



2008年 1月 9日

第26回 CG作品の自由制作 (1)

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/>



今後のスケジュール

- これまでに習得したCG技法を使って、オリジナル作品を制作する

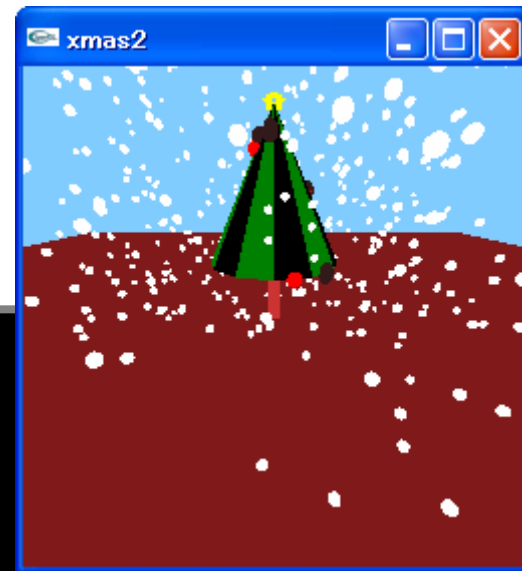
回	日付	内容
26	1月 9日 (水)	CG作品の自由制作(1)
27	1月 11日 (金)	CG作品の自由制作(2)
28	1月 16日 (水)	CG作品の自由制作(3)
29	1月 23日 (水)	作品発表会(1)
30	1月 25日 (金)	作品発表会(2)・まとめ



前回の実習課題

- 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
```





解答例 (1 / 7)

xmas2.c

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

#define PI 3.14159265358979
#define NUM 1024

double er = 4.0, ez = 2.0, theta = 0.0;
int xBegin, yBegin, mButton;
double x[NUM], y[NUM], z[NUM];

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 */
    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();
}
```



解答例 (2/7)

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

/* 葉っぱ */
glPushMatrix();
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 );
        glVertex3f( cos( ( i + 1 ) * 2 * PI / 12 ), sin( ( i + 1 ) * 2 * PI / 12 ), 0.0 );
    glEnd();
    if( i % 3 == 0 ){
        glColor3f( 1.0, 0.0, 0.0 );
        glPushMatrix();
            glTranslatef( cos( ( i + 0.5 ) * 2 * PI / 12 ) / ( i / 3 + 1 ),
                        sin( ( i + 0.5 ) * 2 * PI / 12 ) / ( i / 3 + 1 ), 2.0 - 2.0 / ( i / 3 + 1 ) );
            glutSolidSphere( 0.1, 12, 12 );
        glPopMatrix();
    }
    if( i % 2 == 1 ){
        glColor3f( 0.2, 0.1, 0.1 );
        glPushMatrix();
```



解答例 (3 / 7)

```
glTranslatef( cos( ( i + 0.5 ) * 2 * PI / 12 ) / ( i / 2 + 1 ),
              sin( ( i + 0.5 ) * 2 * PI / 12 ) / ( i / 2 + 1 ), 2.0 - 2.0 / ( i / 2 + 1 ) );
glScalef( 1.0, 1.0, 1.5 );
glutSolidSphere( 0.1, 12, 12 );
glPopMatrix();
}
}
glPopMatrix();

/* 星 */
glPushMatrix();
glTranslatef( 0.0, 0.0, 3.0 );
glScalef( 0.1, 0.1, 0.1 );
glLineWidth( 4.0 );
glColor3f( 1.0, 1.0, 0.0 );
glBegin( GL_LINE_LOOP );
for( i = 0; i < 10; i++ ){
    if( i % 2 == 0 ){
        glVertex3f( 0.0, sin( 2 * PI * i / 10 ), cos( 2 * PI * i / 10 ) );
    }
    else{
        glVertex3f( 0.0, sin( 2 * PI * i / 10 ) / 2, cos( 2 * PI * i / 10 ) / 2 );
    }
}
glEnd();
glPopMatrix();

/* 雪 */
glColor3f( 1.0, 1.0, 1.0 );
```



解答例 (4 / 7)

```
for( i = 0; i < NUM; i++ ){
    glPushMatrix();
    glTranslatef( x[i], y[i], z[i] );
    glutSolidSphere( 0.05, 12, 12 );
    glPopMatrix();
}

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 )
{
    if( state == GLUT_DOWN ){
        mButton = button;
        xBegin = x;
    }
}
```



解答例 (5 / 7)

```
    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;
    }
    xBegin = x;
    yBegin = y;
    glutPostRedisplay();
}

void idle( void )
{
    int i;

    for( i = 0; i < NUM; i++ ){
        if( z[i] > 0.0 ){
```




解答例 (6 / 7)

```
x[i] += ( rand() % 5 - 2 ) / 32.0;
y[i] += ( rand() % 5 - 2 ) / 32.0;
z[i] -= 0.005 * ( rand() % 100 );
}
else{
    z[i] = 10.0;
}
}

glutPostRedisplay();
}

int main( int argc, char *argv[] )
{
    int i;

    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 );
}
```



解答例 (7 / 7)

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

for( i = 0; i < NUM; i++ ){
    x[i] = rand() % 5 - 2;
    y[i] = rand() % 5 - 2;
    z[i] = 10.0;
}

glutMainLoop();

return 0;
}
```



今回の授業内容

- シェーディング
- CG作品の自由制作(1)



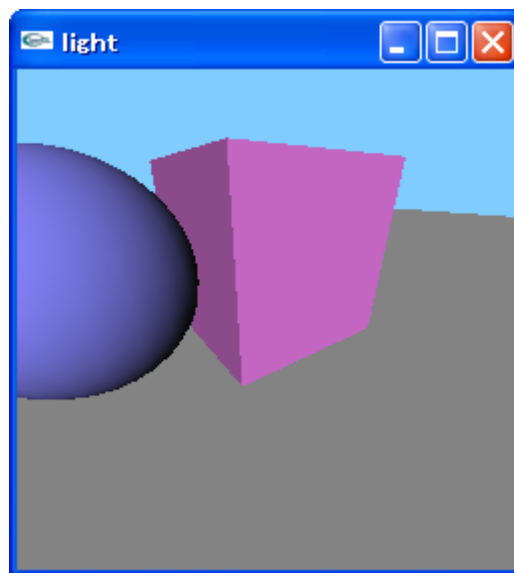
シェーディング


- 光源や物体の材質などの要素からモデルに陰影をつけ、立体感のあるCGを実現する技法



例題

- 物体表面の材質を定義し、シェーディングを実現する。





ソースコード (1 / 6)

light.c

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

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

void display( void )
{
    float diffuse1[] = { 1.0, 1.0, 1.0, 1.0 };
    float diffuse2[] = { 1.0, 0.5, 1.0, 1.0 };
    float diffuse3[] = { 0.5, 0.5, 1.0, 1.0 };

    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 );
    glEnable( GL_LIGHTING );
}
```



ソースコード (2/6)

```
/* テーブル面 */
```

```
glMaterialfv( GL_FRONT, GL_DIFFUSE, diffuse1 );  
glBegin( GL_POLYGON );  
    glNormal3f( 0.0, 0.0, 1.0 );  
    glVertex3f( 20.0, 20.0, 0.0 );  
    glVertex3f( -20.0, 20.0, 0.0 );  
    glVertex3f( -20.0, -20.0, 0.0 );  
    glVertex3f( 20.0, -20.0, 0.0 );  
glEnd();
```

```
/* 立方体 */
```

```
glMaterialfv( GL_FRONT, GL_DIFFUSE, diffuse2 );  
glPushMatrix();  
    glTranslatef( 0.0, 0.0, 1.0 );  
    glutSolidCube( 2.0 );  
glPopMatrix();
```

```
/* 球面 */
```

```
glMaterialfv( GL_FRONT, GL_DIFFUSE, diffuse3 );  
glPushMatrix();  
    glTranslatef( 0.0, 2.0, 1.0 );  
    glutSolidSphere( 1.0, 24, 24 );
```



ソースコード (3/6)

```
glPopMatrix();

glDisable( GL_LIGHTING );
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;
    }
    xBegin = x;
```



ソースコード (5 / 6)

```
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)

```
glEnable( GL_LIGHT0 ); // 0番のライトを使用すると宣言  
  
glutMainLoop();  
  
return 0;  
}
```



解説

- あらかじめ float 型配列

```
float diffuse[] = { R, G, B,    };
```

を定義し、物体の拡散反射成分を

```
glMaterialfv( GL_FRONT, GL_DIFFUSE, diffuse );
```

によって与える。

```
glEnable( GL_LIGHTING );
```

シェーディングの処理

```
glDisable( GL_LIGHTING );
```



実習課題

- これまでに習得したCG技法を使って、オリジナル作品を制作せよ。
 - 宛先: dan@cc.matsuyama-u.ac.jp
 - 件名: 学籍番号_氏名 (例: 12050000_檀裕也)
 - 本文: CG作品の予定



次回の予定

- 日時： 2008年 1月11日 (金)
4時限
- CG作品の自由制作(2)