

```

/* ITTextCG, Chapter 8, Exercise 2, Bezier Curve */
#include <GL/glut.h>

/* ベジエ曲面の制御点 */
GLfloat controlpoints[4][4][3] = { { {0.0, 0.0, 150.0},
                                       {50.0, 50.0, 150.0},
                                       {100.0, 50.0, 150.0},
                                       {150.0, 0.0, 150.0}},
                                     { {0.0, 50.0, 100.0},
                                       {50.0, 70.0, 100.0},
                                       {100.0, 70.0, 100.0},
                                       {150.0, 50.0, 100.0}},
                                     { {0.0, 50.0, 50.0},
                                       {50.0, 70.0, 50.0},
                                       {100.0, 70.0, 50.0},
                                       {150.0, 50.0, 50.0}},
                                     { {0.0, 0.0, 0.0},
                                       {50.0, 50.0, 0.0},
                                       {100.0, 50.0, 0.0},
                                       {150.0, 0.0, 0.0}} };

void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glFrustum(-50.0, 50.0, -50.0, 50.0, 50.0, 1000.0);
    gluLookAt(300.0, 300.0, 300.0, 0.0, 0.0, 0.0, 0.0,
              1.0, 0.0);

    glMatrixMode(GL_MODELVIEW);

```

```

glLoadIdentity();

/* ベジエ曲面の表示 */
glEvalMesh2(GL_FILL, 0, 8, 0, 8);

glFlush();
}

void keyboard(unsigned char key, int x, int y)
{
    switch(key) {
        case 27: exit(0); break;

    }

}

/* 材質と照明の設定のための関数 */
void LightSource()
{
    GLfloat mat_diffuse[] = { 1.0, 1.0, 1.0, 1.0 };
    GLfloat mat_specular[] = { 1.0, 1.0, 1.0, 1.0 };
    GLfloat mat_shininess[] = { 10.0 };
    GLfloat light_diffuse[] = { 0.6, 0.6, 0.6, 1.0 };
    GLfloat light_specular[] = { 1.0, 1.0, 1.0, 1.0 };
    GLfloat light_ambient[] = { 0.4, 0.4, 0.4, 1.0 };
    GLfloat light_position[] = { -0.5, 1.0, 0.5, 0.0 };

    glMaterialfv(GL_FRONT, GL_DIFFUSE, mat_diffuse);
    glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
    glMaterialfv(GL_FRONT, GL_SHININESS, mat_shininess);

```

```

    glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse);
    glLightfv(GL_LIGHT0, GL_SPECULAR, light_specular);
    glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient);
    glLightfv(GL_LIGHT0, GL_POSITION, light_position);

    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
    glEnable(GL_DEPTH_TEST);
}

int main(int argc, char** argv)
{
    glutInitWindowSize(1000,1000);
    glutInitWindowPosition(0,0);
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGBA | GLUT_DEPTH);
    glutCreateWindow("BezierSurface");

    /* ベジエ曲面のための二次元エバリュエータの定義 */
    glMap2f(GL_MAP2_VERTEX_3, 0, 1, 3, 4, 0, 1, 12, 4,
            &controlpoints[0][0][0]);
    glEnable(GL_MAP2_VERTEX_3);
    glEnable(GL_AUTO_NORMAL);
    glMapGrid2f(8, 0.0, 1.0, 8, 0.0, 1.0);

    LightSource();

    glClearColor(0.0, 0.0, 0.0, 0.0);

    glutDisplayFunc(display);
    glutKeyboardFunc(keyboard);
}

```

```
glutMainLoop();
```

```
}
```