



2007年11月 2日

第12回 OpenGL によるグラフィックス (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/>



前回の実習課題

- 次の問題に解答せよ。
 - 例題2における配列 `str` の要素数を答えよ。また、その内容を簡単に説明せよ。
 - ポインタとは何か説明せよ。
- 宛先: `dan@cc.matsuyama-u.ac.jp`
- 件名: プログラミング課題#10



解答

- 文字列の終端記号 '`¥0`' を含む点に注意する

```
char str[] = "Hello, world¥n";
```

```
char str[14];
```

```
str[0] = 'H';
```

```
str[1] = 'e';
```

```
str[2] = 'l';
```

```
str[3] = 'l';
```

```
str[4] = 'o';
```

```
str[5] = ',';
```

```
str[6] = ' ';
```

```
str[7] = 'w';
```

```
str[8] = 'o';
```

```
str[9] = 'r';
```

```
str[10] = 'l';
```

```
str[11] = 'd';
```

```
str[12] = '¥n';
```

```
str[13] = '¥0';
```



今回の授業内容

- OpenGL によるグラフィックス (1)
 - グラフィックス処理の第一歩



OpenGL とは

- 3次元CGに対応したグラフィックスライブラリ
 - プラットフォームに依存しない
 - オープンソース
 - 対応ハードウェアの普及



ヘッダーファイル

- C言語で OpenGL を使うには、ソースコードの冒頭で、そのヘッダーファイルを読み込む

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

標準ライブラリ

OpenGL

C言語のプログラム



座標系

- 直交座標系の範囲を設定する

```
glOrtho( x1, x2, y1, y2, z1, z2 );  
glOrtho( 0.0, 1.0, 0.0, 1.0, -1.0, 1.0 );
```





描画処理

- 描画用のコールバック関数を指定し、イベント待ちの状態にする。

```
glutDisplayFunc( display );  
glutMainLoop();
```

描画の必要があるときに
処理を実行する

イベントの
発生



処理実行

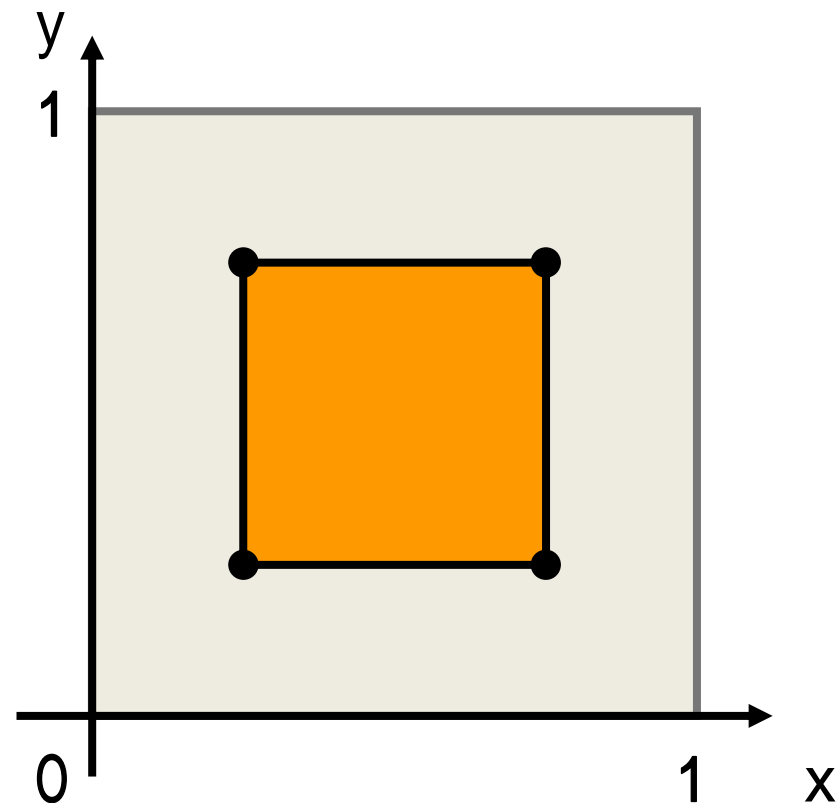
具体的な処理内容を
関数 `display` に記述する




正方形の描画

- 多角形の頂点座標を指定する

```
glBegin( GL_POLYGON );  
glVertex3f( 0.25, 0.25, 0.0 );  
glVertex3f( 0.75, 0.25, 0.0 );  
glVertex3f( 0.75, 0.75, 0.0 );  
glVertex3f( 0.25, 0.75, 0.0 );  
glEnd();
```






例題 (1 / 3)

whitesquare.c

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

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

    // 白い正方形を描く
    glColor3f( 1.0, 1.0, 1.0 );
    glBegin( GL_POLYGON );
        glVertex3f( 0.25, 0.25, 0.0 );
        glVertex3f( 0.75, 0.25, 0.0 );
        glVertex3f( 0.75, 0.75, 0.0 );
```




例題 (2 / 3)

```
        glVertex3f( 0.25, 0.75, 0.0 );
    glEnd();

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

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( "white square" );

    // 背景色を黒にする
```



例題 (3 / 3)

```
glClearColor( 0.0, 0.0, 0.0, 0.0 );

// 座標系を設定する
glMatrixMode( GL_PROJECTION );
glLoadIdentity();
glOrtho( 0.0, 1.0, 0.0, 1.0, -1.0, 1.0 );

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

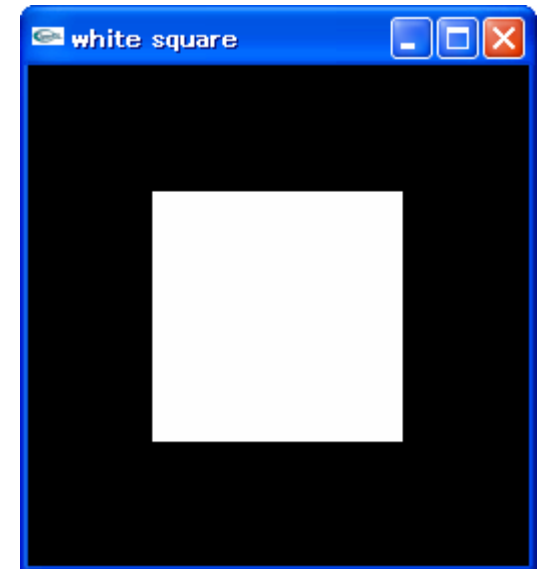
return 0;
}
```



実行結果

- プログラムを実行すると、新しいウィンドウが開き、白い正方形が表示される。
 - 閉じるボタンでプログラムを終了する

```
C:¥>whitesquare.exe
```

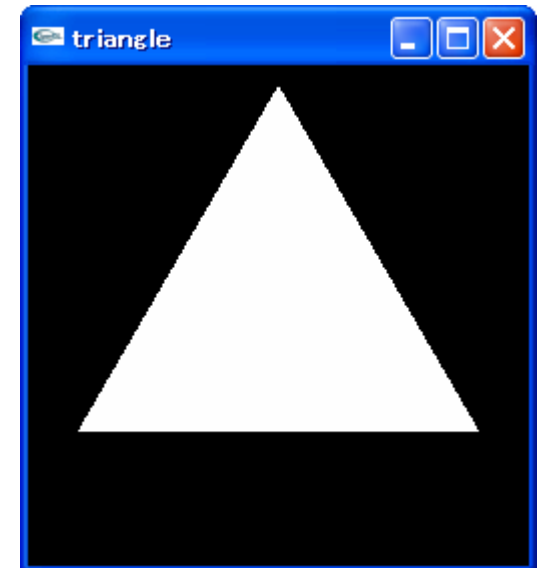




実習課題

- 正三角形を描画するプログラムのソースコード `whitetriangle.c` を提出せよ。
 - 宛先: `dan@cc.matsuyama-u.ac.jp`
 - 件名: CG課題#01

```
C:¥>whitetriangle.exe
```





次回の予定

- 日時： 11月 7日 (水)
5 時限
- OpenGL によるグラフィックス
– 多角形の描画