



2007年12月5日

## 第20回 OpenGL によるグラフィックス (9)

# 情報処理論 (応用)

松山大学 経営学部

檀 裕也

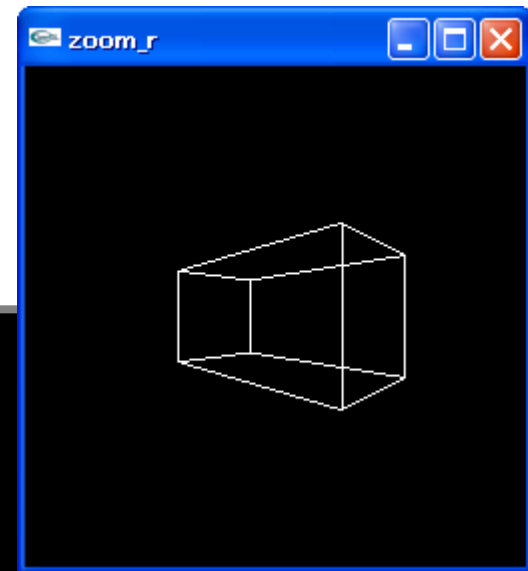
<http://www.cc.matsuyama-u.ac.jp/~dan/education/application/>



# 前回の実習課題

- 例題のプログラムに 'r' または 'R' のキーを押すと視点の位置を回転させる機能を追加せよ
  - 宛先: dan@cc.matsuyama-u.ac.jp
  - 件名: CG課題#08

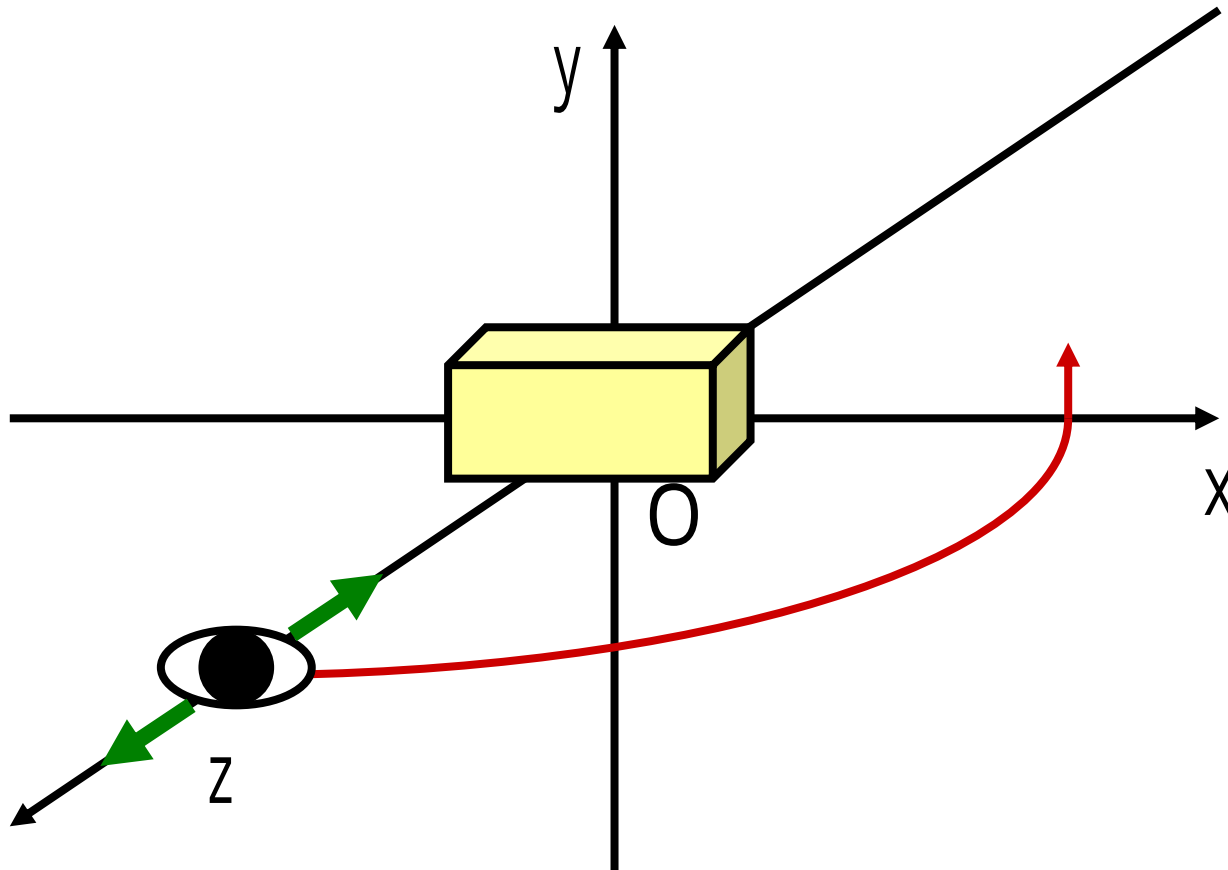
```
C:¥>zoom_r.exe
```





# 考えかた

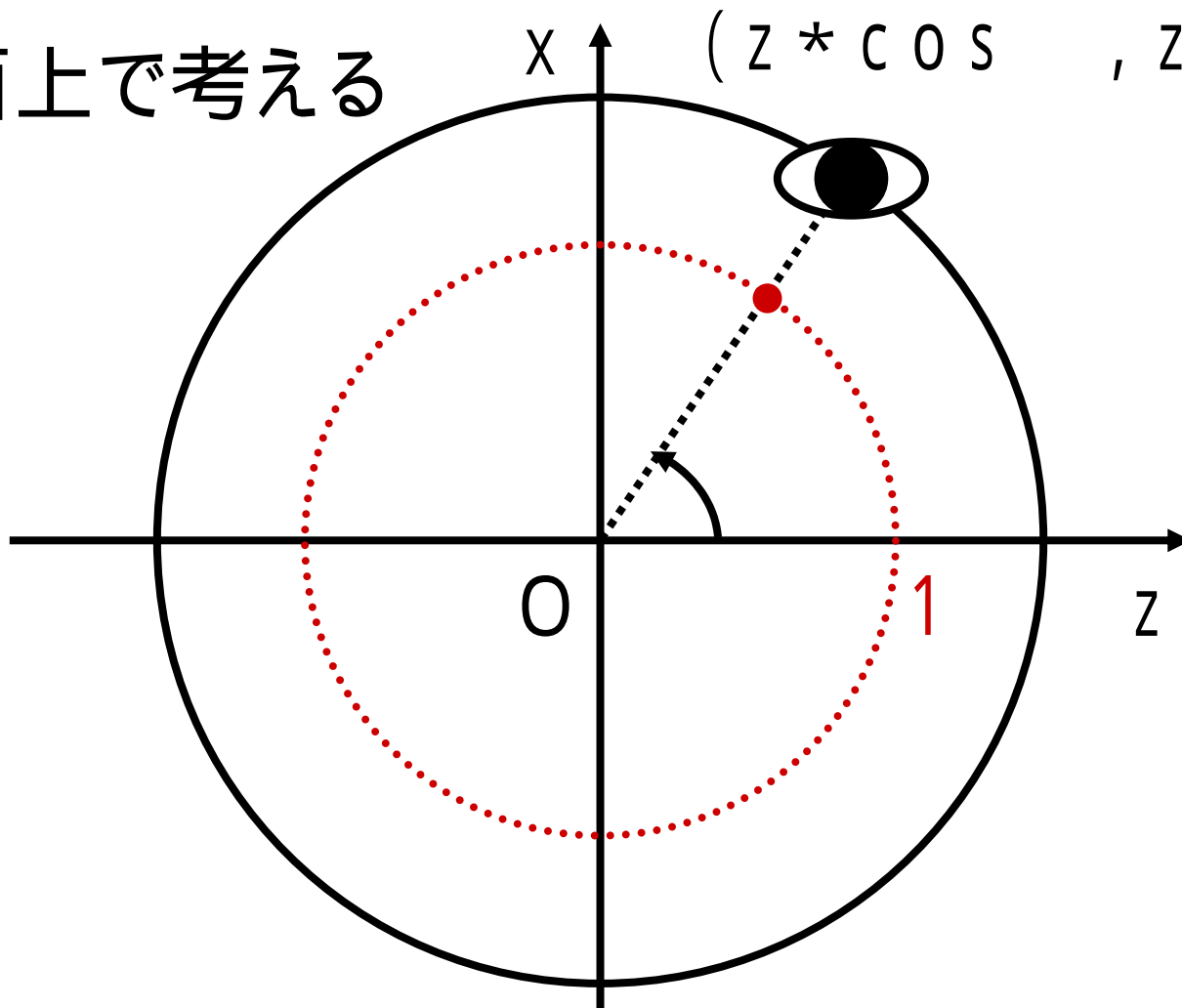
- $y$  軸を中心に回転する ( $xz$  平面での回転)





# 三角関数

- 平面上で考える  $(z * \cos, z * \sin)$





# 解答例 ( 1 / 4 )

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

double z = 5.0;
double theta = 0.0;

void display( void )
{
    glClear( GL_COLOR_BUFFER_BIT );
    glColor3f( 1.0, 1.0, 1.0 );
    glLoadIdentity();
    gluLookAt( z * sin( theta ), 0.0, z * cos( theta ),
              0.0, 0.0, 0.0, 0.0, 1.0, 0.0 );
}
```



# 解答例 (2 / 4)

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



# 解答例 (3 / 4)

```
case 'r':
    theta -= 0.1;
    break;
case 'R':
    theta += 0.1;
    break;
}
glutPostRedisplay();
}

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



# 解答例 (4 / 4)

```
glClearColor( 0.0, 0.0, 0.0, 0.0 );
glMatrixMode( GL_PROJECTION );
glLoadIdentity();
glFrustum( -1.0, 1.0, -1.0, 1.0, 1.0, 20.0 );
glMatrixMode( GL_MODELVIEW );

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

return 0;
}
```





# 今回の授業内容

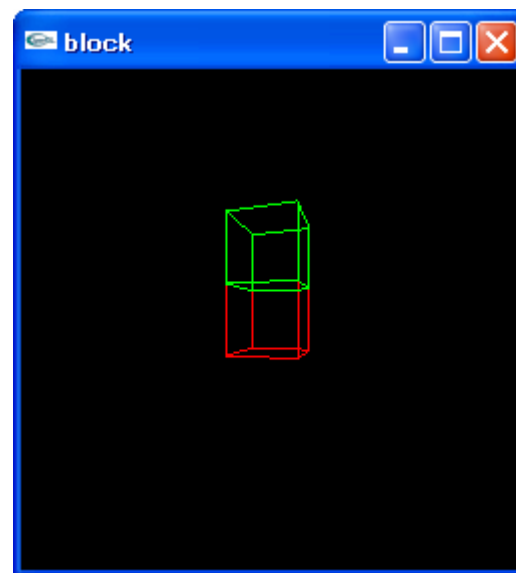
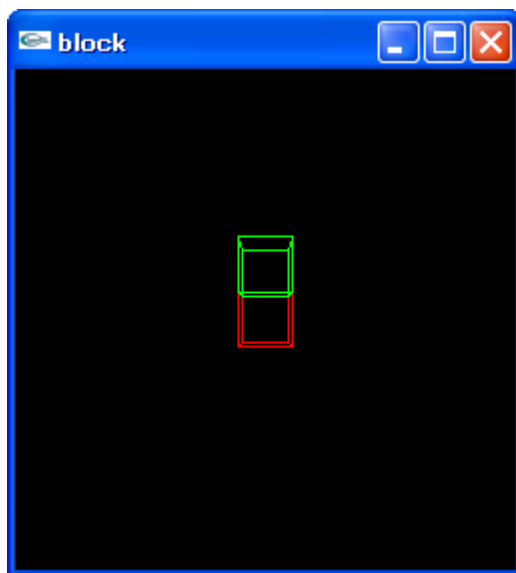
---

- OpenGL によるグラフィックス (9)
  - モデリング



# 例題

- 赤色のブロックの上に、緑色のブロックがある
  - 'z' または 'Z' で物体までの距離を調節する
  - 'r' または 'R' で視点を回転させる





# ソースコード (1/4)

**block2.c**

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

double z = 5.0;
double theta = 0.0;

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 );

    glColor3f( 1.0, 0.0, 0.0 );
    glutWireCube( 1.0 );
}
```



# ソースコード (2/4)

```
glTranslatef( 0.0, 1.0, 0.0 );
glColor3f( 0.0, 1.0, 0.0 );
glutWireCube( 1.0 );

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 (3/4)

```
case 'r':
    theta -= 0.1;
    break;
case 'R':
    theta += 0.1;
    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 );
```



# ソースコード (4 / 4)

```
glMatrixMode( GL_PROJECTION );  
glLoadIdentity();  
glFrustum( -1.0, 1.0, -1.0, 1.0, 1.0, 20.0 );  
glMatrixMode( GL_MODELVIEW );  
  
glutDisplayFunc( display );  
glutKeyboardFunc( myKeyBoard );  
glutMainLoop();  
  
return 0;  
}
```



# glTranslatef 関数

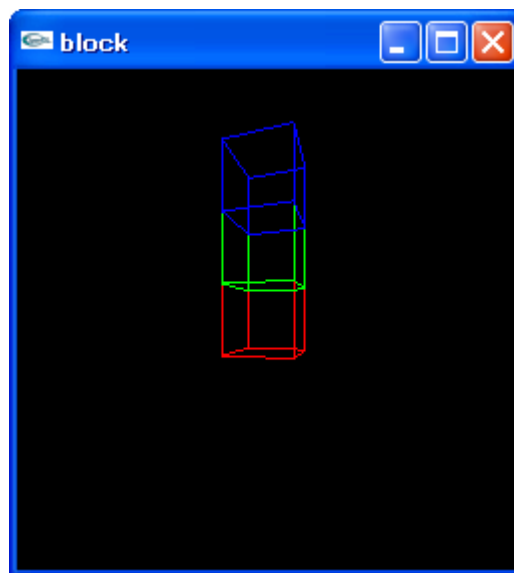
- 立体図形を指定した分だけ平行移動させる
  - *tx* x軸方向の移動量
  - *ty* y軸方向の移動量
  - *tz* z軸方向の移動量立体図形の描画前に設定する

```
glTranslatef( tx, ty, tz );
```



# 練習

- 例題のプログラムに、さらに青色の立方体を積み上げ、下図のように3つの立方体を表示せよ。



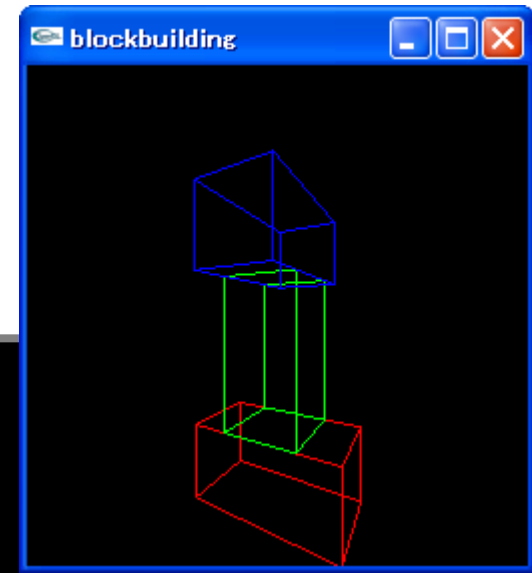




# 実習課題

- 大きさ  $1 \times 1 \times 2$  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
```





# 次回の予定

---

- 日時： 12月 7日 (金)  
4 時限
- OpenGL によるグラフィックス  
– モデリング