PROJECT DESCRIPTION

Project idea.

The idea for this project came from the London New Year's Eve fireworks that just resumed this year, so I wanted to use what I learned in coding1 to simply replicate a fireworks scene.

How to do it.

First, I wanted to use what I learned in week three about Flores geometry to create a geometric shape that would take different forms depending on the movement of the mouse. This part was used to represent fireworks, and I created a total of seven different geometric shapes that displayed different forms during mouse movement. Then, I added sound files to simulate the sound made by the fireworks when they explode. Finally, the visualization of the mouse trajectory was used to simulate the fragments of the fireworks when they explode. The final result can produce different geometric shapes and different color mouse trajectories depending on the mouse movement.

Questions and thoughts.

1. At first, I couldn't make the geometry change color according to the mouse movement, then I tried if, by if an x and y interval to set the different color display conditions.

```
if(20>x3>=0){context.strokeStyle = "#7391E5"; context.lineCap="square";} if(y3<=20){context.strokeStyle = "#DDE573";
if(20>x5>=0){context.strokeStyle = "#58BA19"; context.lineCap="square";}if(y5<=20){context.strokeStyle = "#DB215B"; context.lineWidth=0.2;}
```

2. Seven different geometries were subjected to color changes under different conditions. The geometric forms were also changed by cos and sin functions.

```
var x6 = Math.cos(spacing * i6 * (mouseX/10)) * Math.cos(spacing * i6 * (mouseY/30)) * radius6;var y6 = Math.sin(spacing * i6 * (mouseX/10)) * Math.sin(spacing * i6 * (mouseY/30)) * radius6;.
```

Also, the position of the geometry is changed.

```
context.lineTo(x6+1150+radius5,y6+350+radius6,penSize1); }
```

Seven variations of geometric shapes with widely varying dimensions.

```
function draw() {

var segments = 500;
var spacing = TWO_PI / segments;
var radius = 120;

var segments1 = 2000;
var spacing1 = TWO_PI / segments;
var radius1 = 250;
var segments2 = 800;
var spacing2 = TWO_PI / segments;
var radius2 = 200;

var segments3 = 600;
var spacing3 = TWO_PI / segments;
var radius3 = 270;

var segments4 = 150;
var spacing4 = TWO_PI / segments;
var radius4 = 150;

var segments5 = 150;
var spacing5 = TWO_PI / segments;
var radius5 = 350;

var segments6 = 150;
var spacing6 = TWO_PI / segments;
var radius6 = 450;

var penSize = 0.1;
var penSize1 = 1;
var positionX = 200
```

3. The visualization of the mouse track at the beginning of the path is too tightly connected and not fragmented.

```
if (distance < 30) {
ctx.moveTo(particlesArray[i].x, particlesArray[i].y);
ctx.lineTo(particlesArray[j].x, particlesArray[j].y);
ctx.stroke();
ctx.closePath();
},
```

Also to make fragmentation stronger I tweaked these codes.

*this.size = Math.random() * 7+ 1;this.speedX = Math.random() * 4 + 1; this.speedY = Math.random() * 4 + 1;.*

To visualize the pieces of the mouse track can have random colors.

*ctx.fillStyle = 'hsl('+ Math.random()*300+',80%,50%)';.*

4. The sound file is added directly to the sample in mp3 format.

```
let maxi = maximilian();
let audio = new maxi.maxiAudio();
let myOsc = new maxi.maxiOsc();
let mySample = new maxi.maxiSample();
audio.init();
audio.loadSample('yanhua1.mp3', mySample);
audio.play = function(){
var out = mySample.play();
return out;
}
```

https://mimicproject.com/code/5e20f6f3-eea7-f9d3-d5cc-bc919f114ab0