

第8回 宿題

「第8回」の宿題の雛形プログラムの `cgsample08.c` の最後の部分にある、実際に図形描画を行う関数 `draw()` は次のようになっています。

```
/*
** 図形の描画
*/
void draw(int width, int height)
{
    double rotation[4][4];           /* 物体の回転の変換行列 */
    double spin[4][4];               /* スピン用の変換行列 */
    double model[4][4];              /* モデル変換行列 */
    double view[4][4];               /* ビュー変換行列 */
    double modelview[4][4];          /* モデルビュー変換行列 */
    double perspective[4][4];        /* 透視投影変換行列 */
    double temporary[4][4];          /* 中間結果を保存する */
    double scale[4][4];              /* 拡大縮小の変換行列 */
    double translation[4][4];        /* 平行移動の変換行列 */
    double viewport[4][4];           /* ビューポート変換行列 */
    double matrix[4][4];             /* 合成した変換行列 */
    static int frame = 0;             /* フレーム数 */

    /*
    ** ビューポート変換行列を求める
    */

    /* scale にウィンドウサイズにあわせて図形を拡大縮小する変換行列を求める */
    setScale(scale, width * 0.08, height * 0.08, 1.0);
    /* translation に図形がウィンドウに収まるように平行移動する変換行列を求める */
    setTranslation(translation, width * 0.5, height * 0.5, 0.0);
    /* 拡大縮小の変換行列に平行移動の変換行列を掛けてビューポート変換行列を求める */
    multiply(viewport, translation, scale);

    /*
    ** ビュー変換行列を求める
    */
    lookAt(view, 0.0, 3.0, 7.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);

    /*
    ** 透視投影変換行列を求める
    */
    setPerspective(perspective, 30.0, (double)width / (double)height, 1.0, 10.0);

    /*
    ** 回転アニメーション用の変換行列を求める
    */
    setRotationY(spin, (double)frame);           /* フレーム数=回転角 */

    /*
    ** 一つ目の図形のモデル変換行列を求める
    */
    setRotationZ(rotation, 45.0);               /* z 軸中心に 45° 回転 */

    /*
    ** 一つ目の図形の座標変換と描画
    */
}
```

```

multiply(model, spin, rotation);          /* 回転×回転 */
multiply(modelview, view, model);        /* ×視野 */
multiply(temporary, perspective, modelview); /* ×透視 */
multiply(matrix, viewport, temporary);   /* ×ビューポート */
projection(screen, matrix, vertex, NVERTEX); /* 投影 */
wireframe(screen, edge, NEDGE);          /* 描画 */

/*
** 二つ目の図形のモデリング変換行列を求める
*/
setRotationZ(rotation, -135.0);          /* z 軸中心に -135° 回転 */

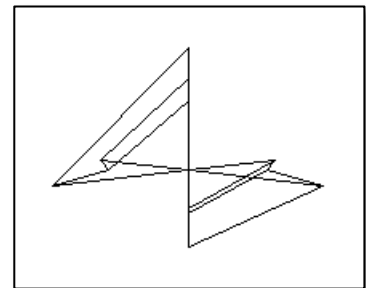
/*
** 二つ目の図形の座標変換と描画
*/
multiply(model, spin, rotation);          /* 回転×回転 */
multiply(modelview, view, model);        /* ×視野 */
multiply(temporary, perspective, modelview); /* ×透視 */
multiply(matrix, viewport, temporary);   /* ×ビューポート */
projection(screen, matrix, vertex, NVERTEX); /* 投影 */
wireframe(screen, edge, NEDGE);          /* 描画 */

/*
** フレーム数のカウント
*/
if ((frame += 5) >= 360) frame = 0;
}

```

このプログラムを完成させるには、透視変換行列を求める `setPerspective()` を実装する必要があります。なお、「第7回」の宿題の `setScale()`, `setTranslation()`, `setRotation[XYZ]()`, および変換行列による投影を行う `projection()`, 任意軸中心の回転変換行列を求める `setRotation()`, 視点や目標の位置を設定する `lookAt()`, そして線分を描く `line()` は既に組み込んであります。

- (1) 関数 `setPerspective()` の中身を実装して、このプログラムを実行したときに右図のような図形が表示されるようにしてください。なお、このプログラムはキー(ESC, q, Q を除く)をタイプすると図形が 5° ずつ回転します。



これが表示できたら、配列変数 `vertex` と `edge` に「第6回」の宿題において自分で作成した図形のデータを設定して、その図形を 45° 回転したものと 135° 回転したものが二つ表示されるようにしてください(2)(3)。このとき、配列変数 `vertex` には座標値を同次座標で設定する必要があります。また、図形が画面にうまく収まるように、ビューポート変換行列の設定を調整してください。

プログラムが期待通り動作したら、作成したソースファイル (`cgsample08.c`) を授業のホームページのアップローダからアップロードしてください。期限は12月8日(木)中です。