HW 4 - Journal

Concept

First, I looked at http://wolframalpha.com/ for generative functions I could use in my project. I found the Rhodonea Curve, which I used to create a single flower. After that, I tried to make multiple flowers. The project has a circle in the middle. The circles have flowers spinning on it. The user can put any number of spinning flowers on the circle, and also change the speed, size, and the number of petals of the flowers.

Moreover, there is a random switch. If you change the random variable to true, it will generate a random set of flowers each time the code is run.

Generative rules

Using flower shapes, spin them on a circle.

Parameters:

- Radius of the circle
- Size of the flower
- Number of petals
- Number of flowers
- Speed of rotation

Relationship of the parameters

Number of petals & Radius of circle: The flowers are placed so that no matter how much flowers there are it will always be on the circle.

The other parameters work independent of each other, giving the user full control of the artwork. I think this approach is very much like the one-of-one technique.

The Rhodonea Curve

$$x = rcos(kt)cos(t)$$

$$y = rcos(kt)sin(t)$$

- k - number of petals.

- r - size of the flower This is the basic flower formula. I have implemented it in the drawFlower() function.

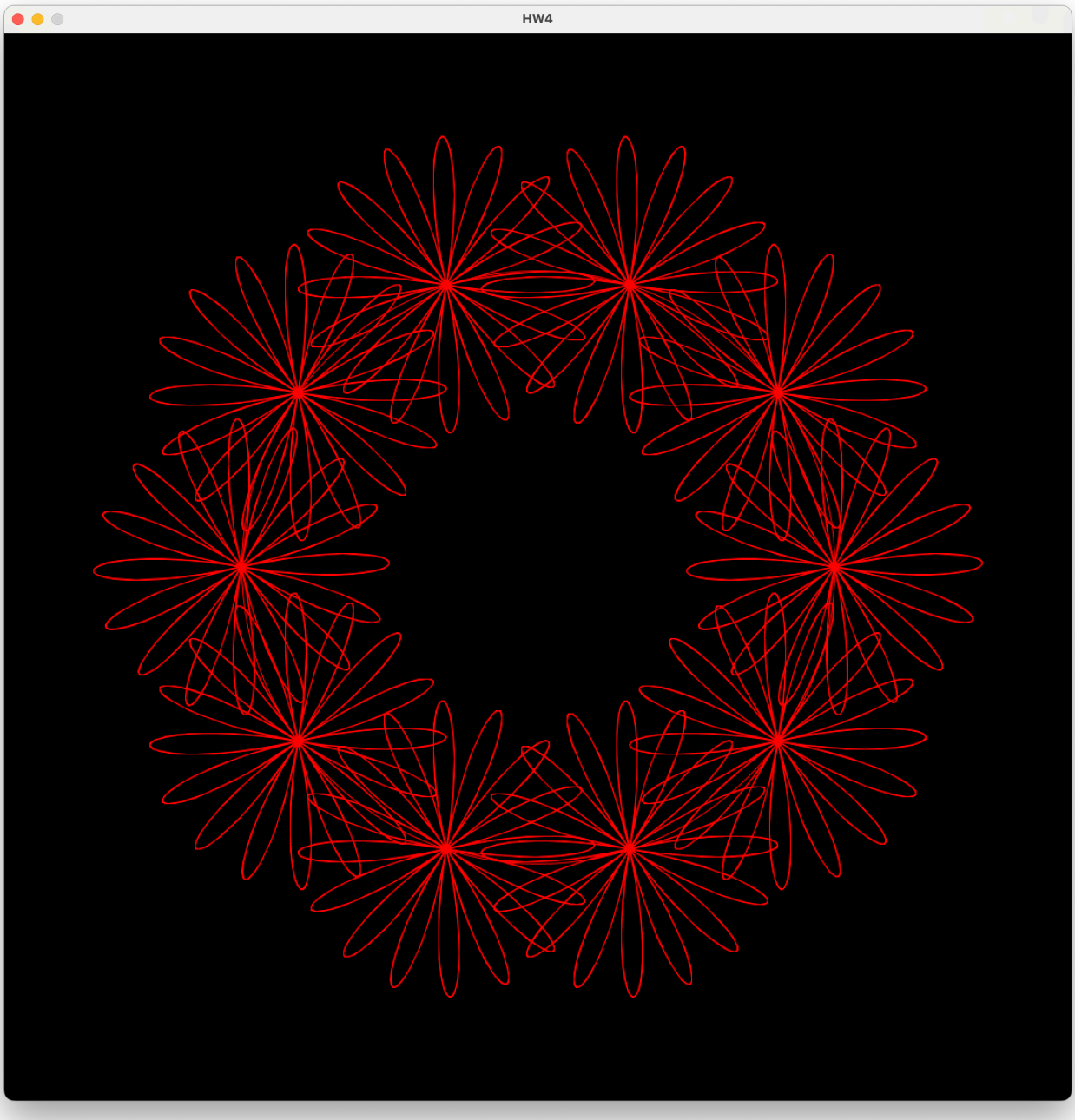
Source code

```
boolean random=false; //make this true to get a random artwork everytime the
code is run

float k = 8; // Number of petals if odd, 2k petals if even
float scale = 250; // flower scale
int num_flowers = 5; //number of flowers
float radius = 64; // radius of sphere
float angle = 0.0;
float aOff = 0.05; // rotation speed

float x,y;

void setup() {
  size(1080, 1080);
  background(0);
  stroke(145, 140, 200);
  noFill();
  smooth(8);
  frameRate(60);
  if (random){
    k = random(15);
    scale = random(200) + 100;
    num_flowers = (int) random(20);
    radius = random(100);
    aOff = random(1);
    stroke(random(255), random(255), random(255));
  }
}

void drawFlower(float xPos, float yPos, float k, float scale, float a) {
  pushMatrix();
  translate(xPos, yPos);
  rotate(a);
  beginShape();
  for (float t = 0; t < TWO_PI * 5; t += 0.02) {
```

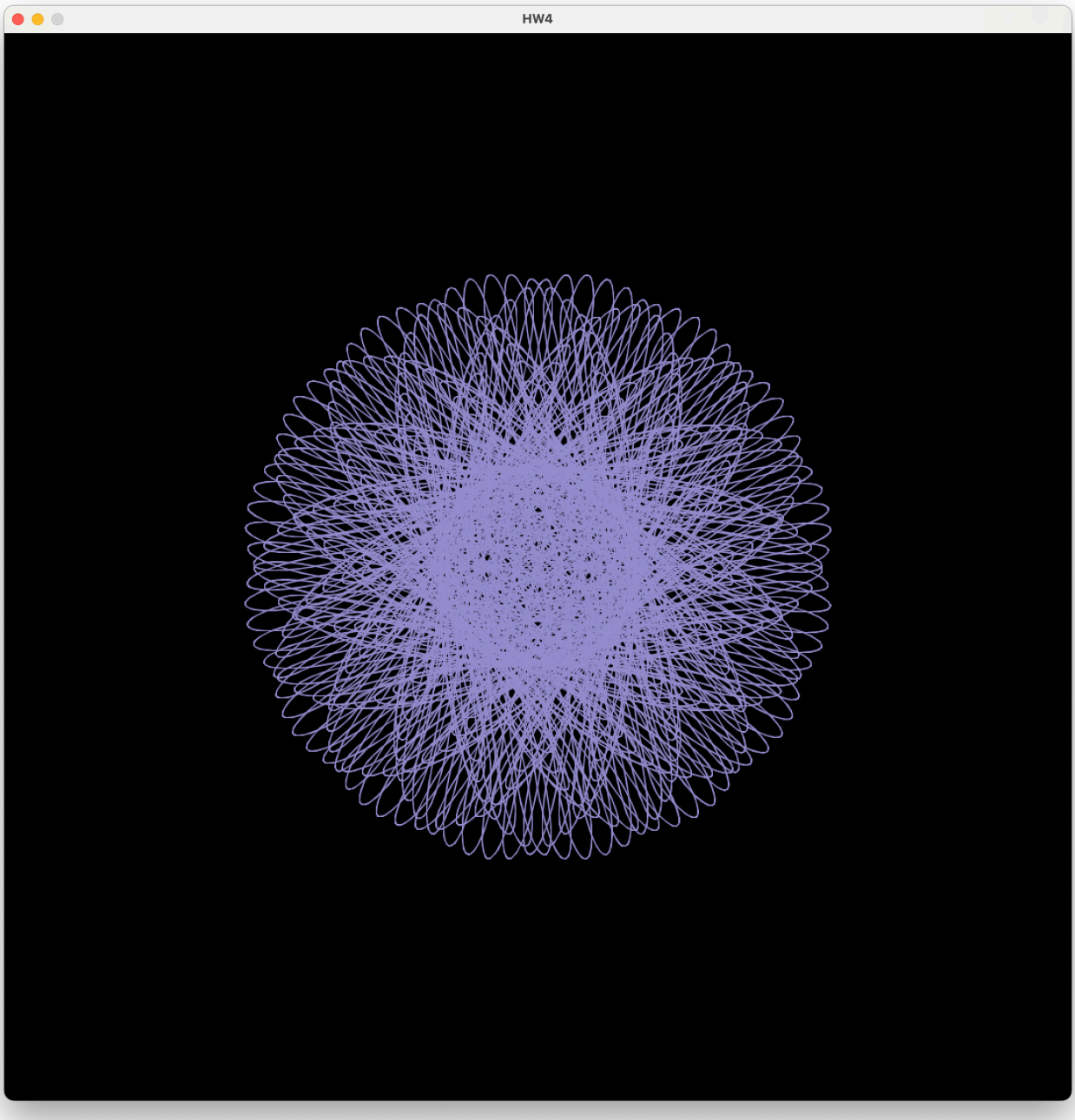
```
    float radius = scale * cos(k * t);
    float x = radius * cos(t);
    float y = radius * sin(t);
    vertex(x, y);
  }
  endShape(CLOSE);
  popMatrix();
}

void draw(){
  background(0);
  ellipse(width/2, height/2, radius*2, radius*2);
  translate(width/2, height/2);
  float rotation = TWO_PI / num_flowers; // angle between each flower
  angle += aOff;
  for (int i=1; i<num_flowers+1;i++){
    y = (float) Math.sin(rotation * i) * radius;
    x = (float) Math.cos(rotation * i) * radius;
    drawFlower(x,y,k, scale, angle);
  }
}
```

Iterative process / Final Outcome

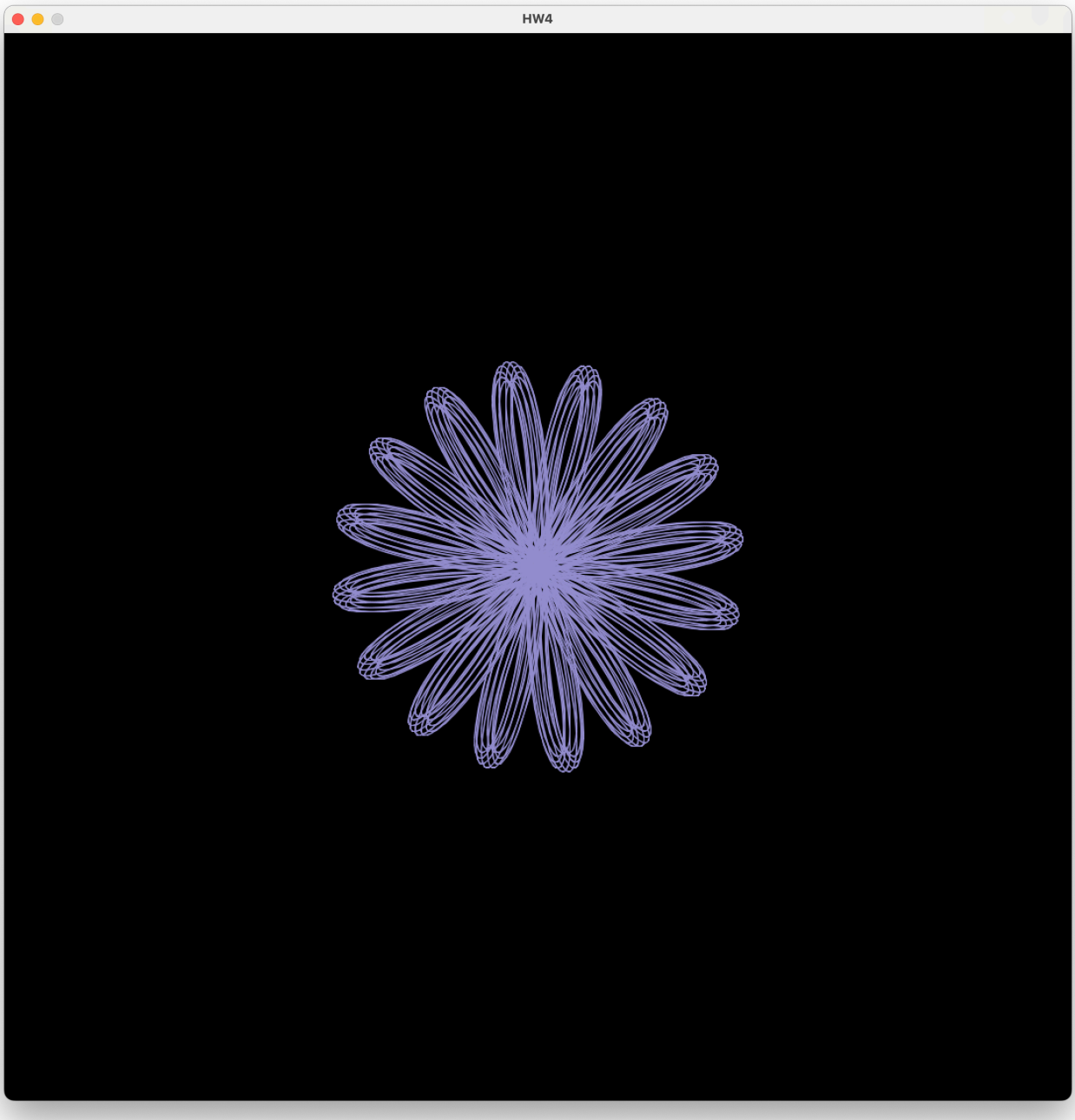
First ever instance



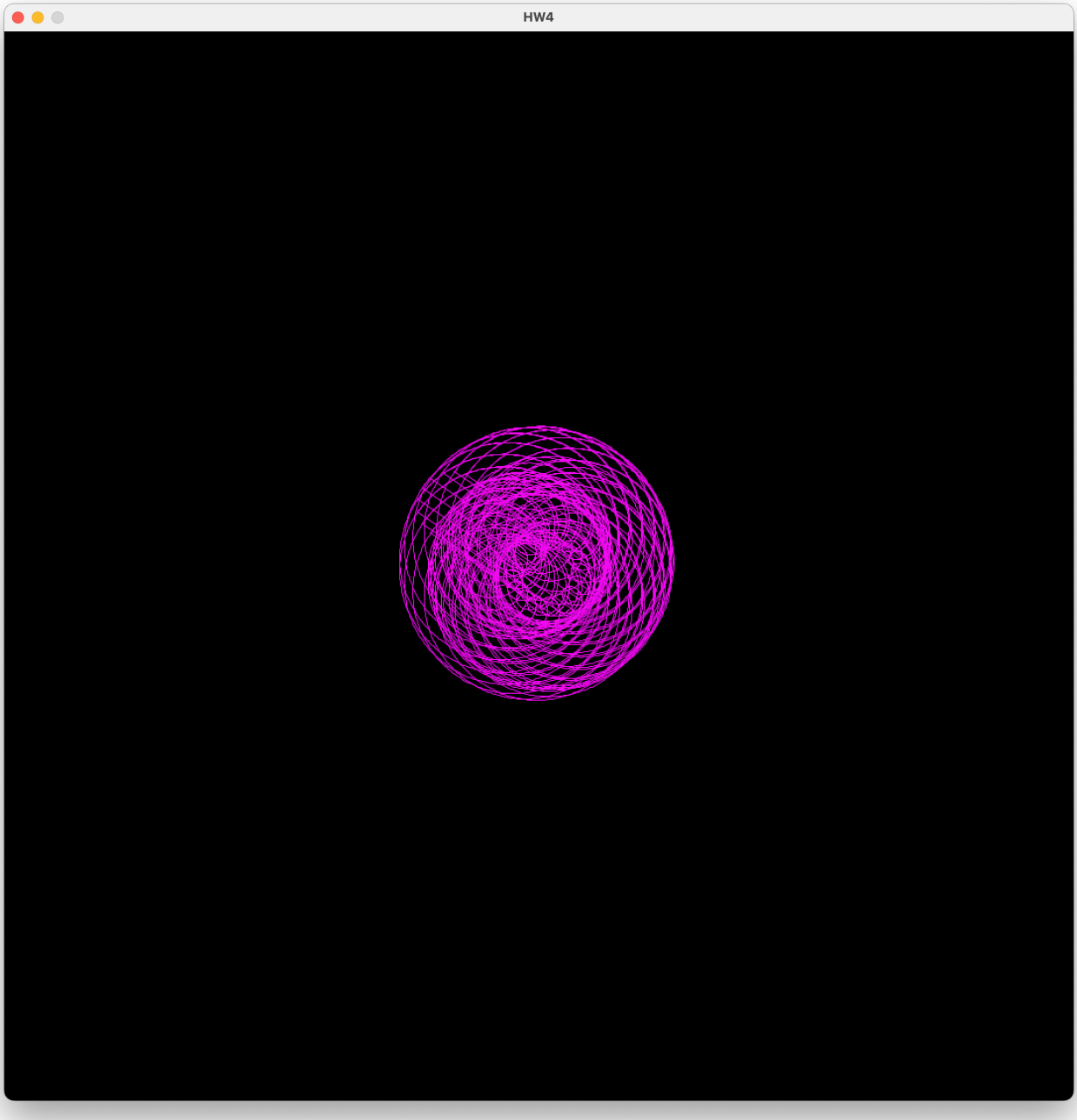
Increasing the number of flowers and reducing the radius

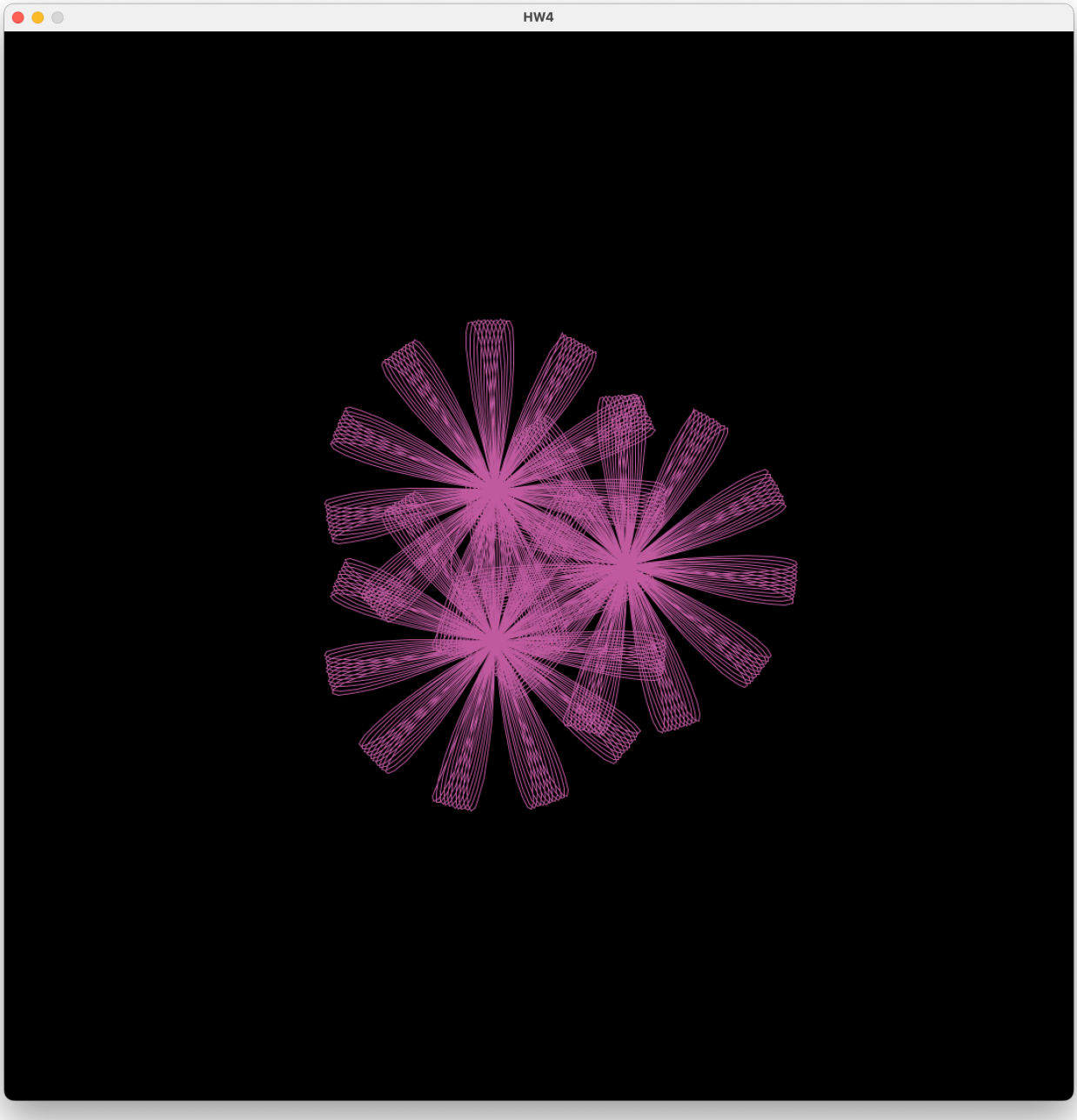


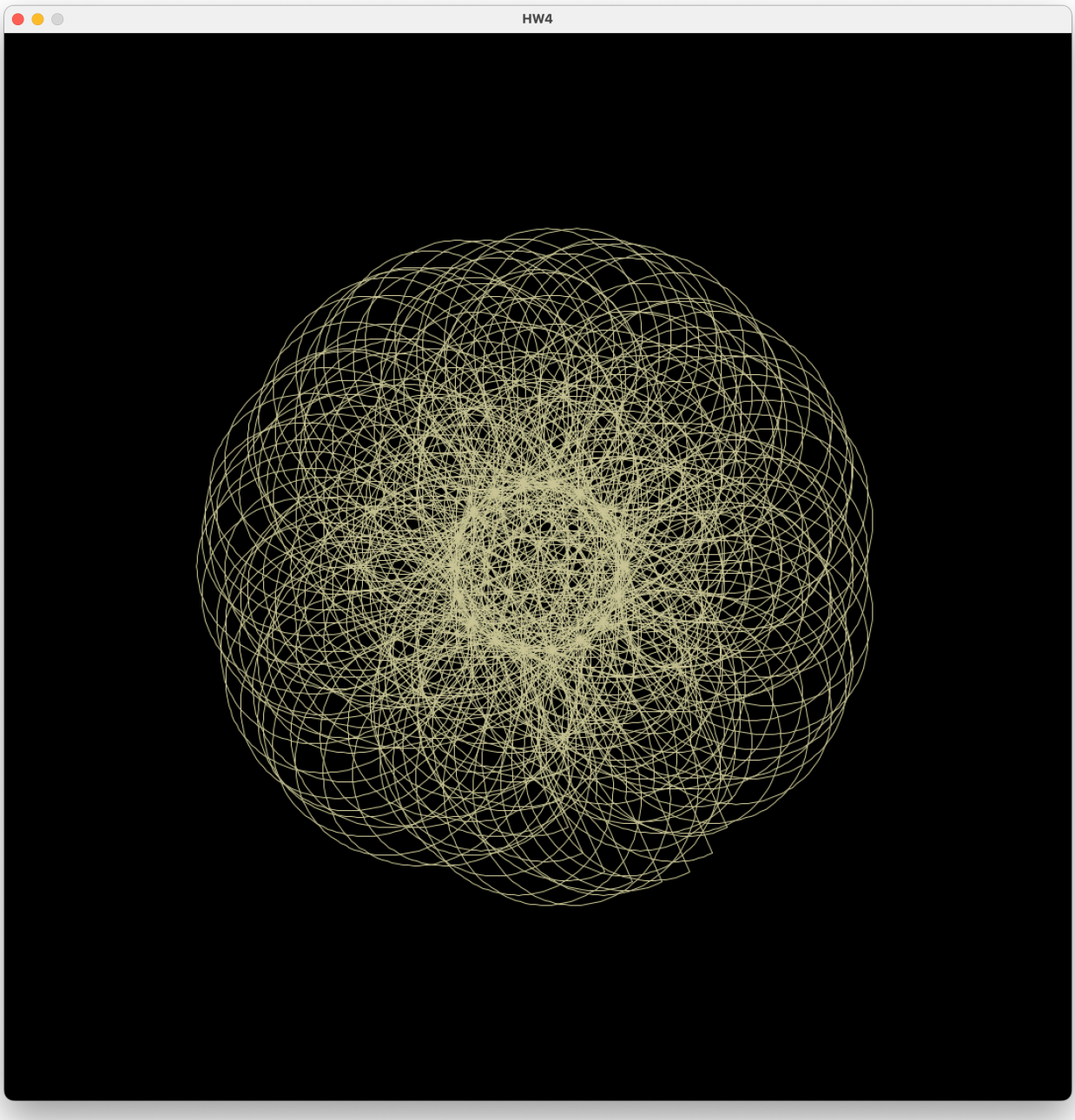
Radius is changed to 5. Seems to create a single flower.

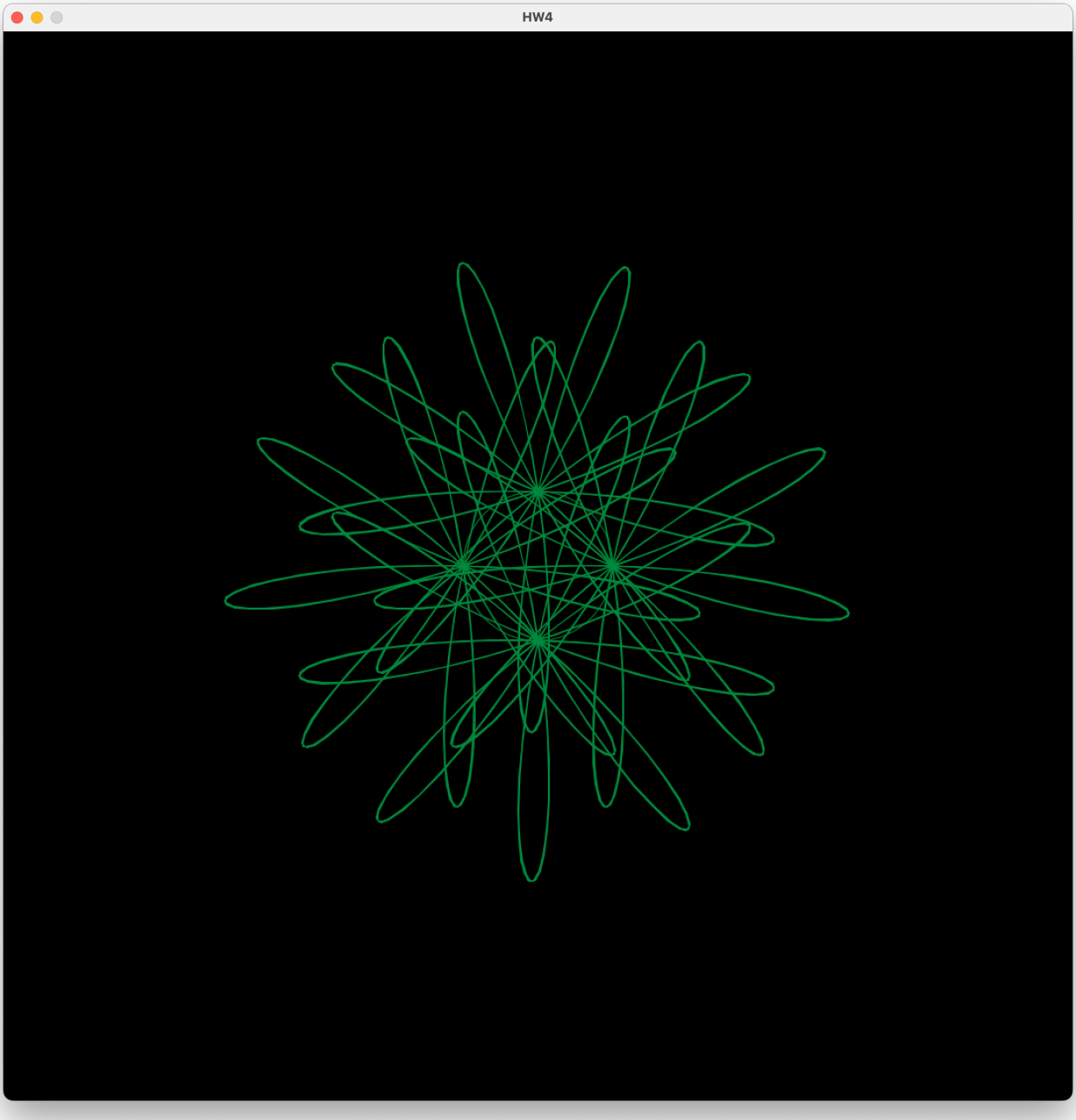


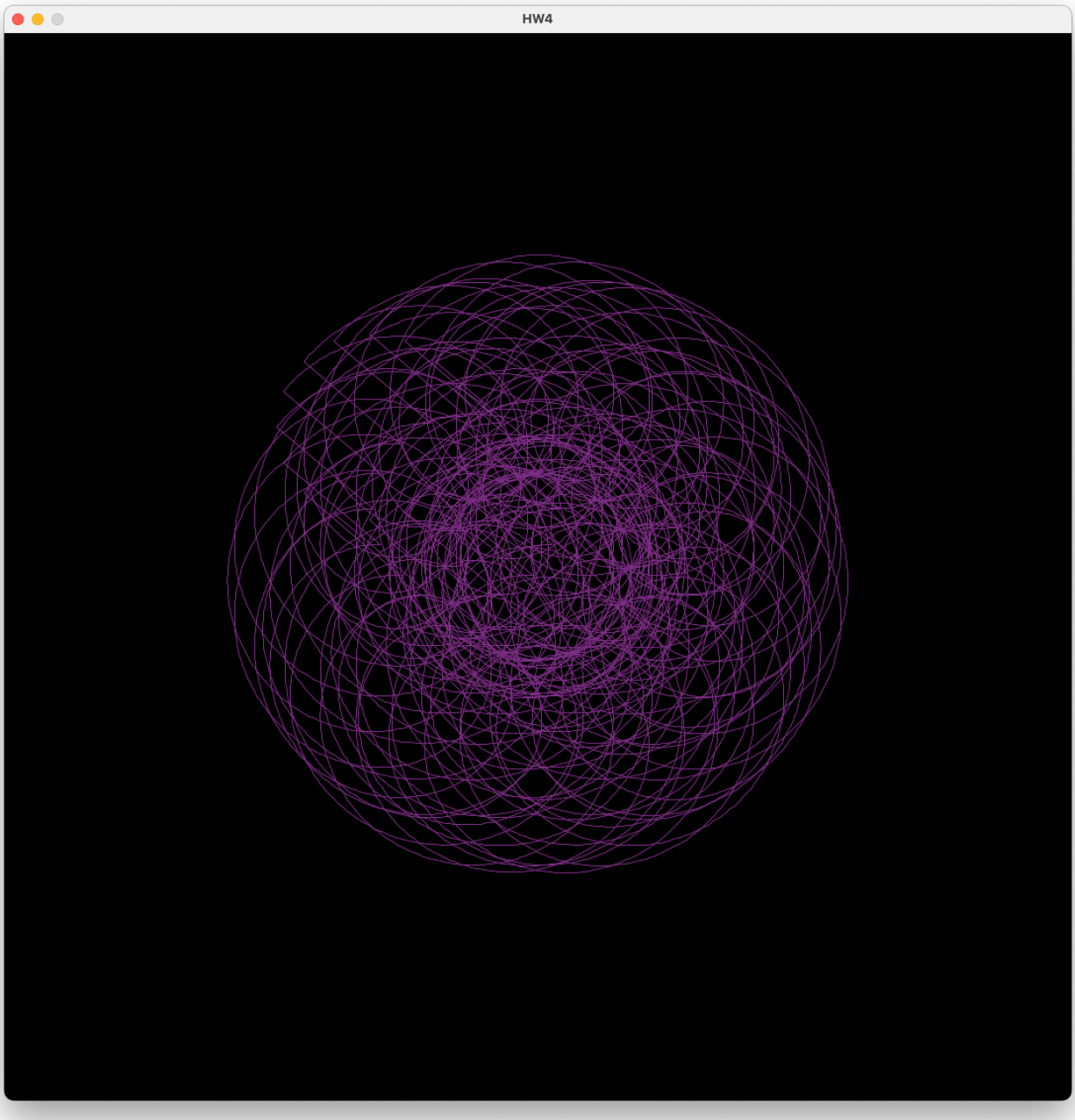
Random instance


HW4


HW4


HW4


HW4


HW4