２００７年１２月　７日

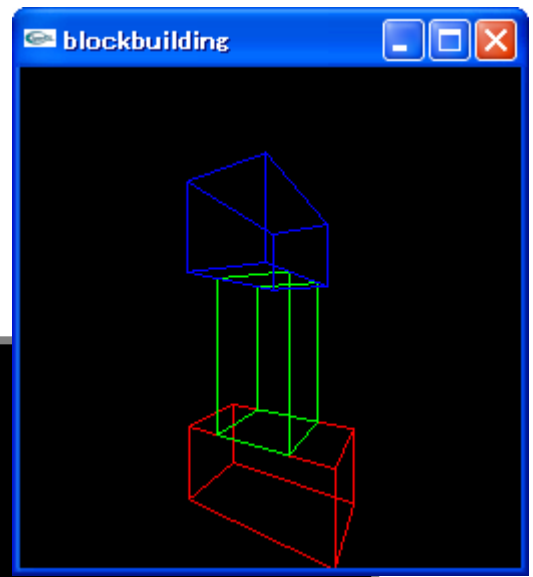
第21回　OpenGL によるグラフィックス（10）

情報処理論（応用）

松山大学 経営学部
檀　裕也
http://www.cc.matsuyama-u.ac.jp/~dan/education/application/

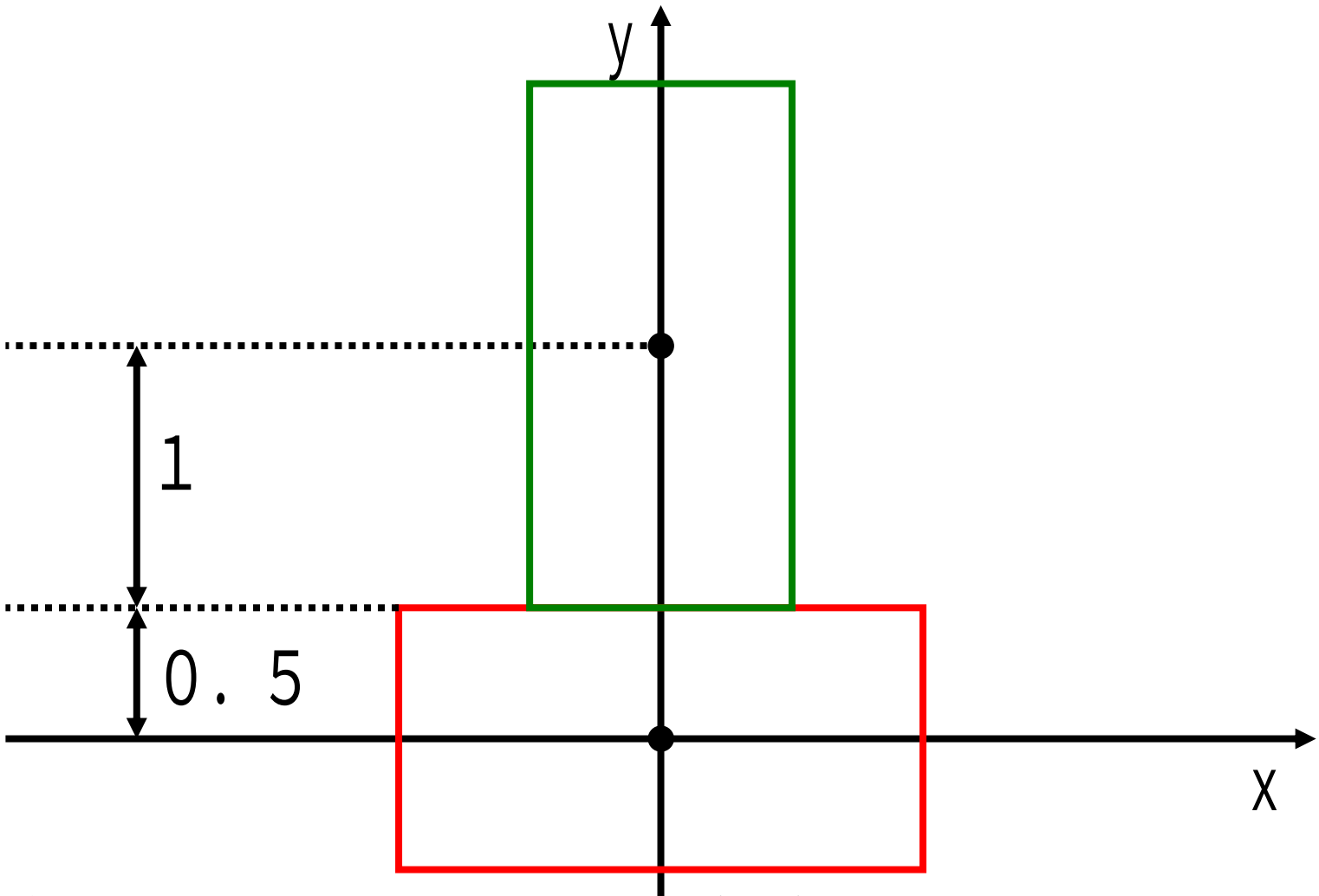
前回の実習課題

- 大きさ１×１×２のブロックを３つ以上使い、積み木遊びをせよ。
  - 宛先：dan@cc.matsuyama-u.ac.jp
  - 件名：CG課題#09

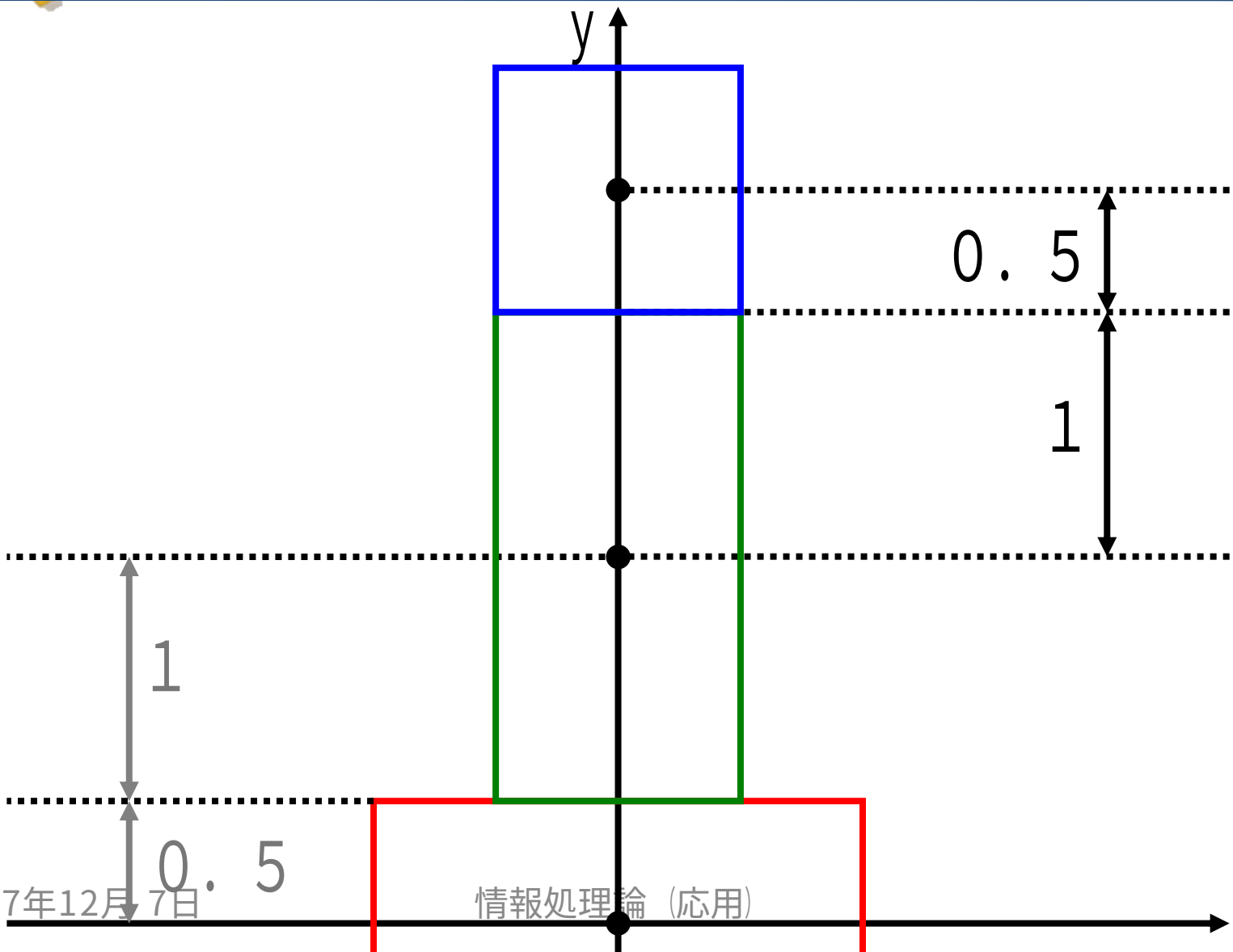



```
C:\>blockbuilding.exe
```

考えかた（緑ブロック）



考えかた　(青ブロック)



解答例（1/2）

blockbuilding.c

```
void display( void )
{
        glClear( GL_COLOR_BUFFER_BIT );
        glLoadIdentity();
        gluLookAt( z * sin( theta ), 0.0, z * cos( theta ),
                   0.0, 0.0, 0.0, 0.0, 1.0, 0.0 );

        glTranslatef( 0.0, -2.0, 0.0 );

        glColor3f( 1.0, 0.0, 0.0 );
        glScalef( 2.0, 1.0, 1.0 );
        glutWireCube( 1.0 );

        glTranslatef( 0.0, 0.5 + 1.0, 0.0 );
        glColor3f( 0.0, 1.0, 0.0 );
        glScalef( 0.5, 2.0, 1.0 );
```

解答例（2/2）

```
        glutWireCube( 1.0 );

        glTranslatef( 0.0, ( 1.0 + 0.5 ) / 2, 0.0 );
        glColor3f( 0.0, 0.0, 1.0 );
        glScalef( 1.0, 0.5, 2.0 );
        glutWireCube( 1.0 );

        glFlush();
}
```

今回の授業内容

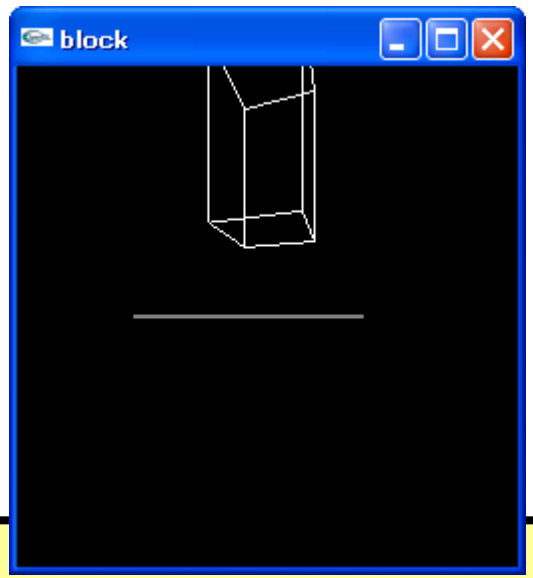
- OpenGL によるグラフィックス（10）
  - 座標系の保存と復元

行列スタック

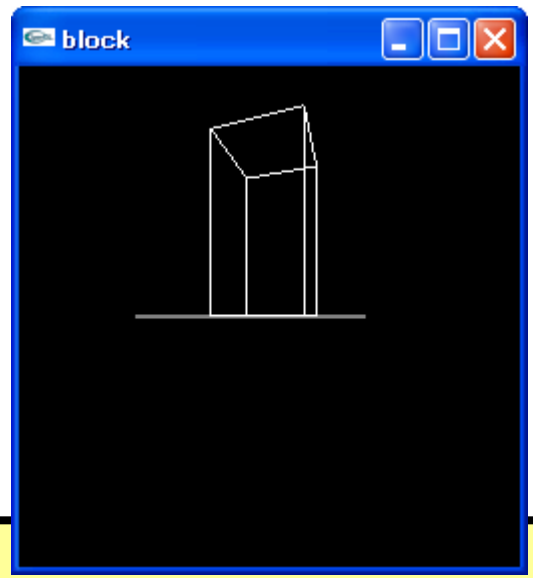
- 座標変換の積み重ね
  - 拡大･縮小　glScalef
  - 平行移動　glTranslatef
  - 回転　glRotatef

座標変換の順序

- 順序が変わると結果が変わる




```
glTranslatef( 0.0, 1.0, 0.0 );
glScalef( 1.0, 2.0, 1.0 );
glutWireCube( 1.0 );
```




```
glScalef( 1.0, 2.0, 1.0 );
glTranslatef( 0.0, 1.0, 0.0 );
glutWireCube( 1.0 );
```

行列スタック

- 座標変換の結果を glPushMatrix で保存し、glPopMatrix で復元する

```
glPushMatrix();
        glScalef( 1.0, 2.0, 1.0 );
        glutWireCube( 1.0 );
glPopMatrix();
```

- 復元した座標系では、その間の座標変換の影響を受けない

変更箇所（1/2）

blockbuilding2.c

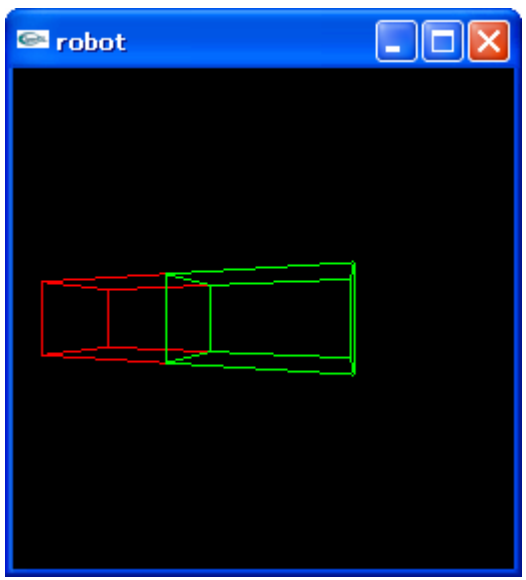
```
void display( void )
{
        glClear( GL_COLOR_BUFFER_BIT );
        glLoadIdentity();
        gluLookAt( z * sin( theta ), 0.0, z * cos( theta ),
                   0.0, 0.0, 0.0, 0.0, 1.0, 0.0 );

        glTranslatef( 0.0, -2.0, 0.0 );

        glColor3f( 1.0, 0.0, 0.0 );
        glPushMatrix();
                glScalef( 2.0, 1.0, 1.0 );
                glutWireCube( 1.0 );
        glPopMatrix();

        glTranslatef( 0.0, 0.5 + 1.0, 0.0 );
```

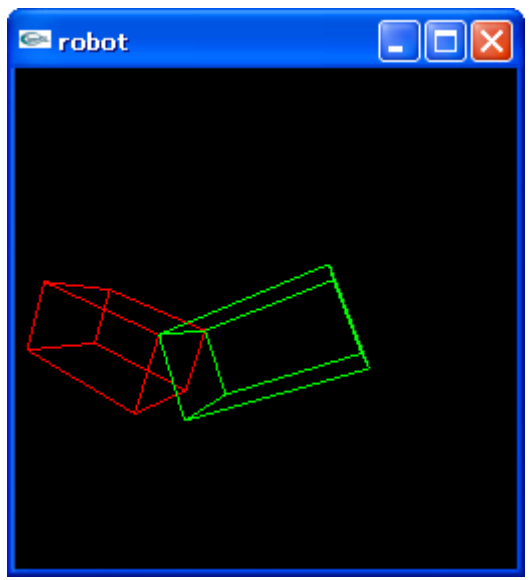
変更箇所（2/2）

```
        glColor3f( 0.0, 1.0, 0.0 );
        glPushMatrix();
                glScalef( 1.0, 2.0, 1.0 );
                glutWireCube( 1.0 );
        glPopMatrix();

        glTranslatef( 0.0, 1.0 + 0.5, 0.0 );
        glColor3f( 0.0, 0.0, 1.0 );
        glPushMatrix();
                glScalef( 1.0, 1.0, 2.0 );
                glutWireCube( 1.0 );
        glPopMatrix();

        glFlush();
}
```

例題

- ロボット・アーム
  - 's' または 'S' で肩を動かす
  - 'e' または 'E' で肘を動かす





ソースコード（1/6）

robot.c

```
#include <stdlib.h>
#include <GL/glut.h>
#include <GL/gl.h>
#include <GL/glu.h>
#include <math.h>

double z = 3.0;
double theta = 0.3;
double shoulder = 0;
double elbow = 0;

void display( void )
{
        glClear( GL_COLOR_BUFFER_BIT );
        glLoadIdentity();
        gluLookAt( z * sin( theta ), 0.0, z * cos( theta ),
                   0.0, 0.0, 0.0, 0.0, 1.0, 0.0 );
```

ソースコード（2/6）

```
glTranslatef( -2.0, 0.0, 0.0 );

glColor3f( 1.0, 0.0, 0.0 );
glTranslatef( -1.0, 0.5, 0.0 );
glRotatef( shoulder, 0.0, 0.0, 1.0 );
glTranslatef( 1.0, -0.5, 0.0 );
glPushMatrix();
        glScalef( 2.0, 1.0, 1.0 );
        glutWireCube( 1.0 );
glPopMatrix();

glTranslatef( 2.0, 0.0, 0.0 );

glTranslatef( -1.0, 0.5, 0.0 );
glRotatef( elbow, 0.0, 0.0, 1.0 );
glTranslatef( 1.0, -0.5, 0.0 );
glColor3f( 0.0, 1.0, 0.0 );
glPushMatrix();
```

ソースコード（3/6）

```
                glScalef( 2.0, 1.0, 1.0 );
                glutWireCube( 1.0 );
        glPopMatrix();

        glFlush();
}

void myKeyBoard( unsigned char key, int x, int y )
{
        switch( key ){
        case 0x1B:
                exit( 0 );
        case 'z':
                z -= 0.2;
                break;
        case 'Z':
                z += 0.2;
                break;
```

ソースコード（4/6）

```
case 'r':
        theta -= 0.1;
        break;
case 'R':
        theta += 0.1;
        break;
case 's':
        shoulder -= 5.0;
        break;
case 'S':
        shoulder += 5.0;
        break;
case 'e':
        if( elbow >= 5.0 ){
                elbow -= 5.0;
        }
        break;
case 'E':
```

ソースコード（5/6）

```
                if( elbow <= 180.0 - 5.0 ){
                        elbow += 5.0;
                }
                break;
        }
        glutPostRedisplay();
}

int main( int argc, char *argv[] )
{
        glutInit( &argc, argv );
        glutInitDisplayMode( GLUT_RGB );
        glutInitWindowSize( 250, 250 );
        glutCreateWindow( argv[0] );

        glClearColor( 0.0, 0.0, 0.0, 0.0 );
        glMatrixMode( GL_PROJECTION );
        glLoadIdentity();
```

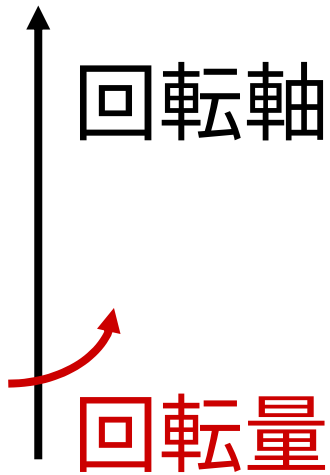
ソースコード（6/6）

```
        glFrustum( -1.0, 1.0, -1.0, 1.0, 1.0, 20.0 );
        glMatrixMode( GL_MODELVIEW );

        glutDisplayFunc( display );
        glutKeyboardFunc( myKeyBoard );
        glutMainLoop();

        return 0;
}
```

glRotatef 関数

- 立体図形を回転させる
  - *theta*　回転量　(度数法)
  - *rx*　　回転軸ベクトルのｘ座標
  - *ry*　　回転軸ベクトルのｙ座標
  - *rz*　　回転軸ベクトルのｚ座標

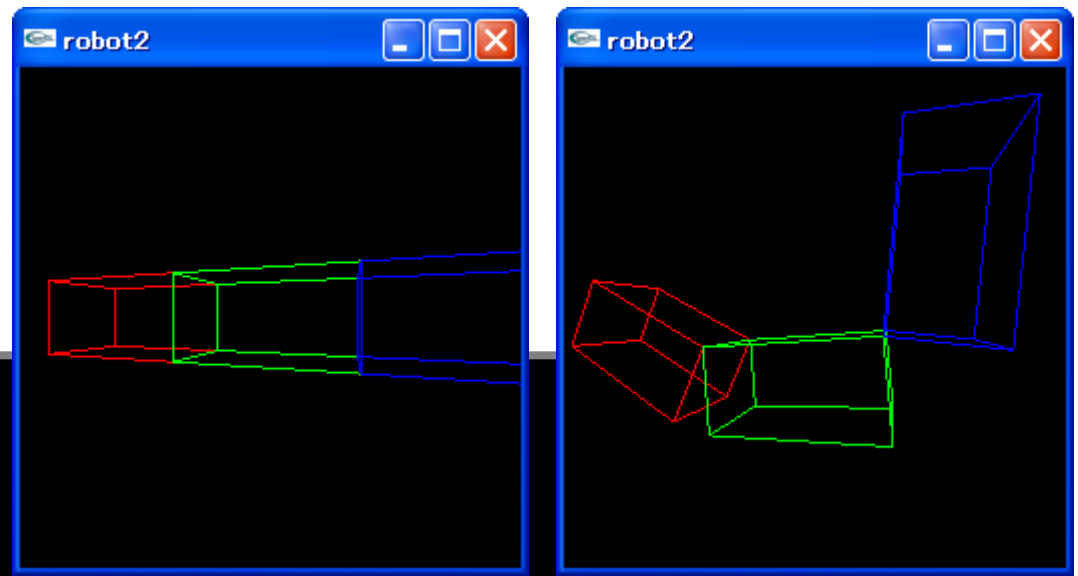
  ※立体図形の描画前に設定する




```
glRotatef( theta, rx, ry, rz );
```

実習課題

- 例題のプログラムにブロックを１つ追加せよ。
  - 宛先：dan@cc.matsuyama-u.ac.jp
  - 件名：CG課題#10




```
C:\>robot2.exe
```

次回の予定

- 日時:　12月12日（水）
  5時限
- OpenGL によるグラフィックス
  - モデリング　(続き)