**Лабораторийн ажил №7-8**

**Shading & lighting & Texturing**

Лабораторийн ажлын даалгавар:

1. Дурын 3D хэмжээст объектыг сонгон авч зурна
2. 1 даалгавар дээр “Lighting” өөрчлөлтүүдийг нэмнэ

Энэхүү лабораторийн ажлаар өмнөх код дээр гэрэлтүүлэгийг тооцоолж, үр дүн гарган авна. Параметр утгыг өөрчлөх замаар жишээ кодыг бүрэн эзэмшинэ үү.

Жишээ:

/\*

\* material.c

\* Энэүү програм нь GL гэрэлтүүлгийн моделийн хэрэглээг харуулдаг.

\* Янз бүрийн материалын шинж чанаруудыг ашиглан хэд хэдэн объектыг зурдаг.

\* Дан ганц light source объектийг гэрэлтүүлнэ.

\*/

#include <cstdlib>

#include <GL/glut.h>

/\* z-buffer, projection matrix, light source болон lighting model Initialize хийнэ.

\* Энд материалын шинж чанарыг тодорхойлохгүй.

\*/

void init(void)

{

GLfloat ambient[] = { 0.0, 0.0, 0.0, 1.0 };

GLfloat diffuse[] = { 1.0, 1.0, 1.0, 1.0 };

GLfloat specular[] = { 1.0, 1.0, 1.0, 1.0 };

GLfloat position1[] = { 0.0, 4.0, 3.0, 1.0 };

GLfloat position2[] = { 0.0, -4.0, 3.0, 1.0 };

GLfloat lmodel\_ambient[] = { 0.4, 0.4, 0.4, 1.0 };

GLfloat local\_view[] = { 0.0 };

glClearColor(0.0, 0.1, 0.1, 0.0);

glEnable(GL\_DEPTH\_TEST);

glShadeModel(GL\_SMOOTH);

glLightfv(GL\_LIGHT0, GL\_AMBIENT, ambient);

glLightfv(GL\_LIGHT0, GL\_DIFFUSE, diffuse);

glLightfv(GL\_LIGHT0, GL\_POSITION, position1);

glLightfv(GL\_LIGHT1, GL\_AMBIENT, ambient);

glLightfv(GL\_LIGHT1, GL\_DIFFUSE, diffuse);

glLightfv(GL\_LIGHT1, GL\_POSITION, position2);

glLightModelfv(GL\_LIGHT\_MODEL\_AMBIENT, lmodel\_ambient);

glLightModelfv(GL\_LIGHT\_MODEL\_LOCAL\_VIEWER, local\_view);

glEnable(GL\_LIGHTING);

glEnable(GL\_LIGHT0);

glEnable(GL\_LIGHT1);

}

//void init(void)

//{

// GLfloat light\_position[] = {0.0, 2.0, 3.0, 1.0};

// GLfloat light\_position2[] = {0.0, -2.0, 3.0, 1.0};

//

// glClearColor (0.0, 0.0, 0.0, 0.0);

// glShadeModel (GL\_SMOOTH);

//

// glEnable(GL\_LIGHTING);

// glEnable(GL\_LIGHT0);

// glEnable(GL\_LIGHT1);

//

// glLightfv(GL\_LIGHT0, GL\_POSITION, light\_position);

// glLightfv(GL\_LIGHT1, GL\_POSITION, light\_position2);

// glEnable(GL\_DEPTH\_TEST);

//}

void display(void)

{

glClear (GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

GLfloat no\_mat[] = { 0.0, 0.0, 0.0, 1.0 };

GLfloat mat\_ambient[] = { 0.7, 0.7, 0.7, 1.0 };

GLfloat mat\_ambient\_color[] = { 0.8, 0.8, 0.2, 1.0 };

GLfloat mat\_diffuse[4] = { 0.0, 0.0, 0.0, 1.0 };

GLfloat mat\_specular[] = { 1.0, 1.0, 1.0, 1.0 };

GLfloat no\_shininess[] = { 0.0 };

GLfloat low\_shininess[] = { 5.0 };

GLfloat high\_shininess[] = { 100.0 };

GLfloat mat\_emission[] = {0.3, 0.2, 0.2, 0.0};

glPushMatrix();

glTranslatef(-3,0,0);

glMaterialfv(GL\_FRONT, GL\_AMBIENT, no\_mat);

mat\_diffuse[1] = 0.5;

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, mat\_diffuse);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, no\_mat);

glMaterialfv(GL\_FRONT, GL\_SHININESS, no\_shininess);

glMaterialfv(GL\_FRONT, GL\_EMISSION, no\_mat);

glScalef(1.5,1.5,1.5);

glutSolidSphere (1, 20, 16);

glPopMatrix();

glPushMatrix();

glTranslatef(3,0,0);

glMaterialfv(GL\_FRONT, GL\_AMBIENT, no\_mat);

mat\_diffuse[1] = 0.0;

mat\_diffuse[2] = 0.5;

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, mat\_diffuse);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, mat\_specular);

glMaterialfv(GL\_FRONT, GL\_SHININESS, high\_shininess);

glMaterialfv(GL\_FRONT, GL\_EMISSION, no\_mat);

glutSolidSphere (1, 20, 16);

glPopMatrix();

glFlush ();

}

void reshape (int w, int h)

{

glViewport(0, 0, w, h);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

if (w <= (h \* 2))

glOrtho (-6.0, 6.0, -3.0\*((GLfloat)h\*2)/(GLfloat)w,

3.0\*((GLfloat)h\*2)/(GLfloat)w, -10.0, 10.0);

else

glOrtho (-6.0\*(GLfloat)w/((GLfloat)h\*2),

6.0\*(GLfloat)w/((GLfloat)h\*2), -3.0, 3.0, -10.0, 10.0);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

}

void keyboard(unsigned char key, int x, int y)

{

switch (key) {

case 27:

exit(0);

break;

}

}

int main(int argc, char\*\* argv)

{

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_SINGLE | GLUT\_RGB | GLUT\_DEPTH);

glutInitWindowSize (500, 500);

glutInitWindowPosition (100, 100);

glutCreateWindow (argv[0]);

init ();

glutDisplayFunc(display);

glutReshapeFunc(reshape);

glutKeyboardFunc(keyboard);

glutMainLoop();

return 0;

}

1. Texture оруулна.

Энэ дасгалд өмнөх код дээр үндэслэн тухайн орчны өнгө (specular colour) болон объектийн эрчмийг тус тусад нь ялгах ambient болон diffuse тодорхойлсон 2 texture нэмж оруулах шаардлагатай.

Тexture оруулах функцуудыг болон хэрхэн хэрэгжүүлэх талаар судлах. <http://cse.csusb.edu/tongyu/courses/cs520/notes/texture.php> линкээр орж дэлгэрэнгүй судална уу.