

CG 基礎 課題 1

情報科学類二年 江畑 拓哉 (201611350)

1 動作環境の説明

- OS など

Windows10 Pro 内の Bash on Ubuntu on Windows
(4.4.0-43-Microsoft #1-Microsoft Wed Dec 31 14:42:53 PST 2014)

- コンパイル

g++ (Ubuntu 5.4.0-6ubuntu1~16.04.4) 5.4.0 20160609

Copyright (C) 2015 Free Software Foundation, Inc.

- コーディング

Spacemacs 0.200.9 (Emacs24.5.1)

2 課題 1

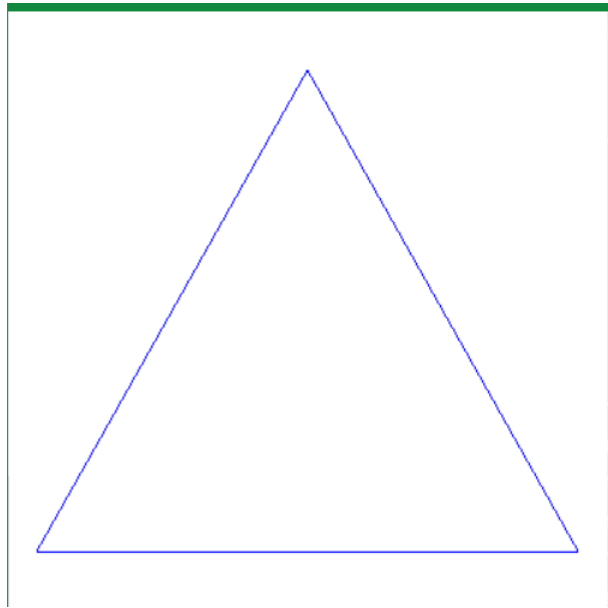
授業用 Web ページにある、それぞれのサンプルコードを実行し、プログラムコードと実行結果の様子を観察しなさい。

サンプルコードについてはこの部分の最後にまとめて記載する。

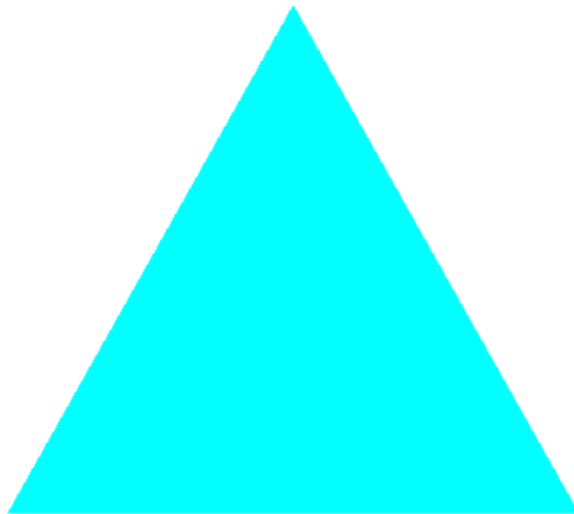
1. ウィンドウの表示



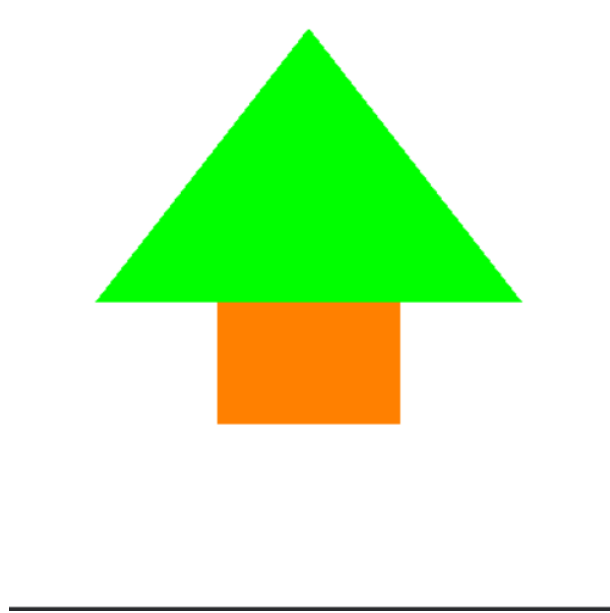
2. 直線の描画



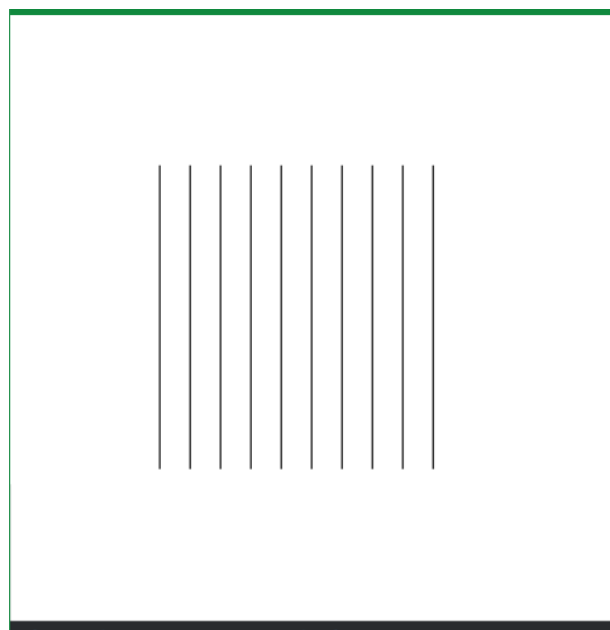
3. 三角形の描画



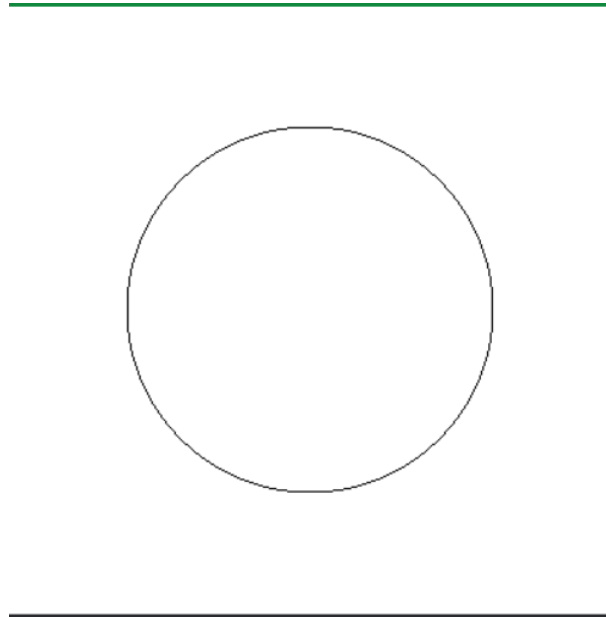
4. 複数の図形の描画



5. ループ処理を用いた描画



6. 円の描画



2.1 サンプルコード

1. ウィンドウの表示

```
1  #include <GL/glut.h> // ライブラリ用ヘッダファイルの読み込み
2
3  // 表示部分をこの関数で記入
4  void display(void) {
5      glClearColor (1.0, 1.0, 1.0, 1.0); // 消去色指定
6      glClear (GL_COLOR_BUFFER_BIT );    // 画面消去
7
8      /* ここに描画に関するプログラムコードを入れる */
9
10     glFlush(); // 画面出力
11 }
12
13 // メインプログラム
14 int main (int argc, char *argv[]) {
15     glutInit(&argc, argv);           // ライブラリの初期化
16     glutInitWindowSize(400 , 400);   // ウィンドウサイズを指定
17     glutCreateWindow(argv[0]);        // ウィンドウを作成
18     glutDisplayFunc(display);         // 表示関数を指定
19     glutMainLoop();                  // イベント待ち
20     return 0;
21 }
```

2. 直線の描画

```
1  #include <GL/glut.h> // ライブラリ用ヘッダファイルの読み込み
2
3  // 表示部分をこの関数で記入
4  void display(void) {
5      glClearColor (1.0, 1.0, 1.0, 1.0); // 消去色指定
6      glClear (GL_COLOR_BUFFER_BIT );    // 画面消去
7
8      glColor3d(0.0, 0.0, 1.0); // 色指定 (R,G,B) で 0~1 まで
9      glBegin(GL_LINE_LOOP);      // 描画するものを指定
10     glVertex2d(-0.9, -0.8); // 頂点位置の指定 (1つめ)
11     glVertex2d( 0.9, -0.8); // 頂点位置の指定 (2つめ)
12     glVertex2d( 0.0,  0.8); // 頂点位置の指定 (3つめ)
13     glEnd();
14
15     glFlush(); // 画面出力
16 }
17
18 // メインプログラム
19 int main (int argc, char *argv[]) {
20     glutInit(&argc, argv); // ライブラリの初期化
21     glutInitWindowSize(400, 400); // ウィンドウサイズを指定
22     glutCreateWindow(argv[0]); // ウィンドウを作成
23     glutDisplayFunc(display); // 表示関数を指定
24     glutMainLoop(); // イベント待ち
25     return 0;
26 }
```

3. 三角形の描画

```
1  #include <GL/glut.h> // ライブラリ用ヘッダファイルの読み込み
2
3  // 表示部分をこの関数で記入
4  void display(void) {
5      glClearColor (1.0, 1.0, 1.0, 1.0); // 消去色指定
6      glClear (GL_COLOR_BUFFER_BIT );    // 画面消去
7
8      glColor3d(0.0, 1.0, 1.0); // 色指定 (R,G,B) で 0~1 まで
9      glBegin(GL_TRIANGLES);      // 描画するものを指定
```

```

10     glVertex2d(-0.9, -0.8); // 頂点位置の指定 (1つめ)
11     glVertex2d( 0.9, -0.8); // 頂点位置の指定 (2つめ)
12     glVertex2d( 0.0,  0.8); // 頂点位置の指定 (3つめ)
13     glEnd();
14
15     glFlush(); // 画面出力
16 }
17
18 // メインプログラム
19 int main (int argc, char *argv[]) {
20     glutInit(&argc, argv); // ライブラリの初期化
21     glutInitWindowSize(400, 400); // ウィンドウサイズを指定
22     glutCreateWindow(argv[0]); // ウィンドウを作成
23     glutDisplayFunc(display); // 表示関数を指定
24     glutMainLoop(); // イベント待ち
25     return 0;
26 }

```

4. 複数の図形の描画

```

1  #include <GL/glut.h> // ライブラリ用ヘッダファイルの読み込み
2
3  // 表示部分をこの関数で記入
4  void display(void) {
5      glClearColor (1.0, 1.0, 1.0, 1.0); // 消去色指定
6      glClear (GL_COLOR_BUFFER_BIT); // 画面消去
7
8      // 1つ目の図形
9      glColor3d(1.0, 0.5, 0.0); // 色指定 (R,G,B) で 0~1 まで
10     glBegin(GL_QUADS); // 描画するものを指定
11         glVertex2d(-0.3, 0.0); // 頂点位置の指定 (1つめ)
12         glVertex2d(-0.3, -0.4); // 頂点位置の指定 (2つめ)
13         glVertex2d( 0.3, -0.4); // 頂点位置の指定 (3つめ)
14         glVertex2d( 0.3, 0.0); // 頂点位置の指定 (4つめ)
15     glEnd();
16
17     // 2つ目の図形
18     glColor3d(0.0, 1.0, 0.0); // 色指定 (R,G,B) で 0~1 まで
19     glBegin(GL_TRIANGLES); // 描画するものを指定
20         glVertex2d( 0.0, 0.9); // 頂点位置の指定 (1つめ)
21         glVertex2d(-0.7, 0.0); // 頂点位置の指定 (2つめ)
22         glVertex2d( 0.7, 0.0); // 頂点位置の指定 (3つめ)

```

```

23     glEnd();
24
25     glFlush(); // 画面出力
26 }
27
28 // メインプログラム
29 int main (int argc, char *argv[]) {
30     glutInit(&argc, argv);           // ライブラリの初期化
31     glutInitWindowSize(400 , 400);   // ウィンドウサイズを指定
32     glutCreateWindow(argv[0]);       // ウィンドウを作成
33     glutDisplayFunc(display);        // 表示関数を指定
34     glutMainLoop();                 // イベント待ち
35     return 0;
36 }

```

5. ループ処理を用いた描画

```

1  #include <GL/glut.h> // ライブラリ用ヘッダファイルの読み込み
2
3  // 表示部分をこの関数で記入
4  void display(void) {
5      glClearColor (1.0, 1.0, 1.0, 1.0); // 消去色指定
6      glClear (GL_COLOR_BUFFER_BIT );    // 画面消去
7
8      glColor3d(0.0, 0.0, 0.0); // 色指定 (R,G,B) で 0~1 まで
9      glBegin(GL_LINES);
10     for(int i = 0; i < 10; i++) {
11         glVertex2d(i * 0.1 - 0.5, 0.5);
12         glVertex2d(i * 0.1 - 0.5, -0.5);
13     }
14     glEnd();
15
16     glFlush(); // 画面出力
17 }
18
19 // メインプログラム
20 int main (int argc, char *argv[]) {
21     glutInit(&argc, argv);           // ライブラリの初期化
22     glutInitWindowSize(400 , 400);   // ウィンドウサイズを指定
23     glutCreateWindow(argv[0]);       // ウィンドウを作成
24     glutDisplayFunc(display);        // 表示関数を指定
25     glutMainLoop();                 // イベント待ち

```



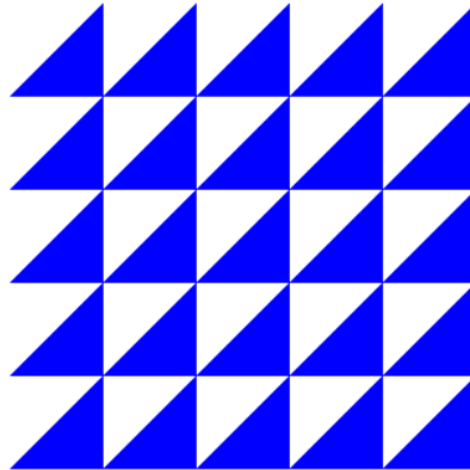
```
26     return 0;
27 }
```

6. 円の描画

```
1  #include <GL/glut.h> // ライブラリ用ヘッダファイルの読み込み
2  #include <math.h>
3
4  // 表示部分をこの関数で記入
5  void display(void) {
6      glClearColor (1.0, 1.0, 1.0, 1.0); // 消去色指定
7      glClear (GL_COLOR_BUFFER_BIT );    // 画面消去
8
9      glColor3d(0.0, 0.0, 0.0); // 色指定 (R,G,B) で 0~1 まで
10     glBegin(GL_LINE_LOOP);
11     for(int i = 0; i < 360; i++) {
12         double x = cos(i * 3.14159 /180.0);
13         double y = sin(i * 3.14159 /180.0);
14         glVertex2d(x * 0.6, y * 0.6);
15     }
16     glEnd();
17
18     glFlush(); // 画面出力
19 }
20
21 // メインプログラム
22 int main (int argc, char *argv[]) {
23     glutInit(&argc, argv); // ライブラリの初期化
24     glutInitWindowSize(400 , 400); // ウィンドウサイズを指定
25     glutCreateWindow(argv[0]); // ウィンドウを作成
26     glutDisplayFunc(display); // 表示関数を指定
27     glutMainLoop(); // イベント待ち
28     return 0;
29 }
```

3 課題 2

下図のように、画面内に縦に 5 つ、横に 5 つ、合計 25 個の三角形を表示するプログラムを作成しなさい。三角形の大きさ、色、配置の間隔は自由に決めてよい。



3.1 コード

コードから分かるように、三角形を 5 つ横に並べたものを次に縦に 5 行並べていくことで課題を解決している。

```
1  #include <GL/glut.h> // ライブラリ用ヘッダファイルの読み込み
2
3  // 表示部分をこの関数で記入
4  void display(void) {
5      float x = -0.6;
6      float y = 0.6;
7      int xcount = 0;
8      int ycount = 0;
9
10     glClearColor (1.0, 1.0, 1.0, 1.0); // 消去色指定
11     glClear (GL_COLOR_BUFFER_BIT );    // 画面消去
12
13     // 三角形 5 つの列を 5 行出力していく
14     while(ycount < 5) { // 行を出力
```

```

15     while (xcount < 5) { // 列を出力
16         glColor3d(0.0, 0.0, 1.0);
17         glBegin(GL_TRIANGLES);
18         glVertex2d( x, y + 0.3); // 頂点位置の指定 (1つめ)
19         glVertex2d( x - 0.3, y); // 頂点位置の指定 (2つめ)
20         glVertex2d( x, y); // 頂点位置の指定 (3つめ)
21         glEnd();
22         x += 0.3;
23         xcount++;
24     }
25     xcount = 0;
26     ycount++;
27     x = -0.6;
28     y -= 0.3;
29 }
30 glFlush(); // 画面出力
31 }
32
33 // メインプログラム
34 int main (int argc, char *argv[]) {
35     glutInit(&argc, argv); // ライブラリの初期化
36     glutInitWindowSize(400 , 400); // ウィンドウサイズを指定
37     glutCreateWindow(argv[0]); // ウィンドウを作成
38     glutDisplayFunc(display); // 表示関数を指定
39     glutMainLoop(); // イベント待ち
40     return 0;
41 }

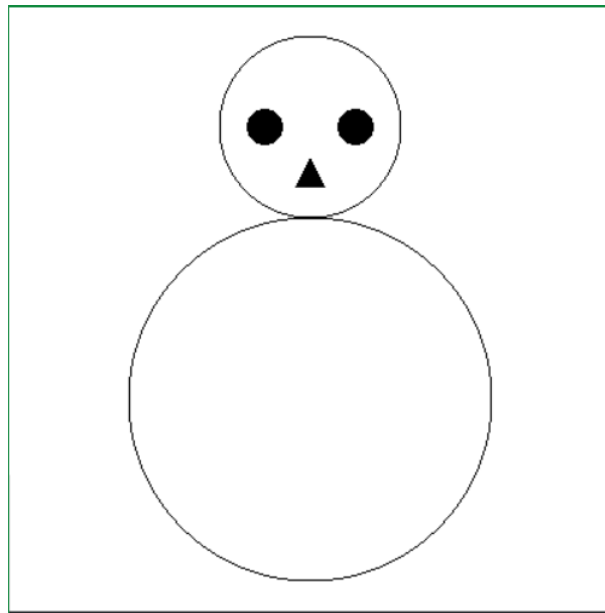
```

4 課題 3

オリジナルの 2 次元図形を画面に表示するプログラムを作成しなさい。ただし、プログラムの中では必ず一度は for ループを用いること。
二種類作成したため、それぞれを紹介する。

4.1 円とその塗りつぶしを用いた簡単な絵の描画

頭と胴と目と口を描画している。



4.1.1 コード

```
1  #include <GL/glut.h> // ライブラリ用ヘッダファイルの読み込み
2  #include <math.h>
3
4  // 表示部分をこの関数で記入
5  void display(void) {
6      glClearColor (1.0, 1.0, 1.0, 1.0); // 消去色指定
7      glClear (GL_COLOR_BUFFER_BIT );      // 画面消去
8
9      // 胴の描画
10     glColor3d(0.0, 0.0, 0.0);
11     glBegin(GL_LINE_LOOP);
12     for(int i = 0; i < 360; i++) {
```

```

13     double x = cos(i * 3.14159 /180.0);
14     double y = sin(i * 3.14159 /180.0) - 0.5;
15     glVertex2d(x * 0.6, y * 0.6);
16 }
17 glEnd();
18
19 // 頭の描画
20 glColor3d(0.0, 0.0, 0.0);
21 glBegin(GL_LINE_LOOP);
22 for(int i = 0; i < 360; i++) {
23     double x = cos(i * 3.14159 /180.0) * 0.5;
24     double y = (sin(i * 3.14159 /180.0) + 2.0) * 0.5;
25     glVertex2d(x * 0.6, y * 0.6);
26 }
27 glEnd();
28
29 // 目の描画
30 glColor3d(0.0, 0.0, 0.0);
31 glBegin(GL_POLYGON);
32 for(int i = 0; i < 360; i++) {
33     double x = cos(i * 3.14159 /180.0) * 0.1 + 0.25;
34     double y = sin(i * 3.14159 /180.0) * 0.1 + 1.0;
35     glVertex2d(x * 0.6, y * 0.6);
36 }
37 glEnd();
38
39 // 目の描画
40 glColor3d(0.0, 0.0, 0.0);
41 glBegin(GL_POLYGON);
42 for(int i = 0; i < 360; i++) {
43     double x = cos(i * 3.14159 /180.0) * 0.1 - 0.25;
44     double y = sin(i * 3.14159 /180.0) * 0.1 + 1.0;
45     glVertex2d(x * 0.6, y * 0.6);
46 }
47 glEnd();
48
49 // 口の描画
50 glColor3d(0.0, 0.0, 0.0);
51 glBegin(GL_POLYGON);
52 glVertex2d( 0.0, 0.5);
53 glVertex2d( -0.05, 0.4);
54 glVertex2d( 0.05, 0.4);
55 glEnd();
56

```

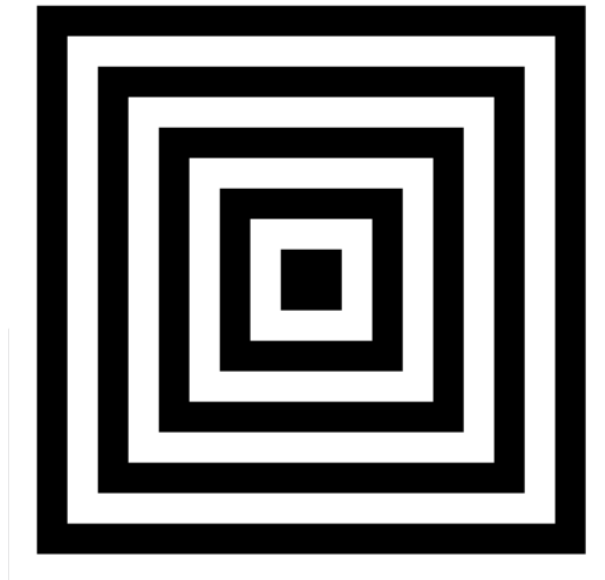
```

57     glFlush(); // 画面出力
58 }
59
60 // メインプログラム
61 int main (int argc, char *argv[]) {
62     glutInit(&argc, argv);           // ライブラリの初期化
63     glutInitWindowSize(400 , 400);   // ウィンドウサイズを指定
64     glutCreateWindow(argv[0]);        // ウィンドウを作成
65     glutDisplayFunc(display);         // 表示関数を指定
66     glutMainLoop();                  // イベント待ち
67     return 0;
68 }

```

4.2 サイズの異なる正方形を重ね合わせた図の描画

広い正方形から色を変えた少し小さな正方形を重ね合わせていく。



4.2.1 コード

```

1  #include <GL/glut.h> // ライブラリ用ヘッダファイルの読み込み
2  #include <math.h>
3
4  // 表示部分をこの関数で記入
5  void display(void) {
6      float x = 1.0;

```

```

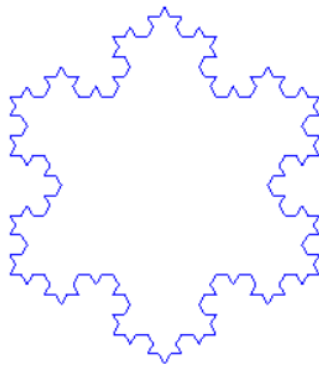
7   float y = 1.0;
8   float color = 1.0;
9
10  glClearColor (1.0, 1.0, 1.0, 1.0); // 消去色指定
11  glClear (GL_COLOR_BUFFER_BIT );    // 画面消去
12
13  for (int i = 0; i < 10; ++i) {
14      // 塗りつぶす色を交互に入れ替える
15      if (i % 2 == 0) {
16          glColor3d(color, color, color);
17      }
18      else {
19          glColor3d(color * 0, color * 0, color * 0);
20      }
21      // 塗り潰す四角形の描画
22      glBegin(GL_QUADS);
23      glVertex2d(x, y);
24      glVertex2d(-1 * x , y);
25      glVertex2d(-1 * x , -1 * y);
26      glVertex2d(x , -1 * y);
27      glEnd();
28      // 正方形のサイズを小さくする。
29      x -= 0.1;
30      y -= 0.1;
31  }
32
33  glFlush(); // 画面出力
34  }
35
36  // メインプログラム
37  int main (int argc, char *argv[]) {
38      glutInit(&argc, argv);           // ライブラリの初期化
39      glutInitWindowSize(400 , 400);   // ウィンドウサイズを指定
40      glutCreateWindow(argv[0]);        // ウィンドウを作成
41      glutDisplayFunc(display);         // 表示関数を指定
42      glutMainLoop();                  // イベント待ち
43      return 0;
44  }

```

5 発展課題

「コッホ曲線」について調べ、ステップ3の時点の図形(右図)を描画するプログラムを作成しなさい。

コッホ曲線の定義をコードにして、繰り返し回数を3にして計算を行い、その結果を線でつないだ。



5.1 コード

```
1  #include <GL/glut.h>
2  #include <math.h>
3
4  // 表示部分をこの関数で記入
5
6  void koch(int level, float p1x, float p1y, float p2x, float p2y) {
7      if(level == 0) {
8          glColor3d(0.0, 0.0, 1.0);
9          glBegin(GL_LINE_LOOP);
10         glVertex2d(p1x, p1y);
11         glVertex2d(p2x, p2y);
12         glEnd();
13         return;
14     }
15     float sx = (2.0 * p1x + 1.0 * p2x) / 3.0;
16     float sy = (2.0 * p1y + 1.0 * p2y) / 3.0;
```



```

17  float tx = (1.0 * p1x + 2.0 * p2x) / 3.0;
18  float ty = (1.0 * p1y + 2.0 * p2y) / 3.0;
19  float ux = (tx - sx) * (1.0 / 2.0) - (ty - sy) * (sqrt(3) / 2.0) + sx;
20  float uy = (tx - sx) * (sqrt(3.0) / 2.0) + (ty - sy) * (1.0 / 2.0) + sy;
21
22  koch(level - 1, p1x, p1y, sx, sy);
23  koch(level - 1, sx, sy, ux, uy);
24  koch(level - 1, ux, uy, tx, ty);
25  koch(level - 1, tx, ty, p2x, p2y);
26 }
27 void display(void) {
28     glClearColor(1.0, 1.0, 1.0, 1.0);
29     glClear (GL_COLOR_BUFFER_BIT );    // 画面消去
30     koch(3, 0.5 * 1.0, 0.5 * -1.0 * (sqrt(3.0) / 2.0),
31         0.5 * -1.0, 0.5 * -1.0 * (sqrt(3.0) / 2.0));
32     koch(3, -1.0 * 0.5, 0.5 * -1.0 * (sqrt(3.0) / 2.0),
33         0.0 * 0.5, (sqrt(3.0) / 2.0) * 0.5);
34     koch(3, 0.0 * 0.5, 0.5 * (sqrt(3.0) / 2.0),
35         0.5 * 1.0, 0.5 * -1.0 * (sqrt(3.0) / 2.0));
36
37     glFlush(); // 画面出力
38 }
39
40 // メインプログラム
41 int main (int argc, char *argv[]) {
42     glutInit(&argc, argv);    // ライブラリの初期化
43     glutInitWindowSize(400 , 400); // ウィンドウサイズを指定
44     glutCreateWindow(argv[0]); // ウィンドウを作成
45     glutDisplayFunc(display);  // 表示関数を指定
46     glutMainLoop();           // イベント待ち
47     return 0;
48 }

```

6 感想

特に課題に関しての感想はないが、開発環境などの構築に戸惑っている生徒が非常に多かったため、なんらかの救済策があればよいと思いました。