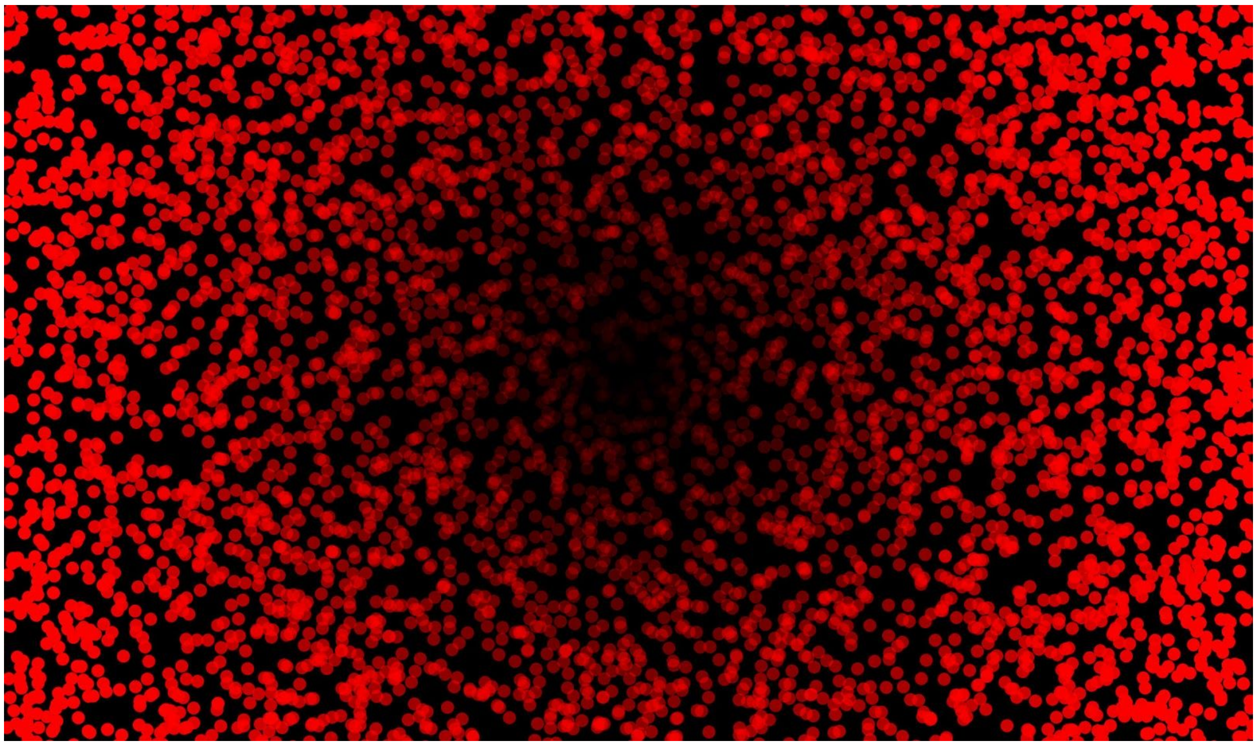
Principles Art Project: Ethan Nichols

Rule 1:
In this rule i wanted the balls to change it’s opacity depending on its distance from the center of the canvas.

Code:
```
  this.render = function(){
    noStroke();
    var ratio = (width/2) / 255;
    var ballDist = int(dist(this.loc.x, this.loc.y,
width/2,height/2))
    var alpha = int(ballDist / ratio);
    var newColor = color(red(this.col), green(this.col),
blue(this.col), alpha);
    var r = 20


    fill(newColor);
    ellipse(this.loc.x, this.loc.y, r, r);
}
```

Photo:


Rule 2:
In this rule balls are moving around the canvas and there is a random radius between 50 and 100, but the rendered ball that the person sees is not the true radius of the ball. If the balls radius intersect each other then a line is drawn in between their centers. Also using html the canvas displays an image that I made and blurs the canvas to create depth.

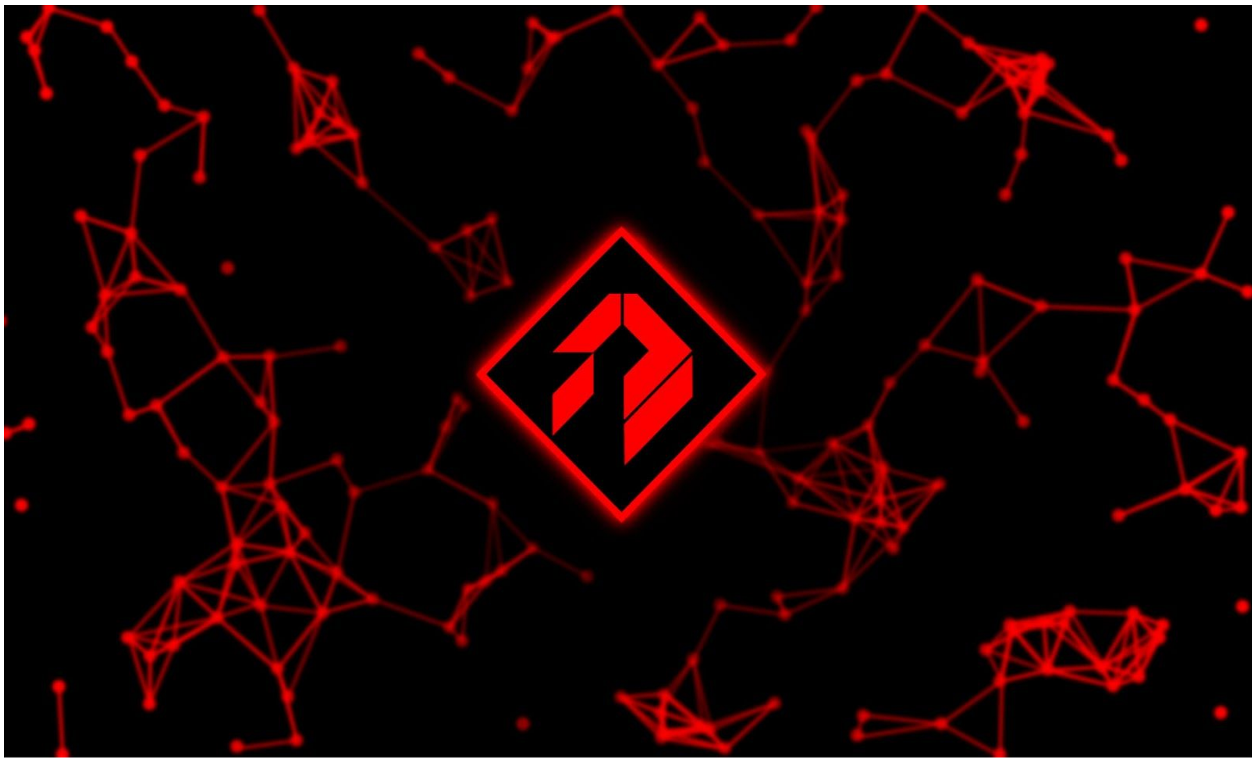
Code:
```
  this.render = function(){
    noStroke();
    var ratio = (width/2) / 255;
    var ballDist = int(dist(this.loc.x, this.loc.y, width/2,
height/2))
    var alpha = int(ballDist / ratio);
    var newColor = color(red(this.col), green(this.col),
blue(this.col), alpha);
    var r = 20


    fill(newColor);
    ellipse(this.loc.x, this.loc.y, r, r);

    for(i = 0; i < balls.length; i++){
      var d = int(dist(this.loc.x, this.loc.y, balls[i].loc.x,
balls[i].loc.y));
      if(d < radius){
        line(this.loc.x, this.loc.y, balls[i].loc.x, balls[i].loc.y);
        stroke(255, 0, 0, alpha);
        strokeWeight(5);
      }
    }
  }
HTML:
<style>
        body {padding: 0; margin: 0;}
        canvas {
                vertical-align: top;
                filter: blur(3px);
               }
       #logo {
         position: absolute;
         top: 0;
         bottom: 0;
         left: 0;
         right: 0;
         display: flex;
         flex-flow: column;
         justify-content: center;
         align-items: center;
         z-index: 100;
            color: red;
       }
       #logo img {
         width: 500px;
         height: 500px;
       }
    </style>
  </head>


  <body>
    <div id="logo">
      <img src="SIVA3.png" />
    </div>
```

Photo:



Rule 3:
For this rule I used something similar to Rule 1 and Rule 2 where the opacity changes but also color. And the balls are invisible and lines are drawn from the rendered balls' centers.

Code:
```
this.render = function(){
    noStroke();
    var ratio = (width/2) / 255;
    var ballDist = int(dist(this.loc.x, this.loc.y, width/2,
height/2))
    var alpha = int(ballDist / ratio);
    var newColor = color(255, alpha, 0, alpha);
    var r = 20


    fill(newColor);
    //ellipse(this.loc.x, this.loc.y, r, r);

    for(i = 0; i < balls.length; i++){
      var d = int(dist(this.loc.x, this.loc.y, balls[i].loc.x,
balls[i].loc.y));
      if(d < r){
        line(this.loc.x, this.loc.y, balls[i].loc.x, balls[i].loc.y);
        stroke(255, alpha, 0, alpha);
        strokeWeight(5);
      }
    }
  }

}
Function draw(){
 background (0,0,0,0);
}
```

Photo: