

リスト 4.4	slCubeReflect2.cpp
<pre> void display(void) { //step1:環境マップの作成 //オブジェクト中心から見た6方向のシーンを作成し、テクスチャ・メモリに転送 //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); // 透視変換行列の設定 (texture作成) glMatrixMode(GL_PROJECTION); glLoadIdentity(); gluPerspective(90.0, 1.0, 0.1, 30.0); //視野角を90° // モデル・ビュー変換行列の設定 glMatrixMode(GL_MODELVIEW); glLoadIdentity(); // オブジェクトから見たシーンをテクスチャに利用 for (int i = 0; i < 6; ++i) { // ビューポートをテクスチャのサイズに設定する glViewport(target[i].x, target[i].y, TEX_WIDTH, TEX_HEIGHT); // オブジェクト中心から見えるものをレンダリング glPushMatrix(); gluLookAt(pos[0], pos[1], pos[2], pos[0]+target[i].cx, pos[1]+target[i].cy, pos[2]+target[i].cz, target[i].ux, target[i].uy, target[i].uz); glLightfv(GL_LIGHT0, GL_POSITION, lightPos); //3D空間の描画 drawRoom(10.0, 10.0, 5.0); glTranslatef(0.0, 1.0, 0.0); pet.draw(frameCount); glPopMatrix(); // レンダリングした画像をテクスチャ・メモリに転送 glCopyTexSubImage2D(target[i].name, 0, 0, 0, target[i].x, target[i].y, TEX_WIDTH, TEX_HEIGHT); } //通常の描画ルーチンに戻す //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); resize(width, height); // シェーダ・プログラムの適用 glUseProgram(shaderProg); //このコマンドの位置が重要 GLuint refLoc = glGetUniformLocation(shaderProg, "reflectivity"); glUniform1f(refLoc, reflectivity0); float m[16] ; glGetFloatv(GL_MODELVIEW_MATRIX, m); GLuint ViewTransLoc = glGetUniformLocation(shaderProg, "ViewTranspose"); glUniformMatrix4fv(ViewTransLoc, 1, GL_TRUE, m); //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); //fragment shaderのユニフォーム変数texのインデックスを取得 GLint samplerLoc = glGetUniformLocation(shaderProg, "sampler"); glUniform1i(samplerLoc, 0); //GL_TEXTURE0を適用 //描画 draw(); // シェーダ・プログラムの適用を解除 glUseProgram(0); </pre>	

```
drawRoom(10.0, 10.0, 5.0); //room2.hにある
glTranslatef(0.0, 1.0, 0.0);
pet.draw(frameCount);

curTime = timeGetTime();
//1コマの所要時間
float dt = (float)(curTime - lastTime) * 0.001; //sec
lastTime = curTime;
elapsedTime += dt;
fps++;
if(elapsedTime >= 1.0)
{
    printf("frame per sec = %d , dt = %f¥n", fps, dt);
    elapsedTime = 0.0;
    fps = 0;
}
frameCount++;
//中略
//終了
glutSwapBuffers();
}
```