

第7章 OpenGL プログラミング

この章では、関数 `opengl.ft` を使用しないでつくった OpenGL プログラムの例を実行する。OpenGL を C 言語で使いやすく定義した関数 `opengl.ft` の構成を示し、OpenGL と `opengl.ft` を比較しながら、`opengl.ft` の考え方を示す。そして、より深く直接 OpenGL プログラミングを勉強する諸君の次へのステップとしたい。

7.1 直線を描く

OpenGL の関数だけを使って、直線を描くプログラムをつくる。

◆ プログラム

line0.c

```
1  /* Opendl の関数だけを使った線の描画 */
2  #include <stdio.h>
3  #include <windows.h>
4  #include <GL/gl.h>
5  #include <GL/glut.h>
6
7  void draw_line(void)
8  {
9      glClear(GL_COLOR_BUFFER_BIT);
10     glColor3d(1.0,1.0,1.0);
11     glBegin(GL_LINES);
12         glVertex2d(0.0,0.0);
13         glVertex2d(100.0,100.0);
14     glEnd();
15     glFlush();
16 }
17
18 void ginit(void)
19 {
20     glClearColor(0.0,0.0,0.0,0.0);
21     glMatrixMode(GL_PROJECTION);
22     glLoadIdentity();
```

```

23         glOrtho(-256.0,256.0,-256.0,256.0,-1.0,1.0);
24     }
25
26     int main(int argc,char **argv)
27     {
28         glutInit(&argc,argv);
29         glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
30         glutInitWindowSize(512,512);
31         glutInitWindowPosition(280,0);
32         glutCreateWindow("Opengl_Window");
33         ginit();
34         glutDisplayFunc(draw_line);
35         glutMainLoop();
36         return 0;
37     }

```

◆ プログラムの解説

本書で一番最初に実行したサンプルプログラム“白い斜めの直線を 1 本描く”と比較しながら読んでほしい。

2 行：標準入出力を定義したヘッダファイル `stdio.h` を取り込む。

3 行：Windows98/NT のためのヘッダファイル `windows.h` を取り込む。

4 行：Visual C++コンパイラのフォルダ GLにある OpenGLのヘッダファイル `gl.h` を取り込む。

5 行：環境の設定で述べた方法で、Visual C++コンパイラのフォルダ GL に格納した Glut (OpenGL Utility Toolkit) ライブラリに関するヘッダファイル `glut.h` を取り込む。

7~16 行：白い斜めの線を 1 本描く関数 `draw_line()` を定義する。

9 行：描画バッファをクリアする。

10 行：描画する絵の色を白色に設定する。

11 行：OpenGL では、点や線、多角形を描く場合、`glBegin(引数)~glEnd()` の構文を使い、～の部分に座標点（頂点）を定義する。`glBegin()` の引数に、点なら `GL_POINTS`、線なら `GL_LINES`、三角形なら `GL_TRIANGLES`、四角形なら `GL_QUADS` などの図形の種類を指定する。

12 行：`glVertex2d()` は、`()` の中に点の座標を指定する。この例では、`(0.0,0.0)` を線の始点にする。

13 行：点の座標を指定する。この例では、`(100.0,100.0)` を線の終点にする。

14 行：`glEnd()` は、点の座標の記述を終了する。`()` の中に引数は何も指定しない。

15 行：バッファの内容をすべて表示する。

18～24 行：描画バッファの視界などの設定を定義する。

20 行：バッファをクリアする。

21 行：変換行列を射影モードにする。

22 行：変換行列に単位行列を設定する。

23 行：glOrtho()は、正射影変換を指定する。

glOrtho(-256.0, 256.0, -256.0, 256.0, -1.0, 1.0)の例では、投影面を四角形で表現し、最初の引数=-256.0 が四角形の左端、2 番目の引数=256.0 が右端、3 番目の引数=-256.0 が四角形の下、4 番目の引数=256.0 が四角形の上を示す。5 番目の引数=-1.0 が手前を示し、6 番目の引数=1.0 が奥行きを示す。一般的に、glOrtho(left,right,bottom,top,near,far)で表現すると、手前のクリップ面が (left,bottom,-near) を左下隅、(right,top,-near) を右下隅とする四角形、奥のクリップ面が (left,bottom,-far) と(right,top,near)の四角形である。

26～37 行：main 関数を定義する。

ここでは、5つの Glut 関数が描画面面の初期化を行い、2つの Glut 関数がイベントを実行している。

28 行：Glut を初期化し、コマンドラインの引数を処理する。

29 行：描画面面は RGB 1 面（バッファはシングルの RGB 面）にする。

30 行：画面サイズを 512×512 にする。

31 行：画面上にウィンドウを表示する位置の左上の座標 (x,y) を指定する。

32 行：描画面面のタイトルバーに名前を付ける。

34～35 行：実行する。

◆ プログラムの実行

実行例を図 7.1 に示す。サンプルプログラムと異なる点の一つだけある。それは、ウィンドウのタイトルバーの名前が"OpenGL"から"Opengl_Window"に変わっているところである。

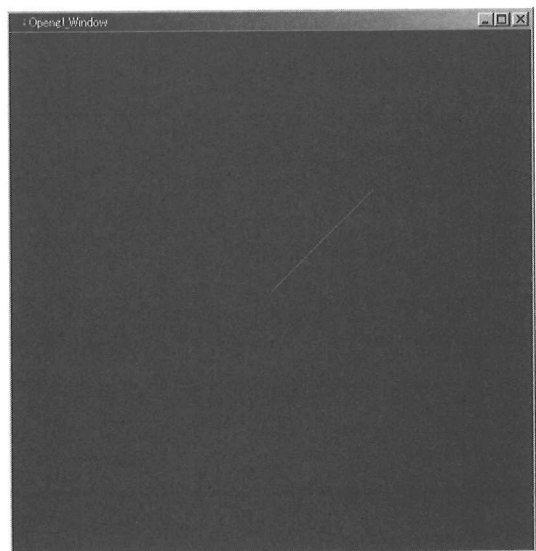


図 7.1 line0.exe の実行画面

● OpenGL と opengl.ft

プログラム line0.c を例にして、OpenGL と opengl.ft の関係を表 7.1 にまとめて示す。

表 7.1 OpenGL と opengl.ft

項 目	opengl.ft による表現	OpenGL による表現
ヘッダファイルの 取り込み	#include<opengl.ft>	#include <windows.h> #include <GL/gl.h> #include <GL/glut.h>
バッファのクリア	gclearbuffer();	glClear(GL_COLOR_BUFFER_BIT);
色の指定	glcolor3d(r,g,b);	glColor3d(r,g,b);
2 点間に直線を描く	gline(x1,y1,x2,y2);	glBegin(GL_LINES); glVertex2d(x1,y1); glVertex2d(x2,y2); glEnd();
バッファの表示	glflush();	glFlush();
平行投影法による 視界の設定	ginit(1);	glClearColor(0.0,0.0,0.0,0.0); glMatrixMode(GL_PROJECTION); glLoadIdentity(); glOrtho(left,right,bottom,top,near,far);
ウィンドウサイズ	window_size(256.0);	glutInitWindowSize(512,512);
イベントの実行	glutdisplayfunc(*func);	glutDisplayFunc(*func);
実行開始	glutmainloop();	glutMainLoop();

表 7.1 は、プログラム line0.c だけを例にしてまとめたものであるが、両者を比較することで、opengl.ft の構成が分かる。C 言語プログラムを学ぶ入門者ができるだけ C 言語ライクなスタイルで OpenGL グラフィックスを利用できる配慮をした。関数 opengl.ft の全プログラムリストを解説抜きで付録に示したので、学習の参考にしてほしい。