リスト2.1 「glGouraud.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 resize(int w, int h);
void keyboard(unsigned char key, int x, int y);
void special(int key, int x, int y);
void setLight();
void setCamera();
void mouse(int button, int state, int x, int y);
void motion(int x, int y);

//affine変換用変数
float pos[] = { 0.0, 0.0, 0.0 };    //オブジェクト中心の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 };
//光源
float lightPos[] = {10.0, 15.0, 10.0, 1.0}; //光源位置
float lightPos0[] = {10.0, 15.0, 10.0, 1.0};//光源位置(初期値)

//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, 0.0, 0.0,//pos(仮設定)
  0.0, 0.0, 0.0,//cnt
  6.0, 0.0, 0.0,//dist, theta, phi
  30.0, 1.0, 100.0//fovY,nearZ, farZ
};
View view0 = view;

//Windowのサイズ
int width = 500;
int height = 500;
//アフィン変換
enum SELECT_KEY {ROTATE, SCALE, TRANSLATE, LIGHT};
SELECT_KEY sKey = TRANSLATE;
//マウス操作
int xStart, yStart;
bool flagMouse = false;
//ヘルプキーフラグ
```

```
bool flagHelp = false;
//光源位置変更フラグ
bool flagLight = false;//追加
//ﾜｲﾔｰﾌﾚｰﾑ<-->ｿﾘｯﾄﾞ切り替えフラグ
bool flagWireframe = false;//追加
//ﾌﾗｯﾄ<--->ｽﾑｰｽﾞｼｪｰﾃﾞｨﾝｸﾞ切り替えﾌﾗｸﾞ
bool flagFlat = false;

int main(int argc, char** argv)
{
  glutInit(&argc, argv);
  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("ｸﾞｰﾛｰｼｪｰﾃﾞｨﾝｸﾞ(GL_Gouraud)"); //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

  //マウス操作
  glutMouseFunc(mouse);
  glutMotionFunc(motion);
  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);
  setCamera(); //視点を求める
  setLight();  //光源設定
  glEnable(GL_DEPTH_TEST);
  glEnable(GL_NORMALIZE);
  glShadeModel(GL_SMOOTH);
  printf("マウス/キー操作の説明には'h'キーをプッシュ¥n");
}

void display(void)
{
  //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)//カメラ仰角度でﾋﾞｭｰｱｯﾌﾟﾍﾞｸﾄﾙ切替
    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);
  //'f'でflatとsmooth shading切り替え
  if(flagFlat) glShadeModel(GL_FLAT);
  else glShadeModel(GL_SMOOTH);

  //描画
  draw();
  if(flagHelp)
  {
//省略
  }
  //終了
  glutSwapBuffers();
}

void draw(void)
{
  float ambient[] = { 0.1, 0.3, 0.3, 1.0};
  float diffuse[] = { 0.2, 0.7, 0.7, 1.0};
  float specular[]= { 0.5, 0.5, 0.5, 1.0};

  glMaterialfv(GL_FRONT,GL_AMBIENT,ambient);
  glMaterialfv(GL_FRONT,GL_DIFFUSE,diffuse);
  glMaterialfv(GL_FRONT,GL_SPECULAR,specular);
  glMaterialf(GL_FRONT,GL_SHININESS,100);

  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]);
  //オブジェクト形状
  glutSolidSphere(1.0, 20, 20);
  glPopMatrix();
}

void resize(int w, int h)
{
  glViewport(0, 0, w, h);     //ビューポート変換
  glMatrixMode(GL_PROJECTION); //プロジェクション行列の指定
  glLoadIdentity();           //プロジェクション行列の初期化
  //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);  //モデル・ビュー行列の宣言
  //表示ウィンドウのサイズ
  width = w;
  height = h;
}

void setLight()
{
  float lightAmbient0[] = {0.5, 0.5, 0.5, 1.0}; //環境光
  float lightDiffuse0[] = {1.0, 1.0, 1.0, 1.0}; //拡散光
  float lightSpecular0[] = {1.0, 1.0, 1.0, 1.0};//鏡面光
  glLightfv(GL_LIGHT0, GL_AMBIENT, lightAmbient0);
  glLightfv(GL_LIGHT0, GL_DIFFUSE, lightDiffuse0);
  glLightfv(GL_LIGHT0, GL_SPECULAR, lightSpecular0);
  glLightfv(GL_LIGHT0, GL_POSITION, lightPos);

  glEnable(GL_LIGHT0);
  glEnable(GL_LIGHTING);
}
```

```
//以後省略
```