

セッソブルーノ

201555648

## 課題2

```
#include <math.h>
#include <GLUT/glut.h> // ライブラリ用ヘッダファイルの読み込み

// 定数 $\pi$ の定義
#ifndef M_PI
#define M_PI 3.14159265358979
#endif

// ディスプレイリストの学習
// 星を描画する描画命令一式を、ディスプレイリストとして作成しておく
// 必要な時に、その命令を呼び出す

#define ID_DRAW_STAR 1 // glGenLists 関数で使用する識別ID。値は何でも構わない

int rotateAngle; // 回転角度を記録しておく変数

// 表示部分をこの関数で記入
void display(void) {
    glClearColor (1.0, 1.0, 1.0, 1.0); // 消去色指定
    glClear(GL_COLOR_BUFFER_BIT);      // 画面と奥行き情報を初期化

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();

    glPushMatrix();
    glColor3d(1.0, 0.0, 0.0);
    glRotated(rotateAngle, 0, 0, 1);
    glTranslated(0.5, 0, 0);
    glRotated(rotateAngle, 0, 0, 1);
    glCallList(ID_DRAW_STAR);
    glPopMatrix();

    glPushMatrix();
    glColor3d(0.0, 1.0, 0.0);
    glRotated(rotateAngle, 0, 0, 1);
    glTranslated(0, 0.5, 0);
    glRotated(rotateAngle, 0, 0, 1);
    glCallList(ID_DRAW_STAR);
    glPopMatrix();

    glPushMatrix();
    glColor3d(0.0, 0.0, 1.0);
    glRotated(rotateAngle, 0, 0, 1);
    glTranslated(-0.5, 0, 0);
    glRotated(rotateAngle, 0, 0, 1);
    glCallList(ID_DRAW_STAR);
    glPopMatrix();

    glutSwapBuffers(); // バッファの入れ替え
}

// 一定時間ごとに呼び出される関数
void timer(int value) {
```

```

    if (rotateAngle == 360)
        rotateAngle = 0;
    rotateAngle++; // 回転角度の更新

    glutPostRedisplay(); // 再描画命令
    glutTimerFunc(10, timer , 0); // 100ミリ秒後に自身を実行する
}

// ディスプレイリストを作成する
void buildDisplayList() {
    glNewList(ID_DRAW_STAR, GL_COMPILE);

    double r0 = 0.15; // 星の内径
    double r1 = 0.4; // 星の外径
    glBegin(GL_TRIANGLES);
    for(int i = 0; i < 5; i++) { // 5つの三角形で星を表現する
        int deg = i * 72;
        glVertex3d(r0 * cos( (deg - 36) * M_PI / 180.0), r0 * sin( (deg -
36) * M_PI / 180.0), 0); // 内側の頂点
        glVertex3d(r1 * cos( deg * M_PI / 180.0), r1 * sin( deg * M_PI /
180.0), 0); // 外側の頂点
        glVertex3d(r0 * cos( (deg + 36) * M_PI / 180.0), r0 * sin( (deg +
36) * M_PI / 180.0), 0); // 内側の頂点
    }
    glEnd();

    glEndList();
}

// メインプログラム
int main (int argc, char *argv[]) {
    glutInit(&argc, argv); // ライブラリの初期化
    glutInitDisplayMode(GLUT_RGBA|GLUT_DOUBLE);

    glutInitWindowSize(400 , 400); // ウィンドウサイズを指定
    glutCreateWindow(argv[0]); // ウィンドウを作成
    glutDisplayFunc(display); // 表示関数を指定

    glutTimerFunc(100 , timer , 0); // 100ミリ秒後に実行する関数の指定

    buildDisplayList();

    rotateAngle = 0; // 変数の初期値の設定

    glutMainLoop(); // イベント待ち
    return 0;
}

```

## 課題3

```
#include <math.h>
#include <GLUT/glut.h> // ライブラリ用ヘッダファイルの読み込み

// 定数πの定義
#ifndef M_PI
#define M_PI 3.14159265358979
#endif

// ディスプレイリストの学習
// 星を描画する描画命令一式を、ディスプレイリストとして作成しておく
// 必要な時に、その命令を呼び出す

#define ID_DRAW_STAR 1 // glGenLists 関数で使用する識別ID。値は何でも構わない

int rotateAngle; // 回転角度を記録しておく変数

// 表示部分をこの関数で記入
void display(void) {
    glClearColor (1.0, 1.0, 1.0, 1.0); // 消去色指定
    glClear(GL_COLOR_BUFFER_BIT);      // 画面と奥行き情報を初期化

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();

    glPushMatrix();
    glRotated(rotateAngle, 0, 0, 1);
    glTranslated(0.5, 0, 0);
    glRotated(rotateAngle, 0, 0, 1);

    glPushMatrix();
    glColor3d(0, 0, 0);
    glRotated(rotateAngle * 3, 0, 0, 1);
    glTranslated(0.5, 0, 0);
    glScaled(0.3, 0.3, 0);
    glRotated(rotateAngle * 10, 0, 0, 1);
    glCallList(ID_DRAW_STAR);
    glPopMatrix();

    glColor3d(1.0, 0.0, 0.0);
    glCallList(ID_DRAW_STAR);
    glPopMatrix();

    glPushMatrix();
    glColor3d(0.0, 1.0, 0.0);
    glRotated(rotateAngle, 0, 0, 1);
    glTranslated(0, 0.5, 0);
    glRotated(rotateAngle, 0, 0, 1);
    glCallList(ID_DRAW_STAR);
    glPopMatrix();

    glPushMatrix();
    glColor3d(0.0, 0.0, 1.0);
    glRotated(rotateAngle, 0, 0, 1);
    glTranslated(-0.5, 0, 0);
    glRotated(rotateAngle, 0, 0, 1);
    glCallList(ID_DRAW_STAR);
    glPopMatrix();

    glutSwapBuffers(); // バッファの入れ替え
```

```

}

// 一定時間ごとに呼び出される関数
void timer(int value) {
    if (rotateAngle == 360)
        rotateAngle = 0;
    rotateAngle++; // 回転角度の更新

    glutPostRedisplay(); // 再描画命令
    glutTimerFunc(10 , timer , 0); // 100ミリ秒後に自身を実行する
}

// ディスプレイリストを作成する
void buildDisplayList() {
    glNewList(ID_DRAW_STAR, GL_COMPILE);

    double r0 = 0.15; // 星の内径
    double r1 = 0.4; // 星の外径
    glBegin(GL_TRIANGLES);
    for(int i = 0; i < 5; i++) { // 5つの三角形で星を表現する
        int deg = i * 72;
        glVertex3d(r0 * cos( (deg - 36) * M_PI / 180.0), r0 * sin( (deg -
36) * M_PI / 180.0), 0); // 内側の頂点
        glVertex3d(r1 * cos( deg * M_PI / 180.0), r1 * sin( deg * M_PI /
180.0), 0); // 外側の頂点
        glVertex3d(r0 * cos( (deg + 36) * M_PI / 180.0), r0 * sin( (deg +
36) * M_PI / 180.0), 0); // 内側の頂点
    }
    glEnd();

    glEndList();
}

// メインプログラム
int main (int argc, char *argv[]) {
    glutInit(&argc, argv); // ライブラリの初期化
    glutInitDisplayMode(GLUT_RGBA|GLUT_DOUBLE);

    glutInitWindowSize(400 , 400); // ウィンドウサイズを指定
    glutCreateWindow(argv[0]); // ウィンドウを作成
    glutDisplayFunc(display); // 表示関数を指定

    glutTimerFunc(100 , timer , 0); // 100ミリ秒後に実行する関数の指定

    buildDisplayList();

    rotateAngle = 0; // 変数の初期値の設定

    glutMainLoop(); // イベント待ち
    return 0;
}

```