Transformation

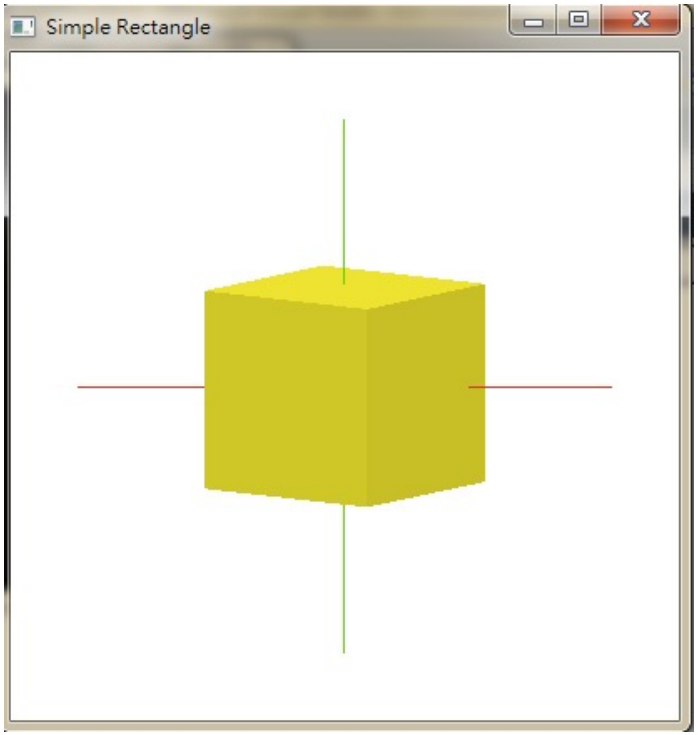
Computer Graphics
Lab02

Exercise - Moving the cube

1. First, make sure your program draws a static cube
2. Try to implement your code:
   - Draw lines to represent x, y, z axis
   - Keyboard control
     - Rotate along x , y, z , respectively
     - Translate along x, y, z, respectively
     - Reset to the original status

Hint: In this execrise, you must do glRotate first and then gltranslate.

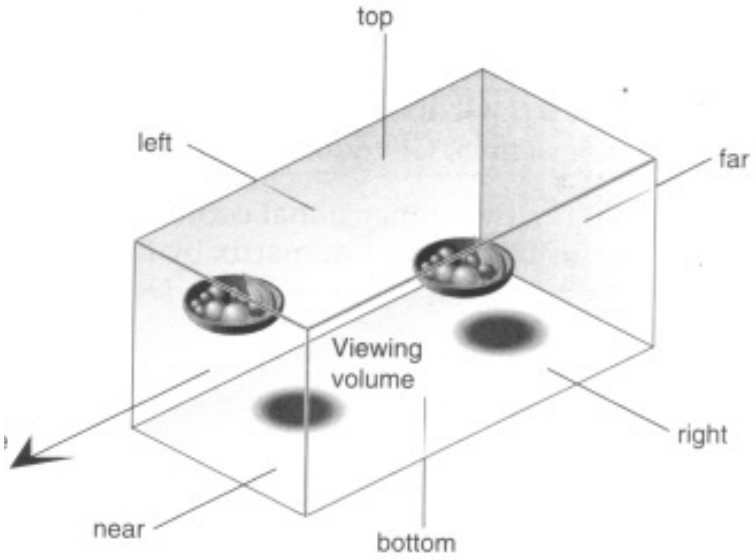
Exercise



glRotate(angle, x, y, z)
glTranslatef(tx,ty,tz)

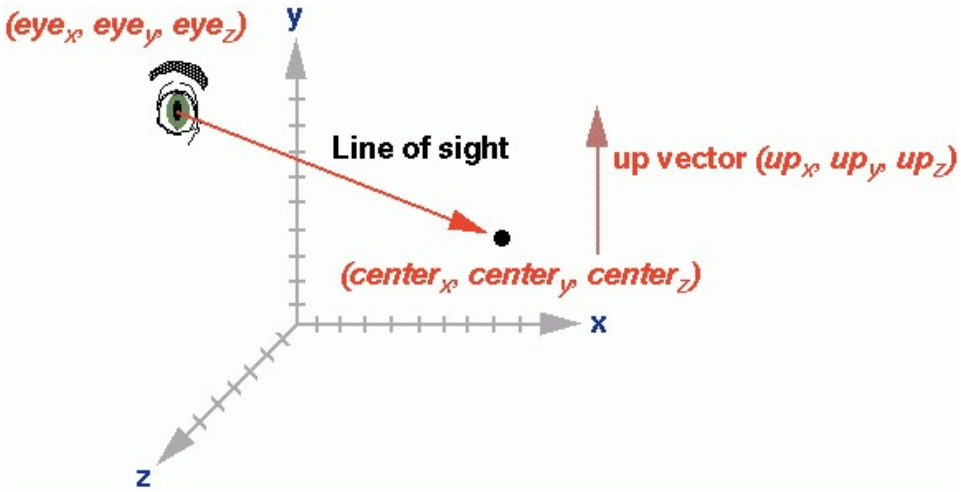
glOrtho

- glOrtho(-10,10,-10,10,-10,20);



gluLookAt

- gluLookAt(0,0,-10.0f ,0,0,0, 0,1,0);



Main

```
int main(int argc, char** argv)
{
    //These are variable that you will need
    //to move your cube
    tx=0; ty=0; tz=0;
    thetaX=0; thetaY=0; thetaZ=0;
    //

    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(400,400);
    glutInitWindowPosition(600,80);
    glutCreateWindow("Moving Cube");
    SetupRC();
    glutReshapeFunc(ChangeSize);
    glutDisplayFunc(RenderScene);
    glutKeyboardFunc(myKeyboard);
    glutSpecialFunc(mySpecialKey);

    glutMainLoop();
    return 0;
}
```

lighting

```
void SetupRC()
    {
    // Light values and coordinates
    GLfloat  whiteLight[] = { 0.45f, 0.45f, 0.45f, 1.0f };
    GLfloat  sourceLight[] = { 0.25f, 0.25f, 0.25f, 1.0f };
    Glfloat   lightPos[] = { 0.f, 25.0f, 20.0f, 0.0f };

    // Enable lighting
    glEnable(GL_LIGHTING);

    // Setup and enable light 0
    glLightModelfv(GL_LIGHT_MODEL_AMBIENT,whiteLight);
    glLightfv(GL_LIGHT0,GL_AMBIENT,sourceLight);
    glLightfv(GL_LIGHT0,GL_DIFFUSE,sourceLight);
    glLightfv(GL_LIGHT0,GL_POSITION,lightPos);
    glEnable(GL_LIGHT0);

    // Enable color tracking
    glEnable(GL_COLOR_MATERIAL);
    glEnable(GL_DEPTH_TEST);

}
```

Reshape Function

```
void ChangeSize(int w, int h)
{
    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION); // load the projection matrix
    glLoadIdentity();
    glOrtho(-10,10,-10,10,-10,20);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}
```

Display Function

```
void RenderScene(void)
{
    glClearColor(1.0, 1.0, 1.0, 1.0);
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glMatrixMode(GL_MODELVIEW); // load the modelview matrix
    glLoadIdentity();
    gluLookAt(0,0,10.0f ,0,0,0, 0,1,0);

    //draw X-axis, Y-axis and Z-axis
    //use:
    //glColor3f( r, g, b);
    //glBegin(GL_LINE);
    //glVertex3f( -x, 0, 0);
    //glVertex3f( x, 0, 0);
    //glEnd();

    //perform transformation for the cube
    //use:
    //glRotatef(theta, x, y, z);
    //glTranslatef(tx,ty,tz)


    //cube
    glColor3f( 1, 1, 0);
    glutSolidCube(6);
    glutSwapBuffers();
}
```

← Implement your code here

keyboard Function

```
void myKeyboard(unsigned char key, int x, int y)
{
  switch (key)
  {
  case 'r':
      //reset translation & rotation
      break;
  case 'a':
      //change the rotation angle thetaX along x-axis
      break;
  case 'd':
      //change the rotation angle thetaX along x-axis
      break;
  case 'w':
      //change the rotation angle thetaY along y-axis
      break;
  case 's':
      //change the rotation angle thetaY along y-axis
      break;
  case 'z':
      //change the rotation angle thetaZ along z-axis
      break;
  case 'x':
      //change the rotation angle thetaZ along z-axis
      break;
  default:
      break;
  }
  glutPostRedisplay();

}
```

For example:

thetaX+=3;

In DisplayFunc:
glRotatef(thetaX, 1,0,0);

Special keyboard Function

```
void mySpecialKey(int key, int x, int y)
{
    switch (key)
  {

  case GLUT_KEY_LEFT:
      //change the translation along x-axis
      break;
  case GLUT_KEY_RIGHT:
      //change the translation along x-axis
      break;
  case GLUT_KEY_UP:
      //change the translation along y-axis
      break;
  case GLUT_KEY_DOWN:
      //change the translation along y-axis
      break;
  default:
      break;

  }
  glutPostRedisplay();

}
```

For example:

tx-=0.2;

In DisplayFunc:
glTranslatef(tx,ty,tz);