Advanced Computer Graphics

Lecture-08 Introduction to OpenGL-5

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

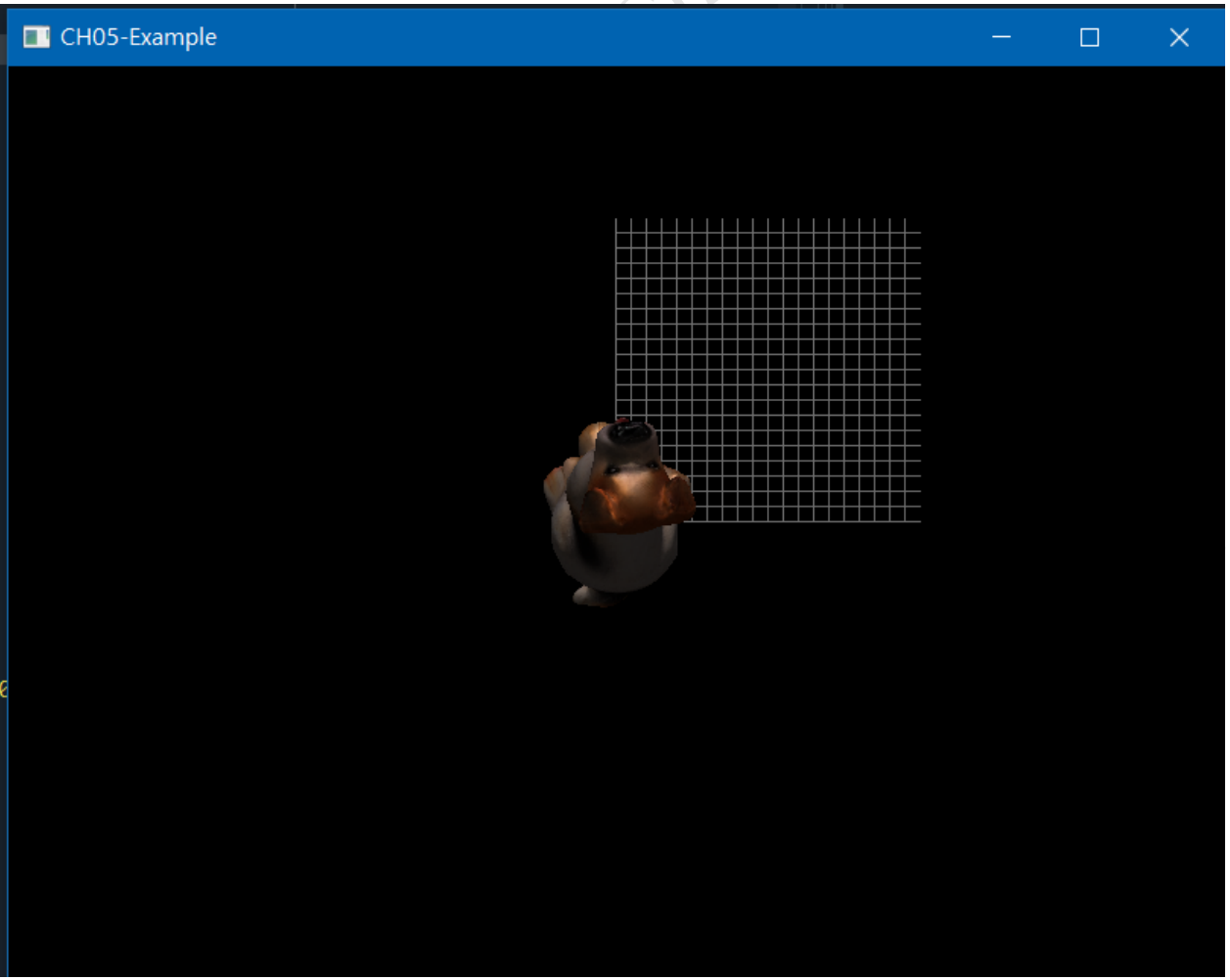
CIT 色彩與照明科技研究所 Graduate Institute of Color and Illumination Technology

Camera/Viewing control

Difference between shading color and solid color

Note: deal with light carefully

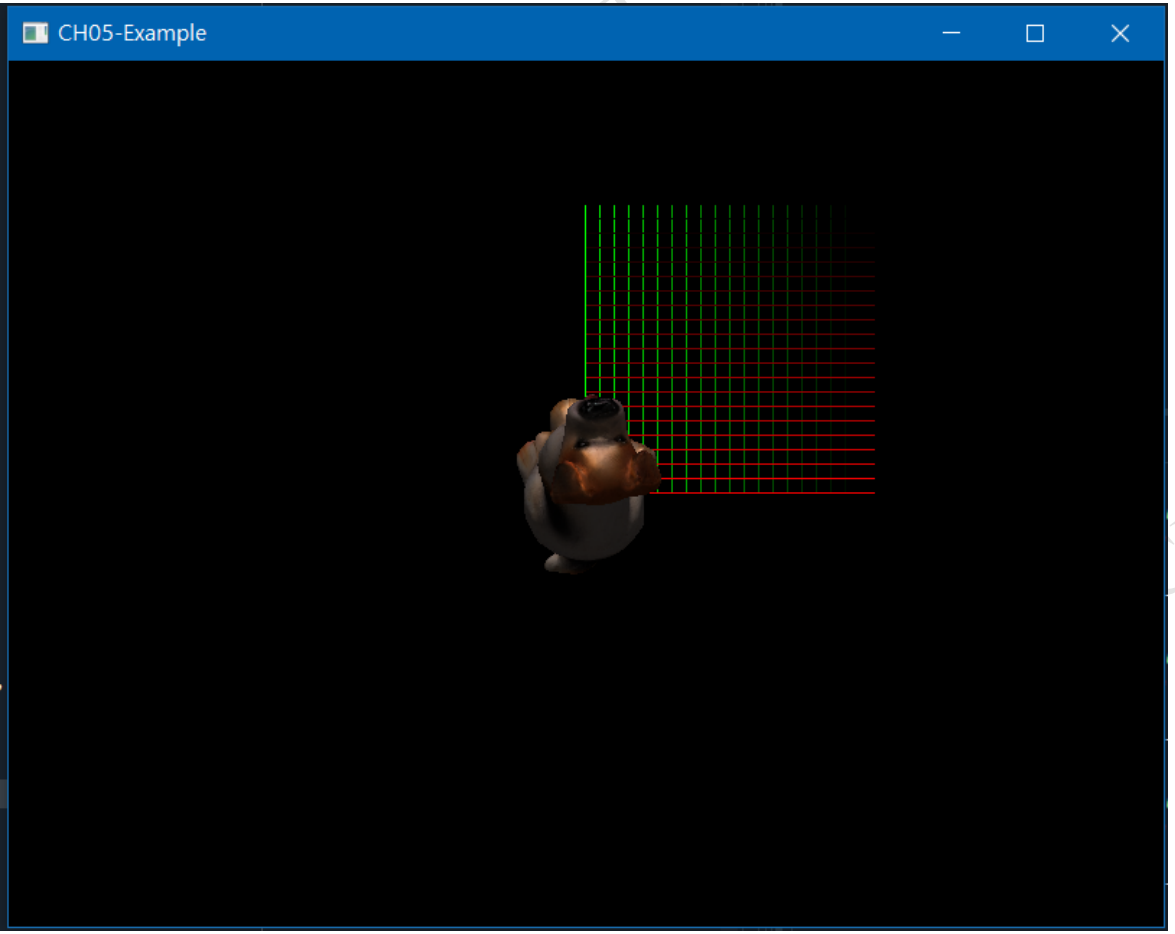
```
24
25  def drawGrid():
26      glLineWidth(1)
27      glBegin(GL_LINES)
28      for y in range(0, 20):
29          glColor3f(1-y/19,0,0)
30          glVertex3f(0,10*y,0)
31          glVertex3f(200,10*y,0)
32      glEnd()
33      glBegin(GL_LINES)
34      for x in range(0, 20):
35          glColor3f(0,1-x/19,0)
36          glVertex3f(10*x,0,0)
37          glVertex3f(10*x,200,0)
38      glEnd()
39
40  def display():
41      glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
42      glMatrixMode(GL_PROJECTION)
43      glLoadIdentity()
44      glLightfv(GL_LIGHT0, GL_POSITION, lightPosition)
45      glViewport(0, 0, windowWidth, windowHeight)
46      glOrtho(-float(windowWidth)/2.0,float(windowWidth)/2.0
47      gluLookAt(0,0,1000,0,0,0,0,1,0)
48      glEnable(GL_LIGHTING)
49      visualization.draw(meshes)
50      drawGrid()
51      glutSwapBuffers()
52
53
54  def reshape(width,height):
55      glViewport(0, 0, width, height)
```



Difference between shading color and solid color

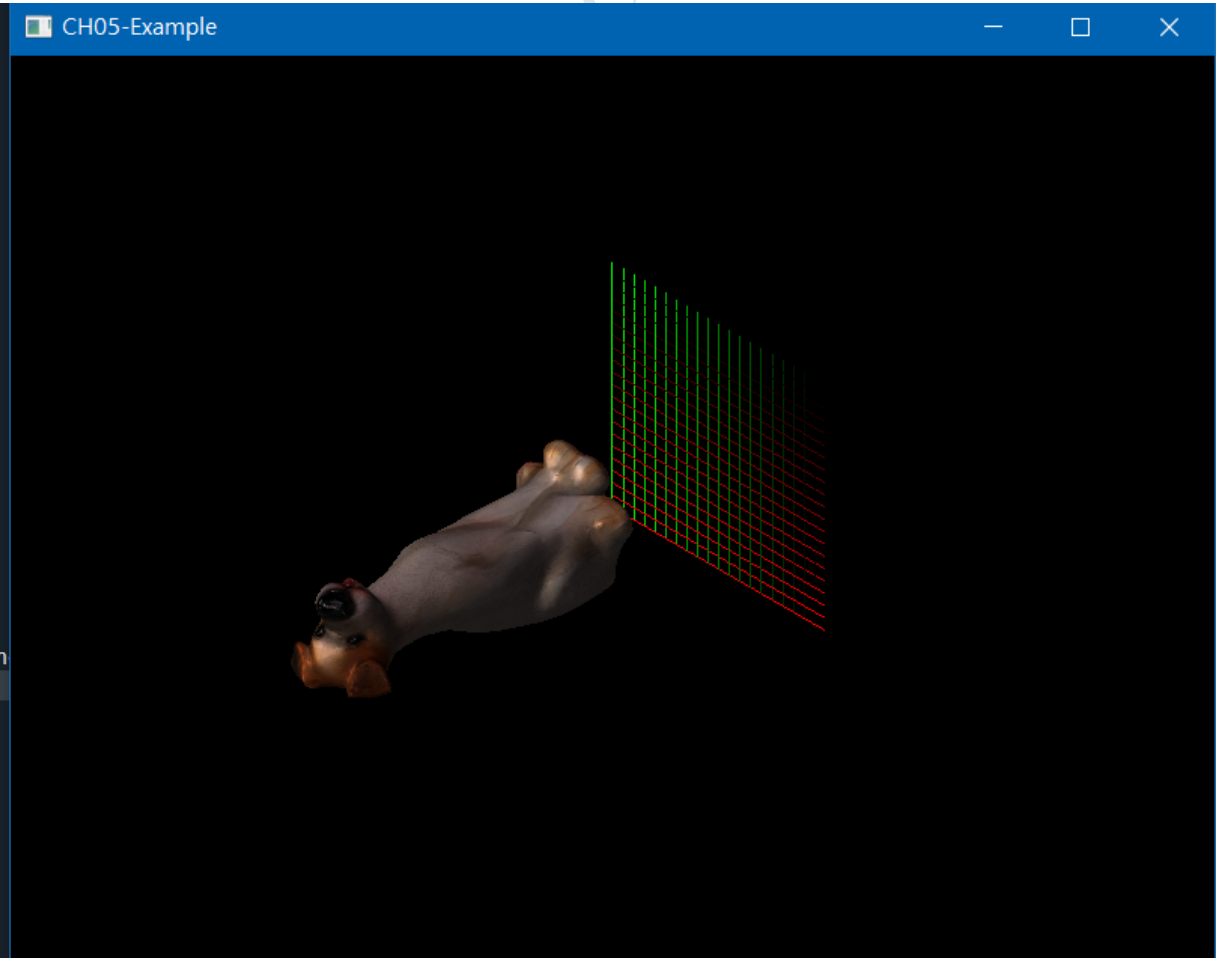
Note: deal with light carefully

```
23
24
25  def drawGrid():
26      glLineWidth(1)
27      glBegin(GL_LINES)
28      for y in range(0, 20):
29          glColor3f(1-y/19,0,0)
30          glVertex3f(0,10*y,0)
31          glVertex3f(200,10*y,0)
32      glEnd()
33      glBegin(GL_LINES)
34      for x in range(0, 20):
35          glColor3f(0,1-x/19,0)
36          glVertex3f(10*x,0,0)
37          glVertex3f(10*x,200,0)
38      glEnd()
39
40  def display():
41      glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
42      glMatrixMode(GL_PROJECTION)
43      glLoadIdentity()
44      glLightfv(GL_LIGHT0, GL_POSITION, lightPosition)
45      glViewport(0, 0, windowWidth, windowHeight)
46      glOrtho(-float(windowWidth)/2.0,float(windowWidth)/2.0,
47      gluLookAt(0,0,1000,0,0,0,0,1,0)
48      glEnable(GL_LIGHTING)
49      visualization.draw(meshes)
50      glDisable(GL_LIGHTING)
51      drawGrid()
52      glutSwapBuffers()
53
54
```



Control the position of camera

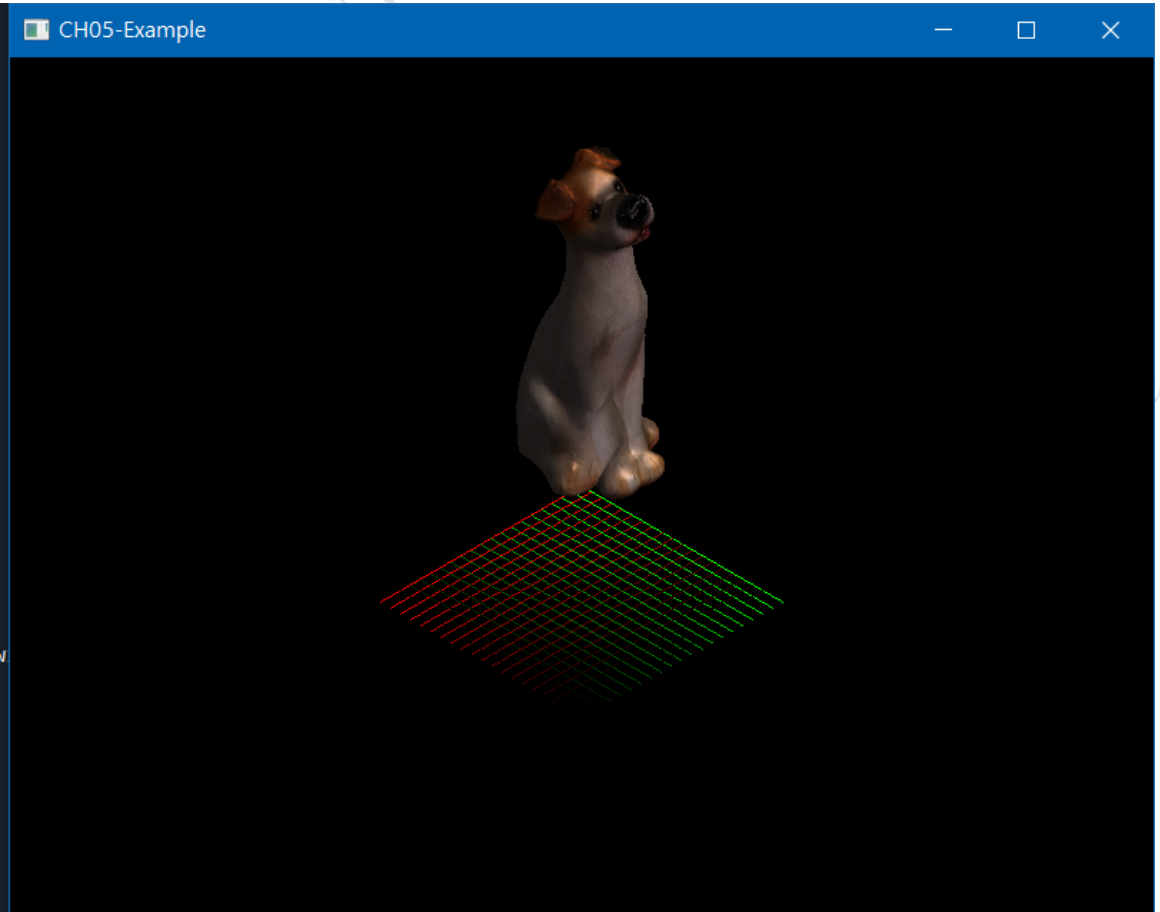
```
25  def drawGrid():
26      glLineWidth(1)
27      glBegin(GL_LINES)
28      for y in range(0, 20):
29          glColor3f(1-y/19,0,0)
30          glVertex3f(0,10*y,0)
31          glVertex3f(200,10*y,0)
32      glEnd()
33      glBegin(GL_LINES)
34      for x in range(0, 20):
35          glColor3f(0,1-x/19,0)
36          glVertex3f(10*x,0,0)
37          glVertex3f(10*x,200,0)
38      glEnd()
39
40  def display():
41      glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
42      glMatrixMode(GL_PROJECTION)
43      glLoadIdentity()
44      glLightfv(GL_LIGHT0, GL_POSITION, lightPosition)
45      glViewport(0, 0, windowWidth, windowHeight)
46      glOrtho(-float(windowWidth)/2.0,float(windowWidth)/2.0,-float(win
47      gluLookAt(1000,1000,1000,0,0,0,0,1,0)
48      glEnable(GL_LIGHTING)
49      visualization.draw(meshes)
50      glDisable(GL_LIGHTING)
51      drawGrid()
52      glutSwapBuffers()
53
54
55  def reshape(width,height):
```



Control the position of camera (Note: up vector)

- Note: changing position of camera is always regarding to three 3D vectors!

```
24
25  def drawGrid():
26      glLineWidth(1)
27      glBegin(GL_LINES)
28      for y in range(0, 20):
29          glColor3f(1-y/19,0,0)
30          glVertex3f(0,10*y,0)
31          glVertex3f(200,10*y,0)
32      glEnd()
33      glBegin(GL_LINES)
34      for x in range(0, 20):
35          glColor3f(0,1-x/19,0)
36          glVertex3f(10*x,0,0)
37          glVertex3f(10*x,200,0)
38      glEnd()
39
40  def display():
41      glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
42      glMatrixMode(GL_PROJECTION)
43      glLoadIdentity()
44      glLightfv(GL_LIGHT0, GL_POSITION, lightPosition)
45      glViewport(0, 0, windowWidth, windowHeight)
46      glOrtho(-float(windowWidth)/2.0,float(windowWidth)/2.0,-float(w
47      gluLookAt(1000,1000,1000,0,0,0,0,0,1)
48      glEnable(GL_LIGHTING)
49      visualization.draw(meshes)
50      glDisable(GL_LIGHTING)
51      drawGrid()
52      glutSwapBuffers()
53
54
55  def reshape(width,height):
```

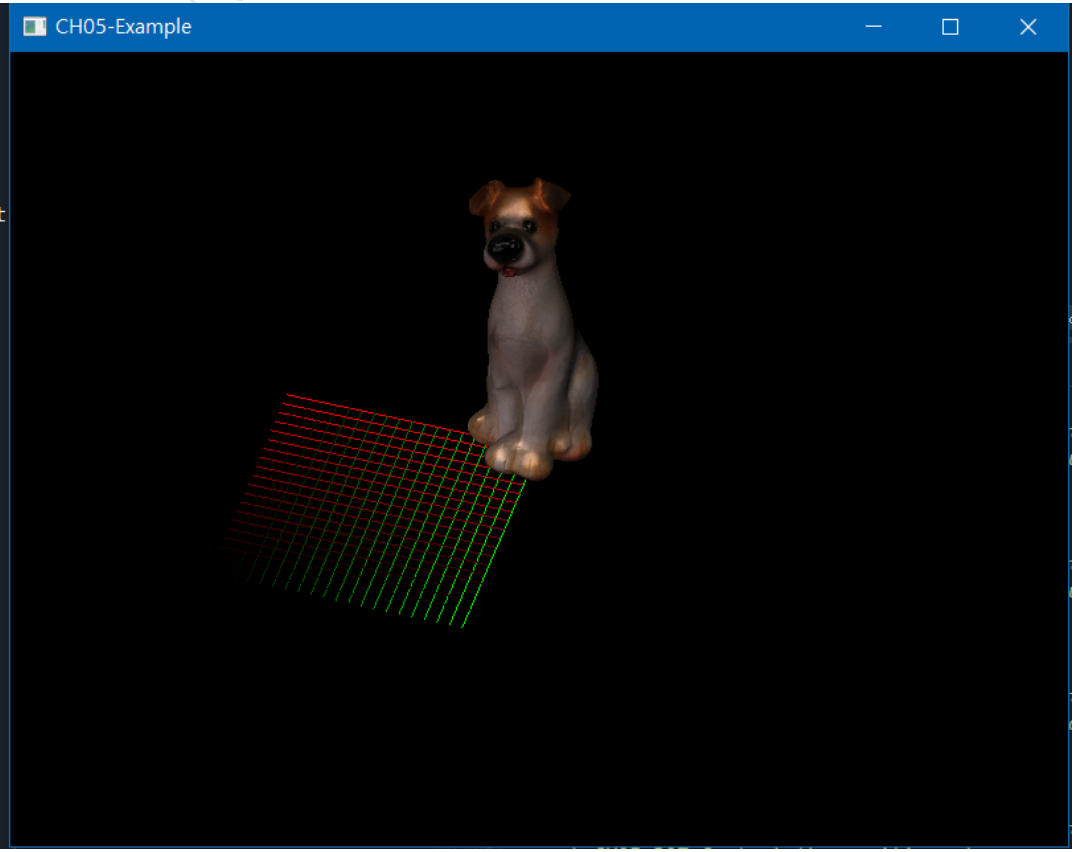


Control the position of camera: dynamic

```python
import sys
from math import *
from OpenGL.GL import *
from OpenGL.GLU import *
from OpenGL.GLUT import *
import numpy as np
from time import *
```

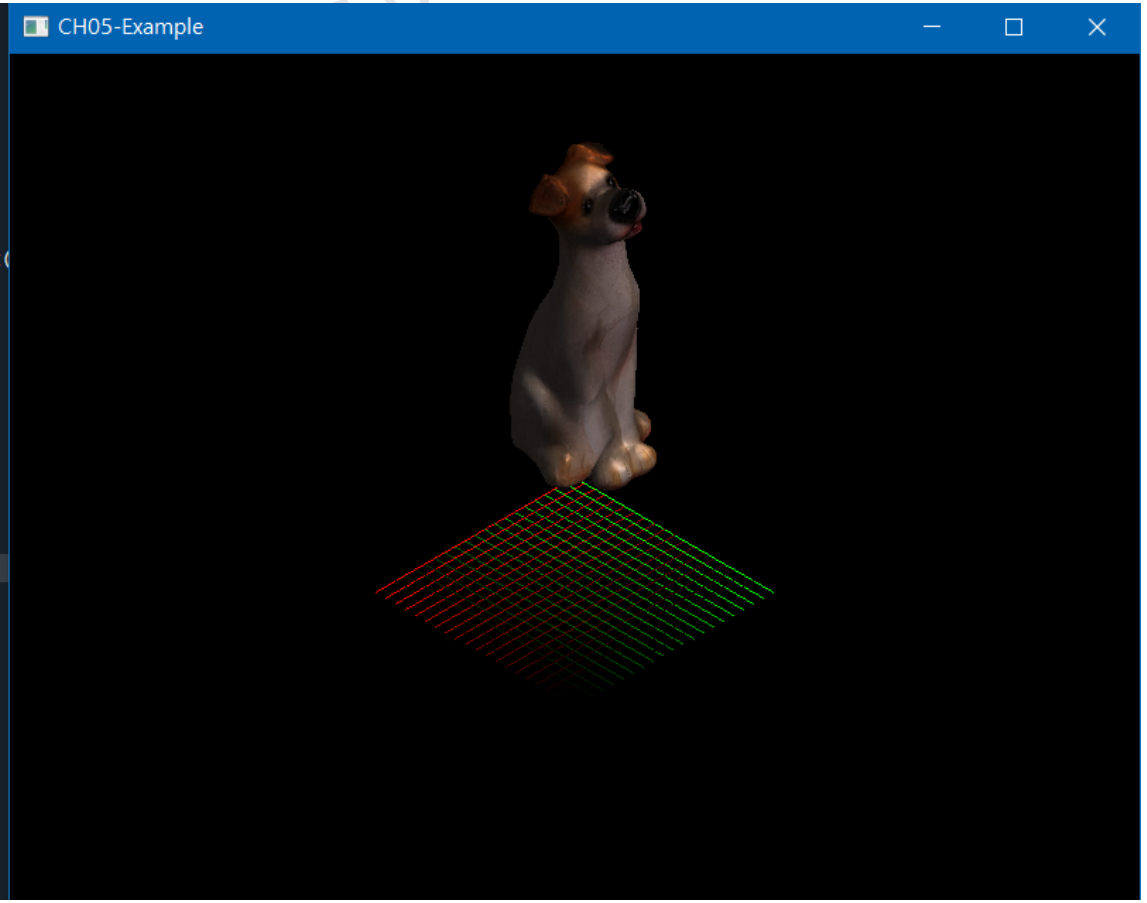
Sleep

```python
def display():
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    glLightfv(GL_LIGHT0, GL_POSITION, lightPosition)
    glViewport(0, 0, windowWidth, windowHeight)
    glOrtho(-float(windowWidth)/2.0,float(windowWidth)/2.0,-float(windowHeight
    global theda
    theda = theda + 1.0/57.3;
    gluLookAt(1000*cos(theda),1000*sin(theda),1000,0,0,0,0,0,1)
    glEnable(GL_LIGHTING)
    visualization.draw(meshes)
    glDisable(GL_LIGHTING)
    drawGrid()
    glutSwapBuffers()
    glutPostRedisplay()
    sleep(0.01)


def reshape(width,height):
    glViewport(0, 0, width, height)

def keyboard( key, x, y ):
    if key == esc:
        sys.exit()


glutInit()
glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA)
glutCreateWindow(b'CH05-Example')
```



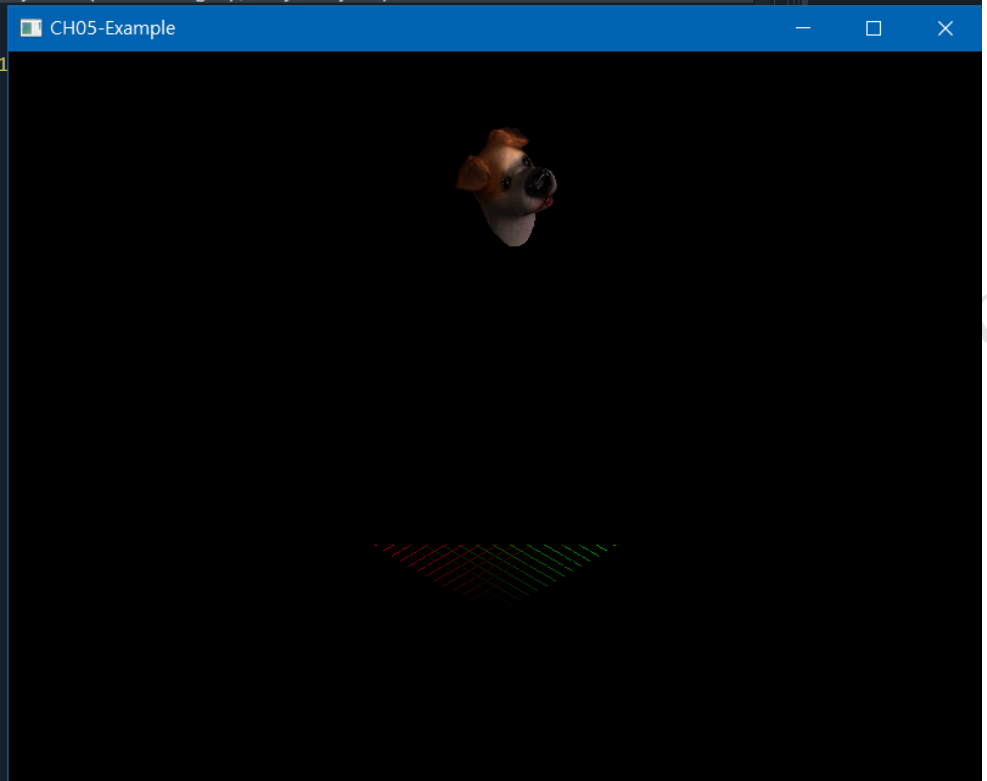
Control the position of camera (distance changes)

```
39      glEnd()
40
41  def display():
42      glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
43      glMatrixMode(GL_PROJECTION)
44      glLoadIdentity()
45      glLightfv(GL_LIGHT0, GL_POSITION, lightPosition)
46      glViewport(0, 0, windowWidth, windowHeight)
47      glOrtho(-float(windowWidth)/2.0,float(windowWidth)/2.0,-float(windowHeight)/2.0,float(
48      global theda
49      theda = theda + 1.0/57.3;
50      gluLookAt(500+400*cos(theda),500+400*cos(theda),500+400*cos(theda),0,0,0,0,0,1)
51      glEnable(GL_LIGHTING)
52      visualization.draw(meshes)
53      glDisable(GL_LIGHTING)
54      drawGrid()
55      glutSwapBuffers()
56      glutPostRedisplay()
57      sleep(0.01)
58
59
60  def reshape(width,height):
61      glViewport(0, 0, width, height)
62
63  def keyboard( key, x, y ):
64      if key == esc:
65          sys.exit()
66
67
68  glutInit()
69  glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA)
```



Control the position of camera distance changes-small view volume

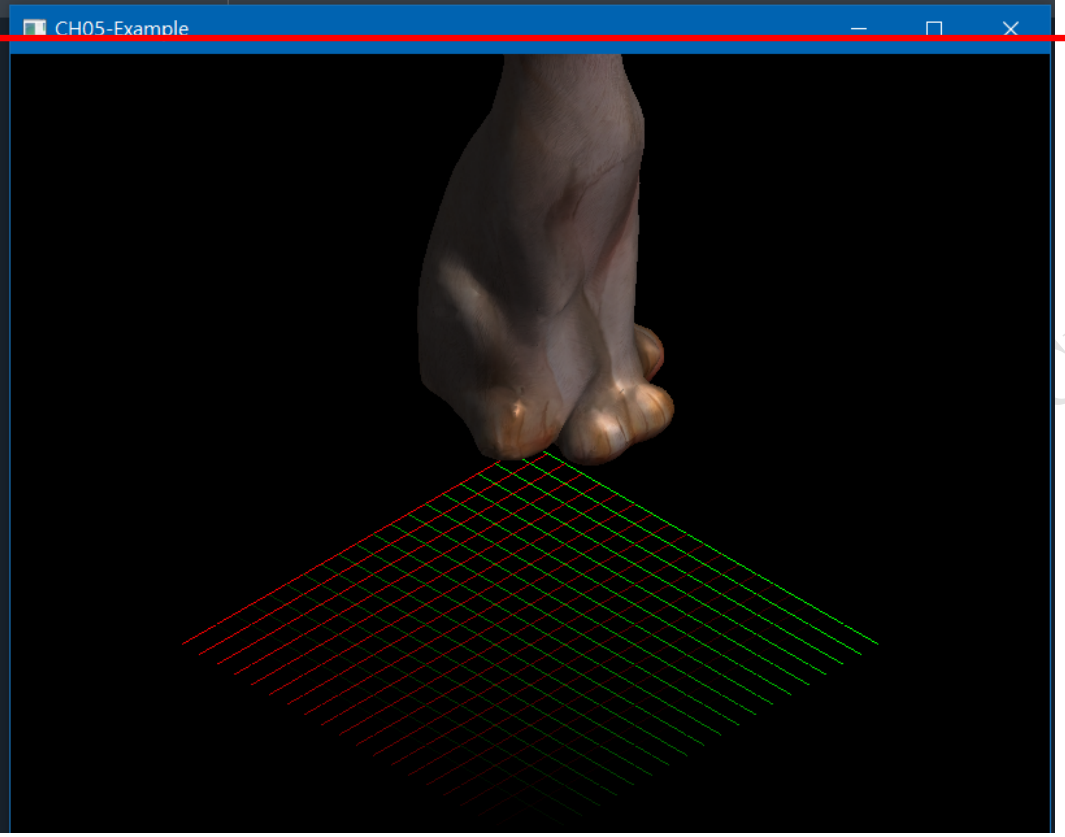
```
41  def display():
42      glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
43      glMatrixMode(GL_PROJECTION)
44      glLoadIdentity()
45      glLightfv(GL_LIGHT0, GL_POSITION, lightPosition)
46      glViewport(0, 0, windowWidth, windowHeight)
47      glOrtho(-float(windowWidth)/2.0,float(windowWidth)/2.0,-float(windowHeight)/2.0,float(windowHeight)/2.0,-100,100)
48      global theda
49      theda = theda + 1.0/57.3;
50      gluLookAt(500+400*cos(theda),500+400*cos(theda),500+400*cos(theda),0,0,0,0,0,1
51      glEnable(GL_LIGHTING)
52      visualization.draw(meshes)
53      glDisable(GL_LIGHTING)
54      drawGrid()
55      glutSwapBuffers()
56      glutPostRedisplay()
57      sleep(0.01)
58
59
60  def reshape(width,height):
61      glViewport(0, 0, width, height)
62
63  def keyboard( key, x, y ):
64      if key == esc:
65          sys.exit()
66
67
68  glutInit()
69  glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA)
70  glutCreateWindow(b'CH05-Example')
71  glutReshapeWindow(windowWidth,windowHeight)
72  glutReshapeFunc(reshape)
73  glutDisplayFunc(display)
74  glutKeyboardFunc(keyboard)
75  glEnable(GL_DEPTH_TEST)
76  glEnable(GL_LIGHTING)
77  glEnable(GL_LIGHT0)
78  glLightfv(GL_LIGHT0, GL_AMBIENT, lightAmbient)
79  glLightfv(GL_LIGHT0, GL_DIFFUSE, lightAmbient)
```



Change viewing size - by viewing volume (Orthographic projection)

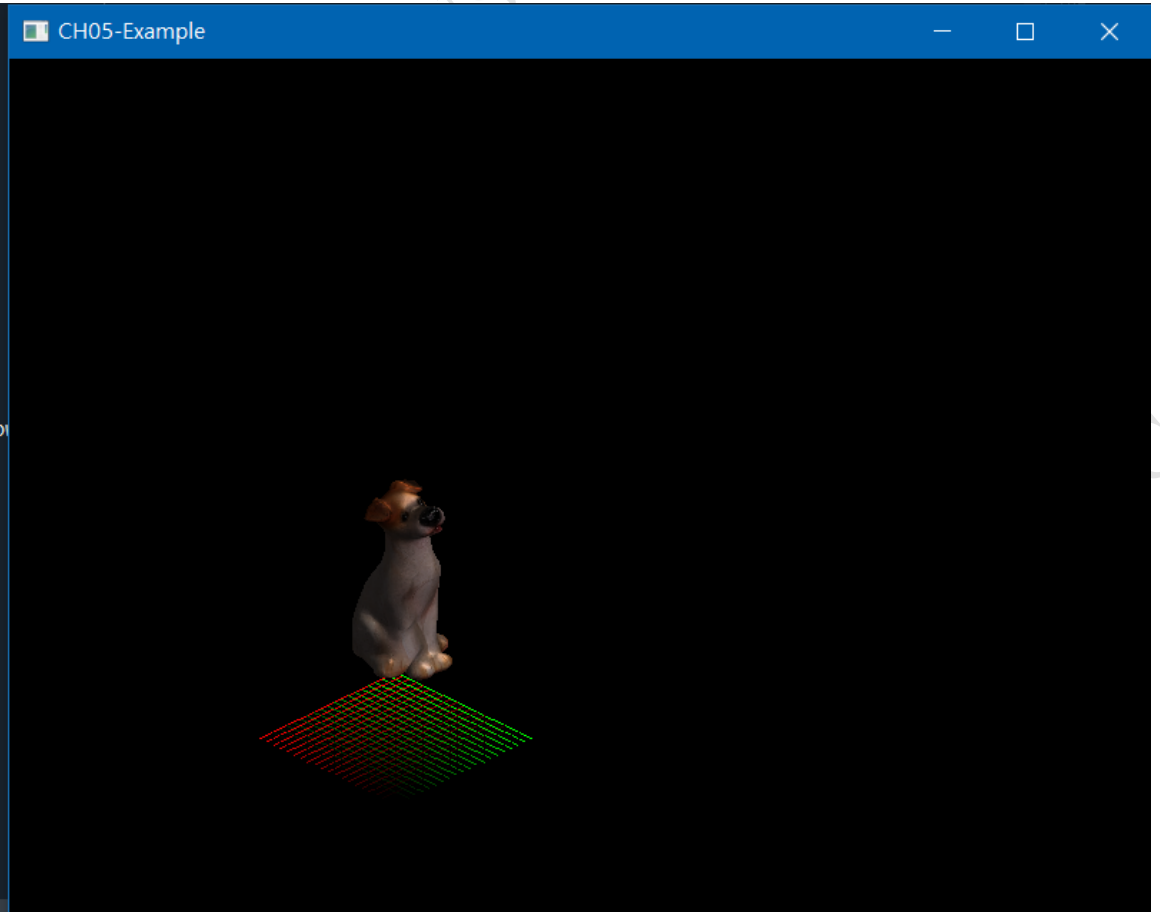
```
41  def display():
42      glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
43      glMatrixMode(GL_PROJECTION)
44      glLoadIdentity()
45      global theda
46      theda = theda + 1/57.3
47      glViewport(0, 0, windowWidth, windowHeight)
48      glOrtho(-float(windowWidth+400*cos(theda))/2.0,float(windowWidth+400*cos(theda))/2.0,-float(windowHeight+300*cos(theda))/2.0,float(windowHeight+
    +300*cos(theda))/2.0,-windowHeight*10.0,windowHeight*10.0)
49      gluLookAt(1000,1000,1000,0,0,0,0,0,1)
50      glEnable(GL_LIGHTING)
51      visualization.draw(meshes)
52      glDisable(GL_LIGHTING)
53      drawGrid()
54      glutSwapBuffers()
55      glutPostRedisplay()
56      sleep(0.01)
57
58
59  def reshape(width,height):
60      glViewport(0, 0, width, height)
61
62  def keyboard( key, x, y ):
63      if key == esc:
64          sys.exit()
65
66
67  glutInit()
68  glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA)
69  glutCreateWindow(b'CH05-Example')
70  glutReshapeWindow(windowWidth,windowHeight)
71  glutReshapeFunc(reshape)
72  glutDisplayFunc(display)
73  glutKeyboardFunc(keyboard)
74  glEnable(GL_DEPTH_TEST)
75  glEnable(GL_LIGHTING)
76  glEnable(GL_LIGHT0)
77  glLightfv(GL_LIGHT0, GL_AMBIENT, lightAmbient)
78  glLightfv(GL_LIGHT0, GL_DIFFUSE, lightAmbient)
79  glLightfv(GL_LIGHT0, GL_SPECULAR, lightSpecular)
```

Note the ratio



Change viewing size - by viewport (Not recommended)

```
    glBegin(GL_LINES)
    for x in range(0, 20):
        glColor3f(0,1-x/19,0)
        glVertex3f(10*x,0,0)
        glVertex3f(10*x,200,0)
    glEnd()

def display():
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    global theda
    theda = theda +1
    glViewport(0, 0, windowWidth-theda%400, windowHeight-theda%400)
    glOrtho(-float(windowWidth)/2.0,float(windowWidth)/2.0,-float(windo
    gluLookAt(1000,1000,1000,0,0,0,0,0,1)
    glEnable(GL_LIGHTING)
    visualization.draw(meshes)
    glDisable(GL_LIGHTING)
    drawGrid()
    glutSwapBuffers()
    glutPostRedisplay()
    sleep(0.01)


def reshape(width,height):
    glViewport(0, 0, width, height)

def keyboard( key, x, y ):
    if key == esc:
        sys.exit()
```



Control the position of camera orthographic mode

glOrtho function

05/31/2018 • 2 minutes to read •

The **glOrtho** function multiplies the current matrix by an orthographic matrix.

Syntax

C++ Copy

```
void WINAPI glOrtho(
   GLdouble left,
   GLdouble right,
   GLdouble bottom,
   GLdouble top,
   GLdouble zNear,
   GLdouble zFar
);
```

https://docs.microsoft.com/en-us/windows/win32/opengl/glortho

Control the position of camera perspective mode

glFrustum function

05/31/2018 • 2 minutes to read •

The **glFrustum** function multiplies the current matrix by a perspective matrix.

Syntax

C++ Copy

```
void WINAPI glFrustum(
   GLdouble left,
   GLdouble right,
   GLdouble bottom,
   GLdouble top,
   GLdouble zNear,
   GLdouble zFar
);
```

https://docs.microsoft.com/en-us/windows/win32/opengl/glfrustum

Control the position of camera perspective mode

gluPerspective function

05/31/2018 • 2 minutes to read •

The **gluPerspective** function sets up a perspective projection matrix.

Syntax

C++ Copy

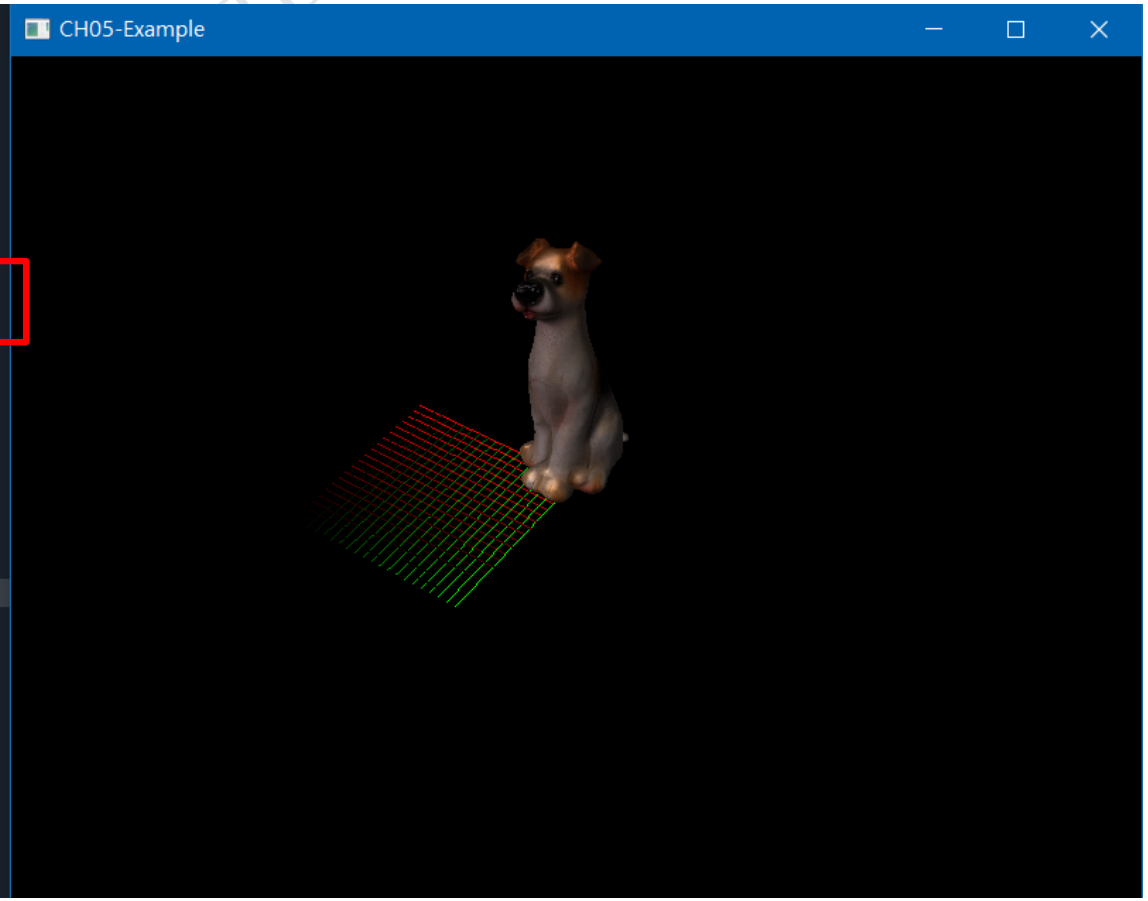
```
void WINAPI gluPerspective(
   GLdouble fovy,
   GLdouble aspect,
   GLdouble zNear,
   GLdouble zFar
);
```

https://docs.microsoft.com/en-us/windows/win32/opengl/gluperspective

Control the position of camera: perspective mode

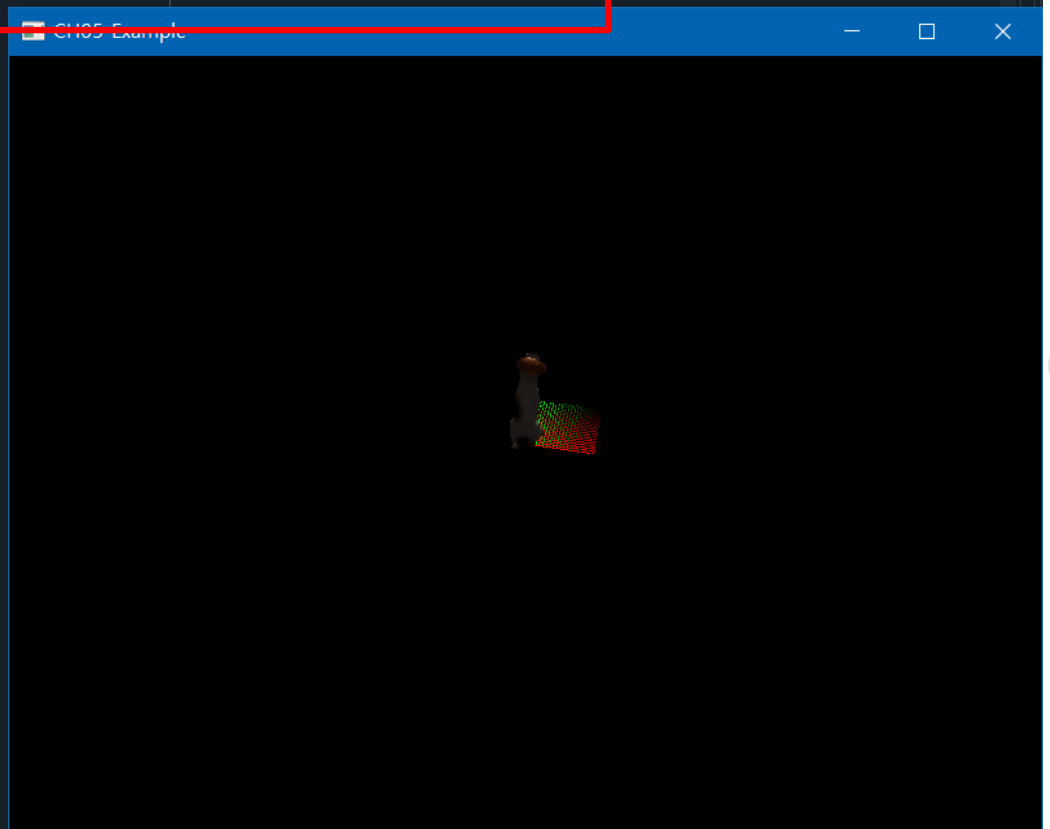
- glFrustum (assign parameters carefully)

```
38          glVertex3f(10*x,200,0)
39      glEnd()
40
41  def display():
42      glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
43      glMatrixMode(GL_PROJECTION)
44      glLoadIdentity()
45      glLightfv(GL_LIGHT0, GL_POSITION, lightPosition)
46      glViewport(0, 0, windowWidth, windowHeight)
47      glFrustum(-windowWidth/400,windowWidth/400,-windowHeight/400,windowHeight/400,5,4000)
48      global theda
49      theda = theda + 1.0/57.3;
50      gluLookAt(1000*cos(theda),1000*sin(theda),1000,0,0,0,0,0,1)
51      glEnable(GL_LIGHTING)
52      visualization.draw(meshes)
53      glDisable(GL_LIGHTING)
54      drawGrid()
55      glutSwapBuffers()
56      glutPostRedisplay()
57      sleep(0.01)
58
59
60  def reshape(width,height):
61      glViewport(0, 0, width, height)
62
63  def keyboard( key, x, y ):
64      if key == esc:
65          sys.exit()
66
67
68  glutInit()
```

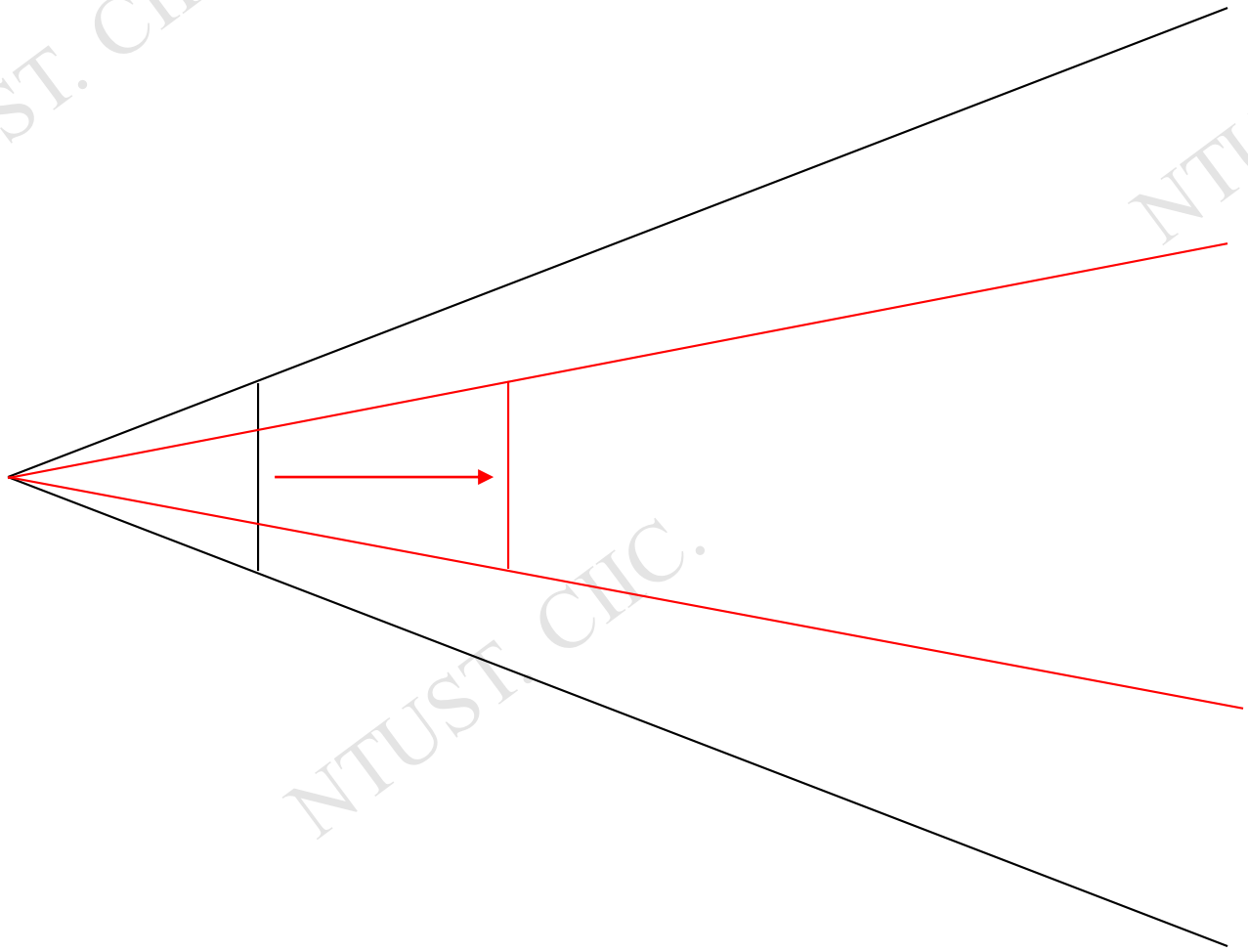


Control the position of camera: zoom by plane distance

```
41  def display():
42      glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
43      glMatrixMode(GL_PROJECTION)
44      glLoadIdentity()
45      glLightfv(GL_LIGHT0, GL_POSITION, lightPosition)
46      glViewport(0, 0, windowWidth, windowHeight)
47      global angle
48      angle = angle + 0.5/57.3
49      glFrustum(-windowWidth/400,windowWidth/400,-windowHeight/400,windowHeight/400,5+4*cos(angle),4000)
50      global theda
51      theda = theda + 1.0/57.3;
52      gluLookAt(1000*cos(theda),1000*sin(theda),1000,0,0,0,0,0,1)
53      glEnable(GL_LIGHTING)
54      visualization.draw(meshes)
55      glDisable(GL_LIGHTING)
56      drawGrid()
57      glutSwapBuffers()
58      glutPostRedisplay()
59      sleep(0.01)
60
61
62  def reshape(width,height):
63      glViewport(0, 0, width, height)
64
65  def keyboard( key, x, y ):
66      if key == esc:
67          sys.exit()
68
69
70  glutInit()
71  glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA)
72  glutCreateWindow(b'CH05-Example')
73  glutReshapeWindow(windowWidth,windowHeight)
74  glutReshapeFunc(reshape)
75  glutDisplayFunc(display)
76  glutKeyboardFunc(keyboard)
77  glEnable(GL_DEPTH_TEST)
78  glEnable(GL_LIGHTING)
79  glEnable(GL_LIGHT0)
80  glLightfv(GL_LIGHT0, GL_AMBIENT, lightAmbient)
81  glLightfv(GL_LIGHT0, GL_DIFFUSE, lightAmbient)
```

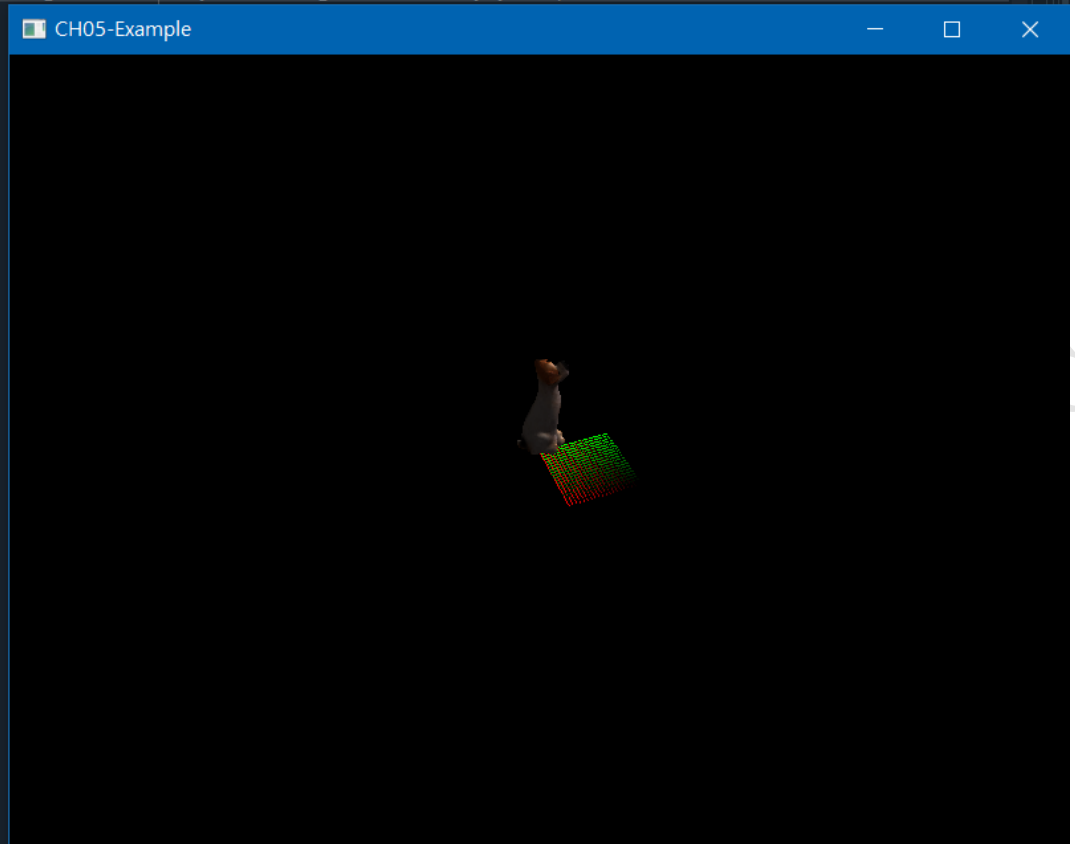


Control the position of camera: zoom by plane distance

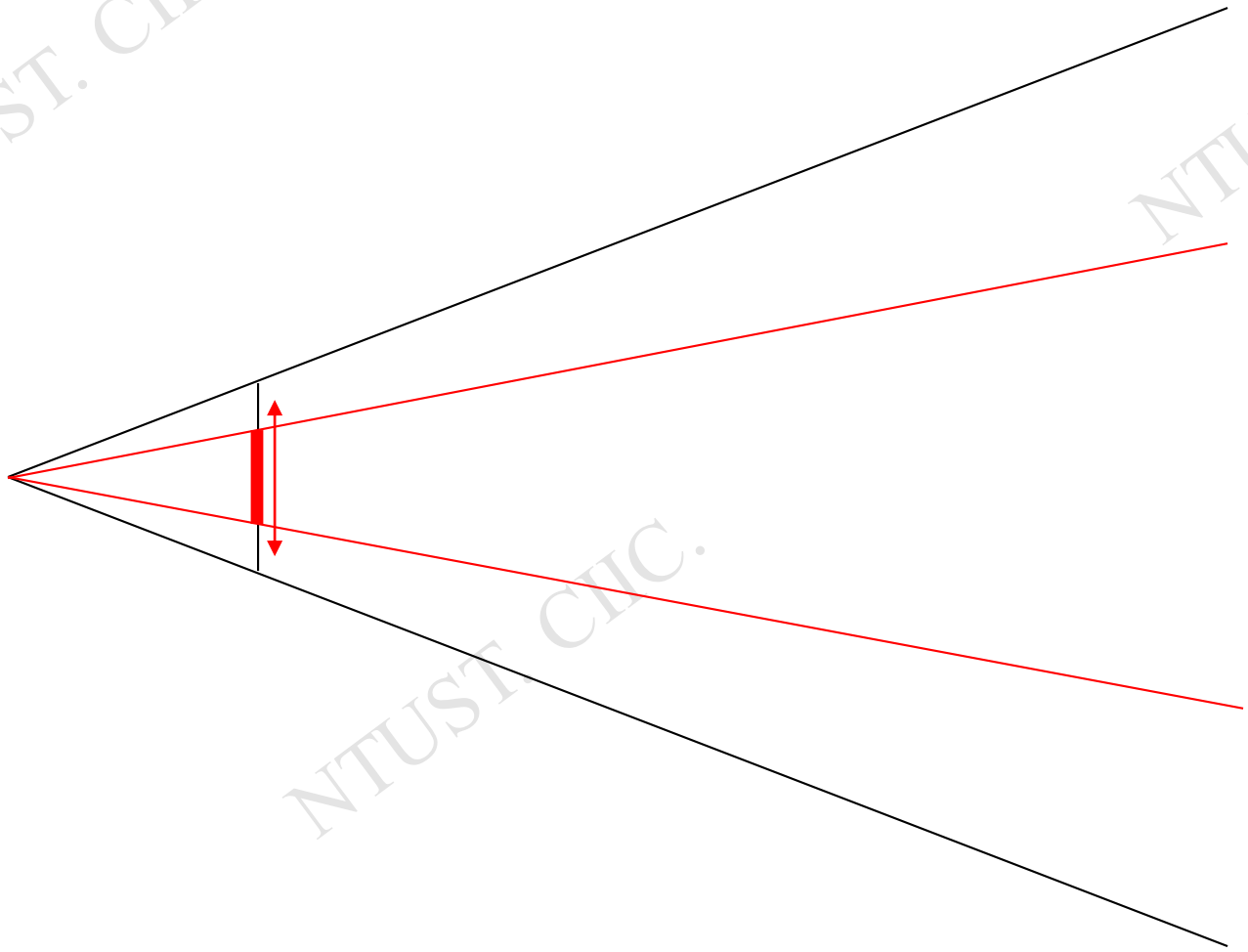


Control the position of camera: zoom by plane size

```
41  def display():
42      glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
43      glMatrixMode(GL_PROJECTION)
44      glLoadIdentity()
45      glLightfv(GL_LIGHT0, GL_POSITION, lightPosition)
46      glViewport(0, 0, windowWidth, windowHeight)
47      global angle
48      angle = angle + 0.5/57.3
49      value = (1.5 + cos(angle) )
50      glFrustum(-windowWidth/400*value,windowWidth/400*value,-windowHeight/400*value,windowHeight/400*value,5,4000)
51      global theda
52      theda = theda + 1.0/57.3;
53      gluLookAt(1000*cos(theda),1000*sin(theda),1000,0,0,0,0,0,1)
54      glEnable(GL_LIGHTING)
55      visualization.draw(meshes)
56      glDisable(GL_LIGHTING)
57      drawGrid()
58      glutSwapBuffers()
59      glutPostRedisplay()
60      sleep(0.01)
61
62
63  def reshape(width,height):
64      glViewport(0, 0, width, height)
65
66  def keyboard( key, x, y ):
67      if key == esc:
68          sys.exit()
69
70
71  glutInit()
72  glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA)
73  glutCreateWindow(b'CH05-Example')
74  glutReshapeWindow(windowWidth,windowHeight)
75  glutReshapeFunc(reshape)
76  glutDisplayFunc(display)
77  glutKeyboardFunc(keyboard)
78  glEnable(GL_DEPTH_TEST)
79  glEnable(GL_LIGHTING)
80  glEnable(GL_LIGHT0)
81  glLightfv(GL_LIGHT0, GL_AMBIENT, lightAmbient)
82  glLightfv(GL_LIGHT0, GL_DIFFUSE, lightAmbient)
```

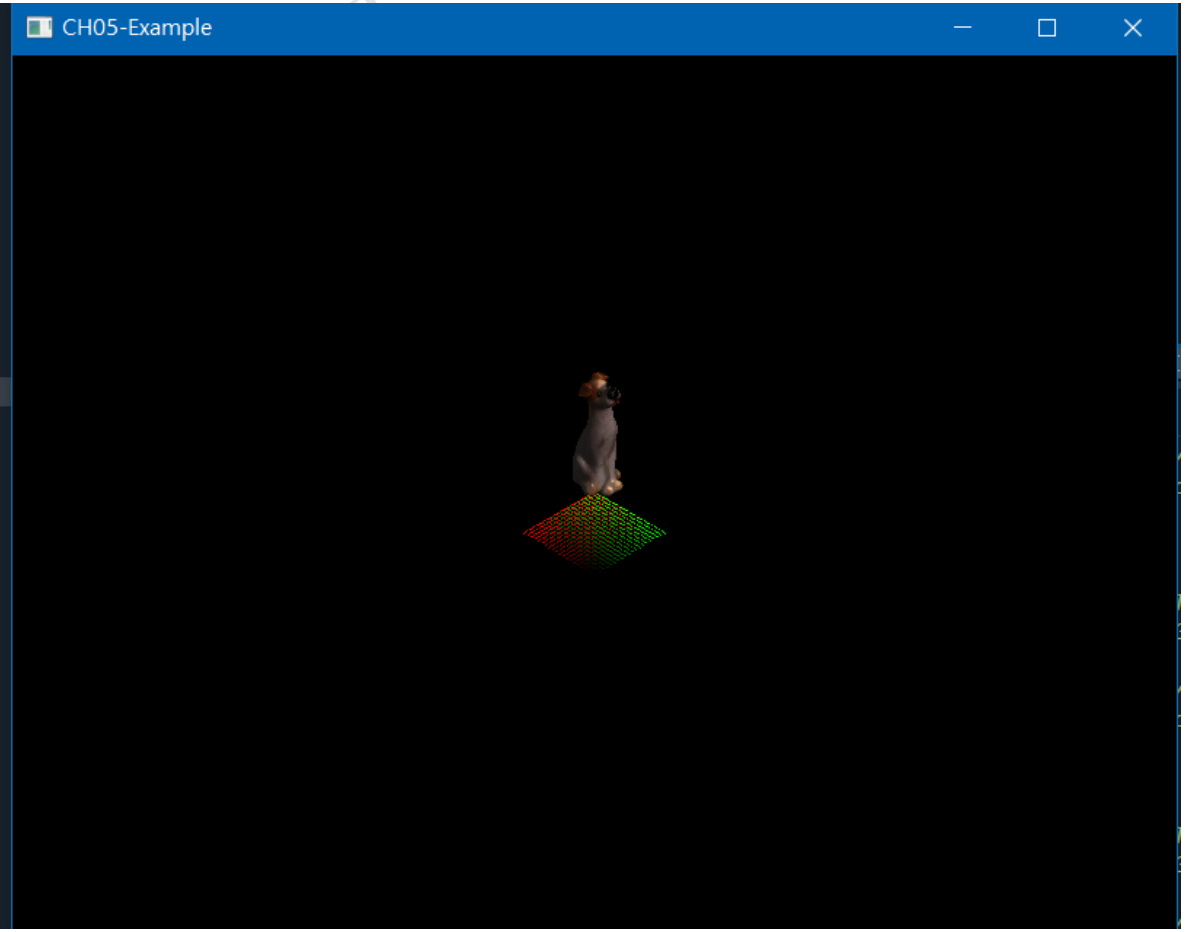


Control the position of camera: zoom by plane size



Control the position of camera: zoom by position

```
38          glVertex3f(10*x,200,0)
39      glEnd()
40
41  def display():
42      glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
43      glMatrixMode(GL_PROJECTION)
44      glLoadIdentity()
45      glLightfv(GL_LIGHT0, GL_POSITION, lightPosition)
46      glViewport(0, 0, windowWidth, windowHeight)
47      glFrustum(-windowWidth/400,windowWidth/400,-windowHeight/400,windowHeight/400,5,4000)
48      global angle
49      angle = angle + 0.5/57.3
50      value = (1.5 + cos(angle) )
51      gluLookAt(1000*value,1000*value,1000*value,0,0,0,0,0,1)
52      glEnable(GL_LIGHTING)
53      visualization.draw(meshes)
54      glDisable(GL_LIGHTING)
55      drawGrid()
56      glutSwapBuffers()
57      glutPostRedisplay()
58      sleep(0.01)
59
60
61  def reshape(width,height):
62      glViewport(0, 0, width, height)
63
64  def keyboard( key, x, y ):
65      if key == esc:
66          sys.exit()
67
68
69  glutInit()
```



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

Find us on: facebook