

```

/* ITTextCG, Chapter 8, Exercise 5, NURBS Circle */
#include <GL/glut.h>

/* NURBS曲線の制御点 */
GLfloat controlpoints[9][3] = {{40.0, 0.0, 0.0},
                                {40.0, 40.0, 0.0},
                                {0.0, 40.0, 0.0},
                                {-40.0, 40.0, 0.0},
                                {-40.0, 0.0, 0.0},
                                {-40.0, -40.0, 0.0},
                                {0.0, -40.0, 0.0},
                                {40.0, -40.0, 0.0},
                                {40.0, 0.0, 0.0}};

/* NURBS曲線のノットベクトル */
GLfloat knots[12] = {0.0, 0.0, 0.0, 0.25, 0.25, 0.5, 0.5,
                    0.75, 0.75, 1.0, 1.0, 1.0};

/* NURBS曲線のポインター */
GLUnurbsObj *theNurbsCurve;

void display(void)
{
    int i;

    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(0.0, 0.0, 200.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);
}

```

```

glMatrixMode(GL_MODELVIEW);
glLoadIdentity();

/* NURBS曲線の制御点多角形の表示 */
glColor3f(1.0, 0.0, 0.0);
glBegin(GL_LINE_STRIP);
for(i = 0; i <= 8; ++i)
{
    glVertex3f((GLfloat)controlpoints[i][0],
               (GLfloat)controlpoints[i][1],
               (GLfloat)controlpoints[i][2]);
} /* for i */
glEnd();

/* NURBS曲線の計算と表示 */
glColor3f(1.0, 1.0, 1.0);
gluBeginCurve(theNurbsCurve);
    gluNurbsCurve(theNurbsCurve, 9+3, knots, 3,
                  &controlpoints[0][0],
                  2+1, GL_MAP1_VERTEX_3);
gluEndCurve(theNurbsCurve);

glFlush();
}

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

```

```

    }

}

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

    /* NURBS曲線ポインターの作成とNURBS曲線プロパティの定義 */
    theNurbsCurve = gluNewNurbsRenderer();
    gluNurbsProperty(theNurbsCurve,
                     GLU_SAMPLING_TOLERANCE, 1.0);

    glClearColor(0.0, 0.0, 0.0, 0.0);

    glutDisplayFunc(display);
    glutKeyboardFunc(keyboard);
    glutMainLoop();
}

```