



2007年12月21日

第25回 OpenGL によるグラフィックス (14)

情報処理論 (応用)

松山大学 経営学部

檀 裕也

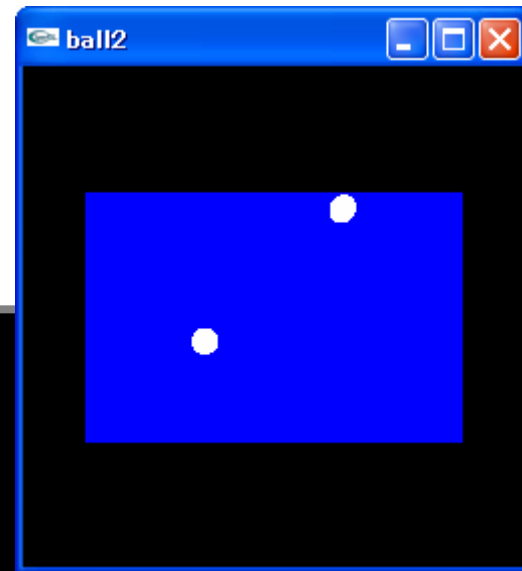
<http://www.cc.matsuyama-u.ac.jp/~dan/education/application/>



前回の実習課題

- 例題のプログラムで、ボールを2個に増やし、動く範囲を長方形にせよ。
 - 宛先: dan@cc.matsuyama-u.ac.jp
 - 件名: CG課題#13

```
C:¥>ball2.exe
```





解答例 (1 / 4)

ball2.c

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

double r = 0.1;
double x1 = 0.0, y1 = 0.5;
double dx1 = 0.001, dy1 = 0.001;
double x2 = 0.0, y2 = -0.5;
double dx2 = -0.001, dy2 = 0.001;

void display( void )
{
    glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );
    glLoadIdentity();
    gluLookAt( 0.0, 0.0, 2.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0 );
    glEnable( GL_DEPTH_TEST );
    glColor3f( 0.0, 0.0, 1.0 );
    glBegin( GL_POLYGON );
        glVertex3f( +1.5, +1.0, 0.0 );
        glVertex3f( -1.5, +1.0, 0.0 );
        glVertex3f( -1.5, -1.0, 0.0 );
```



解答例 (2 / 4)

```
glVertex3f( +1.5, -1.0, 0.0 );  
glEnd();
```

```
glColor3f( 1.0, 1.0, 1.0 );  
glPushMatrix();  
    glTranslatef( x1, y1, r );  
    glutSolidSphere( r, 24, 24 );  
glPopMatrix();
```

```
glColor3f( 1.0, 1.0, 1.0 );  
glPushMatrix();  
    glTranslatef( x2, y2, r );  
    glutSolidSphere( r, 24, 24 );  
glPopMatrix();
```

```
glDisable( GL_DEPTH_TEST );  
glutSwapBuffers();
```

```
}
```

```
void myKeyBoard( unsigned char key, int x, int y )  
{  
    if( key == 0x1B ){
```



解答例 (3 / 4)

```
    exit( 0 );
}

void idle( void )
{
    if( x1 + dx1 < -1.5 + r || 1.5 - r < x1 + dx1 ){
        dx1 = -dx1;
    }
    if( y1 + dy1 < -1.0 + r || 1.0 - r < y1 + dy1 ){
        dy1 = -dy1;
    }
    x1 += dx1;
    y1 += dy1;
    if( x2 + dx2 < -1.5 + r || 1.5 - r < x2 + dx2 ){
        dx2 = -dx2;
    }
    if( y2 + dy2 < -1.0 + r || 1.0 - r < y2 + dy2 ){
        dy2 = -dy2;
    }
    x2 += dx2;
    y2 += dy2;
}
```



解答例 (4 / 4)

```
    glutPostRedisplay();
}

int main( int argc, char *argv[] )
{
    glutInit( &argc, argv );
    glutInitDisplayMode( GLUT_RGB | GLUT_DEPTH | GLUT_DOUBLE );
    glutInitWindowSize( 250, 250 );
    glutCreateWindow( argv[0] );
    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 );
    glutIdleFunc( idle );
    glutMainLoop();

    return 0;
}
```



今回の授業内容


- OpenGL によるグラフィックス (14)
 - クリスマス・スペシャル



例題

- クリスマス・ツリーを表現せよ。また、マウス操作によって視点を変更できるようにせよ。





ソースコード (1 / 6)

xmas.c

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

#define PI 3.14159265358979

double er = 4.0, ez = 2.0, theta = 0.0;
int xBegin, yBegin, mButton;

void display( void )
{
    int i;

    glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );
    glLoadIdentity();
    gluLookAt( er * cos( theta ), er * sin( theta ), ez,
              0.0, 0.0, 0.0, 0.0, 0.0, 1.0 );
    glEnable( GL_DEPTH_TEST );

    /* 地面 */
}
```



ソースコード (2/6)

```
glColor3f( 0.5, 0.1, 0.1 );
glPushMatrix();
    glScalef( 20.0, 20.0, 5.0 );
    glTranslatef( 0.0, 0.0, -0.5 );
    glutSolidCube( 1.0 );
glPopMatrix();

/* 木の幹 */
glColor3f( 0.8, 0.2, 0.2 );
glPushMatrix();
    glTranslatef( 0.0, 0.0, 1.0 );
    glScalef( 0.1, 0.1, 2.0 );
    glutSolidCube( 1.0 );
glPopMatrix();

/* 葉っぱ */
glTranslatef( 0.0, 0.0, 1.0 );
for( i = 0; i < 12; i++ ){
    glColor3f( 0.0, 0.5 * ( i % 2 ), 0.0 );
    glBegin( GL_POLYGON );
        glVertex3f( 0.0, 0.0, 2.0 );
        glVertex3f( cos( i * 2 * PI / 12 ), sin( i * 2 * PI / 12 ), 0.0 );
```



ソースコード (3/6)

```
    glVertex3f( cos( (i+1) * 2 * PI / 12 ), sin( (i+1) * 2 * PI / 12 ), 0.0 );
    glEnd();
}

glDisable( GL_DEPTH_TEST );
glutSwapBuffers();
}

void myKeyBoard( unsigned char key, int x, int y )
{
    if( key ==0x1B ){
        exit( 0 );
    }
    else{
        er = 4.0;
        ez = 2.0;
        theta = 0.0;
    }
    glutPostRedisplay();
}

void myMouse( int button, int state, int x, int y )
```



ソースコード (4/6)

```
{  
    if( state == GLUT_DOWN ){  
        mButton = button;  
        xBegin = x;  
        yBegin = y;  
    }  
}  
  
void myMotion( int x, int y )  
{  
    int xDisp = x - xBegin;  
    int yDisp = y - yBegin;  
  
    switch( mButton ){  
    case GLUT_LEFT_BUTTON:  
        theta += xDisp / 64.0;  
        ez += (float)yDisp / 16.0;  
        break;  
    case GLUT_RIGHT_BUTTON:  
        er += (float)yDisp / 16.0;  
        break;  
    }  
}
```



ソースコード (5/6)

```
xBegin = x;  
yBegin = y;  
glutPostRedisplay();  
}  
  
int main( int argc, char *argv[] )  
{  
    glutInit( &argc, argv );  
    glutInitDisplayMode( GLUT_RGB | GLUT_DEPTH | GLUT_DOUBLE );  
    glutInitWindowSize( 250, 250 );  
    glutCreateWindow( argv[0] );  
  
    glClearColor( 0.5, 0.8, 1.0, 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 );  
    glutMouseFunc( myMouse );  
    glutMotionFunc( myMotion );  
}
```



ソースコード (6/6)

```
glutMainLoop();  
  
return 0;  
}
```



解説

- マウス操作を受け付ける：
 - コールバック関数を設定する

```
glutMouseFunc( myMouse );  
glutMotionFunc( myMotion );
```

- myMouse 関数および myMotion 関数の仮引数
 - button ボタンの種類
 - state ボタンの状態
 - x, y マウスポインタの座標

```
void myMouse( int button, int state, int x, int y )  
void myMotion( int x, int y )
```



実習課題

- 例題のプログラムに、デコレーションを加えよ。
また、自由にアレンジせよ。
 - 宛先: dan@cc.matsuyama-u.ac.jp
 - 件名: CG課題#14

```
C:¥>xmas2.exe
```





次回の予定

- 日時： 2008年 1月 9日 (水)
5 時限
- OpenGL によるグラフィックス
– 照明の効果