

リスト 5.1 「glProjectionMap.cpp」の一部

```

void setTexture()
{
    glPixelStorei(GL_UNPACK_ALIGNMENT, 1);
    //テクスチャの指定

    glTexImage2D(GL_TEXTURE_2D,0,GL_RGBA,TEX_WIDTH,TEX_HEIGHT,0,GL_RGBA,GL_UNSIGNED_BYTE, texImage);
    //テクスチャの繰り返し方法の指定
    glTexParameterf(GL_TEXTURE_2D,GL_TEXTURE_WRAP_S, wrapMode);
    glTexParameterf(GL_TEXTURE_2D,GL_TEXTURE_WRAP_T, wrapMode);
    //テクスチャを拡大・縮小する方法の指定
    glTexParameterf(GL_TEXTURE_2D,GL_TEXTURE_MAG_FILTER,GL_LINEAR); //GL_NEAREST);
    glTexParameterf(GL_TEXTURE_2D,GL_TEXTURE_MIN_FILTER,GL_LINEAR); //GL_NEAREST);
    //テクスチャ生成
    glTexGenf(GL_S, GL_TEXTURE_GEN_MODE, genMode);
    glTexGenf(GL_T, GL_TEXTURE_GEN_MODE, genMode);
    glTexGenf(GL_R, GL_TEXTURE_GEN_MODE, genMode);
    glTexGenf(GL_Q, GL_TEXTURE_GEN_MODE, genMode);

    // テクスチャ生成関数の係数
    static const GLfloat param[][4] = {
        { 1.0, 0.0, 0.0, 0.0 },
        { 0.0, 1.0, 0.0, 0.0 },
        { 0.0, 0.0, 1.0, 0.0 },
        { 0.0, 0.0, 0.0, 1.0 },
    };
    if(genMode == GL_OBJECT_LINEAR)
    {
        glTexGenfv(GL_S, GL_OBJECT_PLANE, param[0]);
        glTexGenfv(GL_T, GL_OBJECT_PLANE, param[1]);
        glTexGenfv(GL_R, GL_OBJECT_PLANE, param[2]);
        glTexGenfv(GL_Q, GL_OBJECT_PLANE, param[3]);
    }
    else
    {
        glTexGenfv(GL_S, GL_EYE_PLANE, param[0]);
        glTexGenfv(GL_T, GL_EYE_PLANE, param[1]);
        glTexGenfv(GL_R, GL_EYE_PLANE, param[2]);
        glTexGenfv(GL_Q, GL_EYE_PLANE, param[3]);
    }
}

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

    //テクスチャ変換行列を設定する
    glMatrixMode(GL_TEXTURE);
    glLoadIdentity();
    //をテクスチャ座標の範囲[0,1]をオブジェクト座標の[-0.5, 0.5]に一致させる
    glTranslatef(0.5, 0.5, 0.0);
    //glRotatef(ang, 0.0, 0.0, 1.0);
    gluPerspective(fov, 1.0, 1.0, 100.0);
    if(projCenter == 0) gluLookAt(0.0, 5.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -1.0); //
    真上
    else if(projCenter == 1) gluLookAt(0.0, 0.0, 5.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0); //z
    軸上
    else if(projCenter == 2) gluLookAt(lightPos[0], lightPos[1], lightPos[2], 0.0,
    0.0, 0.0, 0.0, 0.0, -1.0);
    else gluLookAt(view.pos[0], view.pos[1], view.pos[2], 0.0, 0.0, 0.0, 0.0, 0.0,
    -1.0);
}

```

```

//通常の描画設定に戻す
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
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//'1'を押した後光源位置可変
glLightfv(GL_LIGHT0, GL_POSITION, lightPos);

glEnable(GL_TEXTURE_2D);
glEnable(GL_TEXTURE_GEN_S);
glEnable(GL_TEXTURE_GEN_T);
glEnable(GL_TEXTURE_GEN_R);
glEnable(GL_TEXTURE_GEN_Q);

//描画
draw0();
draw1();
drawFloor0(10.0, 10.0, 10, 10);

glDisable(GL_TEXTURE_GEN_S);
glDisable(GL_TEXTURE_GEN_T);
glDisable(GL_TEXTURE_GEN_R);
glDisable(GL_TEXTURE_GEN_Q);
glDisable(GL_TEXTURE_2D);

//影
drawShadow();

//回転角度更新
ang += dang;
if(ang >= 360.0)
{
    time2 = timeGetTime();
    drawTime = (time2 - time1) * 0.001;
    frame = (360.0 / dang) / drawTime;
    printf("フレーム数= %4.2f[fps] \n", frame);
    //フレームあたりの描画時間
    drawTime = 1.0 / frame;
    printf("描画時間= %4.5f[spf] \n", drawTime);
    ang = 0.0;
}
//中略
//終了
glutSwapBuffers();
}

```