

変更箇所はdisplay関数のみである。

```
void display(void)
{
    float green[] = {0., 1., 0., 1.};
    float blue[] = {0., 0., 1., 1.};

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); // 画面とzバッファをクリア

    glLoadIdentity();
    glRotatef( xrot,1.,0.,0. );
    glRotatef( yrot,0.,1.,0. );
    glScalef( 5.,5.,5. );

    if( axis==1 ) { // 座標軸を描く場合
        glDisable(GL_LIGHT0); // 光源を消す
        glCallList( AXIS ); // 線画を描く
        glEnable(GL_LIGHT0); // 光源を灯す
    }

    glTranslatef( 0.,0.,-1. );

    glMaterialfv( GL_FRONT, GL_DIFFUSE, blue );// これから描く物体は青色
    glCallList( SQUARE ); // 基本四角形表示

    glMaterialfv( GL_FRONT, GL_DIFFUSE, green );// これからは緑色
    glPushMatrix();
    glTranslatef( -1.,0.,0. ); // 追加
    glRotatef( rotangle,0.,1.,0. ); // 追加
    glTranslatef( 1.,0.,0. ); // 追加
    glTranslatef( -2.,0.,0. );
    glCallList( SQUARE );// 左側正方形表示
    glPopMatrix();

    glPushMatrix();
    glTranslatef( 0.,1.,0. ); // 追加
    glRotatef( rotangle,1.,0.,0. ); // 追加
    glTranslatef( 0.,-1.,0. ); // 追加
    glTranslatef( 0.,2.,0. );
    glCallList( SQUARE );// 上側正方形表示
    glPopMatrix();

    glPushMatrix();
    glTranslatef( 0.,-1.,0. ); // 追加
    glRotatef( -rotangle,1.,0.,0. ); // 追加
    glTranslatef( 0.,1.,0. ); // 追加
    glTranslatef( 0.,-2.,0. );
    glCallList( SQUARE );// 下側正方形表示
    glPopMatrix();

    glTranslatef( 1.,0.,0. ); // 追加
    glRotatef( -rotangle,0.,1.,0. ); // 追加
    glTranslatef( -1.,0.,0. ); // 追加
    glPushMatrix();
    glTranslatef( 2.,0.,0. );
    glCallList( SQUARE );// 右側正方形表示
    glPopMatrix();
}
```

```
glMaterialfv( GL_FRONT, GL_DIFFUSE, blue );// 青色
glTranslatef( 3.,0.,0. ); // 追加
glRotatef( -rotangle,0.,1.,0. ); // 追加
glTranslatef( -3.,0.,0. ); // 追加
glTranslatef( 4.,0.,0. );
glCallList( SQUARE ); // 最右側正方形表示

glutSwapBuffers(); // ダブルバッファを切り替える
}
```

感想など

glPushMatrix, glPopMatrixの動作をようやく理解したので、立方体の折りたたみは簡単に作ることができた。しかし簡単というのは本当は嘘で、「最右側正方形」の回転を理解していない。

まず「右側正方形」と同様に、左側の辺を回転軸に重ねて回転させれば、

```
glTranslatef( 1.,0.,0. ); // 追加
glRotatef( -rotangle,0.,1.,0. ); // 追加
glTranslatef( -1.,0.,0. ); // 追加
glTranslatef( 4.,0.,0. );
```

「右側正方形」と一緒に1枚の板のように回転するところまでは分かる。

折りたたまれるようにするためには、glTranslatef(4.,0.,0.); をかける前に、回転軸から正方形1枚分の距離を離して回転させる行列をかける。

```
glTranslatef( 3.,0.,0. ); // 追加
glRotatef( -rotangle,0.,1.,0. ); // 追加
glTranslatef( -3.,0.,0. ); // 追加
```

「最右側正方形」の左側の辺は、回転軸から離れているから、回転させたら、「右側正方形」の右側の辺から離れていってしまうのではないかと思ったが、実行してみると離れないで動いてくれる。回転を理解できていない。

