PROCESSING, FRACTAL SETS, AND ART

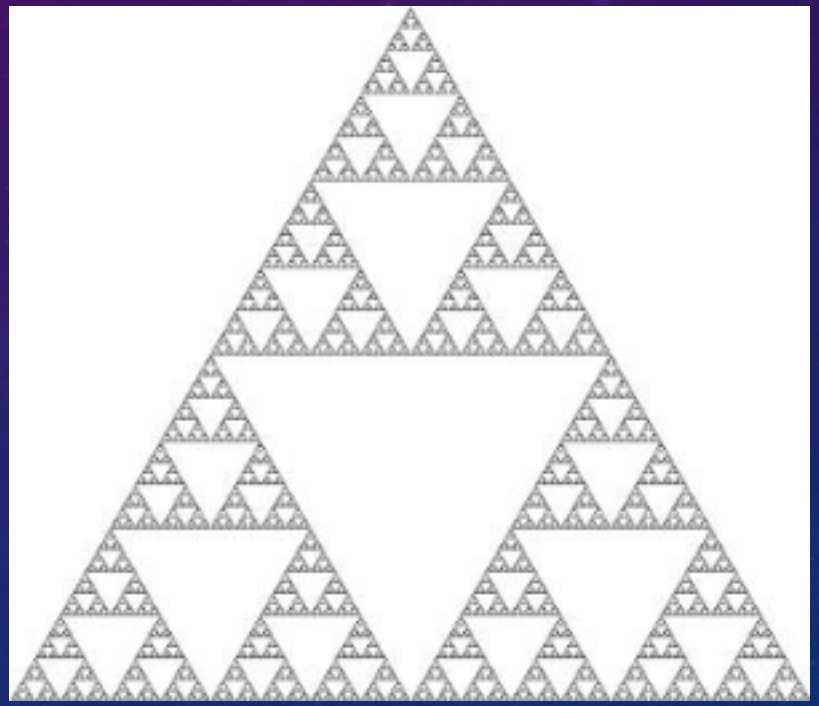
MARGIE FARGAS

ART 185GL

LEGRADY

WHAT IS A FRACTAL SET?

- Fractals are patterns that never end
- Infinitely complex patterns
- Self-similar
- Created by repeating a simple process over and over in an ongoing feedback loop

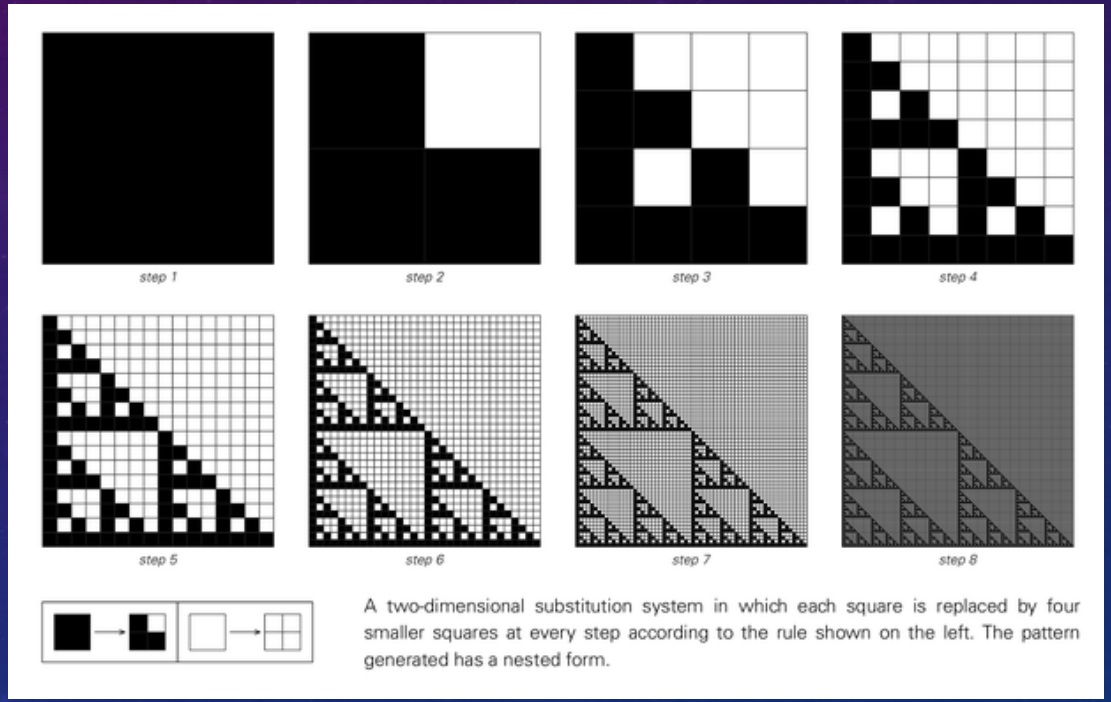


One Dimensional Substitution

- Subdivides each element into several smaller elements

Two Dimensional Substitution

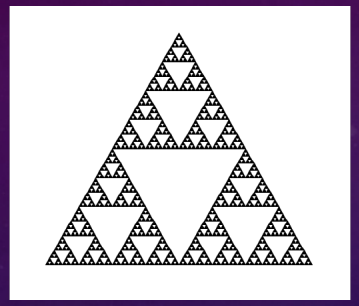
- Essentially work in the same way



Fractal Sets in Nature

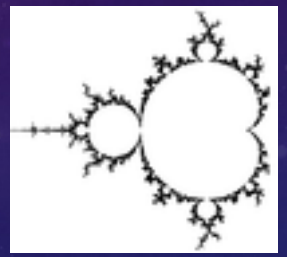


TYPES OF FRACTAL SETS

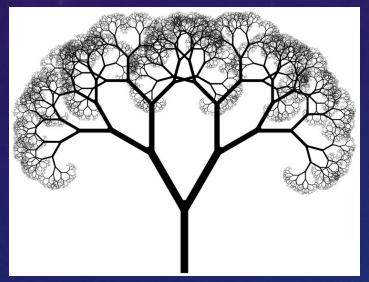


Sierpinski Triangle

Rossler Attractor



Mandelbrot Set



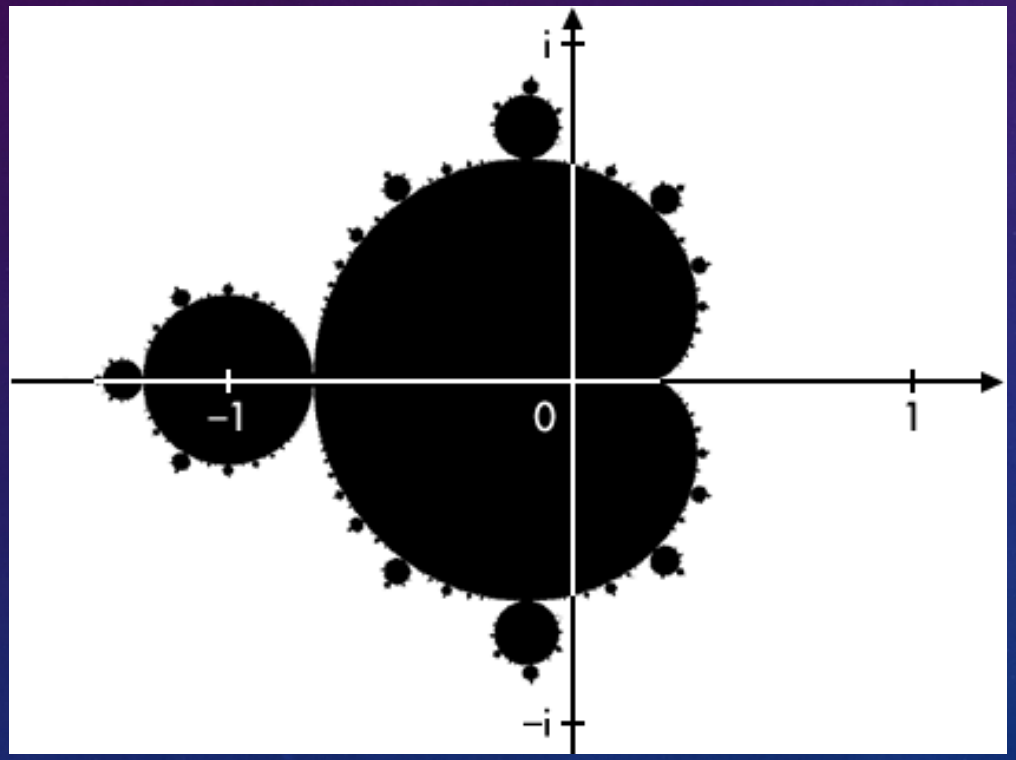
Fractal Tree

Lorenz Attractor

Julia Set

THE MANDELBROT SET AND JULIA SET

$Z_{n+1} = Z_n^2 + C$



- Points C lie in a complex plane
- A complex plane consists of a mix of "real" and imaginary numbers
  - a + bi
  - $i = \sqrt{-1}$

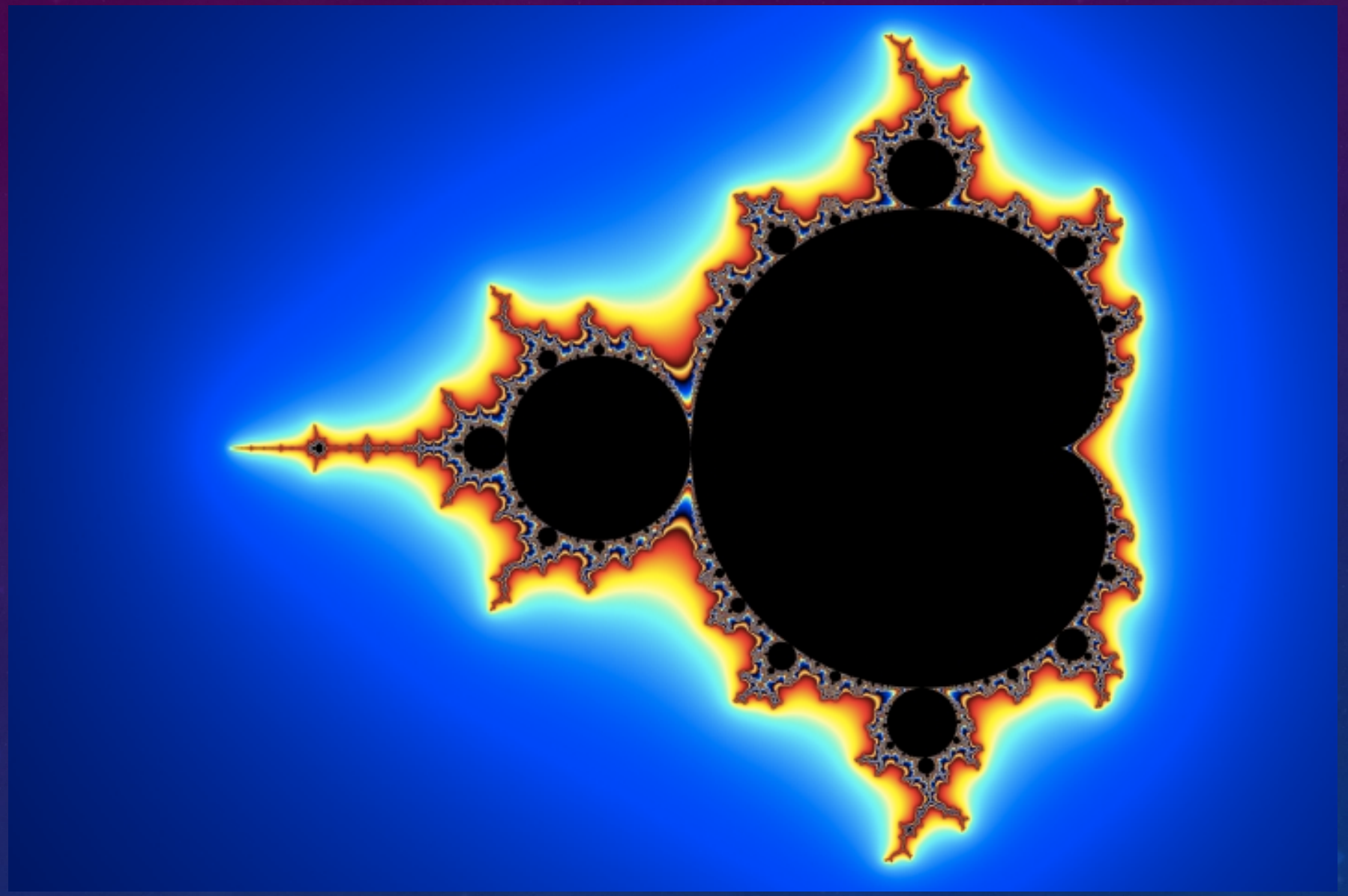
$$Z_{n+1} = Z_n^2 + C$$

$$Z_0 = C$$

$$Z_1 = C^2 + C$$

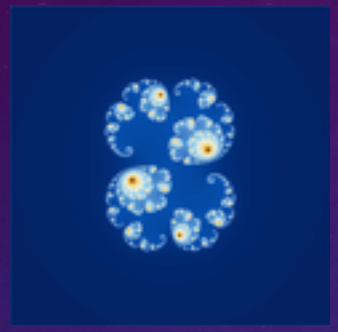
$$Z_2 = (C^2 + C)^2 + C$$

$$Z_3 = ((C^2 + C)^2 + C)^2 + C$$

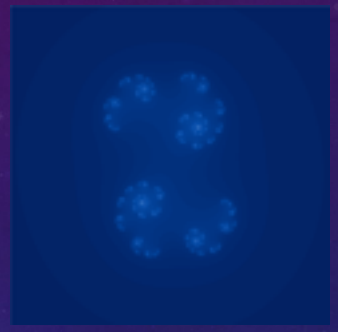


Types of Julia Sets

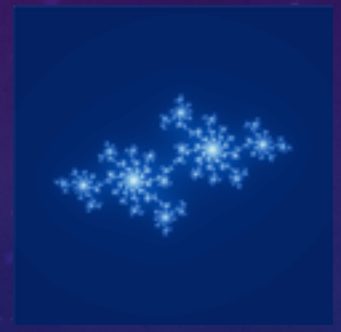
C = 0.285 + 0i



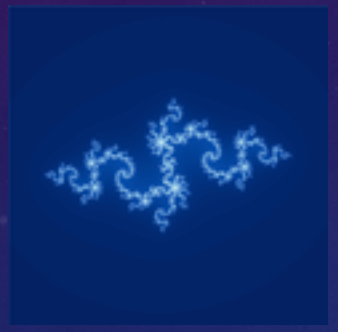
C = 0.285 + 0.01i



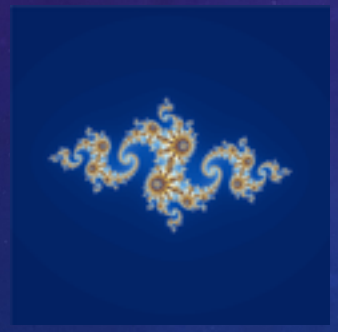
C = 0.45 + 0.1428i



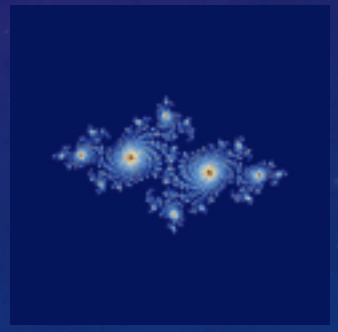
C = -0.70176 – 0.3842i



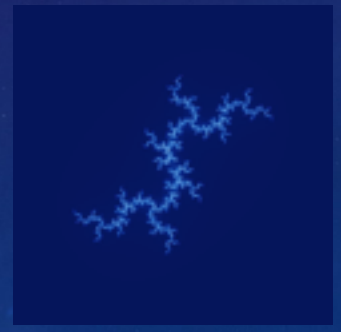
C = -0.835 – 0.2321i



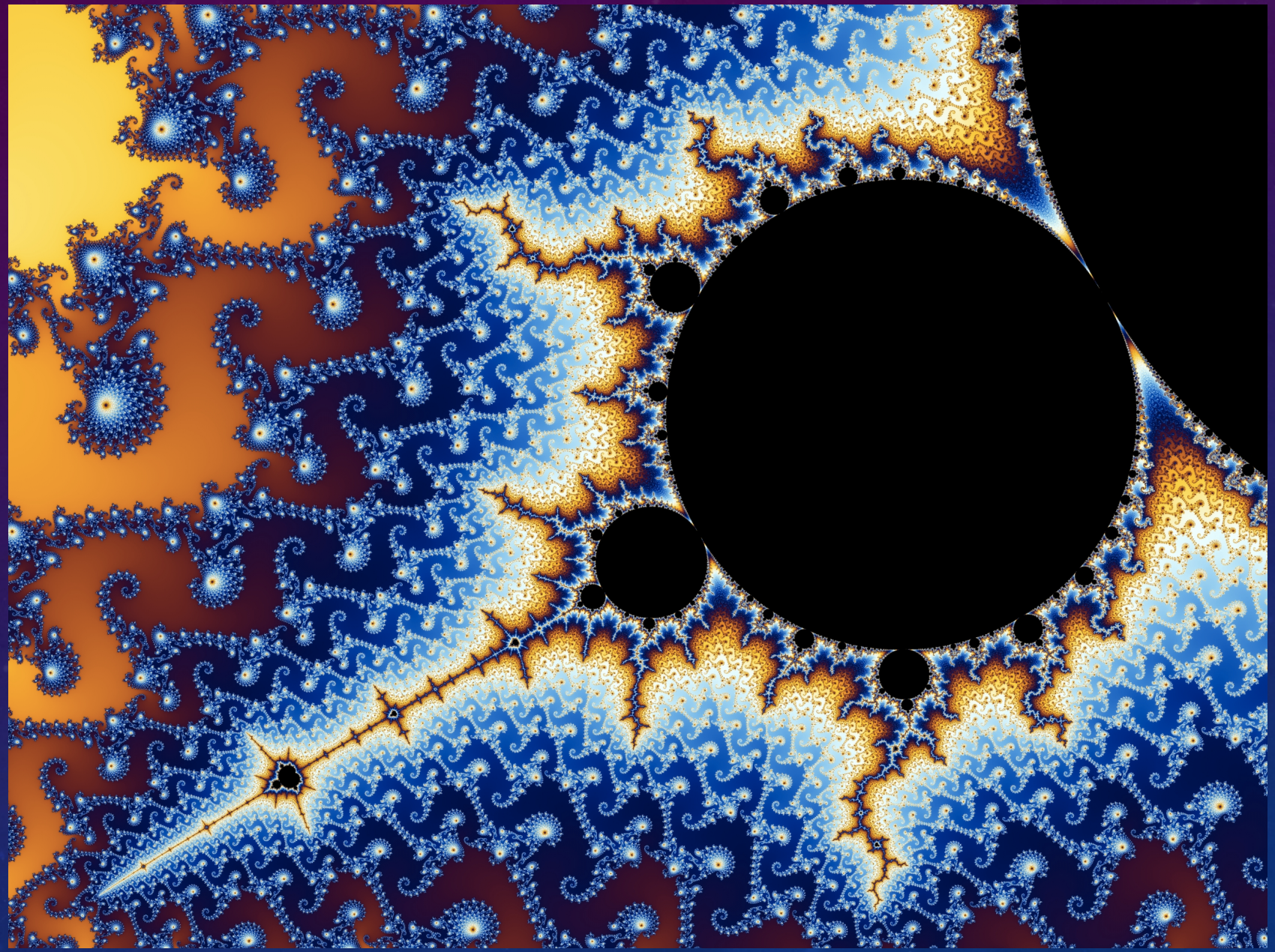
C = -0.8 + 0.156i



C= -0.7269 + 0.1889i



C = -0.8i



Mandelbrot Set in Processing

Java ▾

Mandelbrot_Set_Pwinkle

```
1  void setup() {
2    size(640,480);
3    colorMode(RGB, 1);
4  }
5
6  void draw(){
7    background(255);
8
9    float w = map(mouseX, 0, width, -4, 4) ;
10   float h = (w * height) / width;
11
12   float xmin = -w/2;
13   float ymin = -h/2;
14
15   loadPixels();
16
17   int maxiterations = 100;
18
19   float xmax = xmin + w;
20   float ymax = ymin + h;
21
22   float dx = (xmax - ymin) / (width);
23   float dy = (ymax - ymin) / (height);
24
25   float y = ymin;
26     for (int j = 0; j < height; j++) {
27   float x = xmin;
28     for (int i = 0; i < width; i++) {
29
30   float a = x;
31   float b = y;
32   int n = 0;
```

Size

```
33    while (n < maxiterations) {
34      float aa = a * a;
35      float bb = b * b;
36      float twoab = 2.0 * a * b;
37      a = aa - bb +x;
38      b = twoab + y;
39
40   if (dist(aa, aa, 0, 0) > 18.0) {
41     break;
42   }
43   n++;
44   }
45
46   if (n == maxiterations) {
47     pixels[i+j*width] = color(200);
48       } else{
49     pixels[i+j*width] = color(sqrt(float(n) / maxiterations));
50       }
51      x += dx;
52    }
53    y += dy;
54 }
55 updatePixels();
56 }
```

Convert the equation into a code

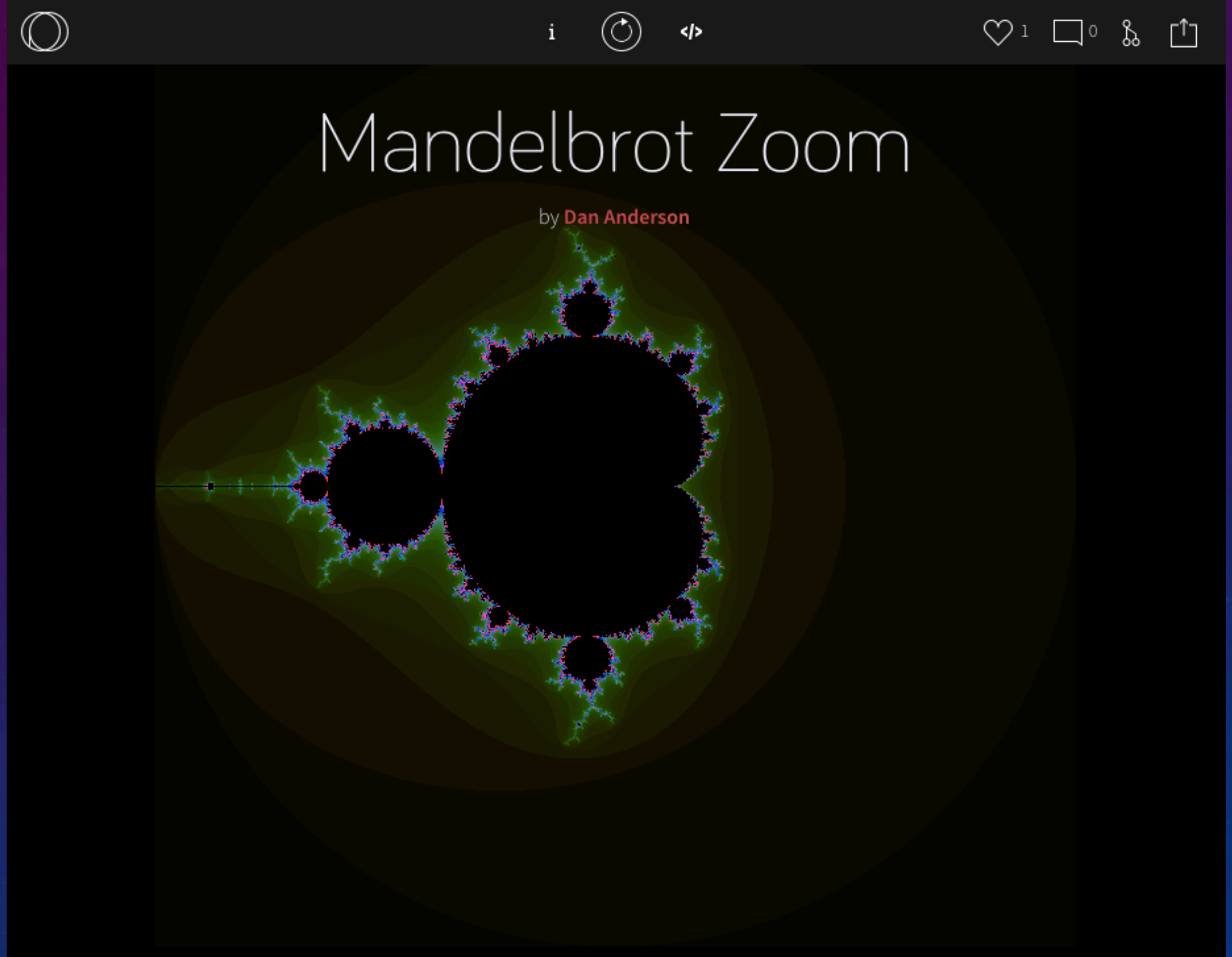
Julia Set in Processing

Julia_Set_Pwinkle

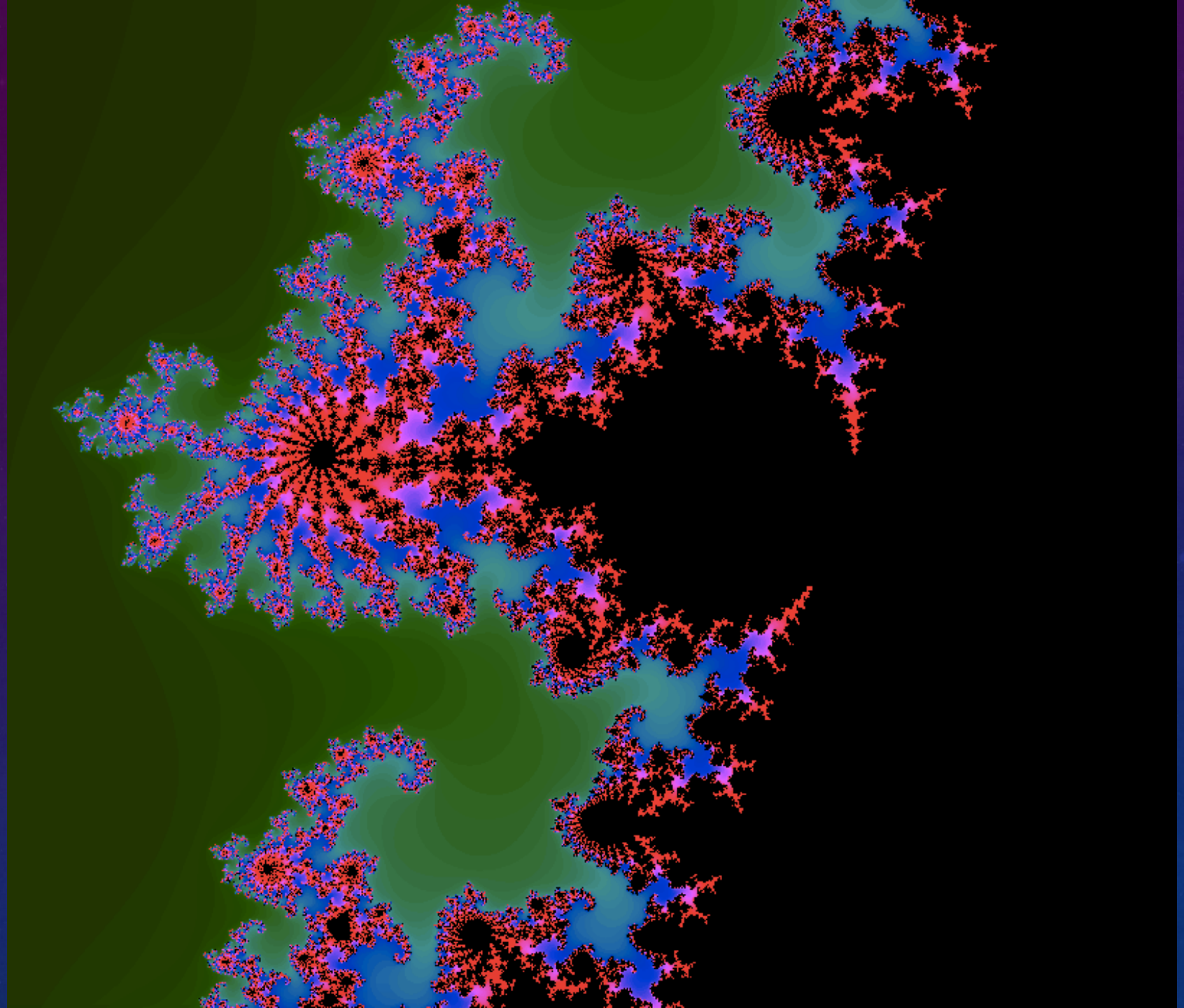
```
void setup() {
  size(640,480);
  colorMode(RGB, 1);
}

void draw(){
  float ca = - 0.8;
  float cb = 0.156;

  background(255);

  float w = map(mouseX, 0, width, -3, 3);
  float h = (w * height) / width;

  float xmin = -w/2;
  float ymin = -h/2;

  loadPixels();

  int maxiterations = 100;

  float xmax = xmin + w;
  float ymax = ymin + h;

  float dx = (xmax - ymin) / (width);
  float dy = (ymax - ymin) / (height);

  float y = ymin;
    for (int j = 0; j < height; j++) {
  float x = xmin;
    for (int i = 0; i < width; i++) {
```

Values of C

```
  float y = ymin;
    for (int j = 0; j < height; j++) {
  float x = xmin;
    for (int i = 0; i < width; i++) {

  float a = x;
  float b = y;
  int n = 0;
  while (n < maxiterations) {
    float aa = a * a;
    float bb = b * b;
    float twoab = 2.0 * a * b;
    a = aa - bb + ca;
    b = twoab + cb;

if (dist(aa, aa, 0, 0) > 18.0) {
  break;
}
n++;
}

if (n == maxiterations) {
  pixels[i+j*width] = color(200);
    } else{
  pixels[i+j*width] = color(sqrt(float(n) / maxiterations));
```



https://www.openprocessing.org/sketch/142856



Further Research

- Study how far one can push the zoom on a fractal set
- They can essentially go on infinitely
- Processing
  - The amount of iterations are high
  - This makes the process slow the more you zoom in
  - Won’t zoom in enough to show the Julia Set aspect
  - Would like to render an interactive code
  - Possibly look into 3D renditions of the fractal

References

- http://mathworld.wolfram.com/MandelbrotSet.html
- http://www.wolframscience.com/nks/p187--substitution-systems-and-fractals/
- https://www.wired.com/2010/09/fractal-patterns-in-nature/
- https://en.wikipedia.org/wiki/List_of_fractals_by_Hausdorff_dimension
- http://world.mathigon.org/Fractals
- https://processing.org/
- http://world.mathigon.org/Fractals