



2007年11月 9日

第14回 OpenGL によるグラフィックス (3)

情報処理論 (応用)

松山大学 経営学部

檀 裕也

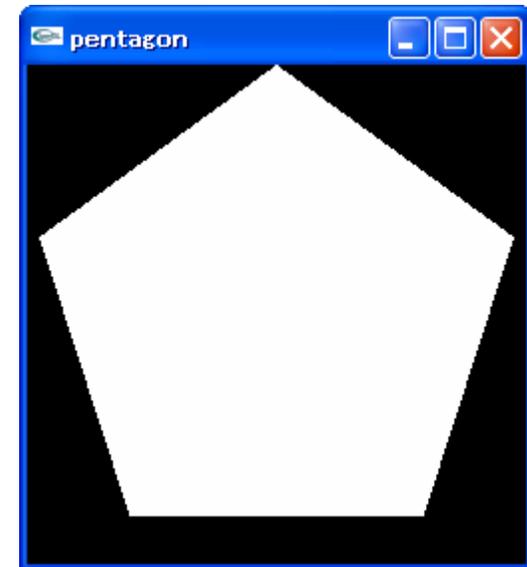
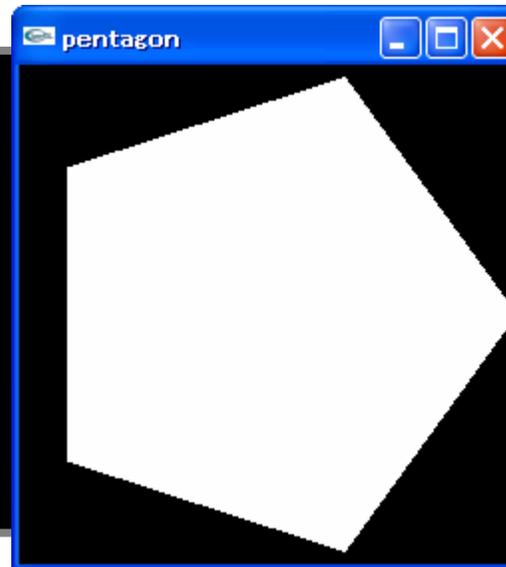
<http://www.cc.matsuyama-u.ac.jp/~dan/education/application/>



前回の実習課題

- 正五角形を描画するプログラムのソースコード `pentagon.c` を提出せよ。
 - 宛先: `dan@cc.matsuyama-u.ac.jp`
 - 件名: CG課題#02

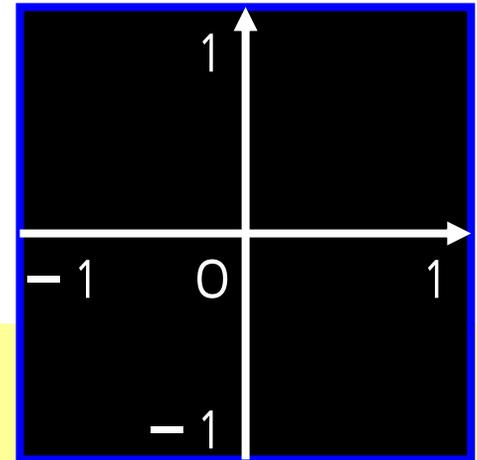
```
C:¥>pentagon.exe
```





ヒント

- main 関数内の座標系の設定で、表示範囲を拡張するとよい。



// 座標系を設定する

```
glMatrixMode( GL_PROJECTION );
```

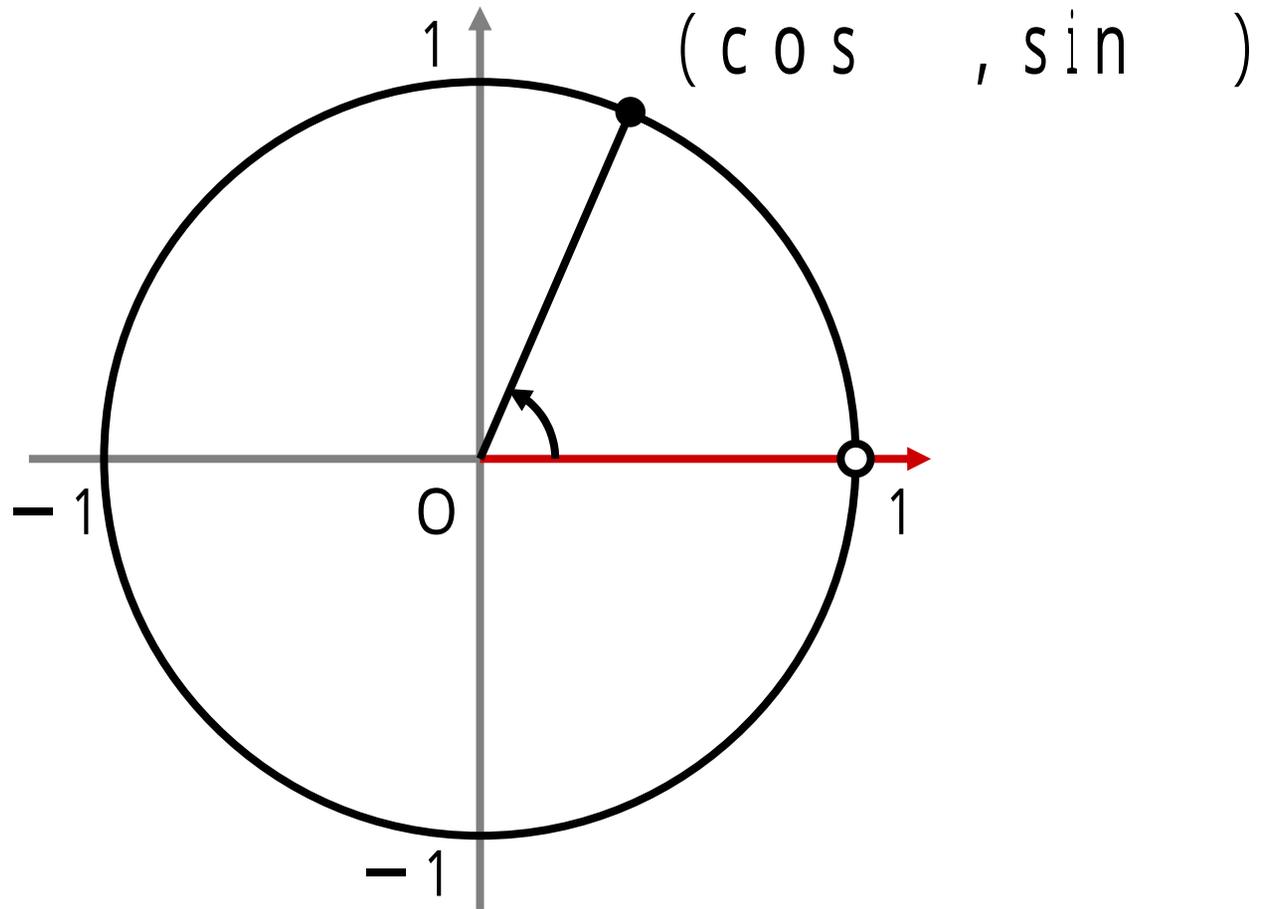
```
glLoadIdentity();
```

```
glOrtho( -1.0, 1.0, -1.0, 1.0, -1.0, 1.0 );
```



三角関数

- 角度を指定すると、単位円上の座標を与える





数学ライブラリ

```
#include <math.h>
```

- 三角関数をはじめ、関数の値を計算する
 - ソースコードの冒頭で `math.h` を読み込む
 - 引数として値を関数に渡すと、計算結果を返す
(例) `x = cos(2.0 * PI / 5.0);`
`y = sin(2.0 * PI / 5.0);`

<code>exp(x)</code>	指数関数 e^x
<code>log(x)</code>	対数関数 $\log x$
<code>sqrt(x)</code>	x の平方根 x
<code>pow(x, y)</code>	x の y 乗 x^y

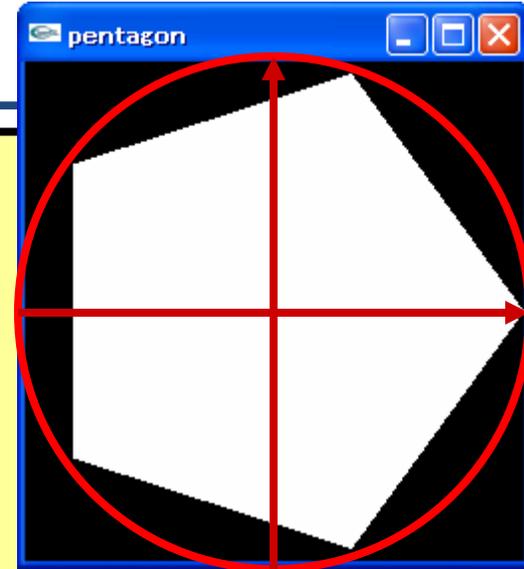


解答例 (一部抜粋)

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

    // 白い正五角形を描く
    glColor3f( 1.0, 1.0, 1.0 );
    glBegin( GL_POLYGON );
        glVertex3f( cos( 2.0 * PI / 5.0 * 0.0 ), sin( 2.0 * PI / 5.0 * 0.0 ), 0.0 );
        glVertex3f( cos( 2.0 * PI / 5.0 * 1.0 ), sin( 2.0 * PI / 5.0 * 1.0 ), 0.0 );
        glVertex3f( cos( 2.0 * PI / 5.0 * 2.0 ), sin( 2.0 * PI / 5.0 * 2.0 ), 0.0 );
        glVertex3f( cos( 2.0 * PI / 5.0 * 3.0 ), sin( 2.0 * PI / 5.0 * 3.0 ), 0.0 );
        glVertex3f( cos( 2.0 * PI / 5.0 * 4.0 ), sin( 2.0 * PI / 5.0 * 4.0 ), 0.0 );
    glEnd();

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





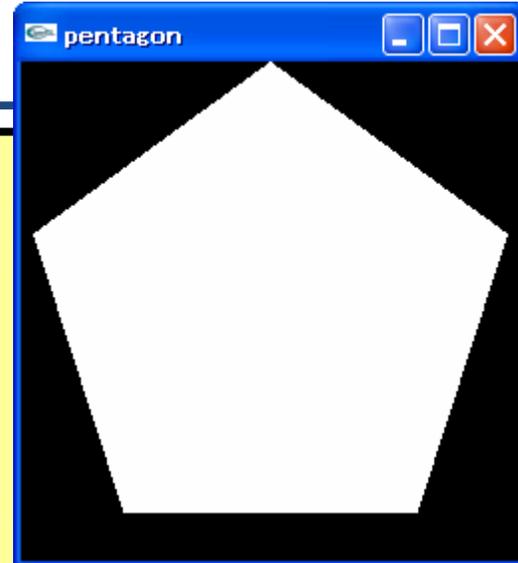
解答例 (一部抜粋)

```
void display( void )
{
    double theta = 2.0 * PI / 5.0;

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

    // 白い正五角形を描く
    glColor3f( 1.0, 1.0, 1.0 );
    glBegin( GL_POLYGON );
    glVertex3f( cos( theta * 0.0 + PI / 2.0 ), sin( theta * 0.0 + PI / 2.0 ), 0.0 );
    glVertex3f( cos( theta * 1.0 + PI / 2.0 ), sin( theta * 1.0 + PI / 2.0 ), 0.0 );
    glVertex3f( cos( theta * 2.0 + PI / 2.0 ), sin( theta * 2.0 + PI / 2.0 ), 0.0 );
    glVertex3f( cos( theta * 3.0 + PI / 2.0 ), sin( theta * 3.0 + PI / 2.0 ), 0.0 );
    glVertex3f( cos( theta * 4.0 + PI / 2.0 ), sin( theta * 4.0 + PI / 2.0 ), 0.0 );
    glEnd();

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





変数の型

- 小数を含む値は float 型または double 型の変数に格納する

変数の型	格納できる値
char	1バイト整数
int	4バイト整数
float	単精度浮動小数点数
double	倍精度浮動小数点数



今回の授業内容

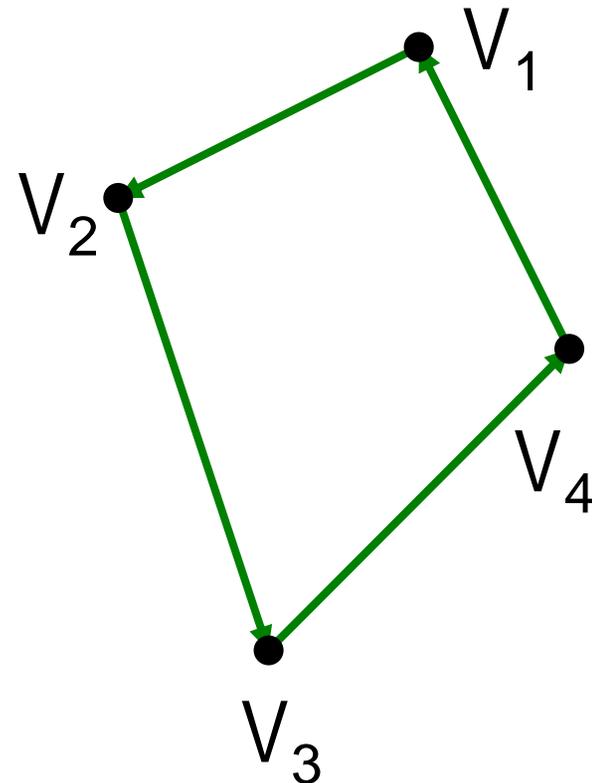
- OpenGL によるグラフィックス (3)
 - 正 n 角形の描画と三角関数



頂点座標データ

- 点、線分、ポリゴンなどの描画図形を指定し、頂点の座標を与える。

```
glBegin( 図形の指定 );  
glVertex3f( x0, y0, z0 );  
glVertex3f( x1, y1, z1 );  
glVertex3f( x2, y2, z2 );  
...  
glEnd();
```





図形の指定

- 頂点座標データを与えるとき、glBegin 関数に描画する図形の種類を指定する

GL_POINTS	点の集合
GL_LINES	線分の集合
GL_LINE_STRIP	連続した線分
GL_LINE_LOOP	ループした線分
GL_POLYGON	ポリゴン



ソースコード (1 / 3)

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

#define PI 3.14159
#define N 6
```



ソースコード (2 / 3)

```
void display( void )
{
    int i;

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

    // 白い正 n 角形を描く
    glColor3f( 1.0, 1.0, 1.0 );
    glBegin( GL_POLYGON );
    
    glEnd();

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



ソースコード (3 / 3)

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

    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, 1.0 );

    glutDisplayFunc( display );
    glutMainLoop();

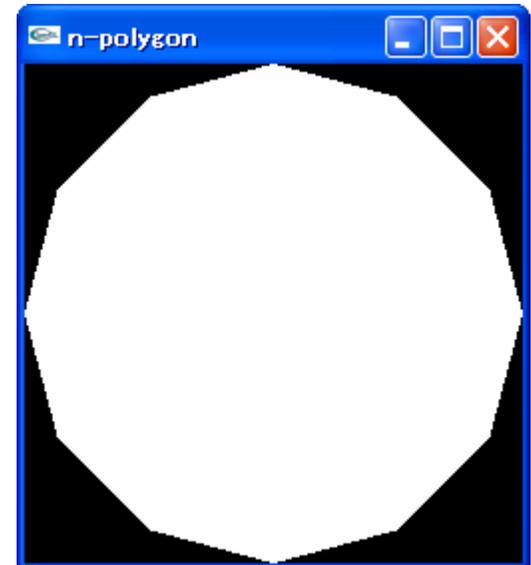
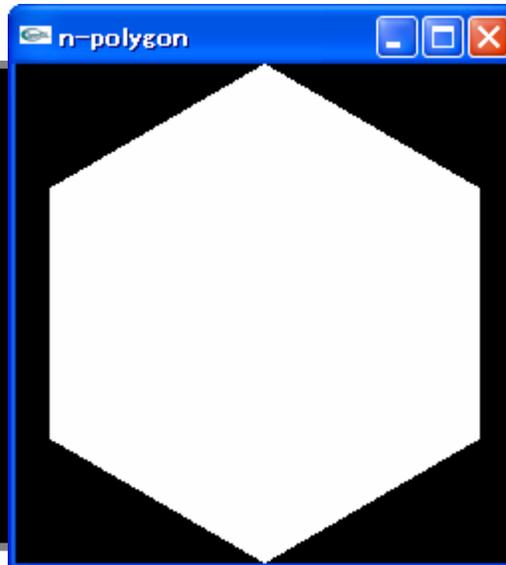
    return 0;
}
```



実習課題

- 正 n 角形を描画するプログラムのソースコード `polygon.c` を提出せよ。
 - 宛先: dan@cc.matsuyama-u.ac.jp
 - 件名: CG課題#03

```
C:¥>polygon.exe
```





ヒント

- ソースコード冒頭で N の値を変えるだけで、正 n 角形が描画されるように、工夫する。
- 頂点座標データを与える繰り返しの処理に着目するとよい。

```
...  
#include <math.h>  
  
#define PI 3.14159  
#define N 6
```



次回の予定

- 日時： 11月14日（水）
5時限
- OpenGL によるグラフィックス
– 描画色