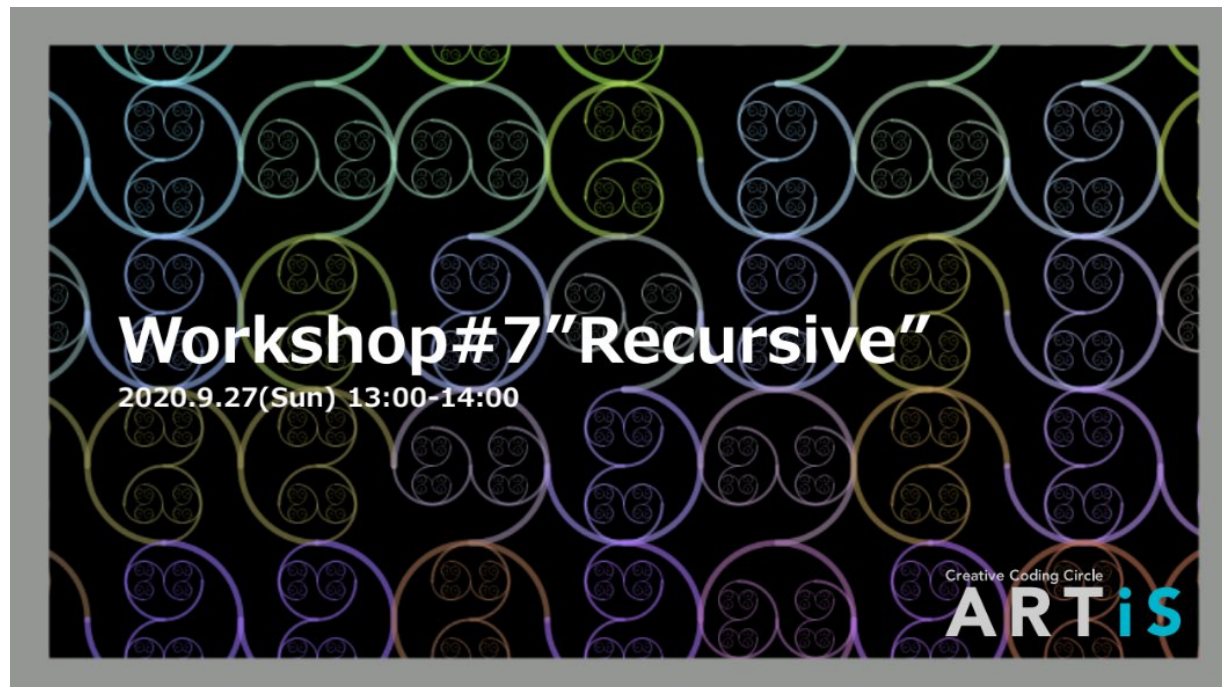


Workshop#7 “Recursive”



Workshop#7は「再帰処理を用いて作品を作ろう」という内容のワークショップでした。再帰的に処理することで、フラクタルの性質を持つ作品を作ることが可能です。

```
// 省略: void setup内でarc_fractalを
// 連続的に並べている
void arc_fractal(float x, float y, float r) {
// arcを利用したフラクタルを作成する
strokeWeight(2*map(r, R, r_min, 2, 0.1));
noFill();
arc(x, y, r, r, 0, 3*PI/2);
r*=scale;
if (r>=r_min) {
for (float i=PI; i<=2*PI; i+=PI) {
push();
translate(scale*r*cos(i)+x,
scale*r*sin(i)+y);
rotate(PI/2);
arc_fractal(0, 0, r);
pop();
}
}
}
```

arcを再帰的に処理しています。一回の処理でランダムにarcを回転させています。これにより、ツタの弦のような模様を描くことができます。

Workshop#8 “Attractor”



Workshop#8は、「アトラクターを使って作品を作ってみよう」という内容のワークショップでした。バナーに使ったプログラムをもとにして学習を進めました。参加者は、リサージュ曲線とアトラクターを組み合わせた作品などをつくりました。

```
void draw() {
fill(0,5);
rect(0, 0, width, height);
for (int i = 0; i < myNodes.length; i++) {
myNodes[i].update();
fill(255, 150, 0, 150);
ellipse(myNodes[i].x,myNodes[i].y,10,10);
}
myAttractor.x = mouseX;
myAttractor.y = mouseY;
for (int i = 0; i < myNodes.length; i++) {
if (mousePressed) {
myAttractor.attract(myNodes[i]);
}
myNodes[i].update();
}
stroke(0, 150, 255);
myAttractor.draw();
noStroke();
fill(0, 100);
}
```

「ノード」という引き寄せられる点、「アトラクター」という引き寄せる点を別のクラスで実装しています。

マウスを押している間はアトラクターがノードを引き寄せます。

アトラクターの位置に規則性を持たせることで、ノードの集合体が図形を描くようにした参加者もいました。