

## 「情報工学」CG 課題2

三浦 憲二郎

令和5年10月25日(水曜日)

提出締め切り 令和5年10月31日(火曜日) 24:00

提出物：提出物は各課題（課題A、課題B、課題C）の最後の状態のプログラムのソースコードとする。

提出：kawai.kota.shizuoka@gmail.com（三浦研川合航太君）

件名：情報工学課題2 学籍番号 名前

提出方法：添付ファイル、本文にも TA が管理しやすくなるので、学籍番号、名前を必ず記入すること。

### 線の描画と四則演算

#### グラフィック

- OpenGL の利用
- 線の描画( GL\_LINES )
- ウィンドウ座標
- 色の指定( glColor, glColor )
- 太さの指定( glLineWidth )

---

#### サンプルプログラム

```
#include <GL/glut.h>
```

```
void display(void) {
    glClearColor(GL_COLOR_BUFFER_BIT);           /* 背景のクリア */
    glBegin(GL_LINES);                           /* 線分を描画する */
    glVertex2f(0.0, 0.0);                         /* 始点の指定 */
    glVertex2f(0.5, 0.5);                         /* 終点の指定 */
    glEnd();                                      /* 線分の描画終了 */
    glFlush();                                    /* 画面を再描画する */
}

void init(void) {
    glClearColor( 0.0, 0.0, 0.0, 0.0 );          /* 背景色の指定 */
    glMatrixMode(GL_PROJECTION);                /* 射影変換を指定 */
    glLoadIdentity();                           /* 射影行列の初期化 */
    glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0); /* 描画のための座標系の指定 */
}

int main(int argc, char **argv) {
    glutInit(&argc, argv);                       /* GLUT の初期化 */
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB); /* 表示モードの指定 */
    glutInitWindowSize(400, 400);               /* ウィンドウサイズの指定 */
    glutInitWindowPosition(100, 100);           /* ウィンドウの位置の指定 */
}
```

```

glutCreateWindow("OpenGL Sample 3");          /* ウィンドウのオープン */
init();                                       /* 初期化処理           */
glutDisplayFunc(display);                   /* 表示の関数の指定     */
glutMainLoop();                             /* GLUT のメインループ  */
return 0;
}

```

### 課題 A

(a) 4×4 の升目を描画するように変更する。

Hint. 縦に 5 本, 横に 5 本の線を描画するように, `display()` 関数の中身を変更する。

(b) 行ごとに線の色を変えて描画するようにする。

Hint. 線の色を変更するには `glColor()` 関数を用いて色を指定した後に線を書く。  
赤色 (1.0, 0.0, 0.0) で線を書く場合 (標準では白色 (1.0, 1.0, 1.0)).

```

glColor(1.0, 0.0, 0.0);
glBegin(GL_LINES);
    glVertex2f(0.0, 0.0);
    glVertex2f(0.5, 0.5);
glEnd();

```

(c) 外枠のみ太い線を書くように変更する。

Hint. 線の太さを変更するには `glLineWidth()` 関数を用いて太さを指定した後に線を書く。

太さを 3 にして線を書く場合 (標準では 1.0).

```

glLineWidth(3.0);
glBegin(GL_LINES);
    glVertex2f(0.0, 0.0);
    glVertex2f(0.5, 0.5);
glEnd();

```

### 課題 B (時間に余裕のある人のために)

(a) `for` 文を用いて縦に 10 本の線を並べて描画するように拡張する。

Hint. 変数を用いて線の座標を変化させながら線の描画を繰り返す。繰り返される度に線の `x` 座標を計算し, その結果を利用して線を描画する。

線の間隔を 0.1 にした場合.

```

int i;                                       /* 繰り返し用の変数     */
double x;                                   /* 座標を保存しておく変数 */
for(i = 0; i < 10; i++) {                  /* 10 回繰り返す       */
    x = 0.1*i;                              /* x 座標を計算する     */
    glVertex2f(x, 0.0);                     /* 始点の指定           */
    glVertex2f(x, 0.5);                     /* 終点の指定           */
}

```

(b) `for` 文を用いて 40×40 の升目を描画するように拡張する。

Hint. 縦の線を描画する `for` 文と横の線を書く `for` 文を作成する。

### 課題 C

(さらに時間に余裕のある人のために)  $n \times m$  の升目を描画するように拡張する。

Hint. `main` 関数の中で, `scanf` 文を用いて `n` と `m` を読み込む。次に, 画面からはみ出さないように縦, 横それぞれの線の間隔を計算する。