

リスト 6.12

「変位マッピング」用の基本立体の例

```

void drawDispPlate(float size, int nDiv, int nRepeatS, int nRepeatT)
{
    //x-y平面
    float tnt[3];
    float s, t;
    float q[4][3]; //平面上的分割頂点
    float s0, s1, t0, t1, d0, d1;
    int i, j;
    float sz = 0.5 * size;
    static float p[4][3] = //z:上方向
    {
        { sz, sz, 0.0}, {-sz, sz, 0.0}, {-sz, -sz, 0.0}, { sz, -sz, 0.0}
    };

    s = (float)nRepeatS;
    t = (float)nRepeatT;
    GLuint tangentLoc = glGetAttribLocation(shaderProg, "tangent");
    glEnable(GL_TEXTURE_2D);
    glBegin(GL_QUADS);
    tnt[0] = 1.0; tnt[1] = 0.0; tnt[2] = 0.0;
    glVertexAttrib3fv(tangentLoc, tnt);
    glNormal3f(0.0, 0.0, 1.0); //z方向の法線
    for(j = 0; j < nDiv; j++)
    {
        d0 = (float)j / (float)nDiv;
        d1 = (float)(j+1) / (float)nDiv;
        //z座標
        q[0][2] = q[1][2] = q[2][2] = q[3][2] = p[0][2];
        //y座標
        q[0][1] = q[1][1] = p[2][1] + (p[1][1] - p[2][1]) * d0;
        q[2][1] = q[3][1] = p[2][1] + (p[1][1] - p[2][1]) * d1;
        //t テクスチャ座標
        t0 = t * d0; t1 = t * d1;
        for(i = 0; i < nDiv; i++)
        {
            d0 = (float)i / (float)nDiv;
            d1 = (float)(i+1) / (float)nDiv;
            //x座標
            q[0][0] = q[3][0] = p[1][0] + (p[0][0] - p[1][0]) * d0;
            q[1][0] = q[2][0] = p[1][0] + (p[0][0] - p[1][0]) * d1;
            //s テクスチャ座標
            s0 = s * d0; s1 = s * d1;
            glTexCoord2f(s0, t0); glVertex3fv(q[0]);
            glTexCoord2f(s1, t0); glVertex3fv(q[1]);
            glTexCoord2f(s1, t1); glVertex3fv(q[2]);
            glTexCoord2f(s0, t1); glVertex3fv(q[3]);
        }
    }
    glEnd();
    glDisable(GL_TEXTURE_2D);
}

```