２００７年１１月２８日

第18回　OpenGL によるグラフィックス（7）

情報処理論（応用）

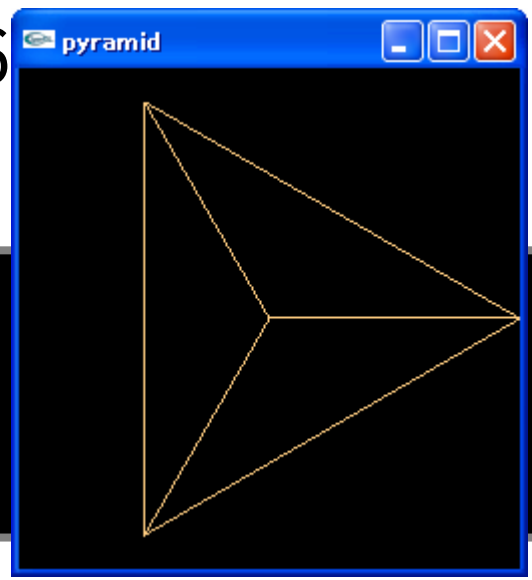
松山大学 経営学部
檀　裕也

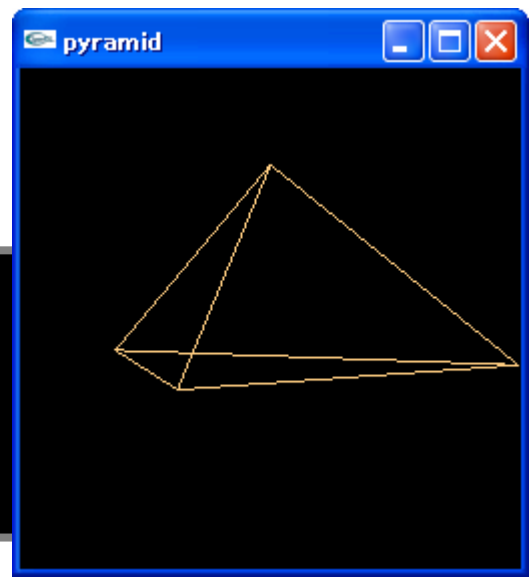
http://www.cc.matsuyama-u.ac.jp/~dan/education/application/

前回の実習課題

- 下図のピラミッドような図形 (四角錐でもよい) を描画するプログラムのソースコード pyramid.c を提出せよ。
  - 宛先：dan@cc.matsuyama-u.ac.jp
  - 件名：CG課題#06

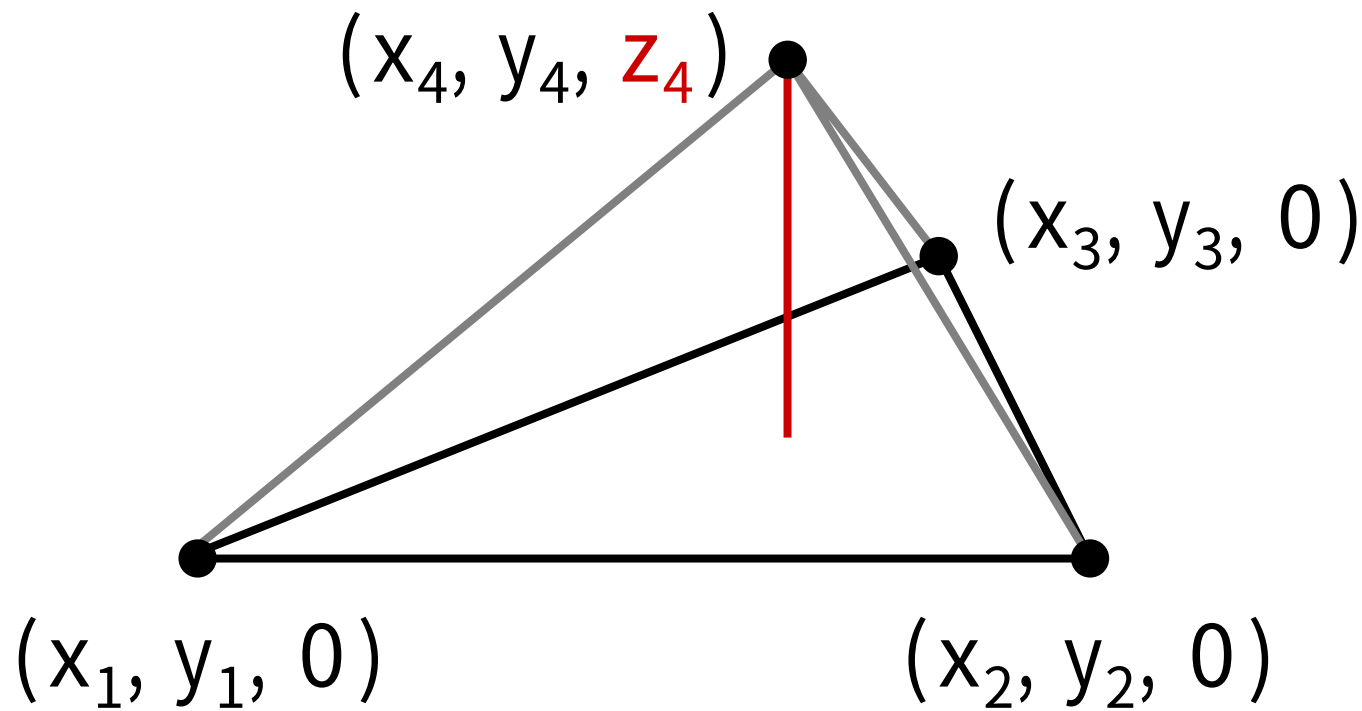
```
C:\>pyramid.exe
```





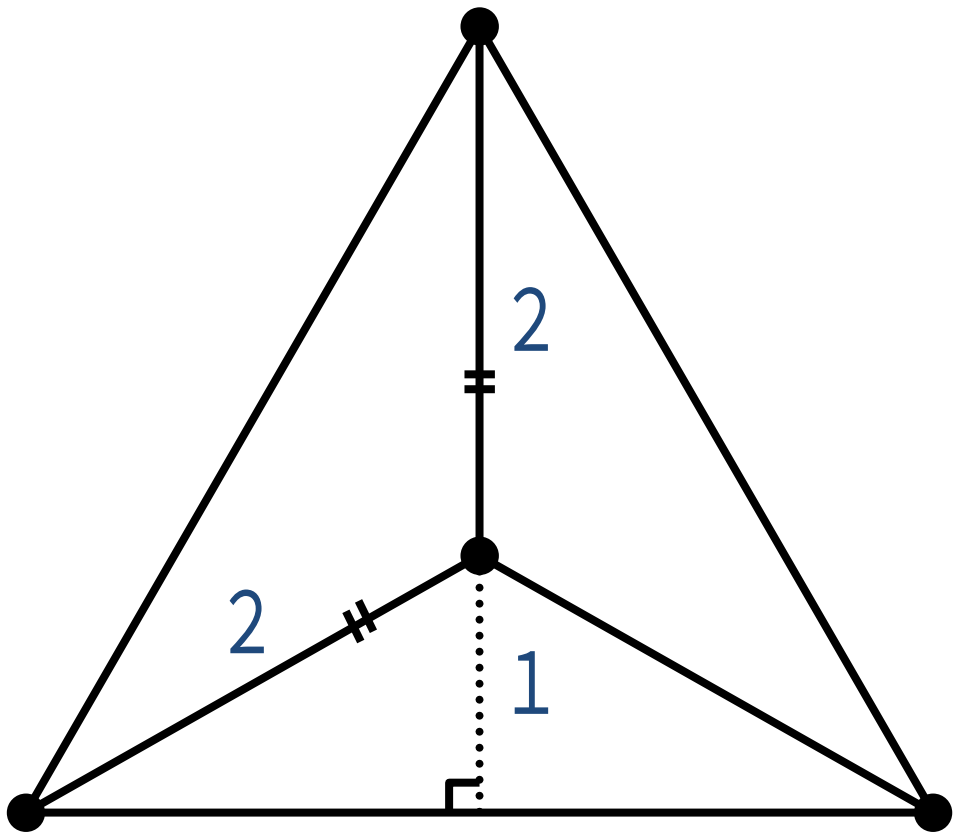
考えかた（1）

- ４つの頂点の３次元座標を求める
  - そのうち３頂点はｘｙ平面上で考えるとよい



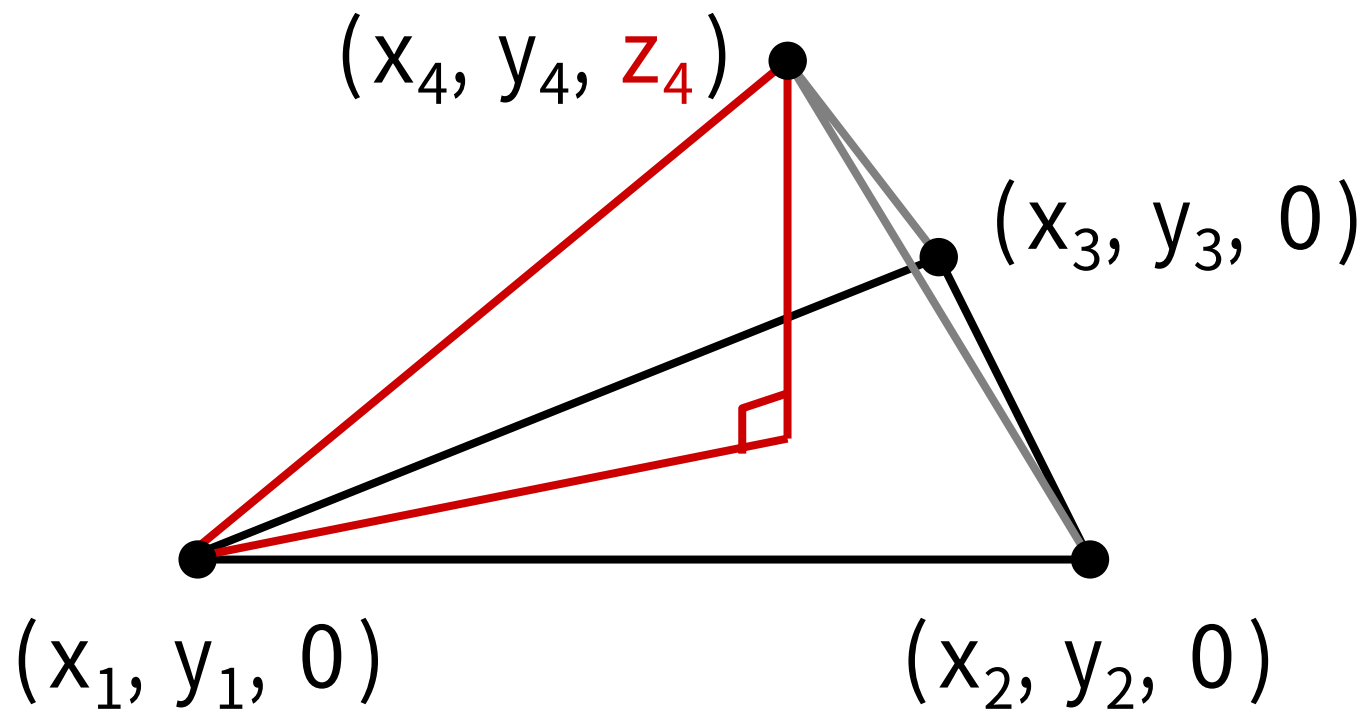
考えかた（2）

- 中線は中心を2：1に分ける



考えかた（3）

- 直角三角形に三平方の定理を適用し、高さ z_4 を求める。



解答例（1/4）

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

#define PI      3.141592653589793

void display( void )
{
        int i;
        double theta;

        // 背景を消去する
        glClear( GL_COLOR_BUFFER_BIT );

        // 砂色のピラミッドを描く
        glColor3f( 1.0, 0.8, 0.5 );
```

解答例（2/4）

```
        glBegin( GL_LINE_LOOP );
        for( i = 0; i < 3; i++ ){
                theta = 2.0 * PI * (double)i / 3.0;
                glVertex3f( cos( theta ), sin( theta ), 0.0 );
        }
        glEnd();

        glBegin( GL_LINES );
        for( i = 0; i < 3; i++ ){
                theta = 2.0 * PI * (double)i / 3.0;
                glVertex3f( 0.0, 0.0, sqrt( 6.0 ) / 3.0 );
                glVertex3f( cos( theta ), sin( theta ), 0.0 );
        }
        glEnd();

        // 発行した OpenGL コマンドを実行する
        glFlush();
}
```

解答例（3/4）

```
int main( int argc, char *argv[] )
{
        // GLUT ライブラリの初期化
        glutInit( &argc, argv );
        glutInitDisplayMode( GLUT_RGB );
        glutInitWindowSize( 250, 250 );
        glutCreateWindow( argv[0] );

        // 背景色を黒にする
        glClearColor( 0.0, 0.0, 0.0, 0.0 );

        // 座標系を設定する
        glMatrixMode( GL_PROJECTION );
        glLoadIdentity();
        glOrtho( -1.0, 1.0, -1.0, 1.0, -1.0, 2.0 );
        gluLookAt( 1.0, 0.4, 0.3, 0.0, 0.0, 0.2, 0.0, 0.0, 1.0 );
```

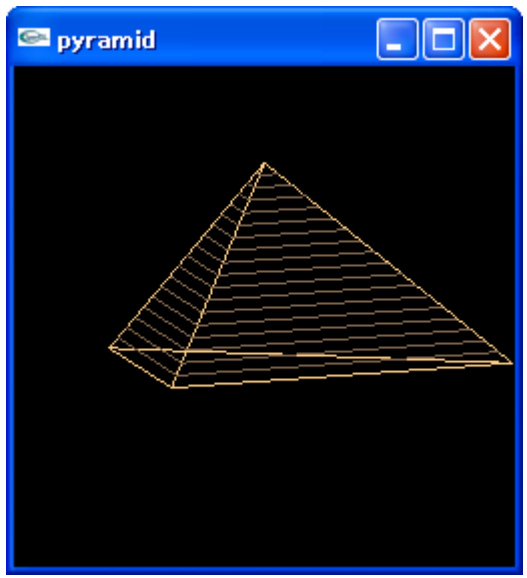
解答例（4/4）

```
        // 描画処理をする
        glutDisplayFunc( display );
        glutMainLoop();

        return 0;
}
```

発展課題

- さらに、横線模様をつけてもよい。



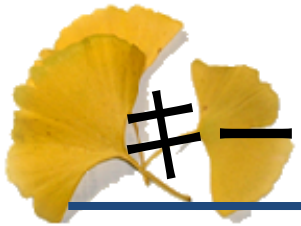
発展課題の解答例

- int 型変数 t と分割数 DIV (解答例は18)を宣言（または定義）した上で、以下のソースコードを display 関数内に追加する。

```
glColor3f( 0.5, 0.4, 0.3 );
for( t = 0; t < DIV; t++ ){
  glBegin( GL_LINE_STRIP );
  for( i = 0; i < 3; i++ ){
    theta = 2.0 * PI * (double)( i - 1 ) / 3.0;
    glVertex3f( cos( theta ) * t / DIV, sin( theta ) * t / DIV,
                sqrt( 6.0 ) / 3.0 * ( DIV - t ) / DIV );
    }
  glEnd();
}
```

今回の授業内容

- OpenGL によるグラフィックス（7）
  - キーボード入力による割り込みの処理



キーボード入力による割り込みの処理

- プログラム実行中にキーが押されたら、あらかじめ用意しておいた処理を実行する。
  - glutKeyboardFunc でコールバック関数を設定

```
glutKeyboardFunc( myKeyBoard );
```

  - 引数 key には入力された文字コードが入る

```
void myKeyBoard( unsigned char key, int x, int y )
{
        ...
```

例題

- 押されたキーに応じて、表示する立体図形を切り替えるプログラム
  - 'p' を押すとピラミッド図形
  - 'c' を押すと立方体
  - 't' を押すとティーポット
  - [ESC] を押すと表示画面を閉じる

ソースコード（1/6）

pyramid2.c

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

#define PI      3.141592653589793
#define DIV     18

char object;

void pyramid( void )
{
        int i, t;
        double theta;

        // 砂色のピラミッドを描く
        glColor3f( 1.0, 0.8, 0.5 );
```

ソースコード（2/6）

```
glBegin( GL_LINE_LOOP );
for( i = 0; i < 3; i++ ){
        theta = 2.0 * PI * (double)i / 3.0;
        glVertex3f( cos( theta ), sin( theta ), 0.0 );
}
glEnd();

glBegin( GL_LINES );
for( i = 0; i < 3; i++ ){
        theta = 2.0 * PI * (double)i / 3.0;
        glVertex3f( 0.0, 0.0, sqrt( 6.0 ) / 3.0 );
        glVertex3f( cos( theta ), sin( theta ), 0.0 );
}
glEnd();

// 表面に縞模様を描く
glColor3f( 0.5, 0.4, 0.3 );
for( t = 0; t < DIV; t++ ){
```

ソースコード（3/6）

```
                glBegin( GL_LINE_STRIP );
                for( i = 0; i < 3; i++ ){
                        theta = 2.0 * PI * (double)( i - 1 ) / 3.0;
                        glVertex3f( cos( theta ) * t / DIV,
                        sin( theta ) * t / DIV,
                        sqrt( 6.0 ) / 3.0 * ( DIV - t ) / DIV );
                }
                glEnd();
        }

        return;
}

void display( void )
{
        // 背景を消去する
        glClear( GL_COLOR_BUFFER_BIT );
```

ソースコード（4/6）

```
        switch( object ){
        case 'p':
                pyramid();
                break;
        case 'c':
                glColor3f( 0.0, 1.0, 0.0 );
                glutWireCube( 1.0 );
                break;
        case 't':
                glColor3f( 0.0, 0.0, 1.0 );
                glutWireTeapot( 0.7 );
                break;
        }

        // 発行した OpenGL コマンドを実行する
        glFlush();
}
```

ソースコード（5/6）

```
void myKeyBoard( unsigned char key, int x, int y )
{
        switch( key ){
        case 0x1B:
                exit( 0 );
        default:
                object = key;
        }
        glutPostRedisplay();
}

int main( int argc, char *argv[] )
{
        // GLUT ライブラリの初期化
        glutInit( &argc, argv );
        glutInitDisplayMode( GLUT_RGB );
        glutInitWindowSize( 250, 250 );
        glutCreateWindow( argv[0] );
```

ソースコード（6/6）

```
        // 背景色を黒にする
        glClearColor( 0.0, 0.0, 0.0, 0.0 );

        // 座標系を設定する
        glMatrixMode( GL_PROJECTION );
        glLoadIdentity();
        glOrtho( -1.0, 1.0, -1.0, 1.0, -1.0, 2.0 );
        gluLookAt( 1.0, 0.4, 0.3, 0.0, 0.0, 0.2, 0.0, 0.0, 1.0 );

        // 描画処理をする
        glutDisplayFunc( display );
        glutKeyboardFunc( myKeyBoard );
        glutMainLoop();

        return 0;
}
```

プログラムの構造

- pyramid 関数
  ピラミッド図形を描画する
- display 関数
  押されたキーに応じて立体図形を表示する
- myKeyBoard 関数
  キーボード・コールバック関数
- main 関数
  プログラム起動時に実行される

キーボード・コールバック

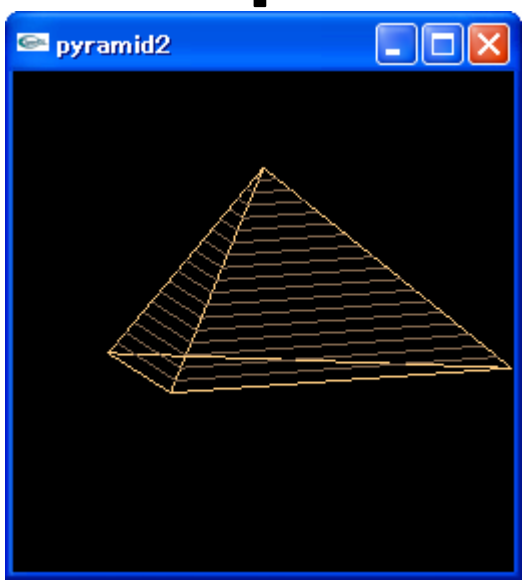
- [ESC]が押されたとき、プログラムを終了する
- それ以外のとき、グローバル変数 object に文字コードを渡す

```
void myKeyBoard( unsigned char key, int x, int y )
{
        switch( key ){
        case 0x1B:      // [ESC] の文字コード（16進数）
                exit( 0 );
        default:
                object = key;
        }
        glutPostRedisplay();    // 再描画させる
}
```

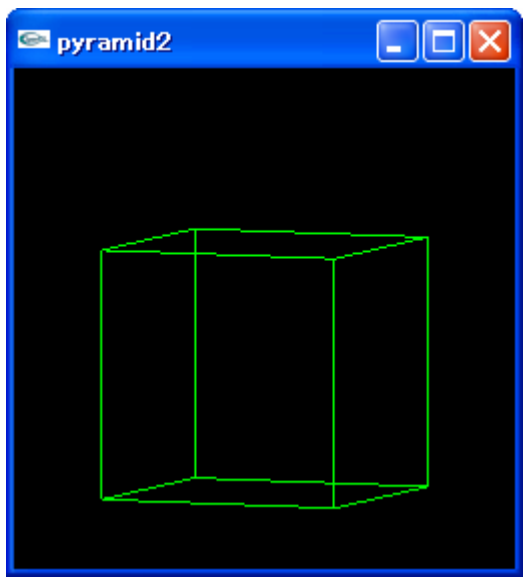
実行例

- キーボードを押すと、立体図形が表示される
  - [ESC] でプログラムを終了する

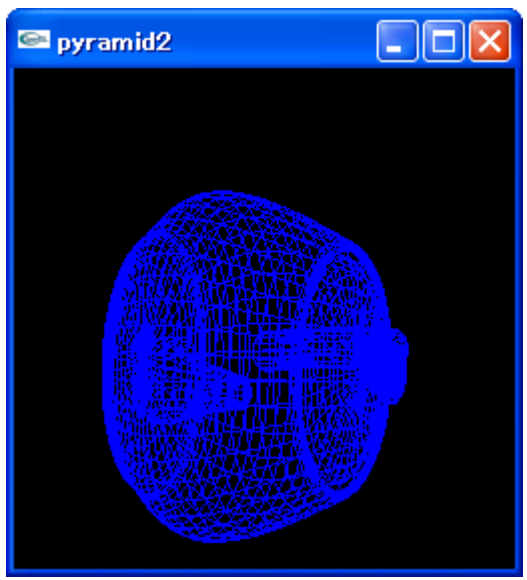
'p'



'c'



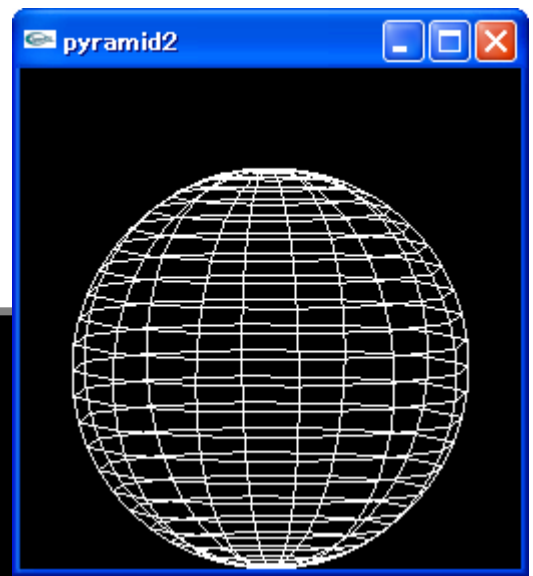
't'



実習課題

- 例題のプログラムに 's' キーを押すと球面を表示する機能を追加せよ。
  - 宛先：dan@cc.matsuyama-u.ac.jp
  - 件名：CG課題#07

```
C:\>pyramid2.exe
```



次回の予定

- 日時:　11月30日（金）
  4時限
- OpenGL によるグラフィックス
  - 立体図形