

リスト 1.5	「glColorSpace.cpp」の一部
---------	-----------------------

```
#include <stdio.h>
#include <GL/glut.h>
#include <math.h>
#define M_PI 3.14159265358979

//関数のプロトタイプ宣言
void init();
void idle();
void display();
void draw();
void drawColorSpace();
void resize(int w, int h);
void keyboard(unsigned char key, int x, int y);
void special(int key, int x, int y);

//affine変換用変数
float pos[] = { 0.0, 0.0, 0.0 }; //図形中心のxyz座標
float scale[] = { 1.0, 1.0, 1.0 }; //大きさ(倍率)
float angle[] = { 0.0, -30.0, 0.0 }; //回転角度
//初期値
float pos0[] = { 0.0, 0.0, 0.0 };
float scale0[] = { 1.0, 1.0, 1.0 };
float angle0[] = { 0.0, -30.0, 0.0 };

//カメラと視体積
struct View
{
    //カメラ
    float pos[3]; //位置(視点)
    float cnt[3]; //注視点
    float dist; //注視点から視点までの距離
    float theta; //仰角(水平面との偏角)
    float phi; //方位角
    //視体積
    float fovY; //視野角
    float nearZ; //前方クリップ面(近平面)
    float farZ; //後方クリップ面(遠平面)
};
//初期値
View view = {
    0.0, 2.0, 4.0, //pos
    0.0, 0.0, 0.0, //cnt
    4.0, 30.0, 20.0, //dist, theta, phi
    30.0, 1.0, 100.0, //fovY, nearZ, farZ
};

//Windowのサイズ
int width = 500;
int height = 500;
//アフィン変換
enum SELECT_KEY {ROTATE, SCALE, TRANSLATE};
SELECT_KEY sKey = TRANSLATE;
//Helpキーフラグ
bool flagHelp = false;

int main(int argc, char** argv)
{
    glutInit(&argc, argv); //GLUTの初期設定
```

```

glutInitDisplayMode(GLUT_RGBA | GLUT_DOUBLE); //表示モード
glutInitWindowSize(width, height);           //表示ウィンドウのサイズ
glutInitWindowPosition(100, 100);            //左上の位置
glutCreateWindow("色空間(GL_ColorSpace)");   //ウィンドウ作成
glutReshapeFunc(resize);                      //ウィンドウのサイズ変更
glutDisplayFunc(display);                    //表示
glutKeyboardFunc(keyboard);                  //キーボードの利用
glutSpecialFunc(special);                    //矢印キーなどの特殊キー利用
glutIdleFunc(idle);                          //再描画
init();                                      //初期設定
glutMainLoop();                              //イベント処理ループに入る
return 0;
}

void idle(void)
{
    //再描画
    glutPostRedisplay();
}

void init(void)
{
    //背景色
    glClearColor(0.2, 0.2, 0.3, 1.0);
}

void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT); //カラーバッファのクリア
    draw();                       //描画
    glutSwapBuffers();           //終了
}

void draw(void)
{
    glPolygonMode(GL_FRONT, GL_FILL); //モード
    glPolygonMode(GL_BACK, GL_POINT);
    glPushMatrix();
    glTranslatef(pos[0], pos[1], pos[2]);
    glRotatef(angle[2], 0.0, 0.0, 1.0); //z軸回転
    glRotatef(angle[1], 0.0, 1.0, 0.0); //y軸回転
    glRotatef(angle[0], 1.0, 0.0, 0.0); //x軸回転
    glScalef(scale[0], scale[1], scale[2]);
    drawColorSpace(); //描画オブジェクト
    glPopMatrix();
}

void drawColorSpace()
{
    float p[8][3] =
    { {0.5, 0.5, 0.5}, {-0.5, 0.5, 0.5}, {-0.5, -0.5, 0.5}, {0.5, -0.5, 0.5},
      {0.5, 0.5, -0.5}, {-0.5, 0.5, -0.5}, {-0.5, -0.5, -0.5}, {0.5, -0.5, -0.5}
    };
    float c[8][3] = //各頂点の色
    { {1.0, 1.0, 1.0}, {1.0, 0.0, 1.0}, {1.0, 0.0, 0.0}, {1.0, 1.0, 0.0},
      {0.0, 1.0, 1.0}, {0.0, 0.0, 1.0}, {0.0, 0.0, 0.0}, {0.0, 1.0, 0.0}
    };

    glBegin(GL_QUADS);

```

```

    glColor3fv(c[0]); glVertex3fv(p[0]);
    glColor3fv(c[1]); glVertex3fv(p[1]);
    glColor3fv(c[2]); glVertex3fv(p[2]);
    glColor3fv(c[3]); glVertex3fv(p[3]);

    glColor3fv(c[0]); glVertex3fv(p[0]);
    glColor3fv(c[3]); glVertex3fv(p[3]);
    glColor3fv(c[7]); glVertex3fv(p[7]);
    glColor3fv(c[4]); glVertex3fv(p[4]);

    glColor3fv(c[0]); glVertex3fv(p[0]);
    glColor3fv(c[4]); glVertex3fv(p[4]);
    glColor3fv(c[5]); glVertex3fv(p[5]);
    glColor3fv(c[1]); glVertex3fv(p[1]);

    glColor3fv(c[1]); glVertex3fv(p[1]);
    glColor3fv(c[5]); glVertex3fv(p[5]);
    glColor3fv(c[6]); glVertex3fv(p[6]);
    glColor3fv(c[2]); glVertex3fv(p[2]);

    glColor3fv(c[2]); glVertex3fv(p[2]);
    glColor3fv(c[6]); glVertex3fv(p[6]);
    glColor3fv(c[7]); glVertex3fv(p[7]);
    glColor3fv(c[3]); glVertex3fv(p[3]);

    glColor3fv(c[4]); glVertex3fv(p[4]);
    glColor3fv(c[7]); glVertex3fv(p[7]);
    glColor3fv(c[6]); glVertex3fv(p[6]);
    glColor3fv(c[5]); glVertex3fv(p[5]);
    glEnd();
}

void resize(int w, int h)
{
    glViewport(0, 0, w, h); //ビューポート変換
    glMatrixMode(GL_PROJECTION); //投影行列の指定
    glLoadIdentity(); //投影行列の初期化
    //透視投影行列の設定 (投影変換)
    gluPerspective(view.fovY, (double)w/(double)h, view.nearZ, view.farZ);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity(); //モデルビュー行列の初期化
    //視点を設定 (視野変換)
    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);
    width = w; height = h; //表示ウィンドウのサイズ
}
//以下省略

```