 void cbMouseMotion(int iX,int iY)
282 void cbMouseButtons(int Button, int State, int iX,int iY)
283 void cbMousePassiveMotion(int iX,int iY)
284 void MenuInit(void)
285 void menuAppsUse(int Value)
286 void menuSimUse(int Value)
287 void menuHelpUse(int Value)
288 void menuMainUse(int Value)
289
290 void ProjOrtho(int iOrthoHdr)  <-- Orthogonal screen & boilerplate info.
291 void PrntOrtho(int nRow,int mColumn,int iColorFG,int iColorBg,char PText[80])
292 void PAhXR( double Xyzhin[4], double Rpyhin[4],double hXRout[4][4]) <-- 6dof
293 void Alpha6D(double Pxyzh[4],double ArpyDh[4],int ModeFlag,             ...etc.
294 void ScreenSelfie(void)
295
296 //Homogeneous vector & matrix functions:
297 void h4Fill( double h4[],double h00, double h01, double h02, double h03)
298 void h44Fill( double h44[4][4],                                         ...etc.
299 void Print4(double hin[4],char LabelIn[80])
300 void Print44(double hin[4][4],char LabelIn[80])
301 void MMply44(double hout[4][4],double hin1[4][4],double hin2[4][4])
302 void Invert44(double hin[4][4],double hinverse[4][4],int iPrint, int iView)
303 void Seeh4d(double hin[4],int nRow,int mCol,int Color,char LabelIn[80])
304 void Seeh44d(double hin[4][4],int nRow,int mCol,int Color,char LabelIn[80])
305
306 void Teapot(void)
307 void CubeGrid(int nCol)
308 int Brk(int LineNum, int n,int m)
309 void HindSight(void)       <-- *** Visualization geometry is resolved here. ***
310 void SeeFrustums(void)
311 void VecText7D( double Pxyzh[4],double ArpyDh[4]                        ...etc.
312 void AppF8(void)
313 void AccessYourActiveApp(void)  <-- Assign your new app to an unused F1 - F7.
314 // ---Subroutines in App-F1.c -F2.c, etc.
315 void AppF1(void)
316 void AppF2(void)
317 void AppF3(void)
318 void AppF4(void)
319 void AppF5(void)
320 void AppF6(void)
321 void AppF7(void)
322 // ---Subroutines in CEnv-Wip.c
323 void openglinfo(
324 void ShowChars(
325 // ---Subroutines in
326 // ---Subroutines in
327 void OverWriter(
328 void OverWriterViewer(
329 // ---
330 void HindSight(double FrustDefIn[8],int iEyeIn, double FrustumsOut[4][4][3]) */
331
332    /* Main program entry point                                    2016.06.24
333             argc, argv[]
334   int main( int , char*                                 );                */
335   int main( void                                        );
336
337    /* Initialization of the GLUT callbacks & handoff:             2016.06.24*/
338   int GlutHandoff(void                                  );
339
340    /* CallBack- User(i.e.your) View - draw scene:                 2016.06.24*/
341  void cbUserView(void                                  );
342
343    /* Present date & time string:                                 2016.06.24*/
344  void DaTime(void                                      );
345
346    /* CEnv colors:  nCol                                          2016.06.24*/
347  void Colors3D(    int                                 );
348
349    /* DOS colors:   nCol                                          2016.06.24*/
350  void ColorStd(     int                                 );
```

```
351
352    /* CallBack- keyboard:   Key, xCursor, yCurs                    2016.06.24*/
353  void cbKeyboard( unsigned char,   int  ,   int          );
354
355    /* special function keys (- F1-F12, arrows, etc.):              2016.06.24
356                              Key, xCursor,yCursor                              */
357  void cbSpecialFunctionKeys( int, int    ,int            );
358
359    /*        - mouse motion:  iX , iY                              2016.06.24*/
360  void cbMouseMotion( int, int                              );
361
362    /* CallBack- mouse "button press":                              2016.06.24
363                       Button, State, iX , iY                       */
364  void cbMouseButtons(  int ,  int , int, int             );
365
366    /*        - mouse motion "without button press"                 2016.06.24*/
367  void cbMousePassiveMotion(int,int                         );
368
369    /* Menu Init.:                                                  2016.06.24*/
370  void MenuInit( void                                       );
371
372    /*  menuApps use  value                                         2016.06.24*/
373   void menuAppsUse(   int                                  );
374
375    /*  menuSim use  value                                          2016.06.24*/
376   void menuSimUse(   int                                   );
377
378    /*  menuHelp use  value                                         2016.06.24*/
379   void menuHelpUse(   int                                  );
380
381    /* menuMain use:          Value                                 2016.06.24*/
382  void menuMainUse( int                                     );
383
384    /* Setup an orthographic projection:                            2016.06.24
385                        iOrthoHdr                                   */
386  void ProjOrtho( int                                       );
387
388    /* Print text on the orthographic screen:                       2016.06.24
389                     nRow, mColumn, iColorFG, iColorBg, PText                  */
390  void PrntOrtho( int , int    , int     , int     , char [80]);
391
392    /*        hPosition    , hAttitude    ,to hXR:                 2016.06.24*/
393  void PAhXR( double [4]   ,double [4]    ,double [4][4]   );
394
395    /* 6dof vector-based characters:                                2016.06.24
396                  Pxyzh    , ArpyDh   , ModeFlag, SizeH ,LineWidth,iCol, Label  */
397  void Alpha6D( double [4], double [4], int   , double, float  , int,char [80]);
398
399    /* Screen dump to .bmp image file:                              2016.06.24*/
400  void ScreenSelfie(void                                    );
401
402    /*Fills a 4 element vector with values                         2016.06.24
403                      H4[4],  h0   ,   h01 ,   h02 ,   h03                    */
404  void h4Fill(  double [4] ,double, double, double, double );
405
406    /* Fills a 4x4 matrix with values                               2016.06.24*/
407  void h44Fill( double [4][4] ,                    // h44[4][4]
408                double, double, double, double,   //h00, h01, h02, h03
409                double, double, double, double,   //h10, h11, h12, h13
410                double, double, double, double,   //h20, h21, h22, h23
411                double, double, double, double ); //h30, h31, h32, h33
412
413    /*Prints h4d[4] followed by LabelIn:
414                      h ,    cLabelIn                              2016.06.24*/
415  void Print4(double [4], char [80]                         );
416
417    /* Prints LabelIn & h44d[4][4]:                                 2016.06.24t
418                       hin  ,     LabelIn                                   */
419  void Print44(double [4][4], char [80]                     );
420
```

```
  /* Solves: hout[4][4]=hin1[4][4]*hin2[4][4]                    2016.06.24
                      hout          ,hin1,       hin2                      */
void MMply44( double [4][4],double [4][4],double [4][4]);

  /* 4x4 Matrix inverter hin ,         hinv ,iPrint,iSee          2016.06.24*/
void Invert44(  double [4][4],double [4][4],  int , int );

  /*Displays h44d[4] at screen location nRow,mColumn & a label.
                  h[4],nRow,mCol,iColr,cLabelIn[80]                2016.06.24*/
void Seeh4d(double [4], int, int, int, char [80]);

  /* Displays a 4x4 array at screen location nRow,mCol:          2016.06.24
                h44d , nRow, mCol, iColr, LabelIn                    */
void Seeh44d(double [4][4], int , int , int  , char [80]);

  /* The GLUT Teapot:                                            2016.06.24*/
void Teapot(void                                        );

  /* A 6dof Origin: nCol                                         2016.06.24*/
void CubeGrid(       int                                );

 /*"Breaker" - a progressive visualization tool - rev. A.        2016.06.24
          LineNum, n , m                                             */
  int Brk(  int  ,int,int                                );

  /* Simulation view control:                                    2016.06.24*/
void HindSight(void                                        );

  /* Draws frustum outlines in 3D:                               2016.06.24*/
void SeeFrustums(void                                   );

  /* 7dof vector-based characters:                               2016.06.24
                  Pxyzh   , ArpyDh   , SizeH ,LineWidth, iCol, Label  */
void VecText7D( double [4],double [4], double,  float  ,  int, char [80]);

  /* User Application #F8:                                       2016.06.24*/
void AppF8(void                                         );

  /* The main Switchboard for calling applications:              2016.06.24*/
 void AccessYourActiveApp(void                           );

  //-------------------------------------------------------------------------
  /* ---Subroutines in App-F1.c -F2.c, etc.---------------       2016.06.24

     User Application #F1: Ortho Projection                      2016.06.24*/
void AppF1(void                                         );

  /* User Application #F2: Mandelbrot Set                      2016.07.04.1600cdt*/
void AppF2(void                                         );


  /* User Application #F3: -- Undefined --                       2016.06.24*/
void AppF3(void                                         );

   /* User Application #F4: VecText7D Demo                       2016.06.24*/
void AppF4(void                                         );

  /* User Application #F5: -- Added in v0.5                    2016.06.27.0840cdt*/
void AppF5(void                                         );
  /* User Application #F5: -- Added in v0.5                    2016.06.27.0840cdt*/
void Read3dvFile(void                                   );

  /* User Application #F6:  Demo-Invert44                        2016.06.24*/
void AppF6(void                                         );

  /* User Application #F7:  -- Not included in v0.5              2016.06.24*/
void AppF7(void                                         );
  /* The Matrix OverWriting Solver:--Not included  in v0.5       2016.06.24*/
 void OverWriter(void                                    );
  /* The OverWriter Viewer  -- Not included in v0.5              2016.06.24*/
```

```
  491   void OverWriterViewer(void                                  );
  492
  493 ////////////////////////////////////////////////////////////////////////////
  494 //-------------------------------------------------------------------------7 9
  495
```