

コンピューターグラフィックス基礎 第一回課題

情報メディア創生学類 3 年 202313625 藤川興昌

実行環境

- Ubuntu 22.04.3 LTS
- gcc version 11.4.0

課題 2

ソースコード

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

#define DIVERGENCE_CHECK_LOOP_NUM 20
#define DIVERGENCE_CHECK_THRESHOLD 2.0

#define DRAW_STEP 0.005

bool doesIncludedInMandelBlotSet(std::complex<float> c) {
    std::complex<float> z(0.0, 0.0);
    for(int i = 0; i < DIVERGENCE_CHECK_LOOP_NUM; i++) {
        if(abs(z) > DIVERGENCE_CHECK_THRESHOLD) return false;
        z = z * z + c;
    }
    return true;
}

void display(void) {
    glClearColor (1.0, 1.0, 1.0, 1.0);
    glClear (GL_COLOR_BUFFER_BIT);

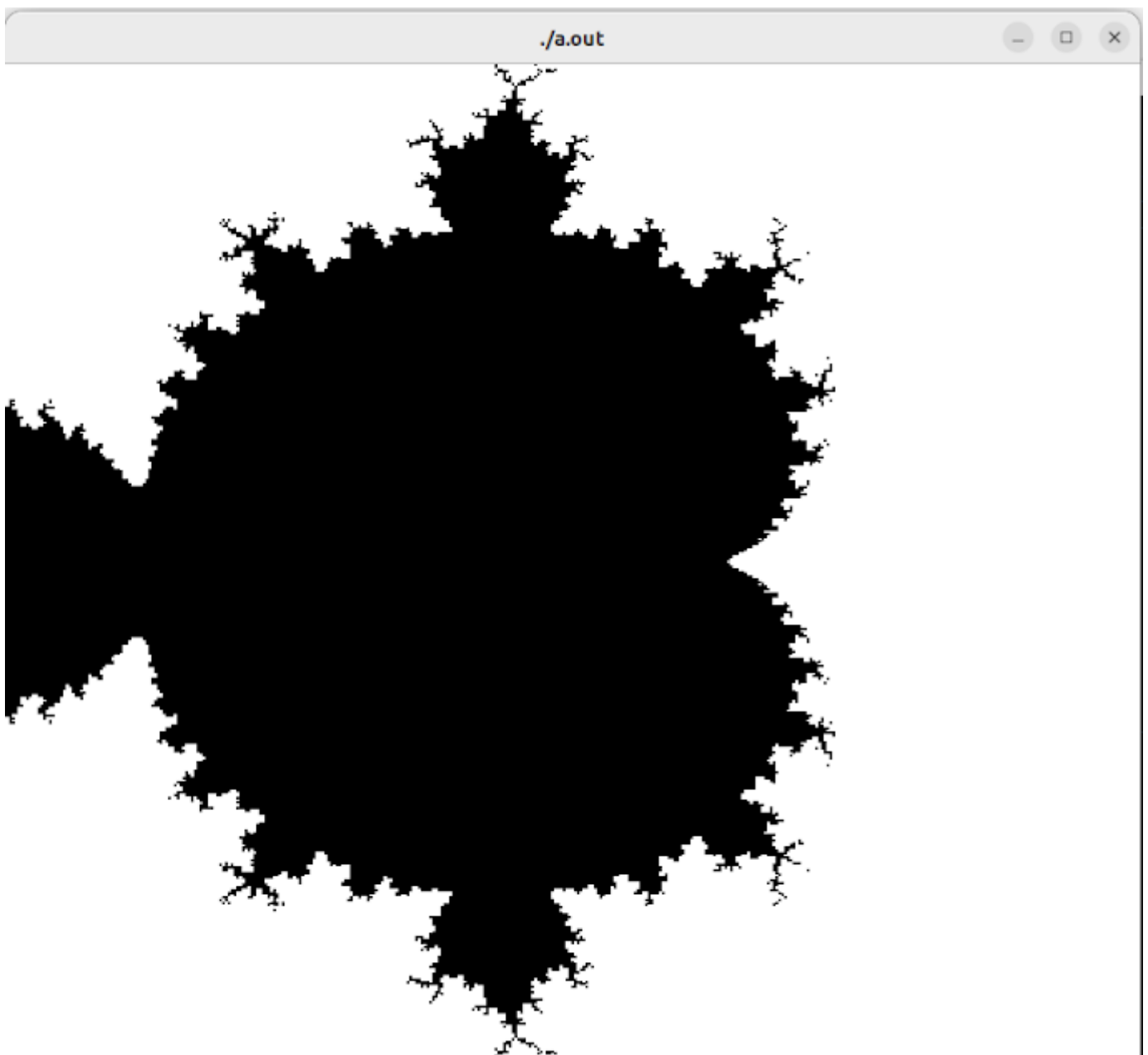
    glColor3d(0.0, 0.0, 0.0);
    glPointSize(1.5);
    glBegin(GL_POINTS);
    for(float x = -1; x < 1; x += DRAW_STEP) {
        for(float y = -1; y < 1; y += DRAW_STEP) {
```

```
        std::complex<float> c(x, y);
        if(doesIncludedInMandelBlotSet(c)) glVertex2d(x, y);
    }
}
glEnd();

glFlush();
}

int main (int argc, char *argv[]) {
    glutInit(&argc, argv);
    glutInitWindowSize(400 , 400);
    glutCreateWindow(argv[0]);
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}
```

スクリーンショット



発展課題

ソースコード

```
#include <GL/glut.h>
#include <utility>
#include <math.h>
#include <vector>
#include <stdio.h>

#define PI 3.14159265359

#define BASE_SIZE 1.0
#define RECURSION_NUM 5

std::vector<std::pair<float, float>> points;
const float PI_60_DEG = 60.0 / 180.0 * PI;
const float BASE_HEIGHT = BASE_SIZE * sin(PI_60_DEG);

void koch(std::pair<float, float> a, std::pair<float, float> b, int n) {
    if(n == 0) return;

    std::pair<float, float> s = std::make_pair(a.first + (b.first - a.first) / 3.0, a.second + (b.second - a.second) / 3.0);
    std::pair<float, float> t = std::make_pair(a.first + (b.first - a.first) * 2.0 / 3.0, a.second + (b.second - a.second) * 2.0 / 3.0);
    std::pair<float, float> u = std::make_pair(s.first + (t.first - s.first) * cos(PI_60_DEG) - (t.second - s.second) * sin(PI_60_DEG),
        s.second + (t.first - s.first) * sin(PI_60_DEG) + (t.second - s.second) * cos(PI_60_DEG));

    koch(a, s, n - 1);
    koch(s, u, n - 1);
    koch(u, t, n - 1);
    koch(t, b, n - 1);

    points.push_back(s);
    points.push_back(u);
    points.push_back(t);
}

void display(void) {
    glClearColor (1.0, 1.0, 1.0, 1.0);
    glClear (GL_COLOR_BUFFER_BIT);

    points.clear();
    std::pair<float, float> a, b, c;
    a = std::make_pair(-1.0 * BASE_SIZE / 2.0, -1.0 * BASE_HEIGHT / 2.0);
    b = std::make_pair(0.0, BASE_HEIGHT / 2.0);
```

```

c = std::make_pair(BASE_SIZE / 2.0, -1.0 * BASE_HEIGHT / 2.0);
koch(a, b, RECURSION_NUM);
koch(b, c, RECURSION_NUM);
koch(c, a, RECURSION_NUM);

glColor3d(0.0, 0.0, 0.0);
glBegin(GL_POINTS);
for(int i = 0; i < points.size(); i++) {
    glVertex2d(points[i].first, points[i].second);
}
glEnd();

glFlush();
}

int main (int argc, char *argv[]) {
    glutInit(&argc, argv);
    glutInitWindowSize(400 , 400);
    glutCreateWindow(argv[0]);
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}

```

スクリーンショット

