

リスト 7.6 「slNoise3D1A.cpp」の一部

```

void setTexture3D(int n)
{
    #ifndef WIN32
    PFNGLTEXIMAGE3DPROC glTexImage3D =
        (PFNGLTEXIMAGE3DPROC)wglGetProcAddress("glTexImage3D");
    #endif

    glPixelStorei(GL_UNPACK_ALIGNMENT, 1);
    glBindTexture(GL_TEXTURE_3D, texName[n]);

    glTexImage3D(GL_TEXTURE_3D, 0, GL_LUMINANCE, NOISE_SIZE, NOISE_SIZE, NOISE_SIZE, 0, GL_LUMINANCE, GL_UNSIGNED_BYTE, texImage);
    glTexParameterf(GL_TEXTURE_3D, GL_TEXTURE_WRAP_S, GL_REPEAT);
    glTexParameterf(GL_TEXTURE_3D, GL_TEXTURE_WRAP_T, GL_REPEAT);
    glTexParameterf(GL_TEXTURE_3D, GL_TEXTURE_WRAP_R, GL_REPEAT);

    glTexParameterf(GL_TEXTURE_3D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
    glTexParameterf(GL_TEXTURE_3D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
    glBindTexture(GL_TEXTURE_3D, 0);
}

void makeTexImage3D(int texPattern)
{
    CNoise3 noise3;
    int i, j, k;

    for (k = 0; k < NOISE_SIZE; k++)
    {
        for (j = 0; j < NOISE_SIZE; j++)
        {
            for (i = 0; i < NOISE_SIZE; i++)
            {
                if (texPattern == PERLIN)
                    texImage[k][j][i] = (GLubyte)((1.0 + noise3.perlin(i, j, k, 5)) * 127.0);
                else // TURBULENCE
                    texImage[k][j][i] = (GLubyte)(noise3.turbulence(i, j, k, 5) * 255.0);
            }
        }
    }
}

void display(void)
{
    //時間計測
    static double time1, time2, drawTime, frame;
    if (ang <= 0.001) time1 = timeGetTime();

    //カラーバッファ, デプスバッファのクリア
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity(); //視点を変えるときはこの位置に必要
    if (cos(M_PI * view.theta / 180.0) >= 0.0) //カメラ仰角度でビューポートベクトル切替
        gluLookAt(view.pos[0], view.pos[1], view.pos[2], view.cnt[0], view.cnt[1], view.cnt[2], 0.0, 1.0, 0.0);
    else
        gluLookAt(view.pos[0], view.pos[1], view.pos[2], view.cnt[0], view.cnt[1], view.cnt[2], 0.0, -1.0, 0.0);

    //光源設定 // 'l' を押した後光源位置可変
    glLightfv(GL_LIGHT0, GL_POSITION, lightPos);

    if (flagWireframe) // 'w' で wireframe と solid model 切り替え
    {

```

```
    glPolygonMode(GL_FRONT, GL_LINE);
    glPolygonMode(GL_BACK, GL_POINT);
}
else glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);

glActiveTexture(GL_TEXTURE0);
glBindTexture(GL_TEXTURE_3D, texName[0]);
glActiveTexture(GL_TEXTURE1);
glBindTexture(GL_TEXTURE_3D, texName[1]);
// シェーダプログラムの適用
glUseProgram(shaderProg);
GLint scaleLoc = glGetUniformLocation(shaderProg, "scale");
glUniform1f(scaleLoc, scaleTex);
GLint coordLoc = glGetUniformLocation(shaderProg, "coord");
if(coord == OBJECT) glUniform1i(coordLoc, 0);
else glUniform1i(coordLoc, 1);
GLint texLoc = glGetUniformLocation(shaderProg, "smp13D");
glUniform1i(texLoc, 0); //GL_TEXTURE0を適用
draw0();
glUniform1i(texLoc, 1); //GL_TEXTURE1を適用
draw1();
// シェーダプログラムの適用を解除
glUseProgram(0);
drawFloor0(10.0, 10.0, 10, 10);
//影
drawShadow();
//中略
//終了
glutSwapBuffers();
}
```