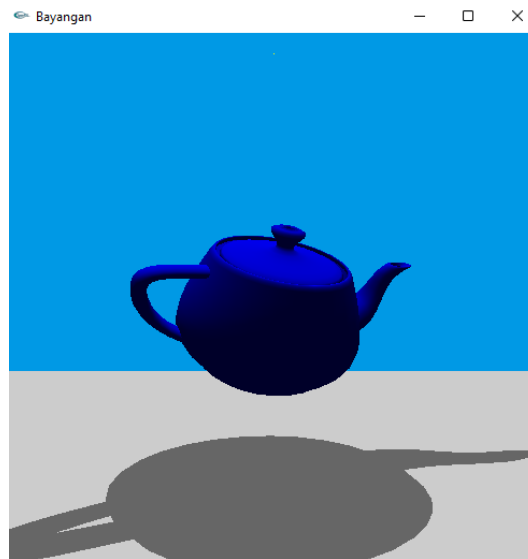


Nama : Adiba Justinian
NIM : 24060120130080
Praktikum GKV B1

Tugas Praktikum 5

Bayangan

1. Ubahlah untuk object yang berbeda



Source code :

```
#include <math.h>
#include <stdio.h>
#include <GL/glut.h>

double rx = 0.0;
double ry = 0.0;
float l[] = { 0.0, 80.0, 0.0 }; // koordinat sumber cahaya
float n[] = { 0.0, -40.0, 0.0 };
float e[] = { 0.0, -60.0, 0.0 };
void help();

// obyek yang akan digambar
void draw() {
    glutSolidTeapot(30.0); //Isi dengan salah satu saja, dan berilah
    //obyek yang selain dalam tabel ini
    //glutSolidTorus(20, 40, 20,10);
    //glutWireSphere(60,10,10);
    //glutSolidCube (40);
    //glutSolidCone(20,50,40,50);
}

//membuat proyeksi bayangan
```

Nama : Adiba Justinian

NIM : 24060120130080

Praktikum GKV B1

```
void glShadowProjection(float * l, float * e, float * n)
{
    float d, c;
    float mat[16];
    d = n[0]*l[0] + n[1]*l[1] + n[2]*l[2];
    c = e[0]*n[0] + e[1]*n[1] + e[2]*n[2] - d;
    mat[0] = l[0]*n[0]+c; // membuat matrik. OpenGL menggunakan
    kolom matrik
    mat[4] = n[1]*l[0];
    mat[8] = n[2]*l[0];
    mat[12] = -l[0]*c-l[0]*d;

    mat[1] = n[0]*l[1];
    mat[5] = l[1]*n[1]+c;
    mat[9] = n[2]*l[1];
    mat[13] = -l[1]*c-l[1]*d;

    mat[2] = n[0]*l[2];
    mat[6] = n[1]*l[2];
    mat[10] = l[2]*n[2]+c;
    mat[14] = -l[2]*c-l[2]*d;

    mat[3] = n[0];
    mat[7] = n[1];
    mat[11] = n[2];
    mat[15] = -d;
    glMultMatrixf(mat); // kalikan matrik
}

void render()
{
    glClearColor(0.0,0.6,0.9,0.0);
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
    glLightfv(GL_LIGHT0, GL_POSITION, l);
    glDisable(GL_CULL_FACE);
    glDisable(GL_LIGHTING);
    glColor3f(1.0,1.0,0.0);
    glBegin(GL_POINTS);
    glVertex3f(l[0],l[1],l[2]);
    glEnd();
    glColor3f(0.8,0.8,0.8);
    glBegin(GL_QUADS);
    glNormal3f(0.0,1.0,0.0);
    glVertex3f(-1300.0,e[1]-0.1, 1300.0);
    glVertex3f( 1300.0,e[1]-0.1, 1300.0);
    glVertex3f( 1300.0,e[1]-0.1,-1300.0);
    glVertex3f(-1300.0,e[1]-0.1,-1300.0);
    glEnd();
    // gambar bayangan
    glPushMatrix();
    glRotatef(ry,0,1,0);
    glRotatef(rx,1,0,0);
    glEnable(GL_LIGHTING);
}
```

Nama : Adiba Justinian

NIM : 24060120130080

Praktikum GKV B1

```
        glColor3f(0.0,0.0,0.8);
        draw();
        glPopMatrix();
        //sekarang gambar bayangan yang muncul
        glPushMatrix();
        glShadowProjection(1,e,n);
        glRotatef(ry,0,1,0);
        glRotatef(rx,1,0,0);
        glDisable(GL_LIGHTING);
        glColor3f(0.4,0.4,0.4);
        draw();
        glPopMatrix();
        glutSwapBuffers();
    }

void keypress(unsigned char c, int a, int b)
{
    if ( c==27 ) exit(0);
    else if ( c=='s' ) l[1]-=5.0;
    else if ( c=='w' ) l[1]+=5.0;
    else if ( c=='a' ) l[0]-=5.0;
    else if ( c=='d' ) l[0]+=5.0;
    else if ( c=='q' ) l[2]-=5.0;
    else if ( c=='e' ) l[2]+=5.0;
    else if ( c=='h' ) help();
}

void help()
{
    printf("proyeksi contoh bayangan sebuah obyek teapot\n");
}

void idle()
{
    rx+=0.1;
    ry+=0.1;
    render();
}

void resize(int w, int h)
{
    glViewport(0, 0, w, h);
}

int main(int argc, char * argv[])
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowPosition(500, 30);
    glutCreateWindow("Bayangan");
    glutReshapeFunc(resize);
    glutReshapeWindow(500, 500);
    glutKeyboardFunc(keypress);
}
```

Nama : Adiba Justinian

NIM : 24060120130080

Praktikum GKV B1

```
glutDisplayFunc(render);
glutIdleFunc(idle);
glEnable(GL_NORMALIZE);
glEnable(GL_LIGHTING);
glEnable(GL_COLOR_MATERIAL);
glEnable(GL_DEPTH_TEST);
glEnable(GL_LIGHT0);
glEnable(GL_TEXTURE_2D);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
gluPerspective(60.0f, 1.0, 1.0, 400.0);
// Reset koordinat sebelum dimodifikasi/diubah
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
glTranslatef(0.0, 0.0, -150.0);
glutMainLoop();
return 0;
}
```

2. Analisis dan Jelaskan secara rinci proses membuat bayangannya.

Jawab :

Dalam pembuatan bayangan, kita melakukan inisialisasi terhadap l (sumber cahaya), n (normal vector), dan e (dasar untuk letak bayangan) terlebih dahulu.

Untuk membuat proyeksi, digunakan prosedur `glShadowProjection(float *l, float *e, float *n)`. Prosedur ini berisi perkalian matriks dengan variabel yang merupakan hasil perkalian koordinat sumber cahaya dan ModelView Matriks. Objek yang digambarkan setelah pemanggilan prosedur ini akan diproyeksikan ke bawah.

Bagian prosedur render digunakan untuk mengatur seluruh adegan dari sudut sumber cahaya dan diletakkan di z . Langkah pada render ini menggunakan 2 buah objek yang sama, objek pertama adalah objek yang tidak terkena proyeksi bayangan (objek aslinya), sedangkan objek kedua adalah objek duplikasi dari objek pertama yang dikenai proyeksi bayangan sehingga nampak sebagai bayangan bendanya, kemudian bayangan akan muncul dengan melakukan `shadowProjection` pada benda aslinya.

Nama : Adiba Justinian

NIM : 24060120130080

Praktikum GKV B1

3. Jelaskan fungsi dan alur `glShadowProjection`.

Jawab :

Prosedur `glShadowProjection` berfungsi untuk membuat proyeksi dari benda/objek membentuk sebuah bayangan. Prosedur ini memiliki 3 parameter masukan, yaitu l , n , dan e .

Dalam prosesnya, prosedur ini berjalan dengan memasukkan ketiga parameter tadi ke dalam matriks, kemudian `glShadowProjection` akan menghasilkan sebuah matriks untuk proyeksi cahaya sebagai bayangan. Ada 2 variabel yang diinisialisasi yaitu variabel d dan c yang mana digunakan untuk mencari dot product antara vector sumber cahaya (l) dengan vector normal (n), dan vector dasar (e). Kemudian `mat[16]` adalah inisialisasi dari matriks yang tadi sudah dibuat dari 0 sampai 15. Matriks tersebut dikalikan dengan menggunakan `glMultMatrixf(mat)` untuk menghasilkan proyeksi cahaya dan membentuk bayangannya