Advanced Computer Graphics

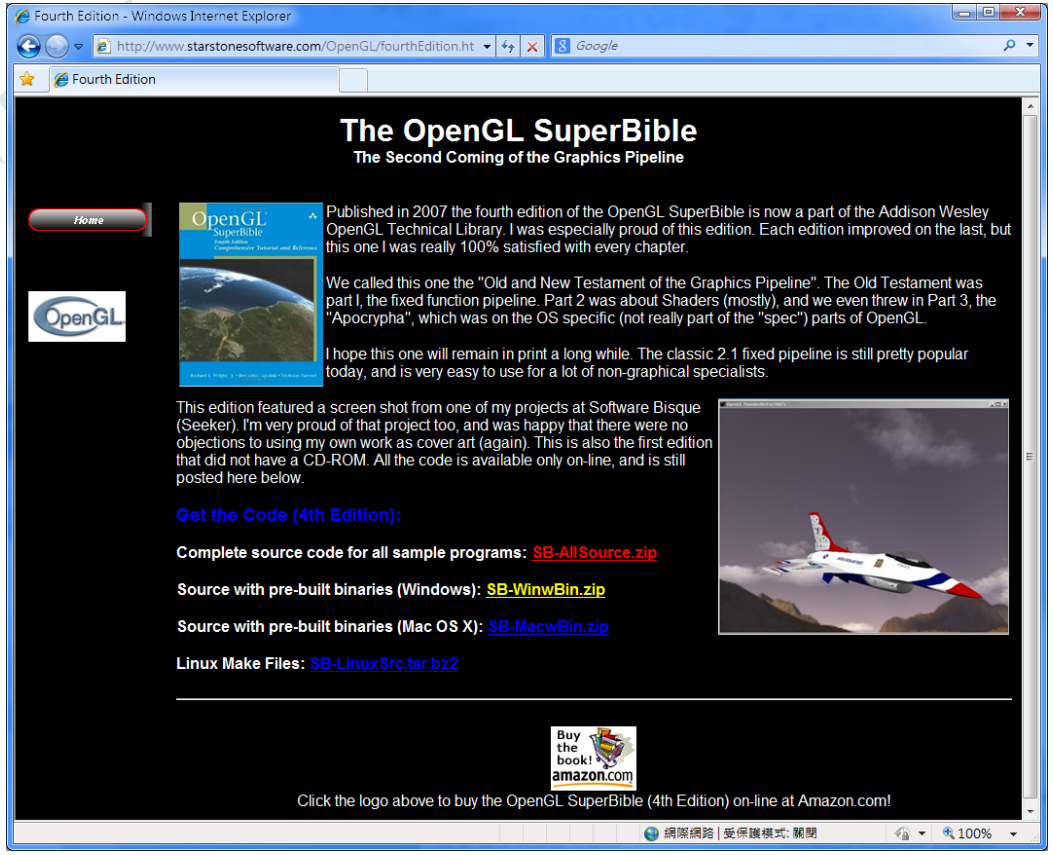
Lecture-08 Introduction to OpenGL-3

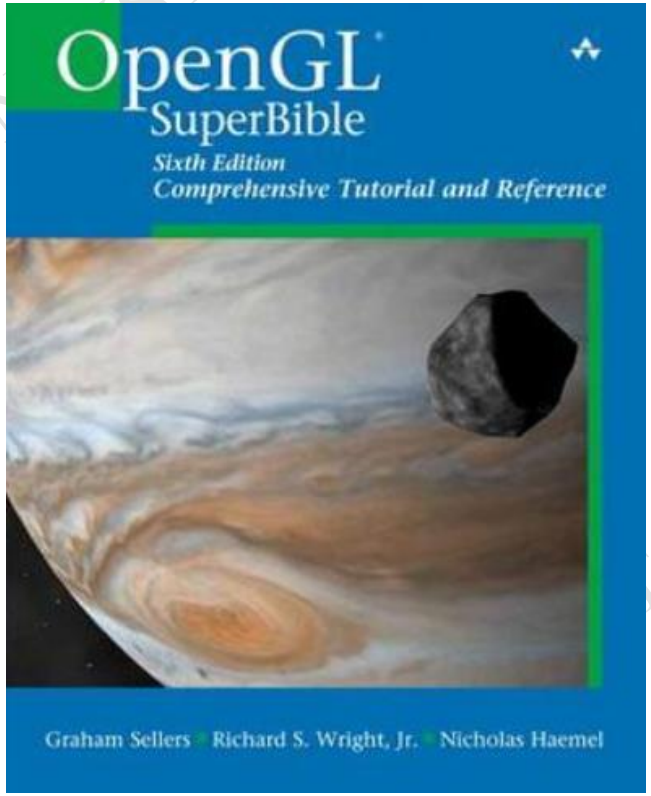
Tzung-Han Lin
National Taiwan University of Science and Technology
Graduate Institute of Color and Illumination Technology
e-mail: thl@mail.ntust.edu.tw

TAIWAN TECH National Taiwan University of Science and Technology

CIT 色彩與照明科技研究所 Graduate Institute of Color and Illumination Technology

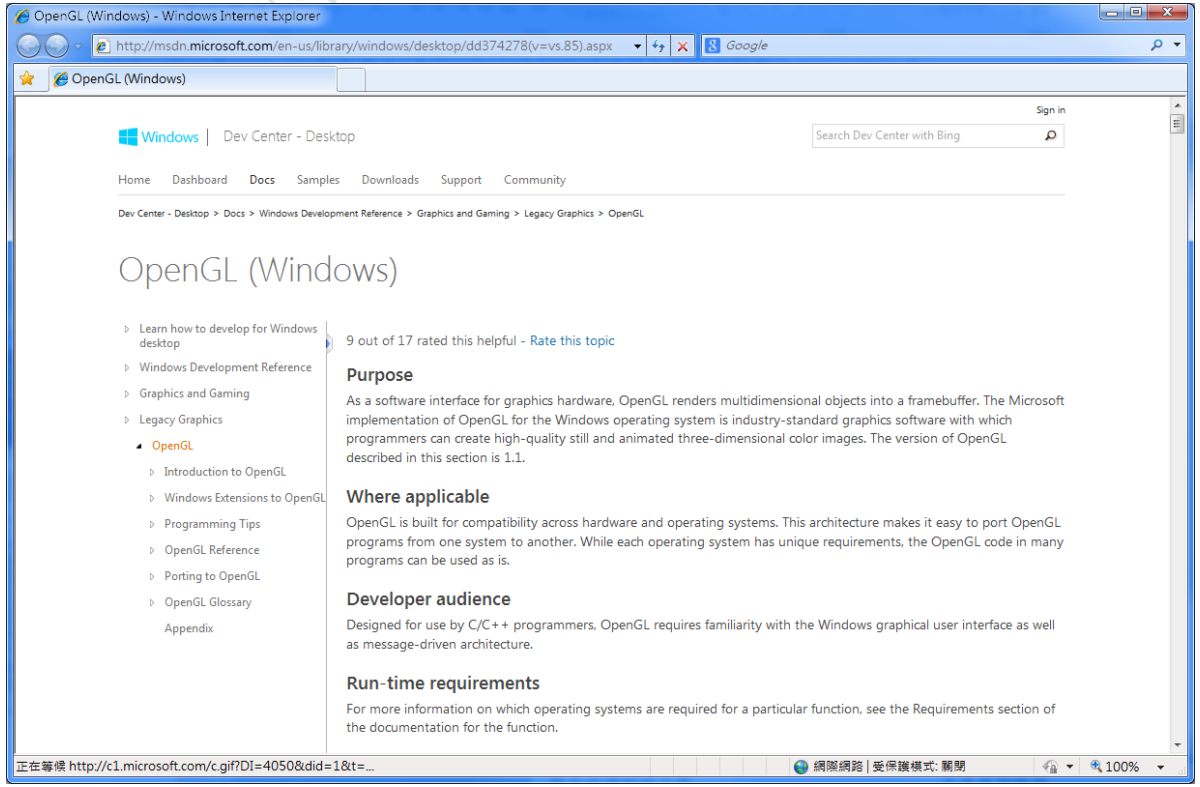
Free source code and example

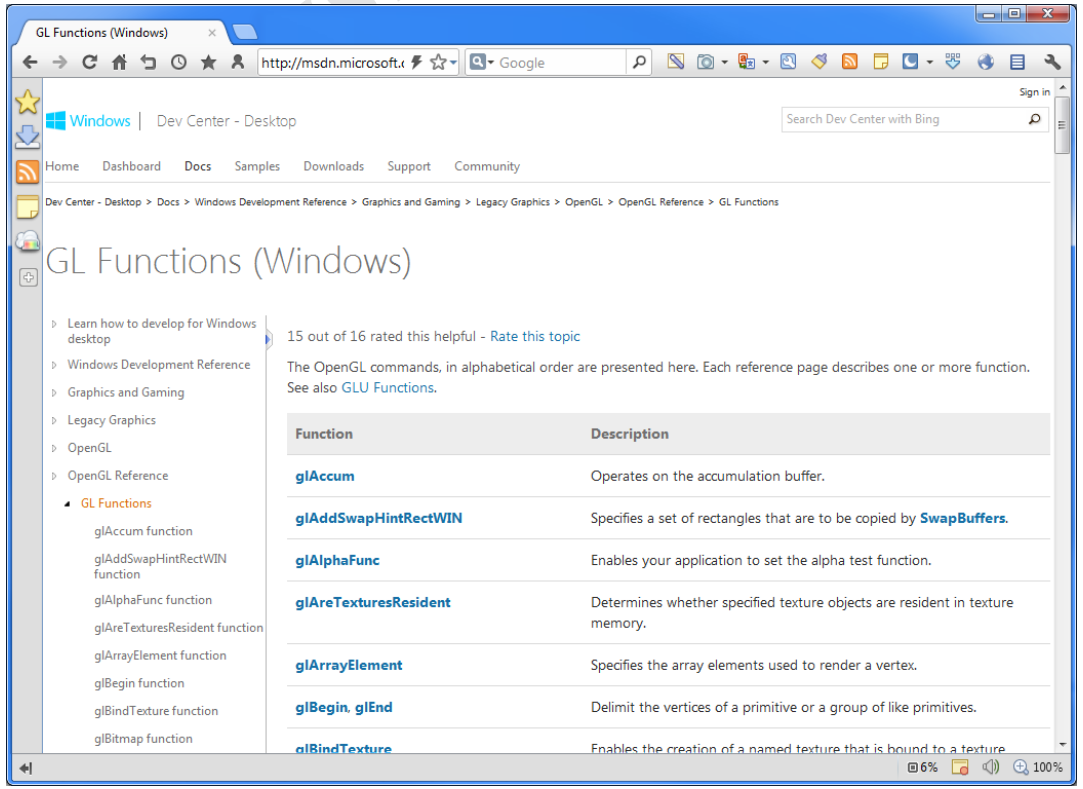




http://www.starstonesoftware.com/OpenGL/index.htm

MSDN documentation





http://msdn.microsoft.com/en-us/library/windows/desktop/dd374278(v=vs.85).aspx

Khronos Group (recommended)

KHRONOS GROUP
CONNECTING SOFTWARE TO SILICON

Developers Conformance Membership News & Events About

Khronos OpenGL® and OpenGL® ES Reference Pages

OpenGL® and OpenGL® ES Reference Pages

Complete descriptions of API commands and shading language functions are provided for the current versions these APIs.

Current API Versions

- OpenGL ES 3.2 and OpenGL ES Shading Language 3.20
- OpenGL 4.5 and OpenGL Shading Language 4.50

Older API Versions

Note that each reference page in the Current Versions pages linked above includes version support information for older versions, so (for example) the OpenGL 3.x reference pages are no longer provided. The older OpenGL ES 3.x and 2.x pages linked here are increasingly out of date, and may eventually be removed.

The OpenGL 2.1 pages are the only source of reference material for GLX, GLU, and the OpenGL Compatibility Profile. They may be retained for that reason, even though they are otherwise useless, or we may eventually incorporate that material in the OpenGL 4.5 pages instead.

- OpenGL ES 3.1 and OpenGL ES Shading Language 3.10
- OpenGL ES 3.0 and OpenGL ES Shading Language 3.00
- OpenGL ES 2.0
- OpenGL ES 1.1
- OpenGL 2.1 (ncluding GLX, GLU, and fixed-function GL compatibility profile APIs)

Reference Page Sources

The Docbook source for the reference pages is available from the OpenGL-Refpages github repository.

API Reference Cards

Quick Reference Cards are available in several formats for OpenGL 3.3 - 4.5 and OpenGL ES 2.0 - 3.2.

Formal Specifications for the OpenGL and OpenGL ES APIs and Shading Languages are available from the OpenGL Registry. These are the authoritative documents for how the APIs and Shading Languages are intended to work. They are also the most difficult to read, being written primarily for device driver implementers, not

https://www.khronos.org/registry/OpenGL-Refpages/

State: Enable / Disable

- glEnalbe() and glDisable()

Example:

- GL_NORMALIZE

  If enabled, normal vectors specified with **glNormal** are scaled to unit length after transformation

- GL_LIGHTING

  If enabled, use the current lighting parameters to compute the vertex color or index. If disabled, associate the current color or index with each vertex

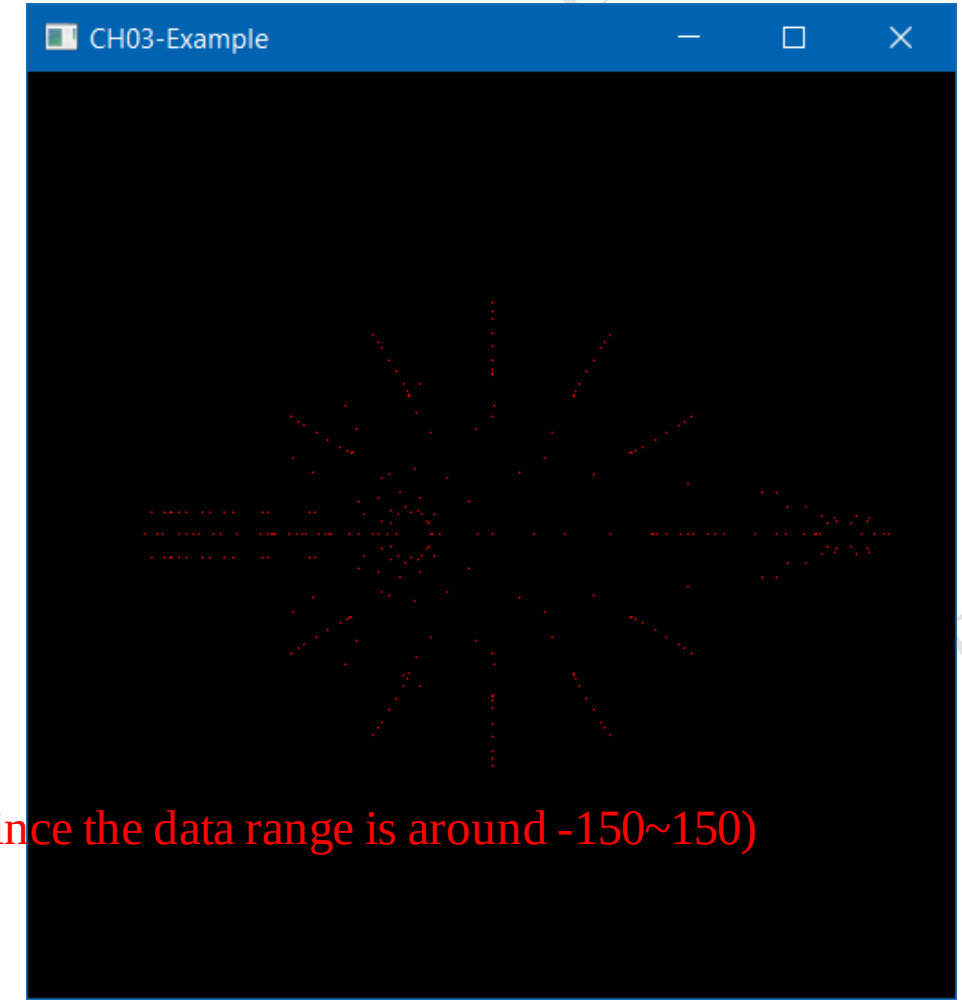
Go though a 2D vector:

```
from OpenGL.GLUT import *

teapotVertex=[[70.0000,0.0000,120.0000],
[60.5630,-35.5704,120.0000],
[59.8820,-35.1704,124.3750],
[69.2130,0.0000,124.3750],
[61.9649,-36.3938,124.3750],
[71.6204,0.0000,124.3750],
[64.8889,-38.1111,120.0000],
[75.0000,0.0000,120.0000],
[35.5704,-60.5630,120.0000],
[35.1704,-59.8820,124.3750],
[36.3938,-61.9649,124.3750],
[38.1111,-64.8889,120.0000],
[0.0000,-70.0000,120.0000],
[0.0000,-69.2130,124.3750],
[0.0000,-71.6204,124.3750],
[0.0000,-75.0000,120.0000],
[-37.5704,-60.5630,120.0000]
```

```
def drawTeapot():
    glColor3f(1,0,0)
    glScalef(0.005,0.005,0.005)
    glBegin(GL_POINTS)
    for pVec in teapotVertex:
        glVertex3fv(pVec)
    glEnd()
```

glScalef (since the data range is around -150~150)



Go though two 2D vectors

```
teapotVertex=[[70.0000,0.0000,120.0000],
[60.5630,-35.5704,120.0000],
[59.8820,-35.1704,124.3750],
[69.2130,0.0000,124.3750],
[61.9649,-36.3938,124.3750],
[71.6204,0.0000,124.3750],
[64.8889,-38.1111,120.0000],
[75.0000,0.0000,120.0000]
```

Vertex Data

```
teapotFace=[[0,1,2],
[2,3,0],
[3,2,4],
[4,5,3],
[5,4,6],
[6,7,5]
```

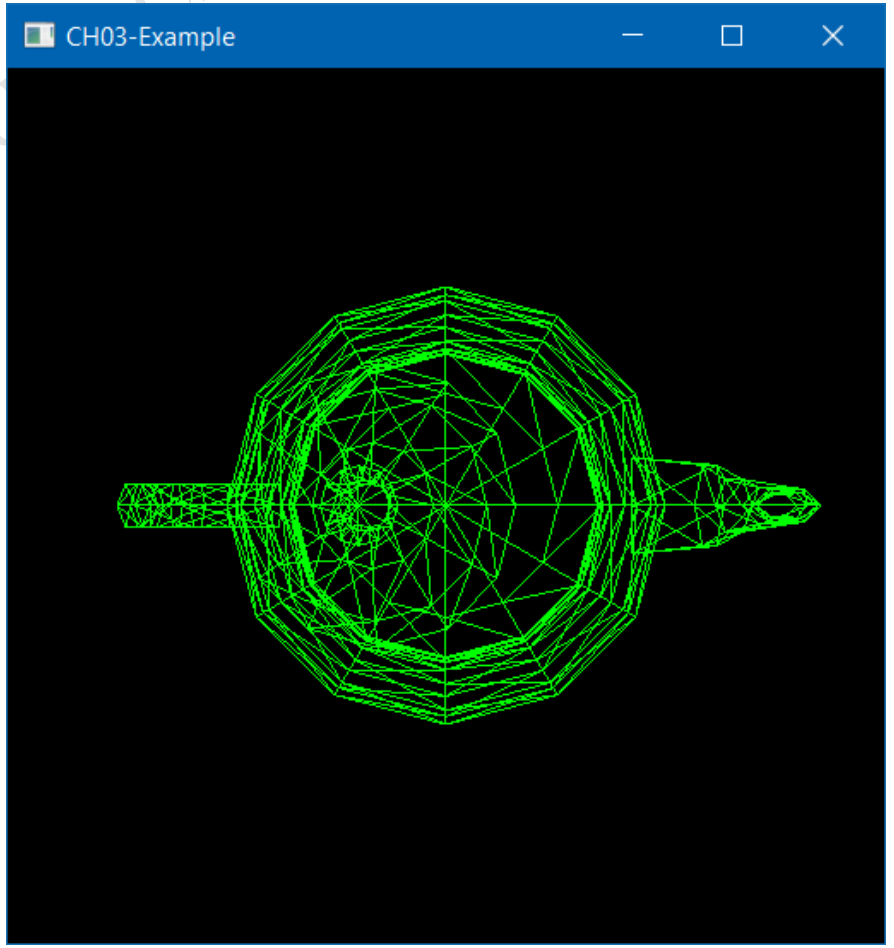
Face ID
(following 3D OBJ file format)

```
def drawTeapot():
    glColor3f(0,1,0)
    glScalef(0.005,0.005,0.005)
    glBegin(GL_LINES)
    for fID in teapotFace:
        glVertex3fv(teapotVertex[fID[0]])
        glVertex3fv(teapotVertex[fID[1]])
        glVertex3fv(teapotVertex[fID[1]])
        glVertex3fv(teapotVertex[fID[2]])
        glVertex3fv(teapotVertex[fID[2]])
        glVertex3fv(teapotVertex[fID[0]])
    glEnd()
```

One line

One line

One line



Go though two 2D vectors (without lighting)

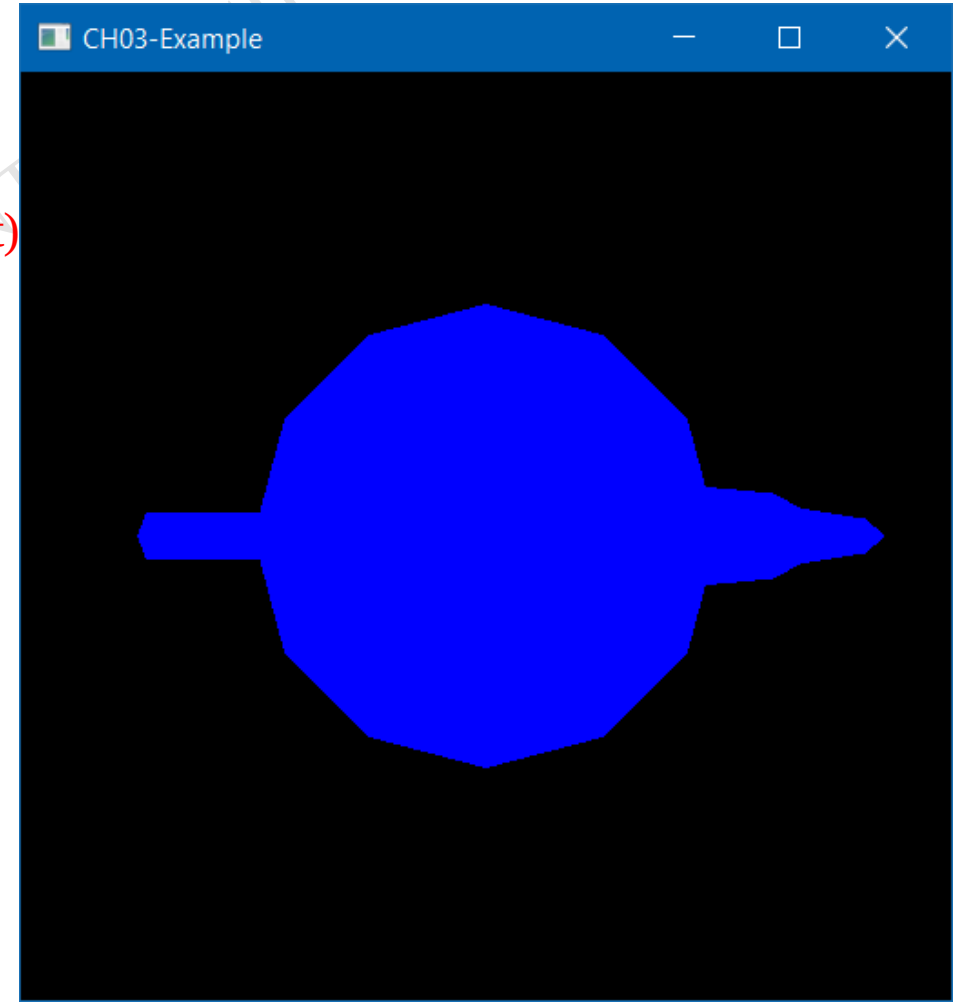
```
teapotVertex=[[70.0000,0.0000,120.0000],
[60.5630,-35.5704,120.0000],
[59.8820,-35.1704,124.3750],
[69.2130,0.0000,124.3750],
[61.9649,-36.3938,124.3750],
[71.6204,0.0000,124.3750],
[64.8889,-38.1111,120.0000],
[75.0000,0.0000,120.0000]
```

Vertex Data

```
teapotFace=[[0,1,2],
[2,3,0],
[3,2,4],
[4,5,3],
[5,4,6],
[6,7,5]
```

Face ID
(following 3D OBJ file format)

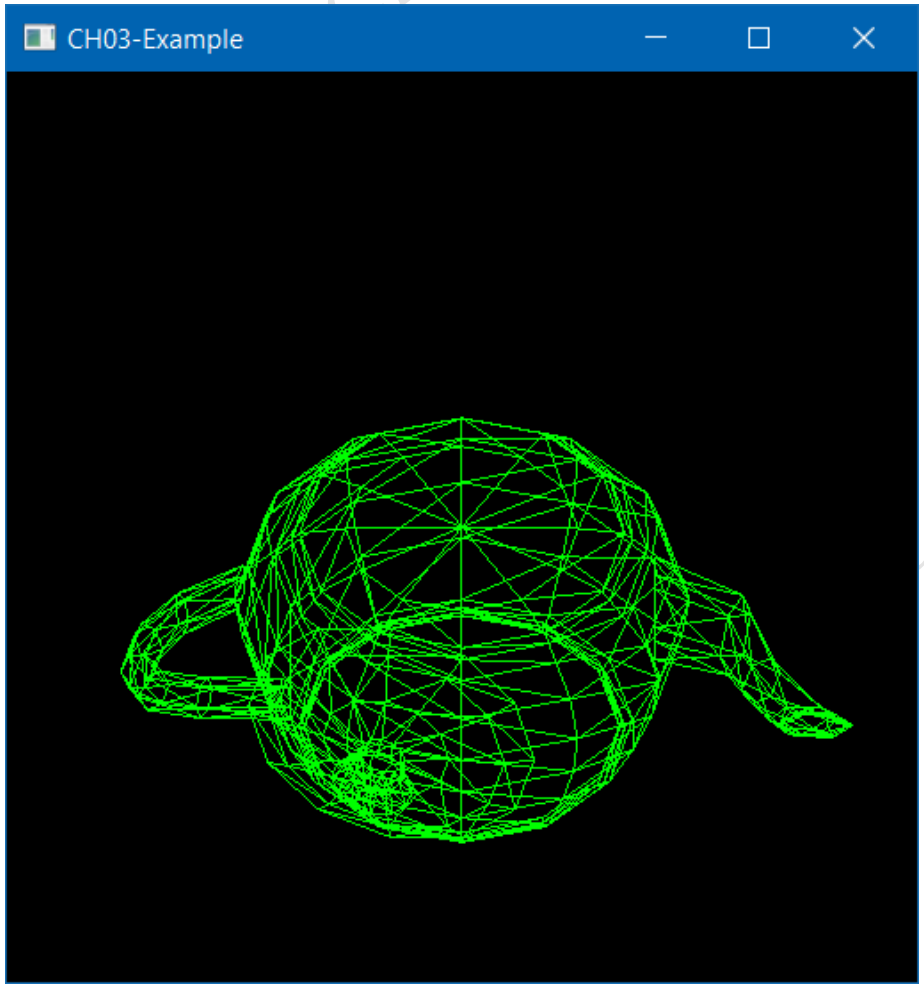
```
def drawTeapot():
    glColor3f(0,0,1)
    glScalef(0.005,0.005,0.005)
    glBegin(GL_TRIANGLES)
    for fID in teapotFace:
        glVertex3fv(teapotVertex[fID[0]])
        glVertex3fv(teapotVertex[fID[1]])
        glVertex3fv(teapotVertex[fID[2]])
    glEnd()
```



Rotation: glRotatef

```
void glRotatef( GLfloat angle,
                GLfloat x,
                GLfloat y,
                GLfloat z);
```

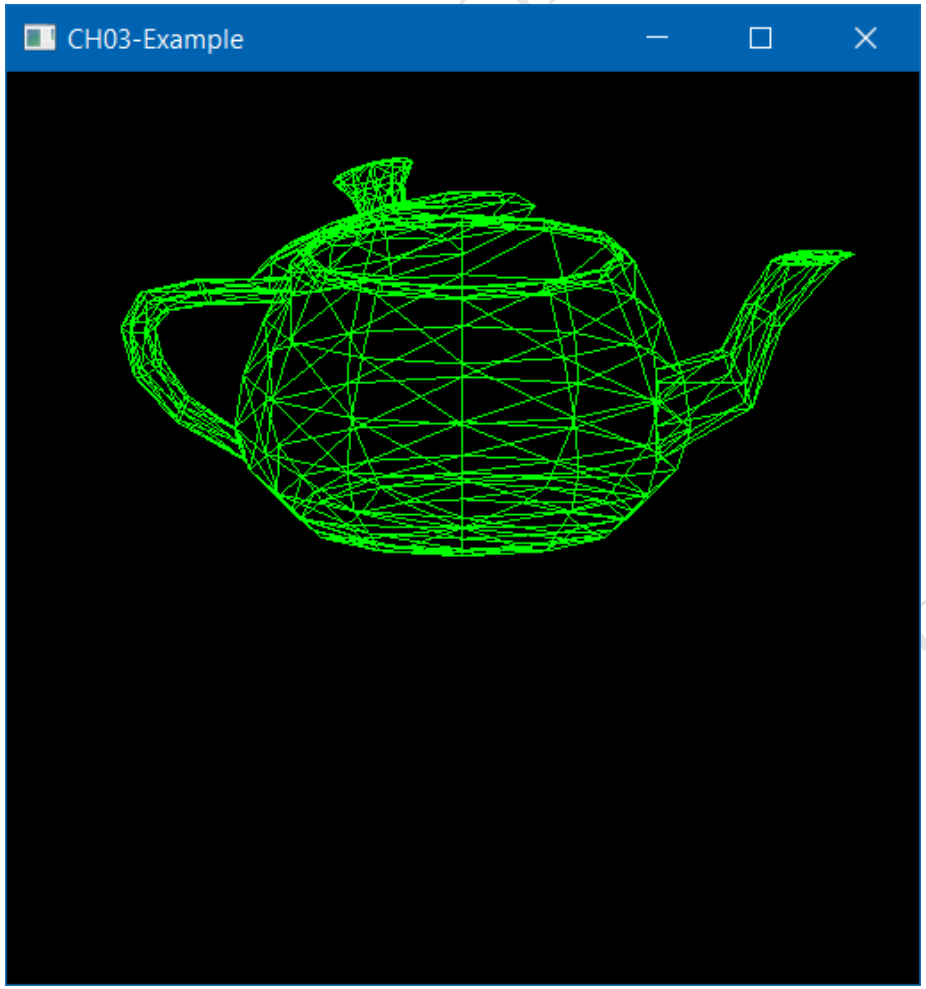
```
def drawTeapot():
    glColor3f(0,1,0)
    glRotatef(45,1,0,0)
    glScalef(0.005,0.005,0.005)
    glBegin(GL_LINES)
    for fID in teapotFace:
        glVertex3fv(teapotVertex[fID[0]])
        glVertex3fv(teapotVertex[fID[1]])
        glVertex3fv(teapotVertex[fID[1]])
        glVertex3fv(teapotVertex[fID[2]])
        glVertex3fv(teapotVertex[fID[2]])
        glVertex3fv(teapotVertex[fID[0]])
    glEnd()
```



Rotation: glutPostRedisplay (dynamic scene)

```
6
7   angle = 0
8
```

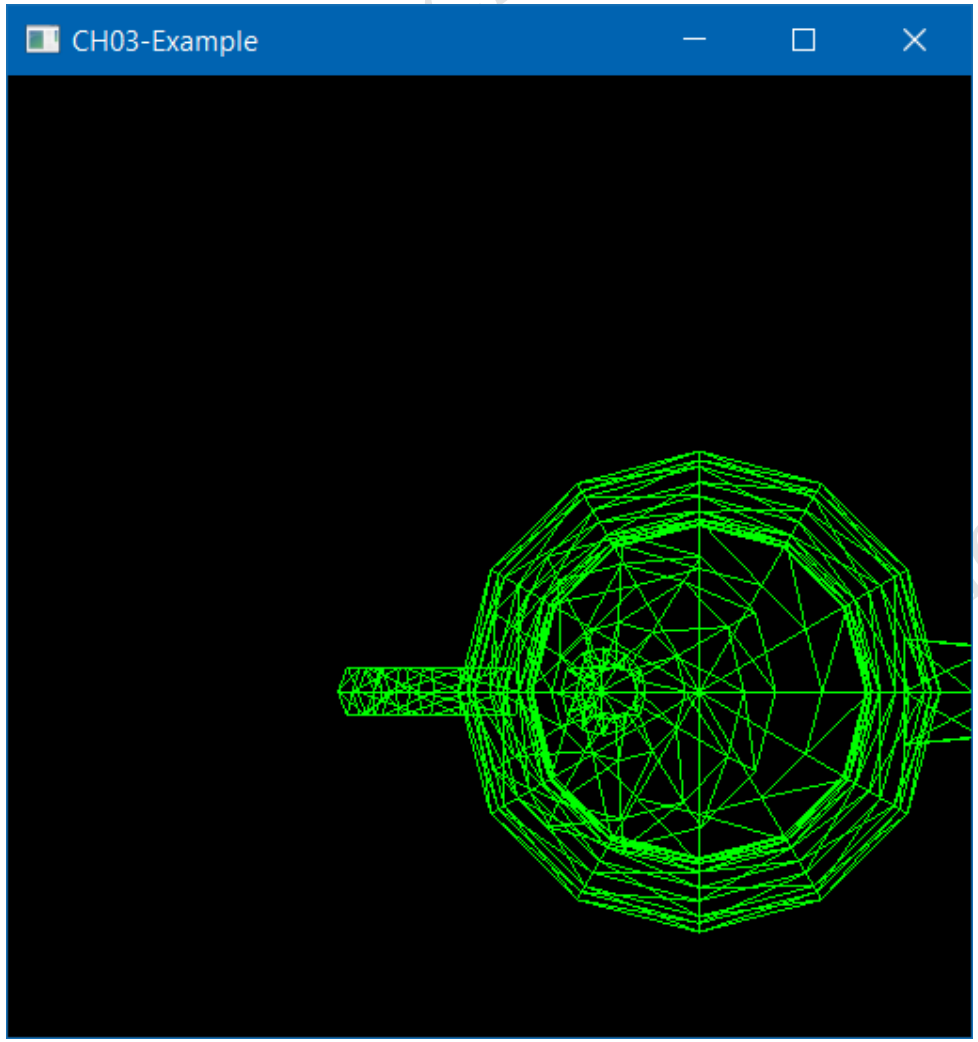
```
866
867 def drawTeapot():
868     glColor3f(0,1,0)
869     glRotatef(angle,1,0,0)
870     glScalef(0.005,0.005,0.005)
871     glBegin(GL_LINES)
872     for fID in teapotFace:
873         glVertex3fv(teapotVertex[fID[0]])
874         glVertex3fv(teapotVertex[fID[1]])
875         glVertex3fv(teapotVertex[fID[1]])
876         glVertex3fv(teapotVertex[fID[2]])
877         glVertex3fv(teapotVertex[fID[2]])
878         glVertex3fv(teapotVertex[fID[0]])
879     glEnd()
880
881
882 def display():
883     glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
884     global angle
885     angle = angle +1
886     glPushMatrix()
887     drawTeapot()
888     glPopMatrix()
889     glutSwapBuffers()
890     glutPostRedisplay()
891
```



Translation: glTranslatef (dynamic scene)

```
7   theda = 0
8   phy = 0
9
```

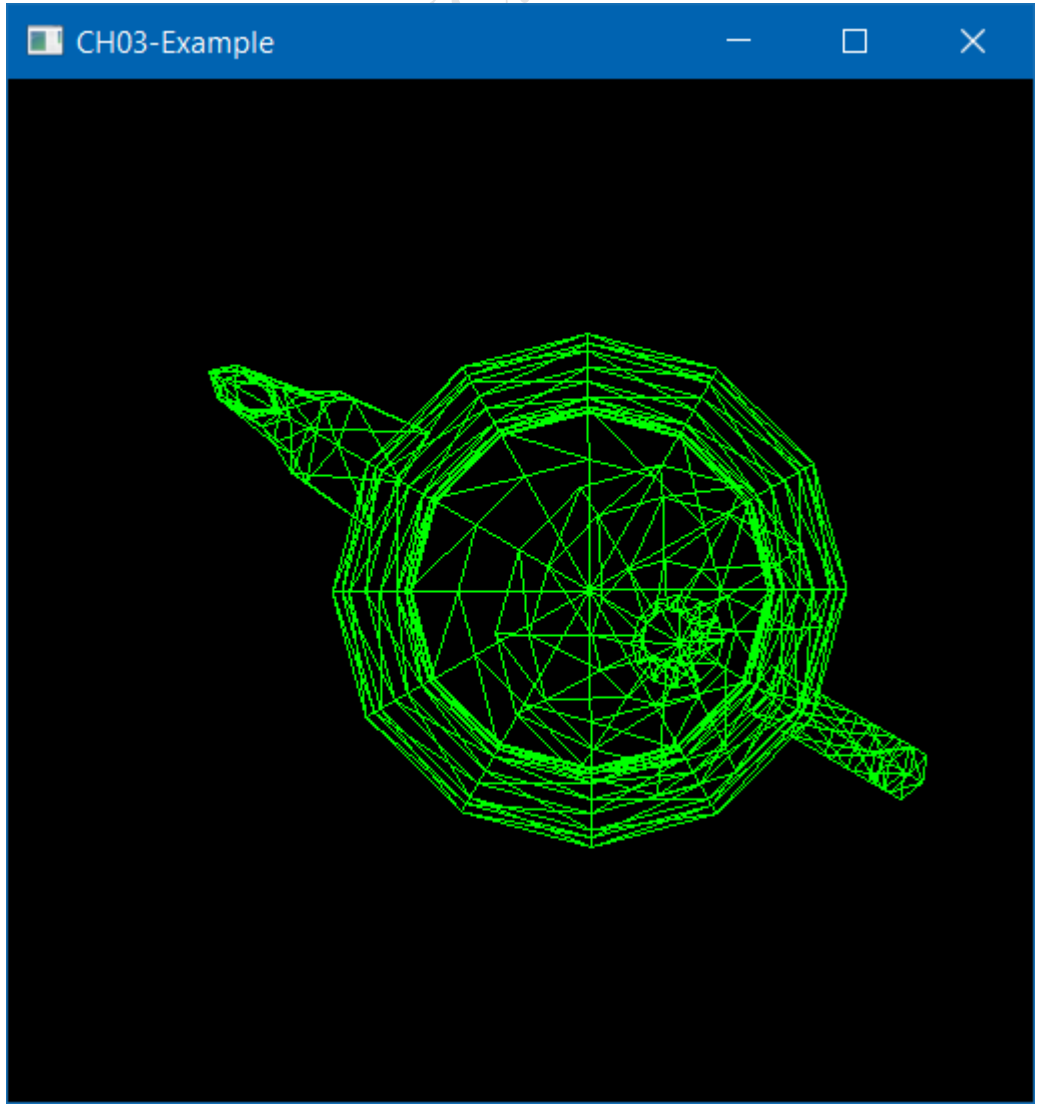
```
869 def drawTeapot():
870     glColor3f(0,1,0)
871     glTranslatef(0.5*cos(theda),0.3*sin(phy),0)
872     glScalef(0.005,0.005,0.005)
873     glBegin(GL_LINES)
874     for fID in teapotFace:
875         glVertex3fv(teapotVertex[fID[0]])
876         glVertex3fv(teapotVertex[fID[1]])
877         glVertex3fv(teapotVertex[fID[1]])
878         glVertex3fv(teapotVertex[fID[2]])
879         glVertex3fv(teapotVertex[fID[2]])
880         glVertex3fv(teapotVertex[fID[0]])
881     glEnd()
882
883
884 def display():
885     glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
886     global theda, phy
887     theda = theda + 0.06
888     phy = phy + 0.02
889     glPushMatrix()
890     drawTeapot()
891     glPopMatrix()
892     glutSwapBuffers()
893     glutPostRedisplay()
894
```



Rotation then Translation

```
6
7   theda = 0
8   angle = 0
9
```

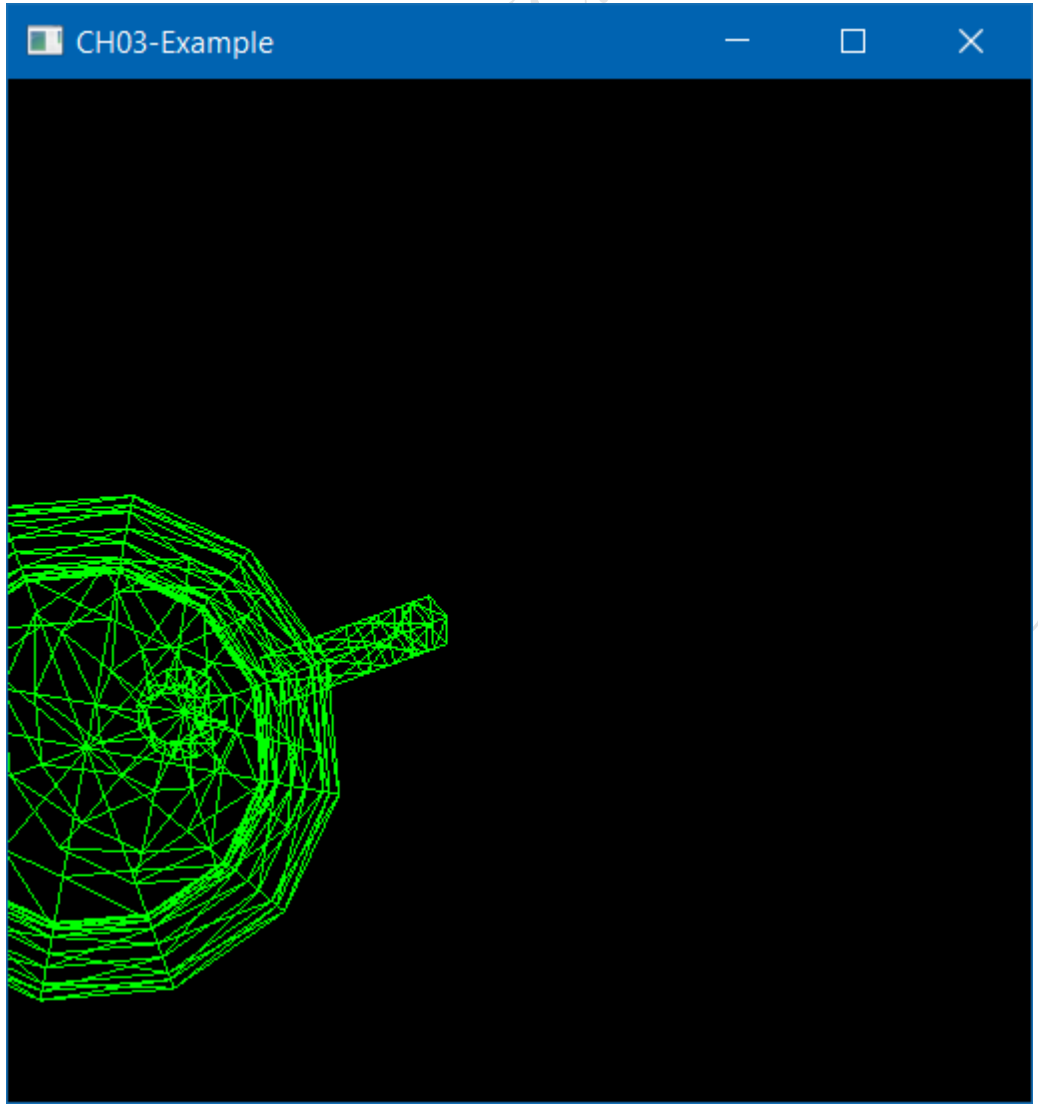
```
868 def drawTeapot():
869     glColor3f(0,1,0)
870     glTranslatef(1*cos(theda),0,0)
871     glRotatef(angle,0,0,1)
872     glScalef(0.005,0.005,0.005)
873     glBegin(GL_LINES)
874     for fID in teapotFace:
875         glVertex3fv(teapotVertex[fID[0]])
876         glVertex3fv(teapotVertex[fID[1]])
877         glVertex3fv(teapotVertex[fID[1]])
878         glVertex3fv(teapotVertex[fID[2]])
879         glVertex3fv(teapotVertex[fID[2]])
880         glVertex3fv(teapotVertex[fID[0]])
881     glEnd()
882
883
884 def display():
885     glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
886     global theda, phy, angle
887     theda = theda + 0.1
888     angle = angle + 5
889     glPushMatrix()
890     drawTeapot()
891     glPopMatrix()
892     glutSwapBuffers()
893     glutPostRedisplay()
894
```



Translation then Rotation

```
6
7   theda = 0
8   angle = 0
9
```

```
868 def drawTeapot():
869     glColor3f(0,1,0)
870     glRotatef(angle,0,0,1)
871     glTranslatef(1*cos(theda),0,0)
872     glScalef(0.005,0.005,0.005)
873     glBegin(GL_LINES)
874     for fID in teapotFace:
875         glVertex3fv(teapotVertex[fID[0]])
876         glVertex3fv(teapotVertex[fID[1]])
877         glVertex3fv(teapotVertex[fID[1]])
878         glVertex3fv(teapotVertex[fID[2]])
879         glVertex3fv(teapotVertex[fID[2]])
880         glVertex3fv(teapotVertex[fID[0]])
881     glEnd()
882
883
884 def display():
885     glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
886     global theda, angle
887     theda = theda + 0.1
888     angle = angle + 5
889     glPushMatrix()
890     drawTeapot()
891     glPopMatrix()
892     glutSwapBuffers()
893     glutPostRedisplay()
894
```

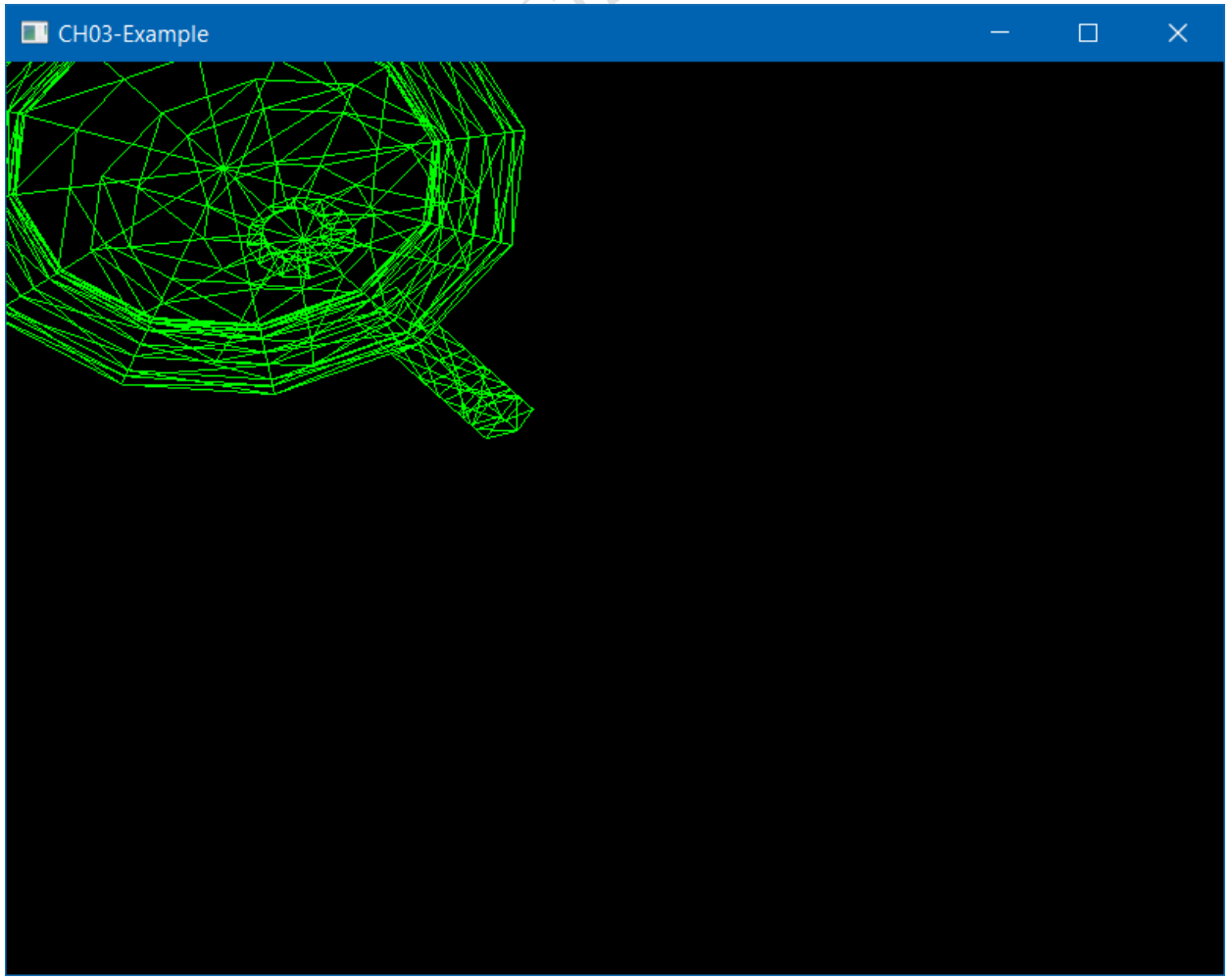


Set canvas size (ratio 4:3)

- Induces compressed images

```
 9
10  windowWidth = 800
11  windowHeight = 600
12
```

```
904
905  glutInit()
906  glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA)
907  glutCreateWindow(b'CH03-Example')
908  glutReshapeWindow(windowWidth,windowHeight)
909  glutReshapeFunc(reshape)
910  glutDisplayFunc(display)
911  glutKeyboardFunc(keyboard)
912  glutMainLoop()
913
```



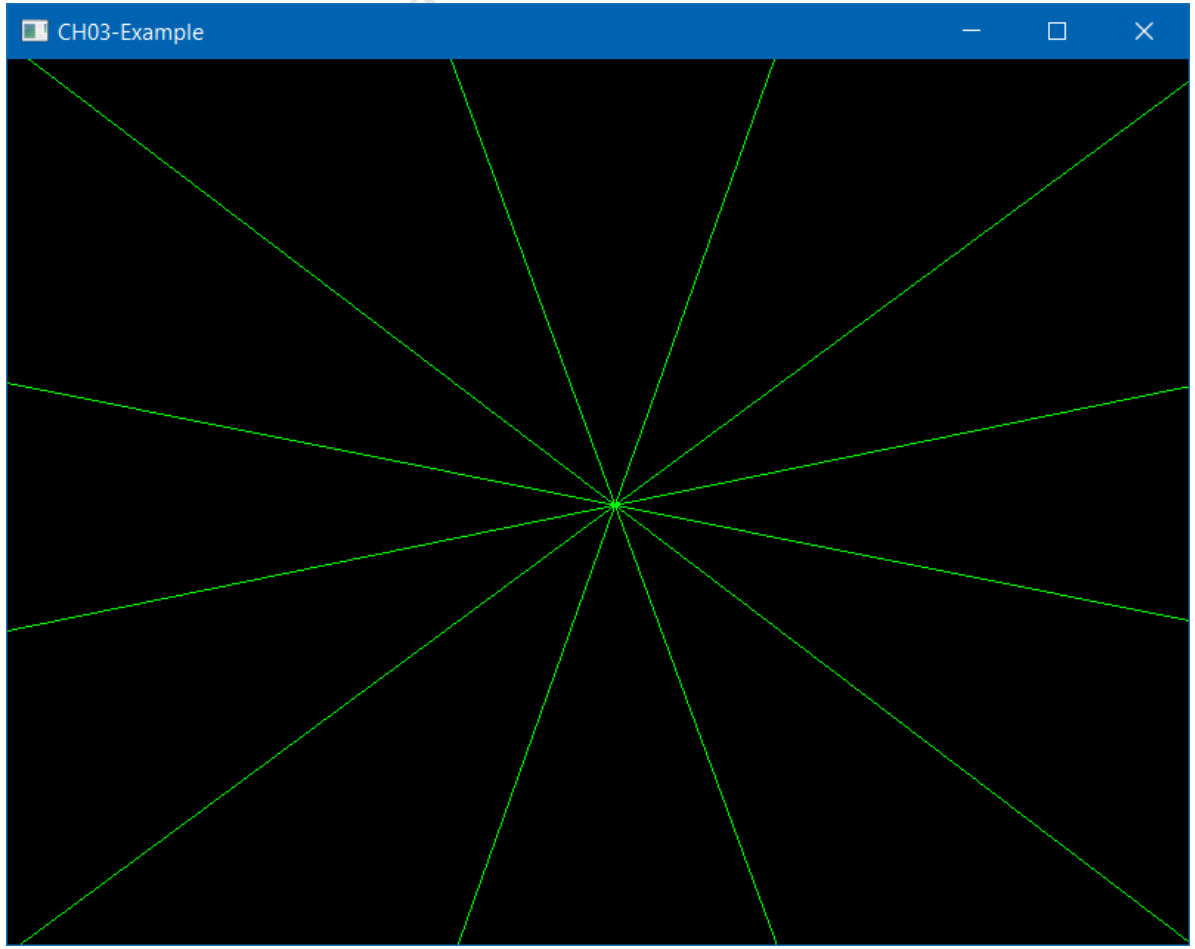
Set regular size of original data

```
teapotVertex=[[70.0000,0.0000,120.0000],
[60.5630,-35.5704,120.0000],
[59.8820,-35.1704,124.3750],
[69.2130,0.0000,124.3750],
[61.9649,-36.3938,124.3750],
[71.6204,0.0000,124.3750],
[64.8889,-38.1111,120.0000],
[75.0000,0.0000,120.0000],
[35.5704,-60.5630,120.0000],
[35.1704,-59.8820,124.3750],
[36.3938,-61.9649,124.3750],
[38.1111,-64.8889,120.0000],
[0.0000,-70.0000,120.0000],
[0.0000,-69.2130,124.3750],
[0.0000,-71.6204,124.3750],
[0.0000,-75.0000,120.0000],
[-37.5704,-60.5630,120.0000],
[-35.7630,-59.8820,124.3750],
[-36.4678,-61.9649,124.3750],
```

Note the range is around -150~150

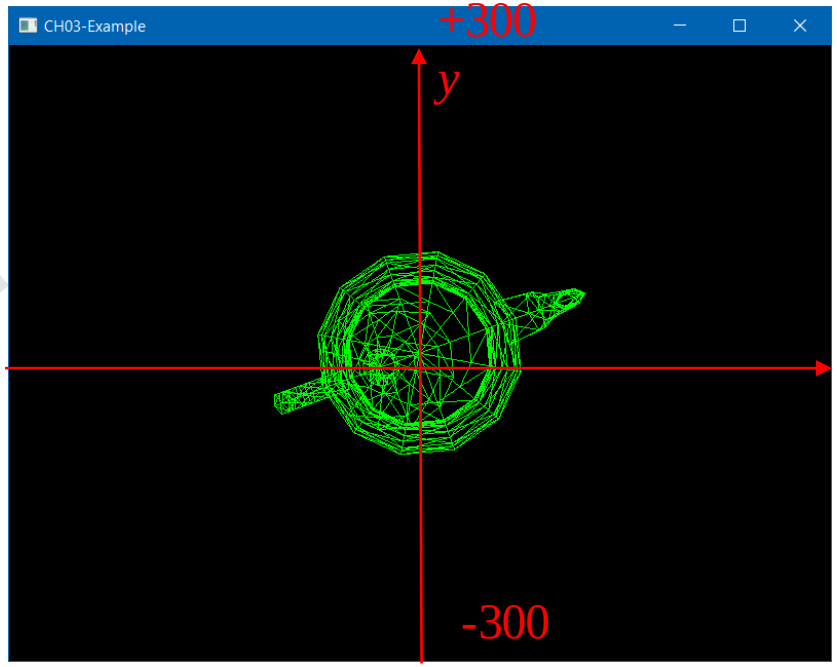
```
def drawTeapot():
    glColor3f(0,1,0)
    glRotatef(angle,0,0,1)
    glTranslatef(1*cos(theda),0,0)
    glBegin(GL_LINES)
    for fID in teapotFace:
        glVertex3fv(teapotVertex[fID[0]])
        glVertex3fv(teapotVertex[fID[1]])
        glVertex3fv(teapotVertex[fID[1]])
        glVertex3fv(teapotVertex[fID[2]])
        glVertex3fv(teapotVertex[fID[2]])
        glVertex3fv(teapotVertex[fID[0]])
    glEnd()
```

We do not do “scaling” operation



Set “View volume”(orthographic projection): glOrtho

```
886  def display():
887      glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
888      glMatrixMode(GL_PROJECTION)
889      glLoadIdentity()
890      glViewport(0, 0, windowWidth, windowHeight)
891      glOrtho(-float(windowWidth)/2.0,float(windowWidth)/2.0,-float(windowHeight)/2.0,float(windowHeight)/2.0,-windowHeight*10.0,windowHeight*10.0)
892      global theda, angle
893      theda = theda + 0.1
894      angle = angle + 5
895      glPushMatrix()
896      drawTeapot()
897      glPopMatrix()
898      glutSwapBuffers()
899      glutPostRedisplay()
900
```



Note ratio
(teapot is not compressed)

+400
x

Set “Camera Position”: gluLookAt

Name

gluLookAt — define a viewing transformation

C Specification

```
void gluLookAt( GLdouble eyeX,
                GLdouble eyeY,
                GLdouble eyeZ,
                GLdouble centerX,
                GLdouble centerY,
                GLdouble centerZ,
                GLdouble upX,
                GLdouble upY,
                GLdouble upZ);
```

Parameters

eyeX, *eyeY*, *eyeZ*

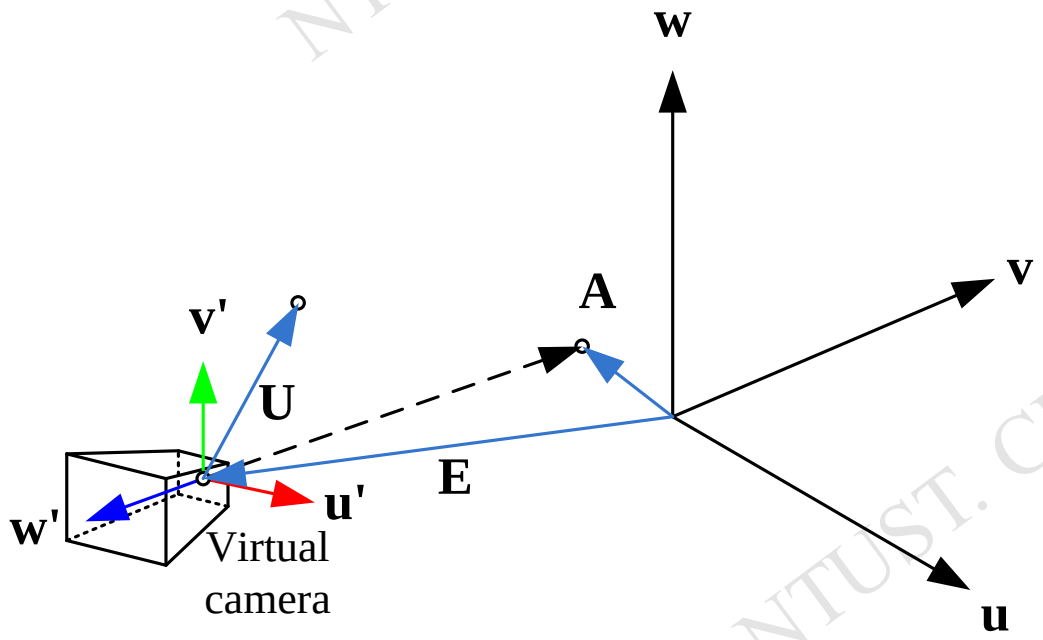
Specifies the position of the eye point.

centerX, *centerY*, *centerZ*

Specifies the position of the reference point.

upX, *upY*, *upZ*

Specifies the direction of the *up* vector.



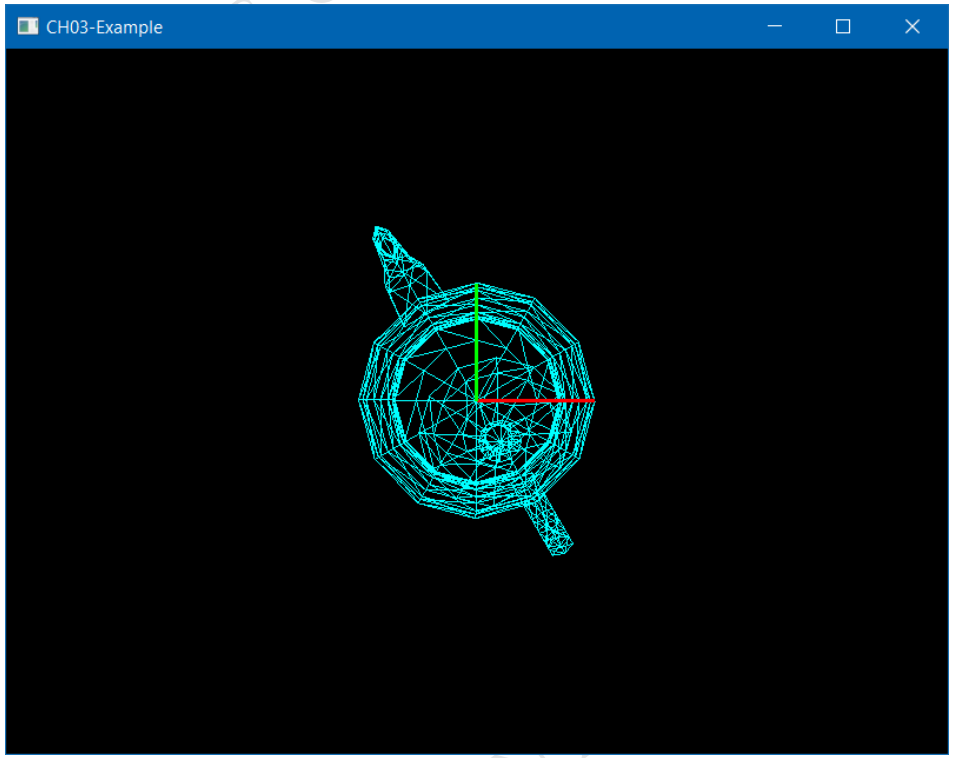
Set camera's position

```
871 def drawTeapot():
872     glLineWidth(1)
873     glColor3f(0,1,1)
874     glRotatef(angle,0,0,1)
875     glTranslatef(1*cos(theda),0,0)
876     glBegin(GL_LINES)
877     for fID in teapotFace:
878         glVertex3fv(teapotVertex[fID[0]])
879         glVertex3fv(teapotVertex[fID[1]])
880         glVertex3fv(teapotVertex[fID[1]])
881         glVertex3fv(teapotVertex[fID[2]])
882         glVertex3fv(teapotVertex[fID[2]])
883         glVertex3fv(teapotVertex[fID[0]])
884     glEnd()
```

Draw cyan teapot

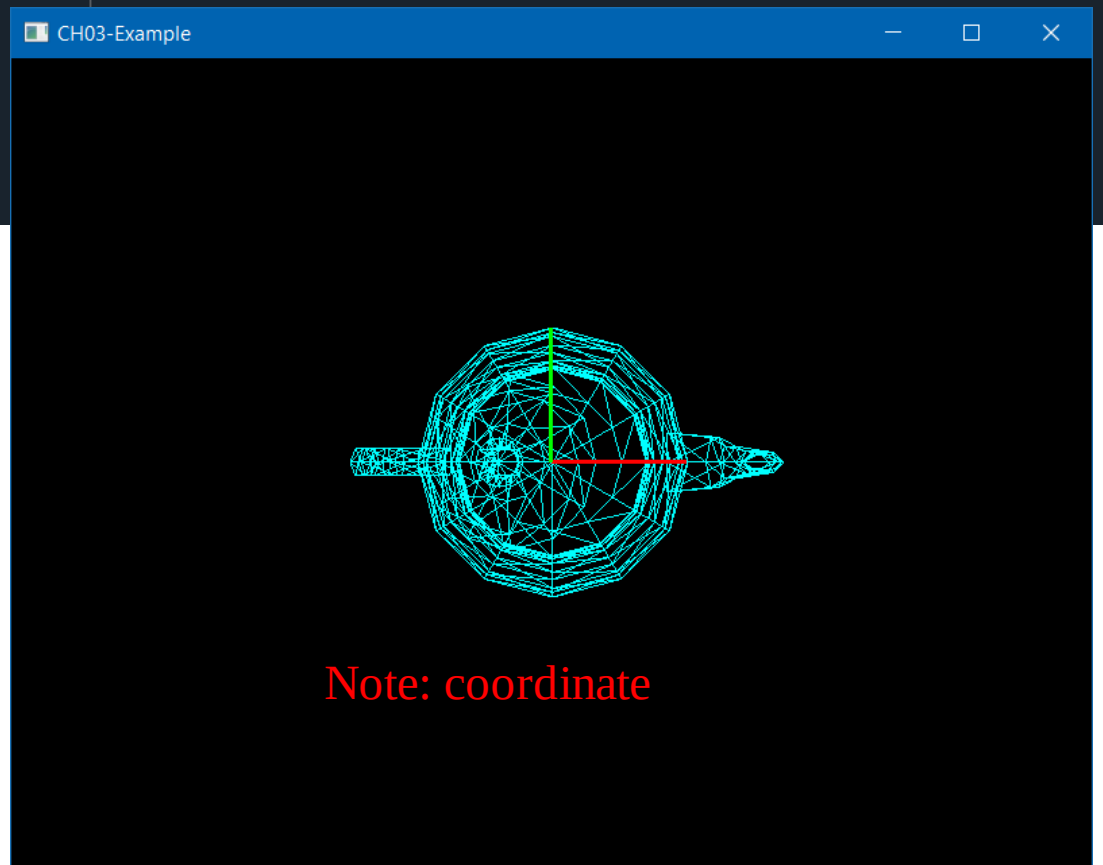
```
886 def drawCoordinate():
887     glLineWidth(3)
888     glBegin(GL_LINES)
889     glColor3f(1,0,0)
890     glVertex3f(0,0,0)
891     glVertex3f(100,0,0)
892     glColor3f(0,1,0)
893     glVertex3f(0,0,0)
894     glVertex3f(0,100,0)
895     glColor3f(0,0,1)
896     glVertex3f(0,0,0)
897     glVertex3f(0,0,100)
898     glEnd()
```

Draw coordinate (three lines)



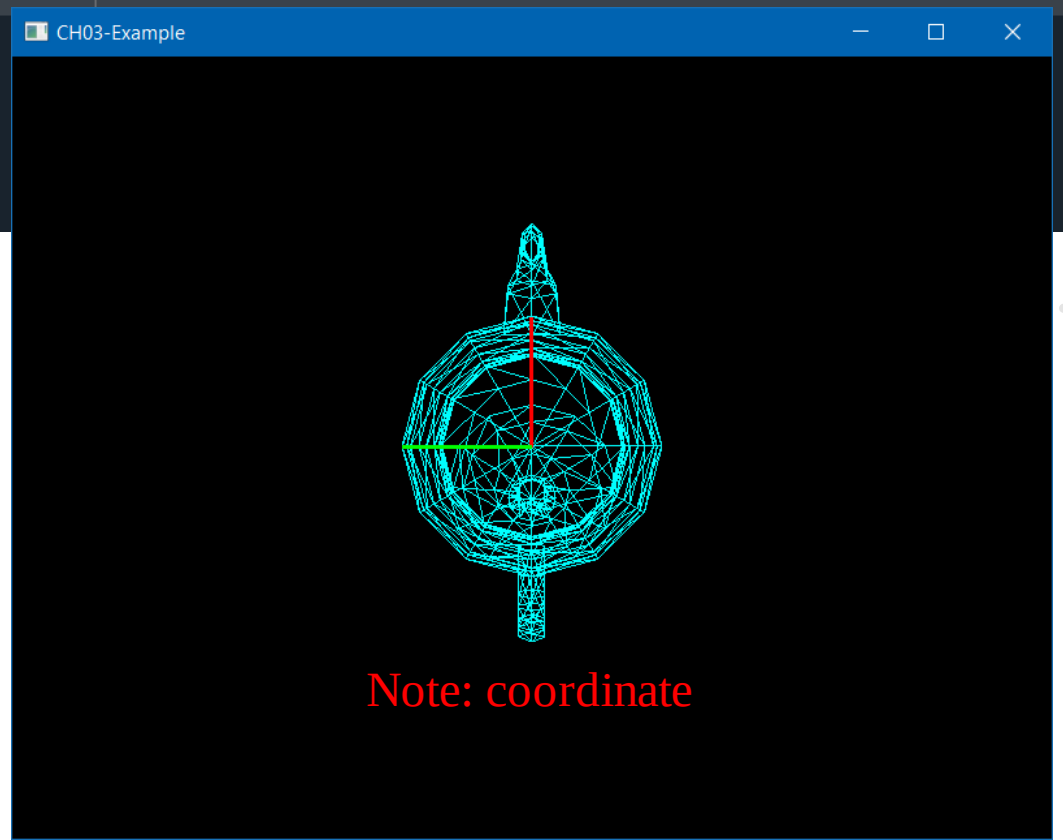
Set camera's position

```
899
900  def display():
901      glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
902      glMatrixMode(GL_PROJECTION)
903      glLoadIdentity()
904      glViewport(0, 0, windowWidth, windowHeight)
905      glOrtho(-float(windowWidth)/2.0,float(windowWidth)/2.0,-float(windowHeight)/2.0,float(windowHeight)/2.0,-windowHeight*10.0,windowHeight*10.0)
906      gluLookAt(0,0,1000,0,0,0,0,1,0)
907      glPushMatrix()
908      drawTeapot()
909      glPopMatrix()
910      drawCoordinate()
911      glutSwapBuffers()
912
```



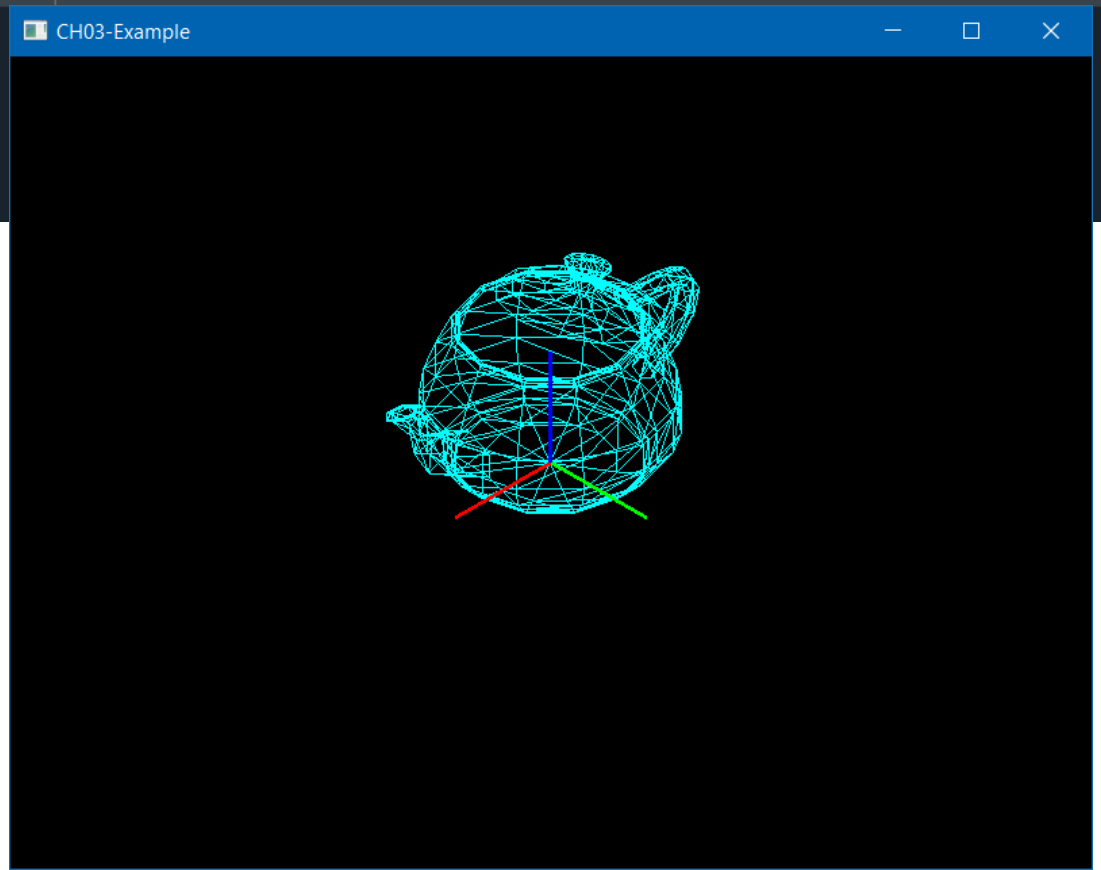
Set camera's position

```
899
900  def display():
901      glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
902      glMatrixMode(GL_PROJECTION)
903      glLoadIdentity()
904      glViewport(0, 0, windowWidth, windowHeight)
905      glOrtho(-float(windowWidth)/2.0,float(windowWidth)/2.0,-float(windowHeight)/2.0,float(windowHeight)/2.0,-windowHeight*10.0,windowHeight*10.0)
906      gluLookAt(0,0,1000,0,0,0,1,0,0)
907      glPushMatrix()
908      drawTeapot()
909      glPopMatrix()
910      drawCoordinate()
911      glutSwapBuffers()
912
```



Set camera's position

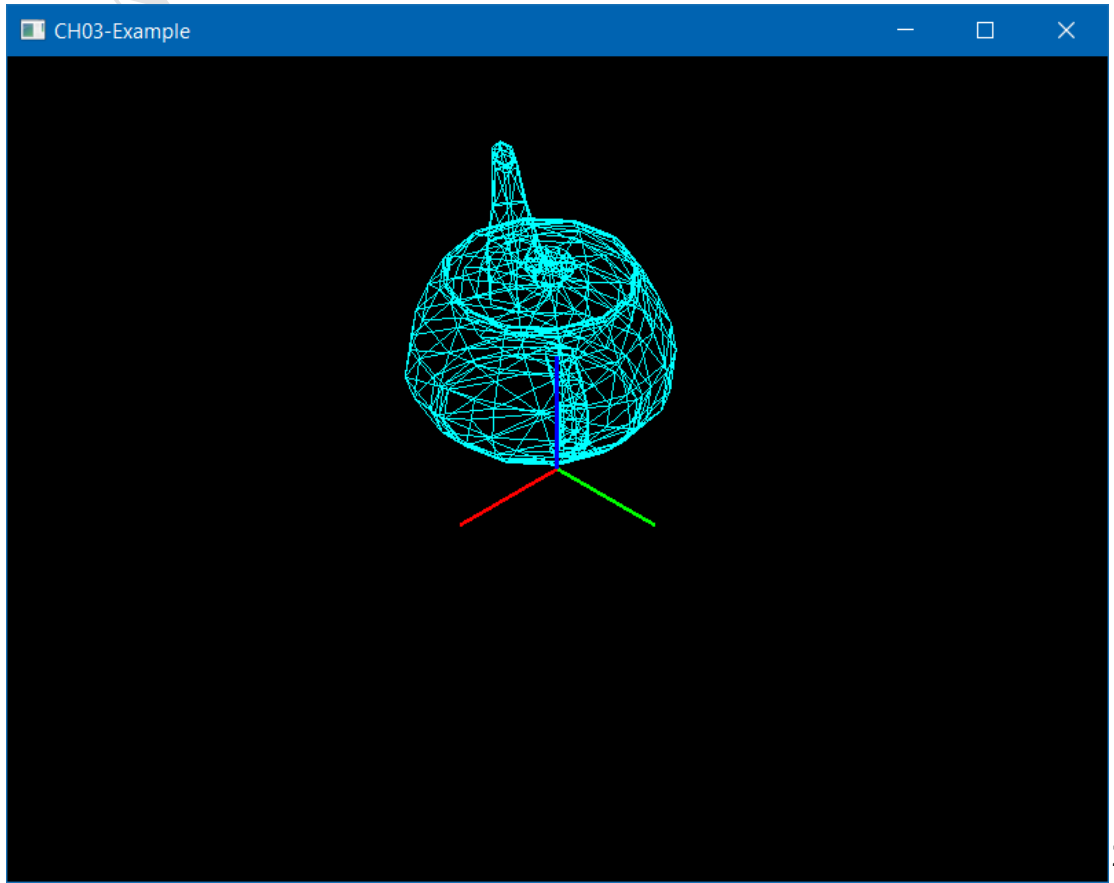
```
899
900  def display():
901      glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
902      glMatrixMode(GL_PROJECTION)
903      glLoadIdentity()
904      glViewport(0, 0, windowWidth, windowHeight)
905      glOrtho(-float(windowWidth)/2.0,float(windowWidth)/2.0,-float(windowHeight)/2.0,float(windowHeight)/2.0,-windowHeight*10.0,windowHeight*10.0)
906      gluLookAt(1000,1000,1000,0,0,0,0,0,1)
907      glPushMatrix()
908      drawTeapot()
909      glPopMatrix()
910      drawCoordinate()
911      glutSwapBuffers()
912
```



glPushMatrix and glPopMatrix (outside block)

```
871  def drawTeapot():
872      glLineWidth(1)
873      glColor3f(0,1,1)
874      glRotatef(angle,0,0,1)
875      glTranslatef(100*cos(theda),0,0)
876      glBegin(GL_LINES)
877      for fID in teapotFace:
878          glVertex3fv(teapotVertex[fID[0]])
879          glVertex3fv(teapotVertex[fID[1]])
880          glVertex3fv(teapotVertex[fID[1]])
881          glVertex3fv(teapotVertex[fID[2]])
882          glVertex3fv(teapotVertex[fID[2]])
883          glVertex3fv(teapotVertex[fID[0]])
884      glEnd()
```

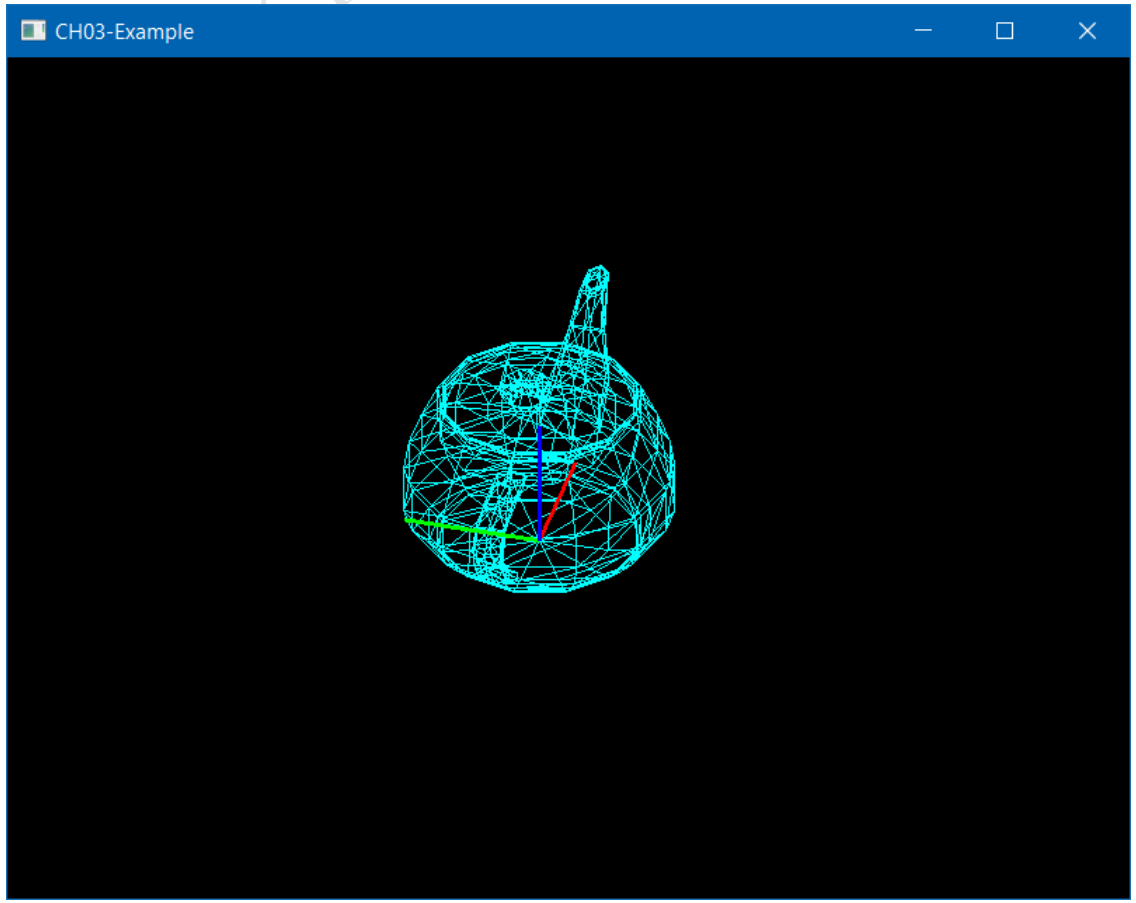
```
899
900  def display():
901      glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
902      glMatrixMode(GL_PROJECTION)
903      glLoadIdentity()
904      glViewport(0, 0, windowWidth, windowHeight)
905      glOrtho(-float(windowWidth)/2.0,float(windowWidth)/2.0,-float(windo
906      gluLookAt(1000,1000,1000,0,0,0,0,0,1)
907      global theda, angle
908      theda = theda + 0.1
909      angle = angle + 5
910      glPushMatrix()
911      drawTeapot()
912      glPopMatrix()
913      drawCoordinate()
914      glutSwapBuffers()
915      glutPostRedisplay()
```



glPushMatrix and glPopMatrix (inside block)

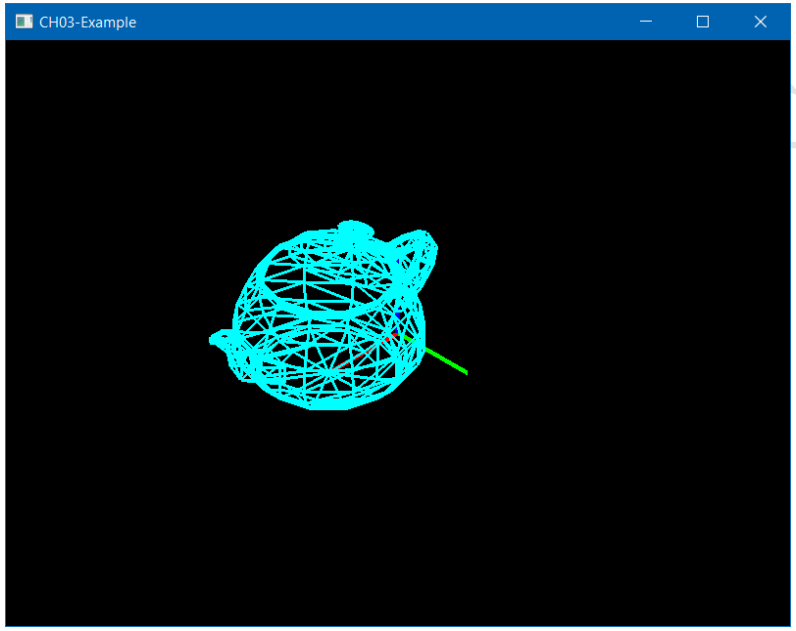
```
900  def display():
901      glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
902      glMatrixMode(GL_PROJECTION)
903      glLoadIdentity()
904      glViewport(0, 0, windowWidth, windowHeight)
905      glOrtho(-float(windowWidth)/2.0,float(windowWidth)/2.0,-float(windowHeight
906      gluLookAt(1000,1000,1000,0,0,0,0,0,1)
907      global theda, angle
908      theda = theda + 0.1
909      angle = angle + 5
910      glPushMatrix()
911      drawTeapot()
912      drawCoordinate()
913      glPopMatrix()
914      glutSwapBuffers()
915      glutPostRedisplay()
916
```

Please compare with previous slide

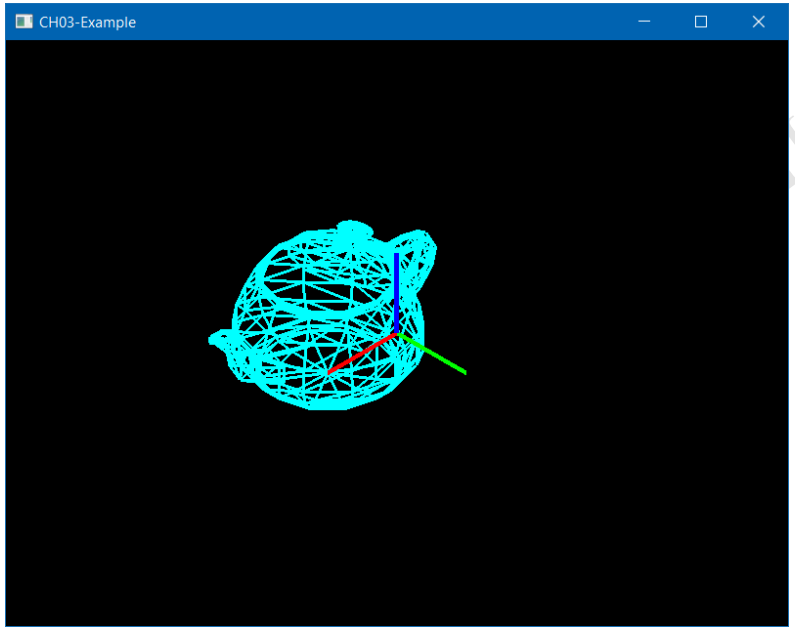


Drawing priority (the earlier the lower priority)

```
900  def display():
901      glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
902      glMatrixMode(GL_PROJECTION)
903      glLoadIdentity()
904      glViewport(0, 0, windowWidth, windowHeight)
905      glOrtho(-float(windowWidth)/2.0,float(windowWidth)/2.0,-fl
906      gluLookAt(1000,1000,1000,0,0,0,0,0,1)
907      drawCoordinate()
908      glPushMatrix()
909      drawTeapot()
910      glPopMatrix()
911      glutSwapBuffers()
912
```




```
900  def display():
901      glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
902      glMatrixMode(GL_PROJECTION)
903      glLoadIdentity()
904      glViewport(0, 0, windowWidth, windowHeight)
905      glOrtho(-float(windowWidth)/2.0,float(windowWidth)/2.0,-float(
906      gluLookAt(1000,1000,1000,0,0,0,0,0,1)
907      glPushMatrix()
908      drawTeapot()
909      glPopMatrix()
910      drawCoordinate()
911      glutSwapBuffers()
912
```

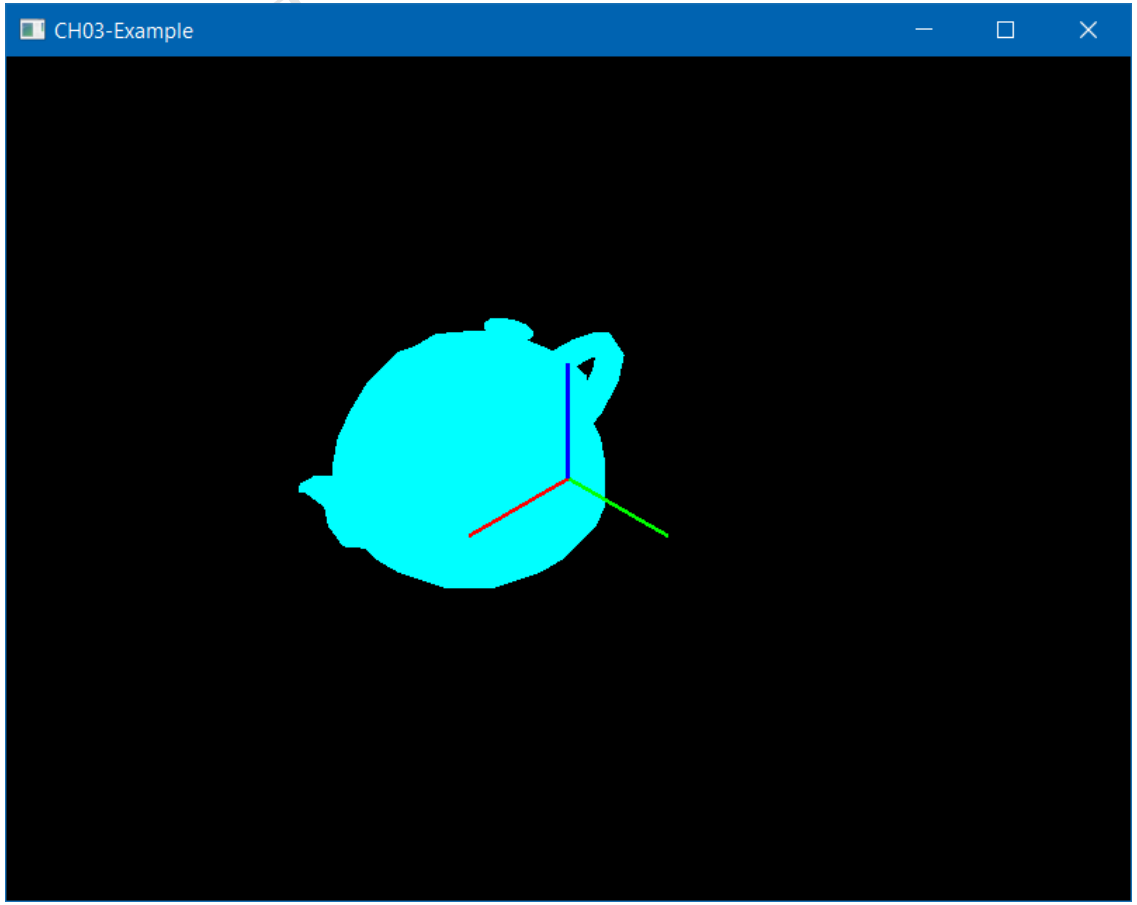


Drawing face (Solid Color → without light)

```
871
872  def drawTeapot():
873      glColor3f(0,1,1)
874      glRotatef(angle,0,0,1)
875      glTranslatef(100*cos(theda),0,0)
876      glBegin(GL_TRIANGLES)
877      for fID in teapotFace:
878          v1 = np.subtract(teapotVertex[fID[1]] ,teapotVertex[fID[0]])
879          v2 = np.subtract(teapotVertex[fID[2]] ,teapotVertex[fID[0]])
880          nv = np.cross(v1,v2)
881          nlen = np.linalg.norm(nv, ord=1)
882          nv = nv / nlen
883          glNormal3f(nv[0],nv[1],nv[2])
884          glVertex3fv(teapotVertex[fID[0]])
885          glVertex3fv(teapotVertex[fID[1]])
886          glVertex3fv(teapotVertex[fID[2]])
887      glEnd()
888
```

Find normal vector

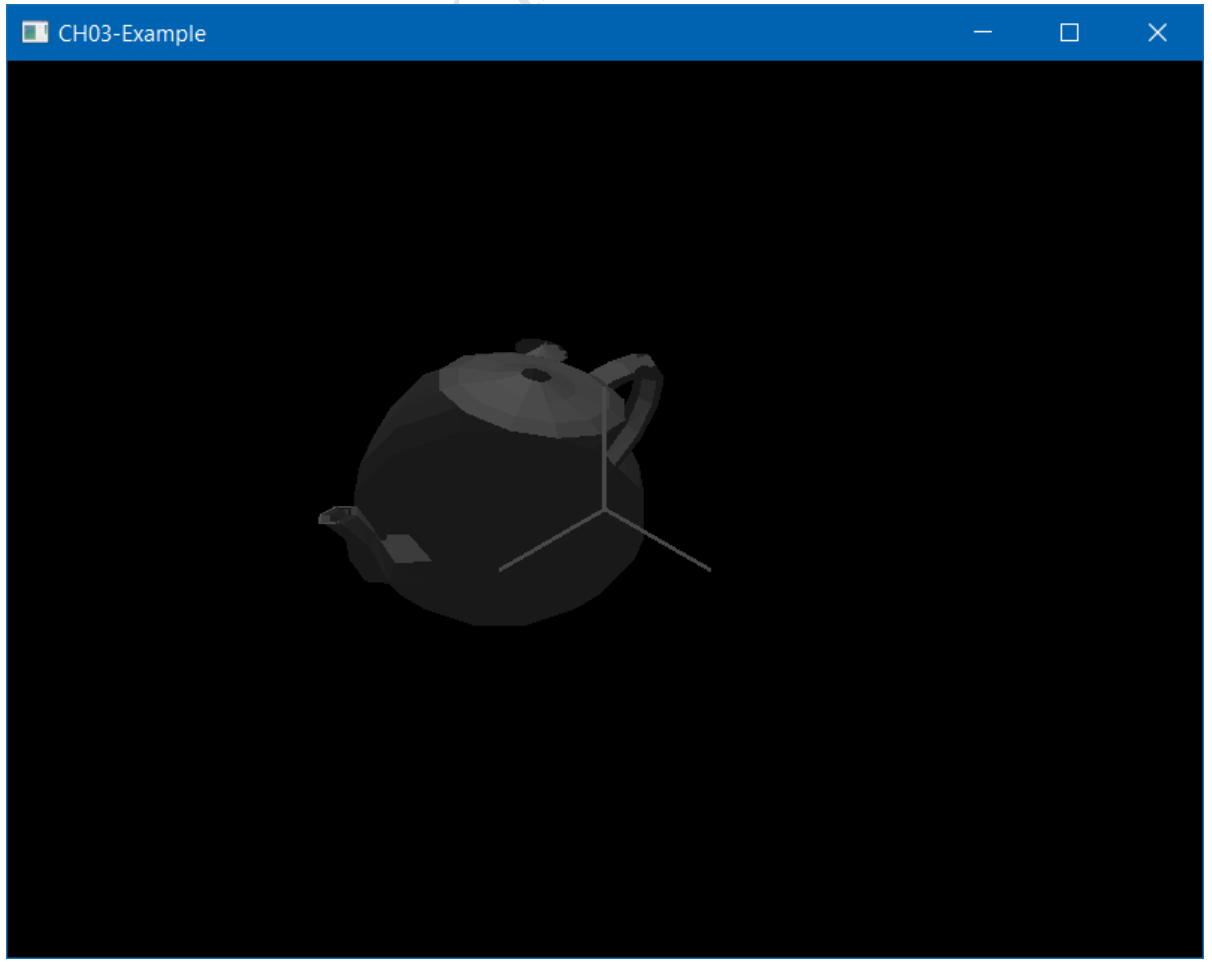
```
903  def display():
904      glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
905      glMatrixMode(GL_PROJECTION)
906      glLoadIdentity()
907      glViewport(0, 0, windowWidth, windowHeight)
908      glOrtho(-float(windowWidth)/2.0,float(windowWidth)/2.
909      gluLookAt(1000,1000,1000,0,0,0,0,0,1)
910      glPushMatrix()
911      drawTeapot()
912      glPopMatrix()
913      drawCoordinate()
914      glutSwapBuffers()
915
```



Drawing face (Shade Color → with light)

Setup lighting

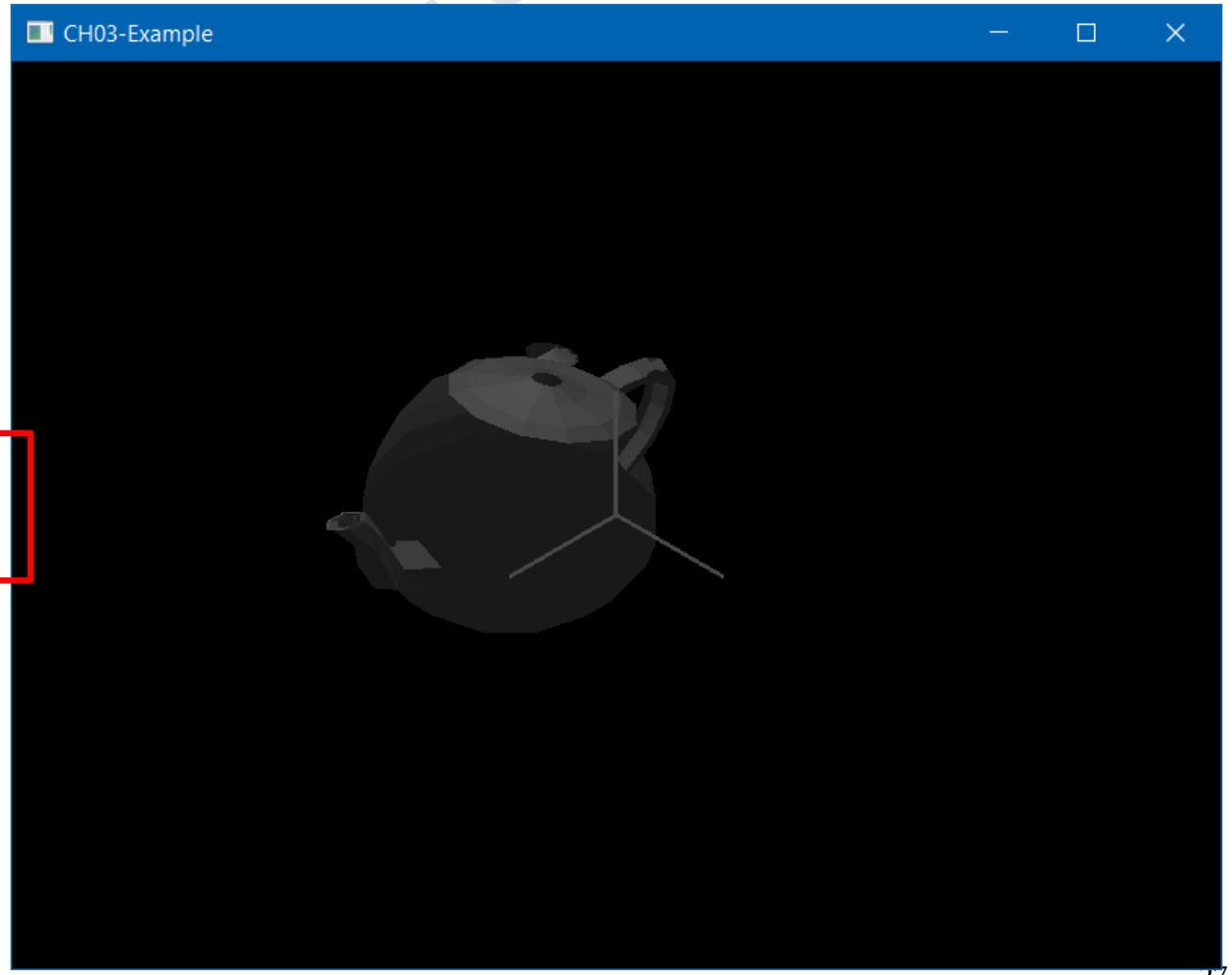
```
923
924     glutInit()
925     glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA)
926     glutCreateWindow(b'CH03-Example')
927     glutReshapeWindow(windowWidth,windowHeight)
928     glutReshapeFunc(reshape)
929     glutDisplayFunc(display)
930     glutKeyboardFunc(keyboard)
931     glEnable(GL_LIGHTING)
932     glEnable(GL_LIGHT0)
933     lightAmbient = [ 0.3,0.3,0.3,1.0 ]
934     lightDiffuse = [ 0.7,0.7,0.7,1.0 ]
935     lightSpecular = [ 1.0,1.0,1.0, 1.0 ]
936     lightPosition = [ 0,0,1000,1.0 ]
937     glLightfv(GL_LIGHT0, GL_AMBIENT, lightAmbient)
938     glLightfv(GL_LIGHT0, GL_DIFFUSE, lightAmbient)
939     glLightfv(GL_LIGHT0, GL_SPECULAR, lightSpecular)
940     glLightfv(GL_LIGHT0, GL_POSITION, lightPosition)
941     glutMainLoop()
942
```



Drawing face (Shade Color → with light)
Setup lighting

```
903  def display():
904      glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
905      glMatrixMode(GL_PROJECTION)
906      glLoadIdentity()
907      glViewport(0, 0, windowWidth, windowHeight)
908      glOrtho(-float(windowWidth)/2.0,float(windowWidth)/2.0,-float(w
909      gluLookAt(1000,1000,1000,0,0,0,0,0,1)
910      glPushMatrix()
911      drawTeapot()
912      glPopMatrix()
913      drawCoordinate()
914      glutSwapBuffers()
915
```

All affected by Lighting, including "lines"

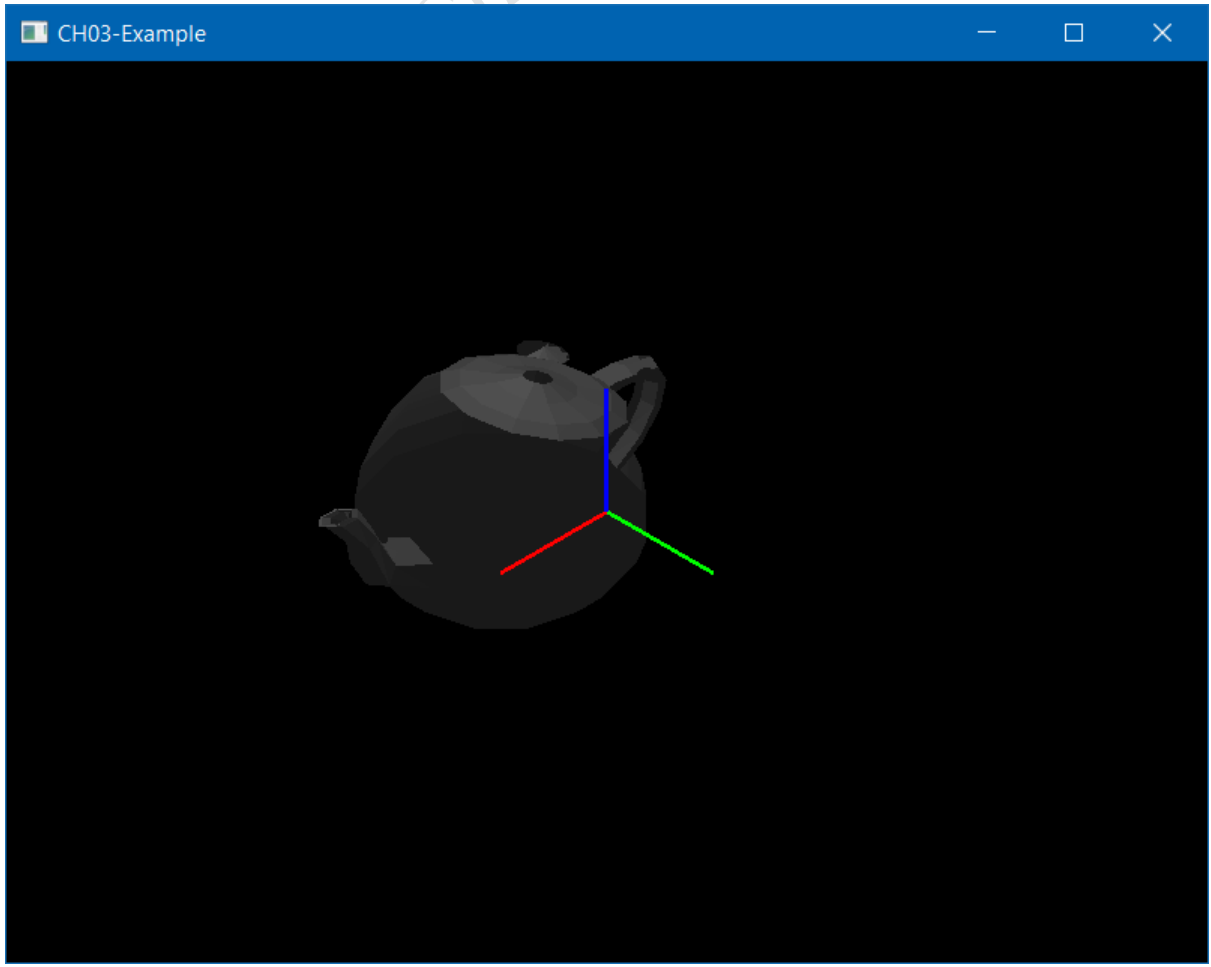


Drawing shade face (separated solid and shade pipeline)

```
902
903  def display():
904      glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
905      glMatrixMode(GL_PROJECTION)
906      glLoadIdentity()
907      glViewport(0, 0, windowWidth, windowHeight)
908      glOrtho(-float(windowWidth)/2.0,float(windowWidth)/2.0,-float(wi
909      gluLookAt(1000,1000,1000,0,0,0,0,0,1)
910      glEnable(GL_LIGHTING)
911      glPushMatrix()
912      drawTeapot()
913      glPopMatrix()
914      glDisable(GL_LIGHTING)
915      drawCoordinate()
916      glutSwapBuffers()
917
```

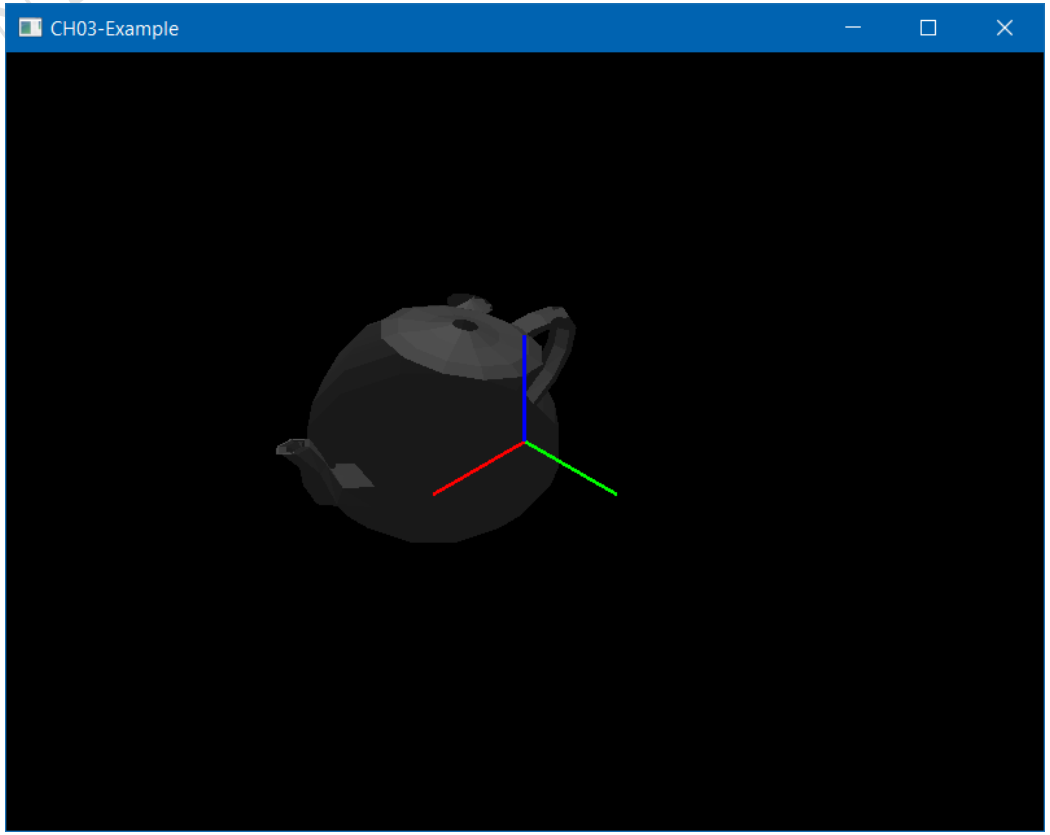
Draw with lighting

Draw without lighting

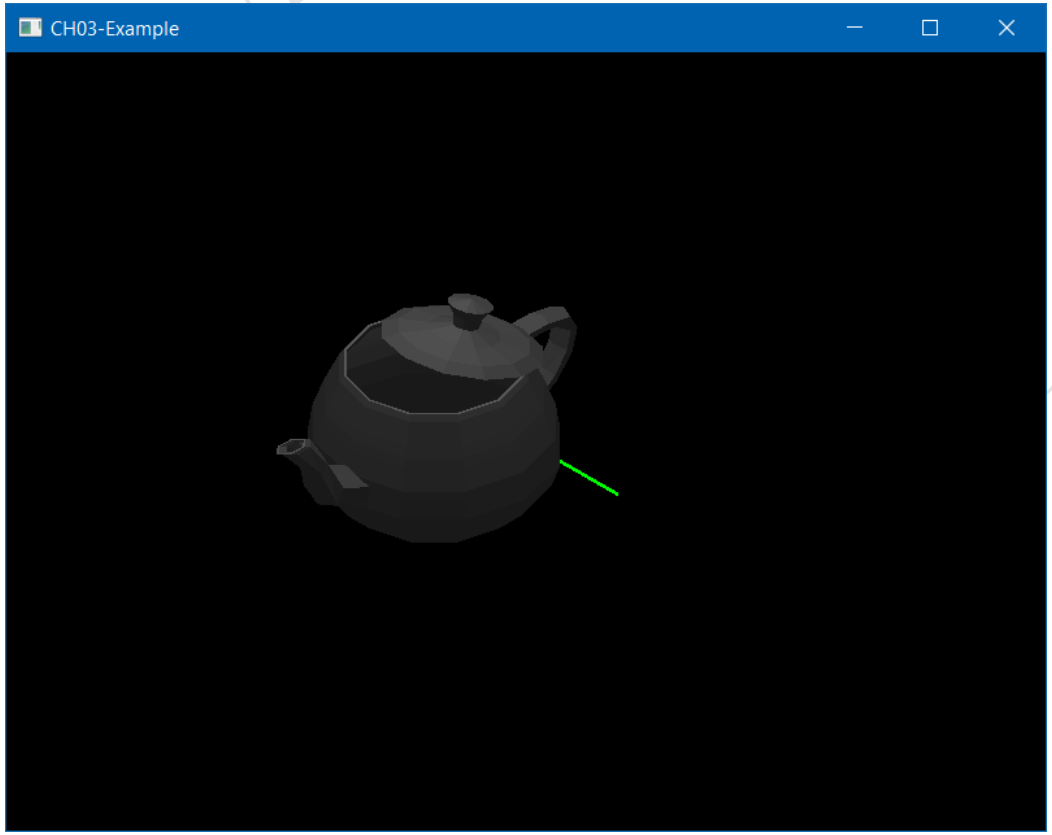


Enable Depth Test

```
931    glutDisplayFunc(display)
932    glutKeyboardFunc(keyboard)
933    glDisable(GL_DEPTH_TEST)
934    glEnable(GL_LIGHTING)
935    glEnable(GL_LIGHT0)
```




```
931    glutDisplayFunc(display)
932    glutKeyboardFunc(keyboard)
933    glEnable(GL_DEPTH_TEST)
934    glEnable(GL_LIGHTING)
935    glEnable(GL_LIGHT0)
```



色彩與照明科技研究所
Graduate Institute of
Color and Illumination Technology

Find us on: facebook