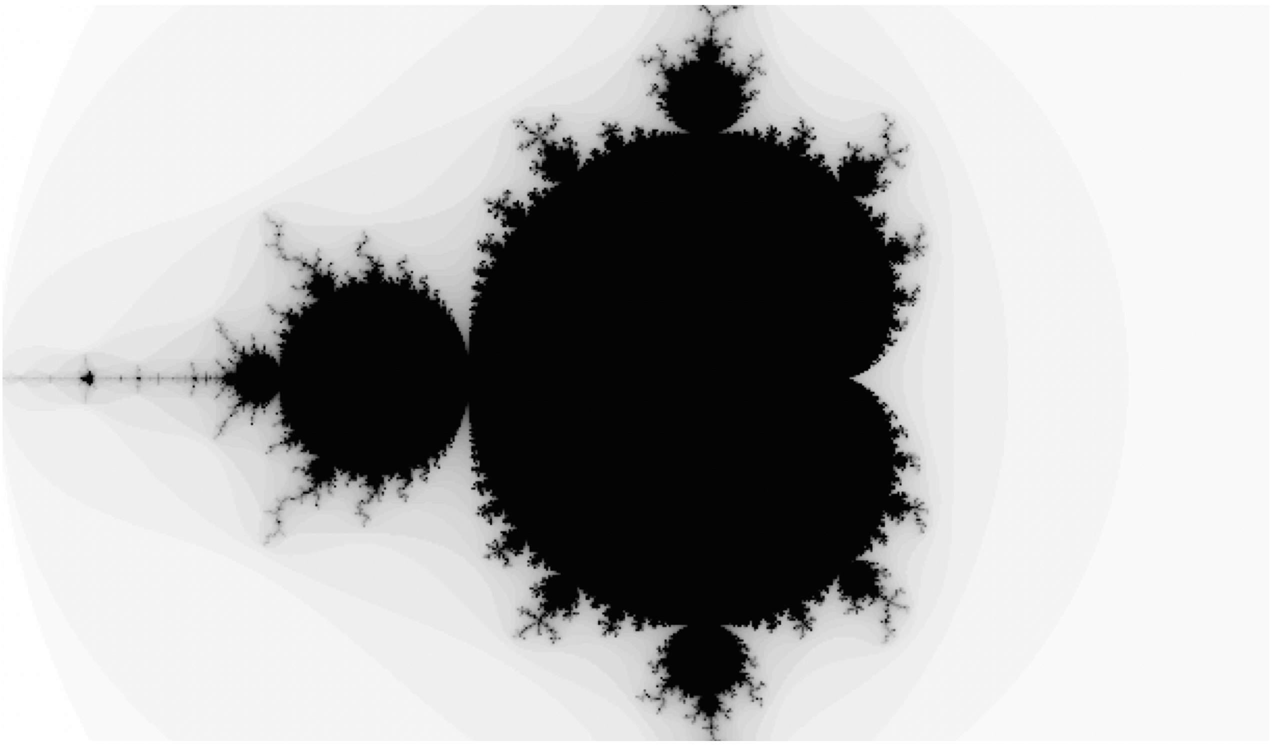
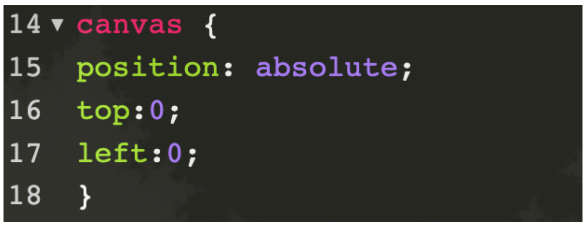


Coding One： Advanced Creative Coding

Mandelbrot set
Shaohua Yang
01/11/2022
Week 4 homework
Words: 1452



```
14 ▾ canvas {
15   position: absolute;
16   top:0;
17   left:0;
18   }
```

14 to 18
Canvas {
Position: absolute;
top: 0;
Left: 0;
{

Make sure the picture is in the right position.

—Explain:
Absolutely positioned elements are computed with a position value that is absolute. The Left, top, and properties specify the offset bottom amount of the edge of the element containing the block. (The containing block is the ancestor of the element relative to its positioning) If the element has margins, add them to the offset.

42.
var canvas = document.querySelector(“canvas”);
Context = canvas.getContext(‘2d’);

—Explain:
To start drawing on canvas the first step is to grab it with the above function and retrieve a specific context.
you grab hold of it and retrieve a particular context - either 2d or webgl:

43.44
var width = window.innerWidth;
var height = window.innerHeight;

—Explain:
InnerHeight and InnerWidth get the height and width of the form.

45.46
canvas.setAttribute("width", width);
canvas.setAttribute("height", height);

—Explain:
The length and width of the canvas are reconfigured according to the image.

The formal is represented the var width and height name.

49, 50, 51

```
var z,zx,zy=0;
var cx,cy=0;
var maxIterations=50;
```

—Explain:

Seen from visual images and MIMIC's notes, the figure is a fractal, which refers to the mathematical concept of the Mandelbrot set.

Judging by the visual image and MIMIC's remarks, the figure is a fractal, which refers to the mathematical concept of the Mandelbrot set.

The Mandelbrot set is established by f(Z) = Z^2 + c, and it occurs entirely in a world of complex numbers, a+bi (where "a" is a real number, "bi" is an imaginary number, and "i^" 2 = -1"), a+bi appears as a point in the Cartesian coordinate system, and the distance from the (0, 0) coordinate to a+bi is also expressed as la+bil.

```
function： z^2=（zx ^ 2-zy ^ 2, 2 * zx * zy），z=a+bi
```

—Explain:

We call numbers of the form z=a+bi (a and b are real numbers) complex numbers. Among them, a is called the real part, b is called the imaginary part, and i is called the imaginary unit. When the imaginary part of z is b=0, then z is a real number; when the imaginary part of z is b≠0, and the real part is a=0, z is often called a purely imaginary number. The complex field is an algebraic closure of the real field, i.e. any polynomial with complex coefficients always has a root in the complex field.

In programming, a is the x-axis of the canvas, and b is the y-axis. When Z is iterated from 0, it does not diverge, the sequence is fc(0), fc(fc(0)), etc.,), it will be calculated over and over again in the following mathematical operations.

f(Z) = Z^2 + c; c = a + b1; i^2 = -1
Z = 0
f(Z) = Z1^2 + c = c
Z = c^2 + c
c^2 = (a + bi)(a + bi)
= a^2 + abi + abi + (bi)^2
= a^2 + 2abi - b^2
= a^2 - b^2 +2a1b1

```
var z, zx, zy = 0
```

—Explain:

This variable is include absolute z and the value of x， y. Because all of the antecedents in lines 14 to 17 are absolute.

`var cx， cy = 0` the center point of the image — the block of pixel
These all things have represented the variable of the complex number C.

After determining the length and width of the canvas, set the x,y values of z and the x,y values of c, the starting value of z is (0,0) the starting value of c (0,0) pixel block, then in Line 51 sets the maximum number of iterations to 50, and sets a variable with a resolution of 2.

51.var maxIterations=50

—Explain:

This means the maximum number of iterations to 50, running 50 times.

52.var res=2

—Explain:

sets a variable with a resolution of 2.（but after tests, The higher the res value, the lower the resolution）

54. context.translate(width*0.5,height*0.5)

—Explain:

Let's shift the coordinate（0,0） to the middle. Use the function of cxt. Translate (w, h)

Next, in order to test all pixels on the screen, the nest for loop statements (check x-axis and y-axis, i=x, j=y)

56. for (var i = —(width/2); i<width/2; I+=res）{
57. for (var j = —(width/2); j<width/2; I+=res）{

—Explain:

+= is the value operator
I = -(width/2) It means the negative semi-axis area of the x-axis， operates between the -(width/2) and width/2 ranges， and the same goes for the y-axis.

58-59

cx=I/(width/4)

—Explain:

Here, width/4 refers to 25% of the canvas

cy=j/(width/4)

Then, set cx=I cy=j to adjust the value of x, y so that it is between -2 and 2, thus we get the coordinate parameters of cx,cy. Because this is where Mandelbrot is located.

60. Set the for loop statement to test the preceding code to see if it can be run, (run infinitely, 50 times a closed loop, and then do the loop).

61. Set the condition z<2.0 (because there is an absolute precedent condition on line 15, so the absolute value of z referred to here must be less than 2 to run the test (if conditional statement).

For example:

Here the test starts from 0 e.g. c=1

F(0)=0^2+1=1

F(1)=1^2+1=2

F(2)=2^2+1=5

F(5)=5^2+1=26

Because f(z)=z^2+c; c=a+bi, if we calculate z^2, the method is as follows z^2=(zx^2-zy^2,2*zx*zy)

In other words we start with zx=(zx*zx)-(zy*zy) and zy=(2*zx*zy).

67-70

var x=(zx*zx)-(zy*zy);//this is z(x)

var y=2*(zx*zy);//this is z(y)

zx=x+cx;//then we add cx to x to get the next value of zx

zy=y+cy;//and then add cy to y to get the next value of zy

67-68

refer to the position of the x and y axis points in z^2,

and 69 - 70 refer to the position of c.

The substituting zx = x+cx and zy = y+cy into the f(z) = z^2+c formula is tested with line 71 z=Math.sqrt((zx*zx)+(zy*zy)) and Get the square of the absolute value of z and the sequence so we can see if it is greater than 2 the next time we run the test.

71. z=Math.sqrt((zx*zx)+(zy*zy))

—Explain:

Calculate the square root function, Find the distance from the origin to Z.

72. var col represents the function of color, we use 255–(255/maxIterations)*test;
The test has given the loop condition and limits range in the front, so when we capture the color of the pixel, the color is automatically incremented or decremented by itself, similar to the automatic filling of the color block.

73. After that, we use the context.fillStyle equation to determine the fill color. Because clo is a variable, we need + on both sides. +col+ is to connect strings and dynamically output array subscripts.
The range of colors we use depends on how many tests we do before the value of z is greater than 2.

74. In theory, the absolute value of all pixel positions in the set can never be greater than 2.0.
Now that all the conditions are done, to make sure it automatically goes in and out every time, we need to clear the variables we are using before running the next test. Why only clear z, zx, zy? Because we only make variables for their three digits, and cx and cy are equivalent to an imaginary number of pixels, we don't need to care. We can clearly see that the values of zx and zy are the ones that change the most in the code running later.

75. create a path
76. create a rectangle path (x,y, width, height) x and y are the x and y axes of the starting point of the rectangle.

Here are x and y are i and j, and its width and height are the variables of the pixel point we set, because this variable was set to 2 before, but after the "for" function statement runs, it is constantly running. increased, so it means that how many pixels we capture are how wide and high.

77. The last step is to fill the rectangle, context.fill() is suitable for use with rect. Equivalent to ending the path.