Nama : aliefian rizky subagia

Npm : 06.2022.1.90527

Jawaban uas grafika komputer

Soal

untuk project akhirnya nanti membuat video presentasi saja, jadi tidak perlu presentasi langsung ke saya. Bisa di upload di youtube masing2. Nanti saya buatkan tugas di classroom, diisi link youtube nya.

Yang di presentasikan adalah gambar 2D/3D apa yang dibuat, cara membuat tiap bentuknya, teknik grafika apa yang dibuat untuk membuat bentuknya (misalnya rotasi/scalling/translasi)

1. Link github program :
2. Link youtube part 1 :

Link youtube part 2 :

1. Program project akhir (program project uas )

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <GL/glut.h>

GLenum doubleBuffer;

float rotX = 0.0, rotY = 0.0;

int teaList;

long patchData[][16] = {

{102,103,104,105,4,5,6,7,8,9,10,11,12,13,14,15},

{12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27},

{24,25,26,27,29,30,31,32,33,34,35,36,37,38,39,40},

{96,96,96,96,97,98,99,100,101,101,101,101,0,1,2,3,},

{0,1,2,3,106,107,108,109,110,111,112,113,114,115,116,117},

{118,118,118,118,124,122,119,121,123,126,125,120,40,39,38,37},

{41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56},

{53,54,55,56,57,58,59,60,61,62,63,64,28,65,66,67},

{68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83},

{80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95}

};

float cpData[][3] = {

{0.2,0,2.7},{0.2,-0.112,2.7},{0.112,-0.2,2.7},{0,-0.2,2.7},

{1.3375,0,2.53125},{1.3375,-0.749,2.53125},{0.749,-1.3375,2.53125},

{0,-1.3375,2.53125},{1.4375,0,2.53125},{1.4375,-0.805,2.53125},

{0.805,-1.4375,2.53125},{0,-1.4375,2.53125},{1.5,0,2.4},{1.5,-0.84,2.4},

{0.84,-1.5,2.4},{0,-1.5,2.4},{1.75,0,1.875},{1.75,-0.98,1.875},

{0.98,-1.75,1.875},{0,-1.75,1.875},{2,0,1.35},{2,-1.12,1.35},

{1.12,-2,1.35},{0,-2,1.35},{2,0,0.9},{2,-1.12,0.9},{1.12,-2,0.9},

{0,-2,0.9},{-2,0,0.9},{2,0,0.45},{2,-1.12,0.45},{1.12,-2,0.45},

{0,-2,0.45},{1.5,0,0.225},{1.5,-0.84,0.225},{0.84,-1.5,0.225},

{0,-1.5,0.225},{1.5,0,0.15},{1.5,-0.84,0.15},{0.84,-1.5,0.15},

{0,-1.5,0.15},{-1.6,0,2.025},{-1.6,-0.3,2.025},{-1.5,-0.3,2.25},

{-1.5,0,2.25},{-2.3,0,2.025},{-2.3,-0.3,2.025},{-2.5,-0.3,2.25},

{-2.5,0,2.25},{-2.7,0,2.025},{-2.7,-0.3,2.025},{-3,-0.3,2.25},

{-3,0,2.25},{-2.7,0,1.8},{-2.7,-0.3,1.8},{-3,-0.3,1.8},{-3,0,1.8},

{-2.7,0,1.575},{-2.7,-0.3,1.575},{-3,-0.3,1.35},{-3,0,1.35},

{-2.5,0,1.125},{-2.5,-0.3,1.125},{-2.65,-0.3,0.9375},{-2.65,0,0.9375},

{-2,-0.3,0.9},{-1.9,-0.3,0.6},{-1.9,0,0.6},{1.7,0,1.425},

{1.7,-0.66,1.425},{1.7,-0.66,0.6},{1.7,0,0.6},{2.6,0,1.425},

{2.6,-0.66,1.425},{3.1,-0.66,0.825},{3.1,0,0.825},{2.3,0,2.1},

{2.3,-0.25,2.1},{2.4,-0.25,2.025},{2.4,0,2.025},{2.7,0,2.4},

{2.7,-0.25,2.4},{3.3,-0.25,2.4},{3.3,0,2.4},{2.8,0,2.475},

{2.8,-0.25,2.475},{3.525,-0.25,2.49375},{3.525,0,2.49375},

{2.9,0,2.475},{2.9,-0.15,2.475},{3.45,-0.15,2.5125},{3.45,0,2.5125},

{2.8,0,2.4},{2.8,-0.15,2.4},{3.2,-0.15,2.4},{3.2,0,2.4},{0,0,3.15},

{0.8,0,3.15},{0.8,-0.45,3.15},{0.45,-0.8,3.15},{0,-0.8,3.15},

{0,0,2.85},{1.4,0,2.4},{1.4,-0.784,2.4},{0.784,-1.4,2.4},{0,-1.4,2.4},

{0.4,0,2.55},{0.4,-0.224,2.55},{0.224,-0.4,2.55},{0,-0.4,2.55},

{1.3,0,2.55},{1.3,-0.728,2.55},{0.728,-1.3,2.55},{0,-1.3,2.55},

{1.3,0,2.4},{1.3,-0.728,2.4},{0.728,-1.3,2.4},{0,-1.3,2.4},{0,0,0},

{1.425,-0.798,0},{1.5,0,0.075},{1.425,0,0},{0.798,-1.425,0},

{0,-1.5,0.075},{0,-1.425,0},{1.5,-0.84,0.075},{0.84,-1.5,0.075}

};

void Teapot(long grid)

{

float p[4][4][3], q[4][4][3], r[4][4][3], s[4][4][3];

long i, j, k, l;

teaList = 1;

glNewList(teaList, GL\_COMPILE);

glPushMatrix();

glRotatef(270.0, 1.0, 0.0, 0.0);

for (i = 0; i < 10; i++) {

for (j = 0; j < 4; j++) {

for (k = 0; k < 4; k++) {

for (l = 0; l < 3; l++) {

p[j][k][l] = cpData[patchData[i][j\*4+k]][l];

q[j][k][l] = cpData[patchData[i][j\*4+(3-k)]][l];

if (l == 1) {

q[j][k][l] \*= -1.0;

}

if (i < 6) {

r[j][k][l] = cpData[patchData[i][j\*4+(3-k)]][l];

if (l == 0) {

r[j][k][l] \*= -1.0;

}

s[j][k][l] = cpData[patchData[i][j\*4+k]][l];

if (l == 0) {

s[j][k][l] \*= -1.0;

}

if (l == 1) {

s[j][k][l] \*= -1.0;

}

}

}

}

}

glMap2f(GL\_MAP2\_VERTEX\_3, 0, 1, 3, 4, 0, 1, 12, 4, &p[0][0][0]);

glEnable(GL\_MAP2\_VERTEX\_3);

glMapGrid2f(grid, 0.0, 1.0, grid, 0.0, 1.0);

glEvalMesh2(GL\_FILL, 0, grid, 0, grid);

glMap2f(GL\_MAP2\_VERTEX\_3, 0, 1, 3, 4, 0, 1, 12, 4, &q[0][0][0]);

glEvalMesh2(GL\_FILL, 0, grid, 0, grid);

if (i < 6) {

glMap2f(GL\_MAP2\_VERTEX\_3, 0, 1, 3, 4, 0, 1, 12, 4, &r[0][0][0]);

glEvalMesh2(GL\_FILL, 0, grid, 0, grid);

glMap2f(GL\_MAP2\_VERTEX\_3, 0, 1, 3, 4, 0, 1, 12, 4, &s[0][0][0]);

glEvalMesh2(GL\_FILL, 0, grid, 0, grid);

}

}

glDisable(GL\_MAP2\_VERTEX\_3);

glPopMatrix();

glEndList();

}

static void Init(void)

{

float position[] = {0.0, 3.0, 3.0, 0.0};

float local\_view[] = {0.0};

float ambient[] = {0.1745, 0.01175, 0.01175};

float diffuse[] = {0.61424, 0.04136, 0.04136};

float specular[] = {0.727811, 0.626959, 0.626959};

glEnable(GL\_DEPTH\_TEST);

glDepthFunc(GL\_LESS);

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

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

glFrontFace(GL\_CW);

glEnable(GL\_LIGHTING);

glEnable(GL\_LIGHT0);

glEnable(GL\_AUTO\_NORMAL);

glEnable(GL\_NORMALIZE);

glMaterialfv(GL\_FRONT, GL\_AMBIENT, ambient);

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, diffuse);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, specular);

glMaterialf(GL\_FRONT, GL\_SHININESS, 0.6\*128.0);

glClearColor(0.5, 0.5, 0.5, 1.0);

glColor3f(1.0, 1.0, 1.0);

Teapot(14);

}

static void Reshape(int w, int h)

{

glViewport(0, 0, (GLint)w, (GLint)h);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(-6.0, 6.0, -6.0, 6.0, -1.0, 10.0);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

}

static void Key(unsigned char key, int x, int y)

{

switch (key) {

case 27:

exit(0);

}

}

static void SpecialKey(int key, int x, int y)

{

switch (key) {

case GLUT\_KEY\_UP:

rotX -= 20.0;

glutPostRedisplay();

break;

case GLUT\_KEY\_DOWN:

rotX += 20.0;

glutPostRedisplay();

break;

case GLUT\_KEY\_LEFT:

rotY -= 10.0;

glutPostRedisplay();

break;

case GLUT\_KEY\_RIGHT:

rotY += 20.0;

glutPostRedisplay();

break;

}

}

static void Draw(void)

{

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

glPushMatrix();

glTranslatef(0.0, 0.0, -5.0);

glRotatef(rotY, 0.0,1.0,0.0);

glRotatef(rotX, 1.0,0.0,0.0);

glCallList(teaList);

glPopMatrix();

if (doubleBuffer) {

glutSwapBuffers();

} else {

glFlush();

}

}

static void Args(int argc, char \*\*argv)

{

GLint i;

doubleBuffer = GL\_FALSE;

for (i = 1; i < argc; i++) {

if (strcmp(argv[i], "-sb") == 0) {

doubleBuffer = GL\_FALSE;

} else if (strcmp(argv[i], "-db") == 0) {

doubleBuffer = GL\_TRUE;

}

}

}

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

{

GLenum type;

glutInit(&argc, argv);

Args(argc, argv);

type = GLUT\_RGB | GLUT\_DEPTH;

type |= (doubleBuffer) ? GLUT\_DOUBLE : GLUT\_SINGLE;

glutInitDisplayMode(type);

glutInitWindowSize(300, 300);

glutCreateWindow("TeaPot");

Init();

glutReshapeFunc(Reshape);

glutKeyboardFunc(Key);

glutSpecialFunc(SpecialKey);

glutDisplayFunc(Draw);

glutMainLoop();

}

1. Hasil running program nya

