RECREATE MASTERPIECES OF MODERN ART WITH JAVASCRIPT

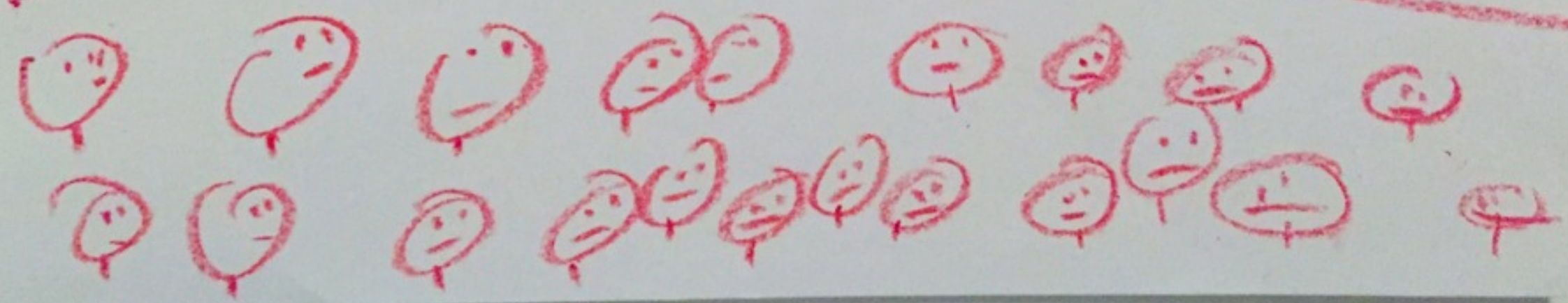
Amy Cheng

- web developer with New York Magazine
- artist


Strawberry
Almond
Scone
from Flour and Twenty
Blackbird
canelé
from Woops Bakery

JS

amy →

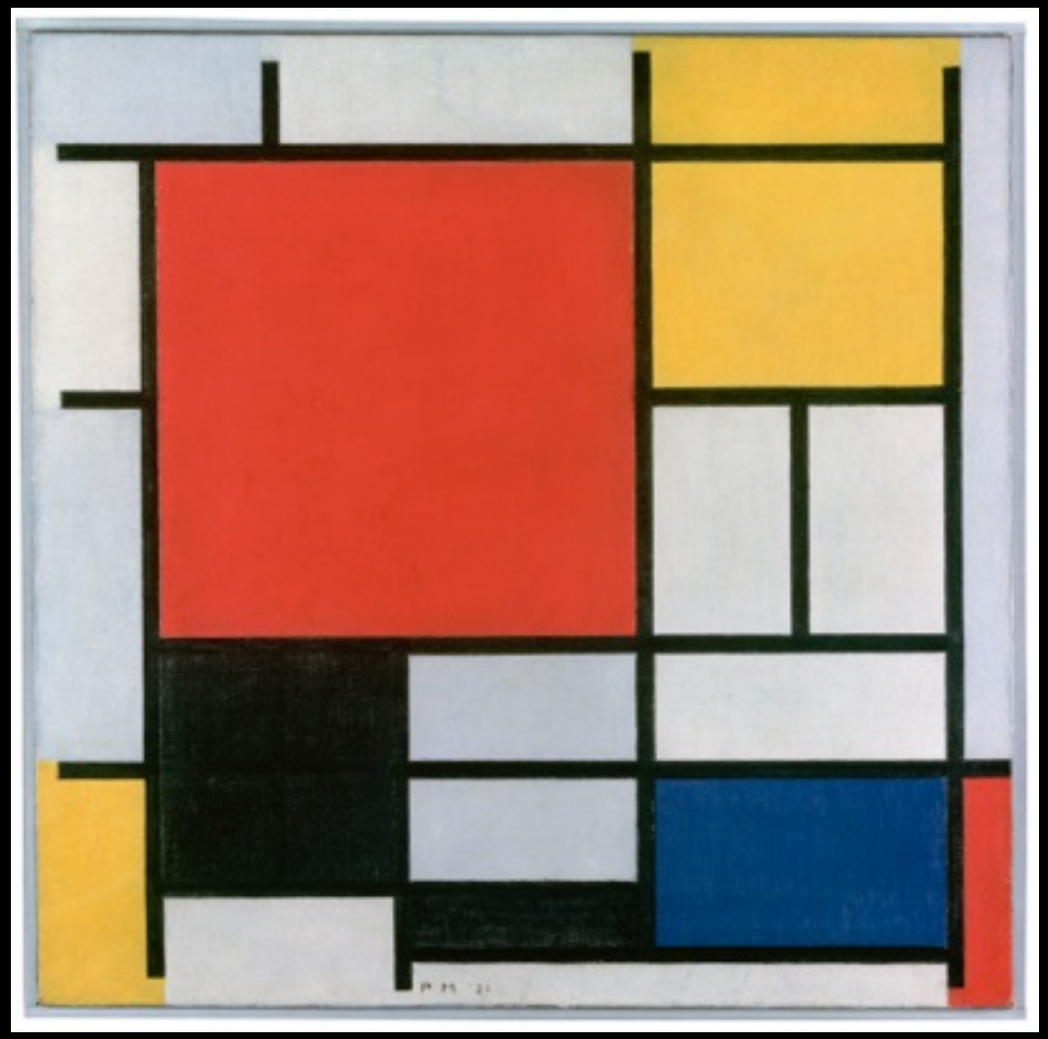


By Frida Kahlo

By Claude Monet

If you know JavaScript,
you can make art!

Spot Painting
Damien Hirst



Composition
Piet Mondrian

HTML5 `CANVAS API` = JavaScript's blank canvas

```
var xPosition = window.innerHeight/2;
var yPosition = window.innerWidth/2;
var size  = 25;

var canvas=document.getElementById("canvas");
var ctx=canvas.getContext("2d");

var draw= function(){
  ctx.beginPath();
  ctx.arc(xPosition, yPosition, size, 0, 2*Math.PI);
  ctx.stroke();
  window.requestAnimationFrame(draw);
};

draw();
```

```
var xPosition = window.innerHeight/2;
var yPosition = window.innerWidth/2;
var size  = 25;

var canvas=document.getElementById("canvas");
var ctx=canvas.getContext("2d");

var draw= function(){
  ctx.beginPath();
  ctx.arc(xPosition, yPosition, size, 0, 2*Math.PI);
  ctx.stroke();
  window.requestAnimationFrame(draw);
};

draw();
```


p5*

Concepts, not tools.

```
// P5
function draw() {
  fill(color);
  ellipse(x, y, horizontalRadius, verticalRadius);
}

// Fabric
var spot = new fabric.Circle(({
  radius: _radius, fill: color, left: x, top: y
}));
canvas.add(spot);

// Easel

var spot = new createjs.Shape();
spot.graphics.beginFill(color).drawspot(0, 0, _radius);
spot.x = x;
spot.y = y;
stage.addChild(spot);
```

Spot Painting

```
p.setup = function() {
  p.createCanvas(800, 500);
  p.background(255);
  p.noLoop();
};
```

```
p.draw = function() {
  for ( var i = 1; i < p.width/spacing ; i++) {
    for ( var j = 1; j< p.height/spacing; j++) {
      color();
      p.noStroke();
      p.ellipse(i*spacing, j*spacing, 25, 25);
    }
  };
};
```

```
p.setup = function() {
  p.createCanvas(800, 500);
  p.background(255);
  p.noLoop();
};
```

```
var color = function() {
  return p.fill( p.random(0, 255), p.random(0,255), p.random(0,255));
};
```

JavaScript is an artistic medium that can do:

- procedural generation

```
for ( var i = 1; i < p.width/spacing ; i++) {
   for ( var j = 1; j< p.height/spacing; j++) {
      color();
      p.noStroke();
      p.ellipse(i*spacing, j*spacing, 25, 25);
   }
};
```

JavaScript is an artistic medium that can do:

- procedural generation
- parameterization

```
var color = function() {
  return p.fill( p.random(0, 255), p.random(0,255), p.random(0,255));
};
```

Color Modes:

- how code interpret colors
- RGB -> red, green, blue
- HSB -> hue, saturation, brightness

Spot Painting Machine

```
//state object for dat.gui
var machineState = {
  colorMode: 'RGB',
  colorVal1: 100,
  colorVal2: 100,
  colorVal3: 100,
  colorVal1Rando: true,
  colorVal2Rando: true,
  colorVal3Rando: true,
  render: generateColors,
  save: saveImage
};
```

```
var color = function() {
   var color1 = machineState.colorVal1;
   var color2 = machineState.colorVal2;
   var color3 = machineState.colorVal3;

   if( machineState.colorMode === 'RGB' ){
     p.colorMode(p.RBG, 255);
     color1 = normalize(machineState.colorVal1);
     color2 = normalize(machineState.colorVal2);
     color3 = normalize(machineState.colorVal3);
   }else{
     p.colorMode(p.HSB, 100);
   }

   var _color1 = p.random(0, color1);
   var _color2 = p.random(0, color2);
   var _color3 = p.random(0, color3);

   // freeze values (don't randomize)
   if(!machineState.colorVal1Rando){_color1 = color1;}
   if(!machineState.colorVal2Rando){_color2 = color2;}
   if(!machineState.colorVal3Rando){_color3 = color3;}

   return p.color(
     _color1,
     _color2,
     _color3
     );
};
```

```
var color = function() {
   var color1 = machineState.colorVal1;
   var color2 = machineState.colorVal2;
   var color3 = machineState.colorVal3;

   if( machineState.colorMode === 'RGB' ){
     p.colorMode(p.RBG, 255);
     color1 = normalize(machineState.colorVal1);
     color2 = normalize(machineState.colorVal2);
     color3 = normalize(machineState.colorVal3);
   }else{
     p.colorMode(p.HSB, 100);
   }

   var _color1 = p.random(0, color1);
   var _color2 = p.random(0, color2);
   var _color3 = p.random(0, color3);

   // freeze values (don't randomize)
   if(!machineState.colorVal1Rando){_color1 = color1;}
   if(!machineState.colorVal2Rando){_color2 = color2;}
   if(!machineState.colorVal3Rando){_color3 = color3;}

   return p.color(
     _color1,
     _color2,
     _color3
     );
};
```

What if we randomize other visual elements?

More randomness

Preparatory Paintings
Elaine De Kooning

```
function setup(){
  createCanvas(800, 500);
  background(255);

  for (var i = 0; i < numberOfSwoosh; i++) {

    noFill()
    stroke(0);
    strokeWeight(random(75));
    strokeCap(ROUND);
    curve(
      randomGaussian(width/2, deviation), randomGaussian(height/2, deviation),
      randomGaussian(width/2, deviation), randomGaussian(height/2, deviation),
      randomGaussian(width/2, deviation), randomGaussian(height/2, deviation),
      randomGaussian(width/2, deviation), randomGaussian(height/2, deviation)
    );
  };

  for (var i = 0; i < numberOfStrokes; i++) {
    stroke(0);
    strokeCap(PROJECT);
    strokeWeight(noise(i)*20);
    line(
      randomGaussian(width/2, deviation), randomGaussian(height/2, deviation),
      randomGaussian(width/2, deviation), randomGaussian(height/2, deviation)
    );
  };
}
```

```
function setup(){
  createCanvas(800, 500);
  background(255);

  for (var i = 0; i < numberOfSwoosh; i++) {

    noFill()
    stroke(0);
    strokeWeight(random(75));
    strokeCap(ROUND);
    curve(
      randomGaussian(width/2, deviation), randomGaussian(height/2, deviation),
      randomGaussian(width/2, deviation), randomGaussian(height/2, deviation),
      randomGaussian(width/2, deviation), randomGaussian(height/2, deviation),
      randomGaussian(width/2, deviation), randomGaussian(height/2, deviation)
    );
  };

  for (var i = 0; i < numberOfStrokes; i++) {
    stroke(0);
    strokeCap(PROJECT);
    strokeWeight(noise(i)*20);
    line(
      randomGaussian(width/2, deviation), randomGaussian(height/2, deviation),
      randomGaussian(width/2, deviation), randomGaussian(height/2, deviation)
    );
  };
}
```

```
p5.prototype.randomGaussian = function(mean, sd)  {
  var y1,x1,x2,w;
  if (previous) {
    y1 = y2;
    previous = false;
  } else {
    do {
      x1 = this.random(2) - 1;
      x2 = this.random(2) - 1;
      w = x1 * x1 + x2 * x2;
    } while (w >= 1);
    w = Math.sqrt((-2 * Math.log(w))/w);
    y1 = x1 * w;
    y2 = x2 * w;
    previous = true;
  }

  var m = mean || 0;
  var s = sd || 1;
  return y1*s + m;
};
```

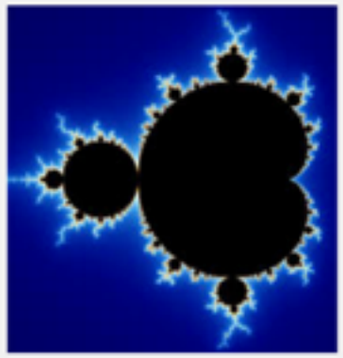
```
p5.prototype.randomGaussian = function(mean, sd)  {
  var y1,x1,x2,w;
  if (previous) {
    y1 = y2;
    previous = false;
  } else {
    do {
      x1 = this.random(2) - 1;
      x2 = this.random(2) - 1;
      w = x1 * x1 + x2 * x2;
    } while (w >= 1);
    w = Math.sqrt((-2 * Math.log(w))/w);
    y1 = x1 * w;
    y2 = x2 * w;
    previous = true;
  }

  var m = mean || 0;
  var s = sd || 1;
  return y1*s + m;
};
```

JavaScript is an artistic medium that can do:

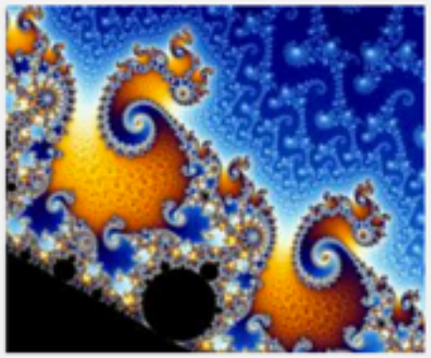
- procedural generation
- parameterization
- painting with algorithms

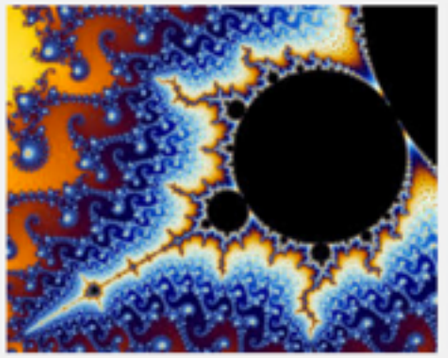
$$z_{n+1} = z_n^2 + c \mid$$

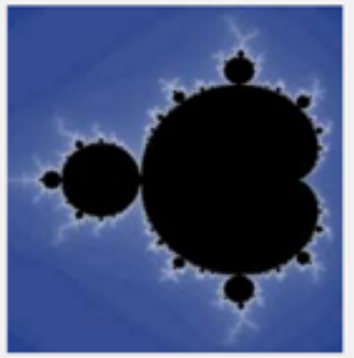
$$z_0 = 0$$

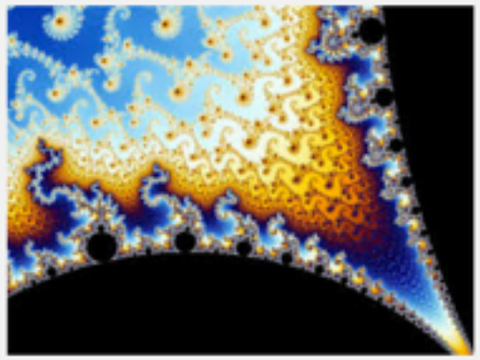


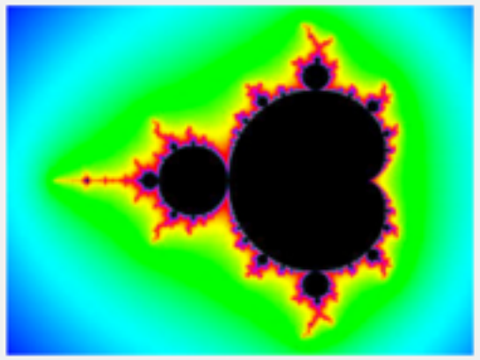

2560 × 1920 - en.wikipedia.org

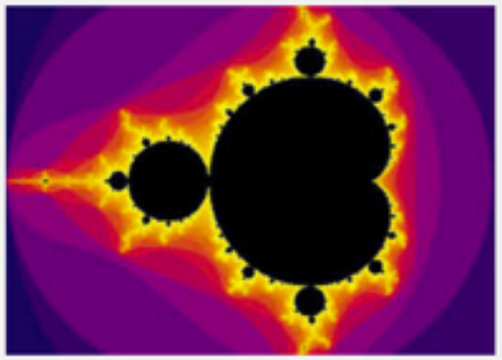


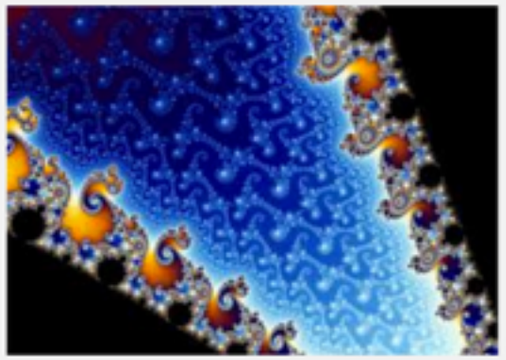


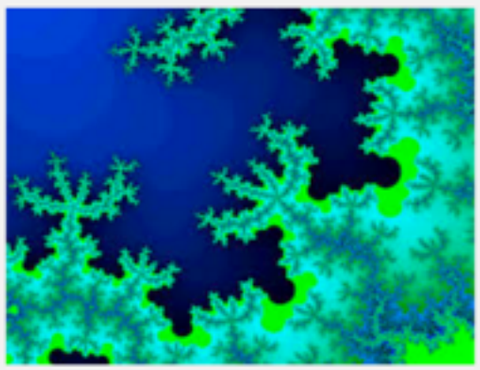


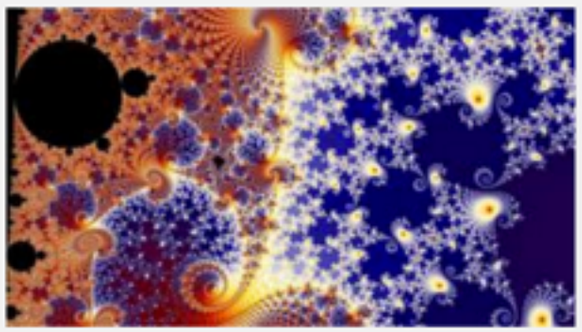


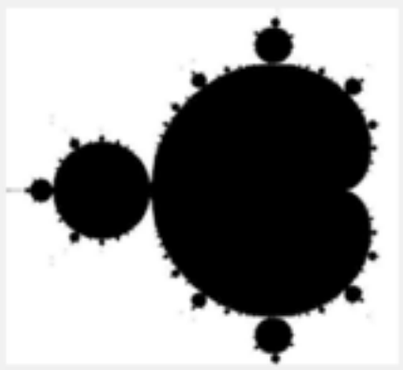


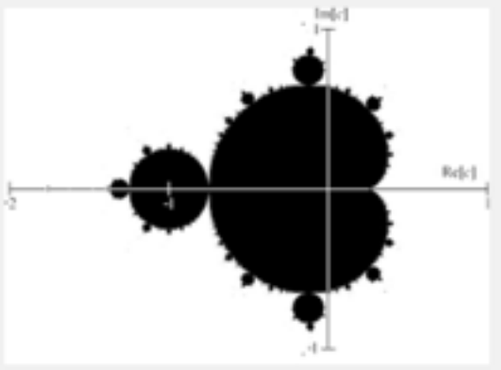











Im[c]
Re[c]

Mathematics is one expression of the world and art is another.

Computation can act as a bridge between the two!

Mandelbrot

JavaScript is an artistic medium that can do:

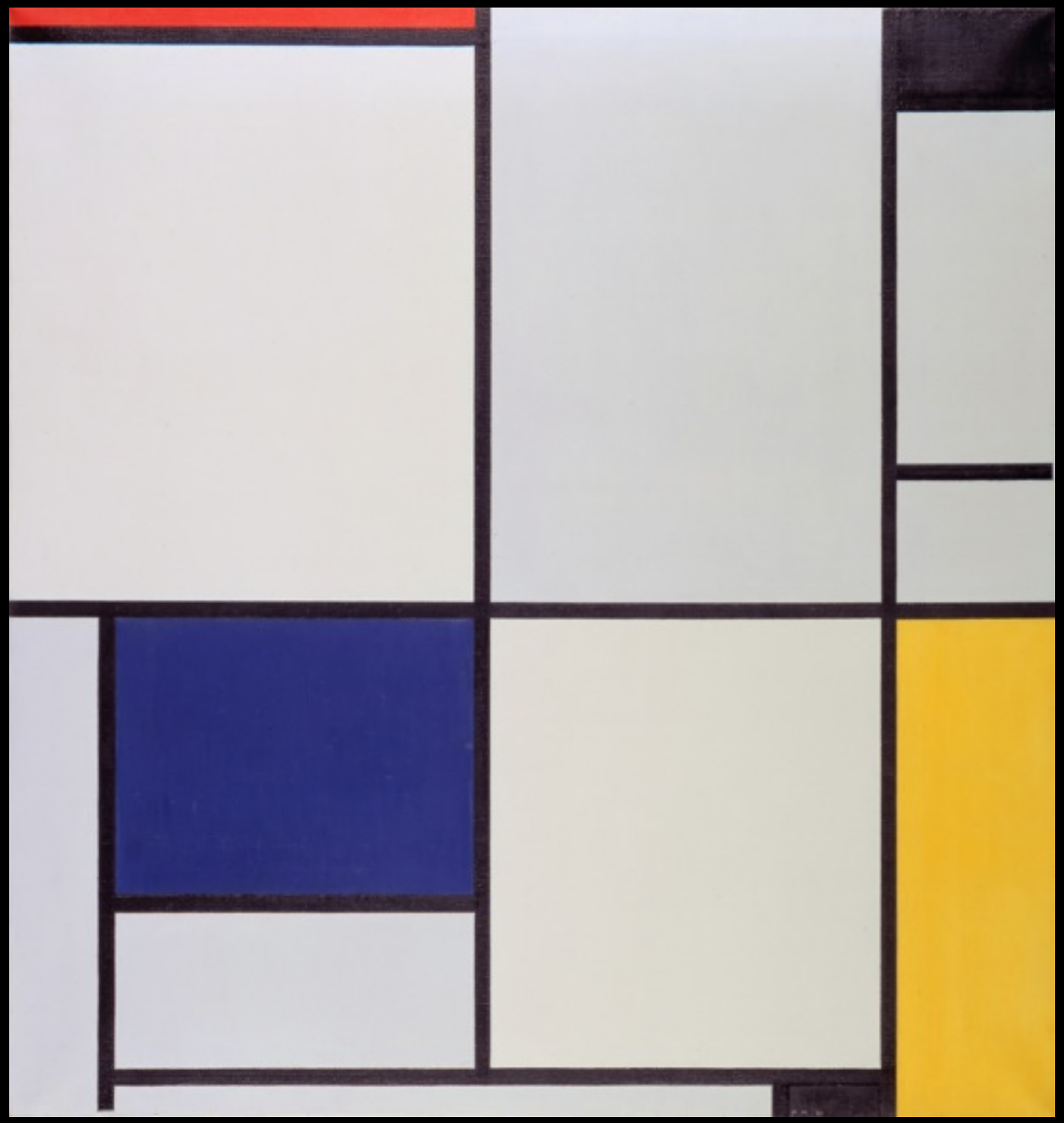
- procedural generation
- parameterization
- painting with algorithms
- creating and running systems

Spot Painting:

- parameters
- random function
- color function
- ellipse function (for loops)
- GUI

Prepartory Painting:

- curve
- line
- random function
- Gaussian function



Composition

Mondrian Automata

Mondrian Automata Pt. 2

```
var Cell = function( initX, initY, height,
width, color){

  // variables for position and vectors

  this.update = function() {
    // ...
  };

  this.render = function(){
    // ...
};

// ...
```

```
        this.update = function() {
          var currentLocation = p.createVector(xPos, yPos);
          var steering;

          if(Math.floor(targetLocation.x) !== Math.floor(currentLocation.x) &&
    Math.floor(targetLocation.y) !== Math.floor(currentLocation.y)){
            steering = p5.Vector.sub(targetLocation,currentLocation);
            steering.normalize();
            xPos += steering.x;
            yPos += steering.y;
          }else{
            targetLocation = p.createVector(p.random(0,600), p.random(0,300));
          }
        };
```

```
callback =
```

1 frame of animation

Spot Painting Pt. 4

```
// Separation
// Method checks for nearby boids and steers away
this.separate = function(boids) {
  var desiredseparation = 50.0;
  var steer = createVector(0,0);
  var count = 0;
  // For every boid in the system, check if it's too close
  for (var i = 0; i < boids.length; i++) {
    var d = p5.Vector.dist(this.position,boids[i].position);
    // If the distance is greater than 0 and less than an arbitrary amount (0 when you are yourself)
    if ((d > 0) && (d < desiredseparation)) {
      // Calculate vector pointing away from neighbor
      var diff = p5.Vector.sub(this.position,boids[i].position);
      diff.normalize();
      diff.div(d);        // Weight by distance
      steer.add(diff);
      count++;            // Keep track of how many
    }
  }
  // Average -- divide by how many
  if (count > 0) {
    steer.div(count);
  }

  // As long as the vector is greater than 0
  if (steer.mag() > 0) {
    // Implement Reynolds: Steering = Desired - Velocity
    steer.normalize();
    steer.mult(this.maxspeed);
    steer.sub(this.velocity);
    steer.limit(this.maxforce);
  }
  return steer;
};
```

Source:
The Nature of Code
Daniel Shiffman
http://natureofcode.com

```
this.cohesion = function(boids) {
  var neighbordist = 50;
  var sum = createVector(0,0);
  var count = 0;
  for (var i = 0; i < boids.length; i++) {
    var d = p5.Vector.dist(this.position,boids[i].position);
    if ((d > 0) && (d < neighbordist)) {
      sum.add(boids[i].position); // Add location
      count++;
    }
  }
  if (count > 0) {
    sum.div(count);
    return this.seek(sum);  // Steer towards the location
  } else {
    return createVector(0,0);
  }
};
```

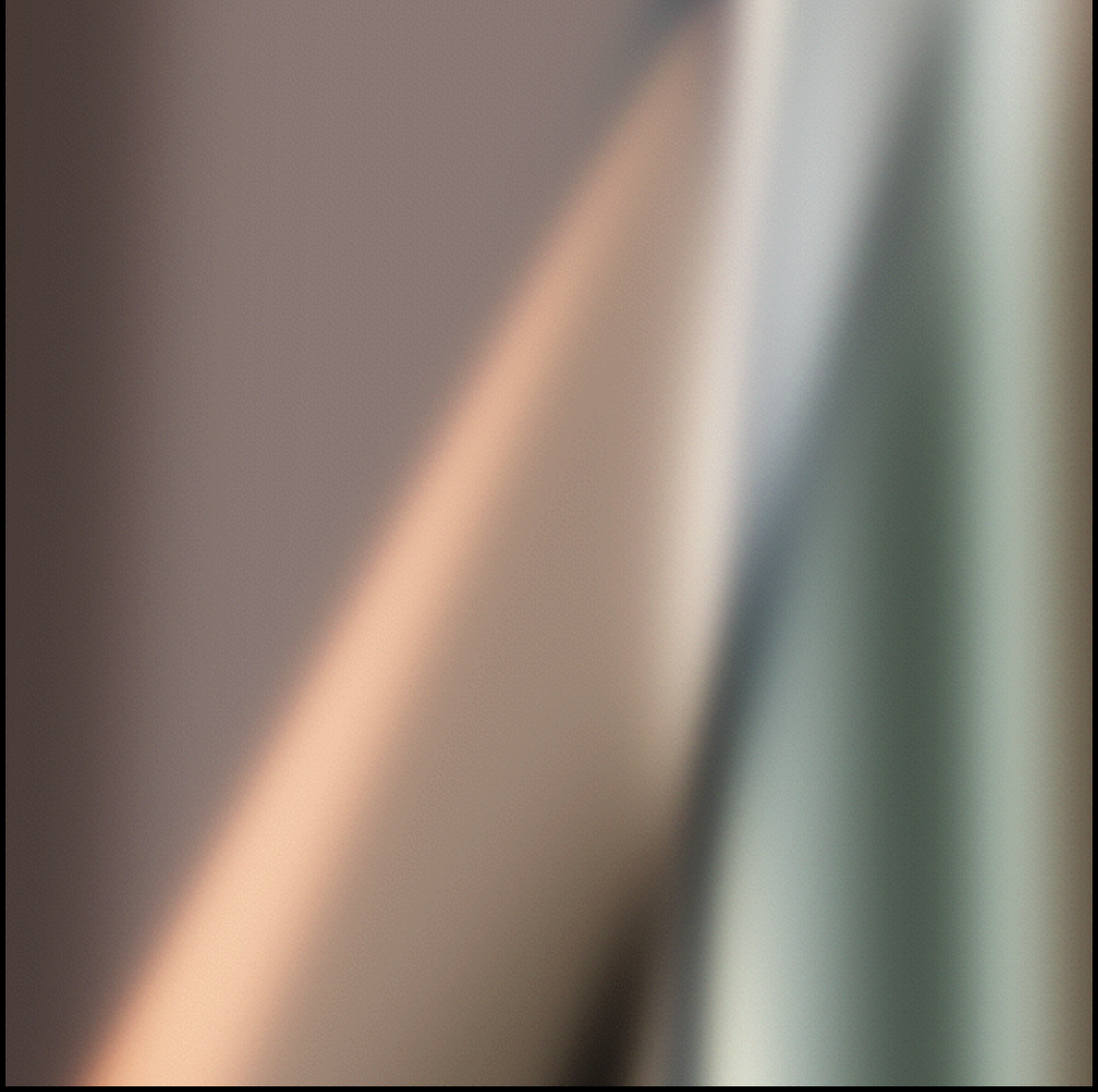
Source:
The Nature of Code
Daniel Shiffman
http://natureofcode.com

JavaScript is an artistic medium that can do:

- procedural generation
- parameterization
- painting with algorithms
- creating and running systems
- self-learning

machine learning =
system that learns from
its inputs

Code Painting

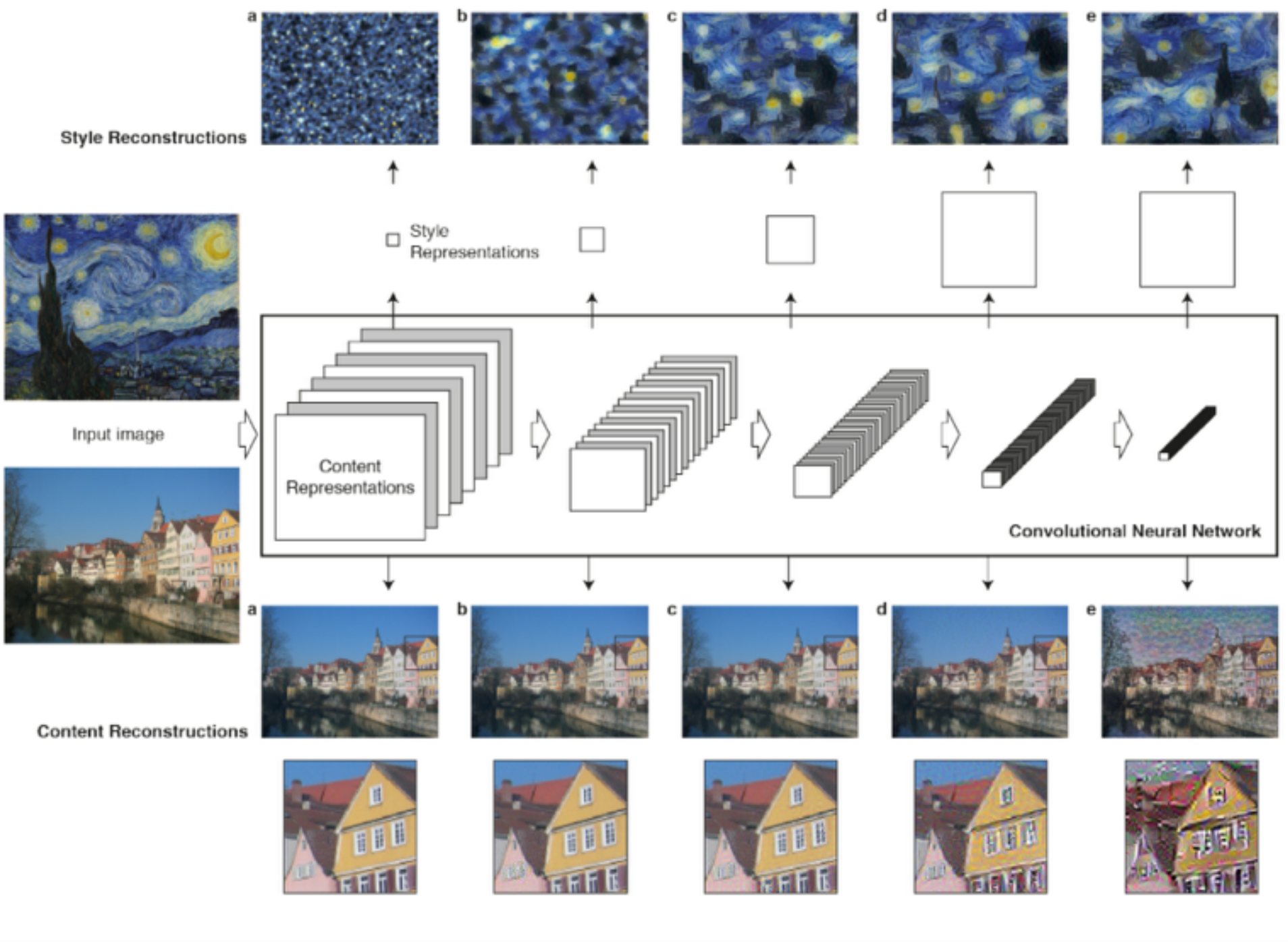


synaptic.js

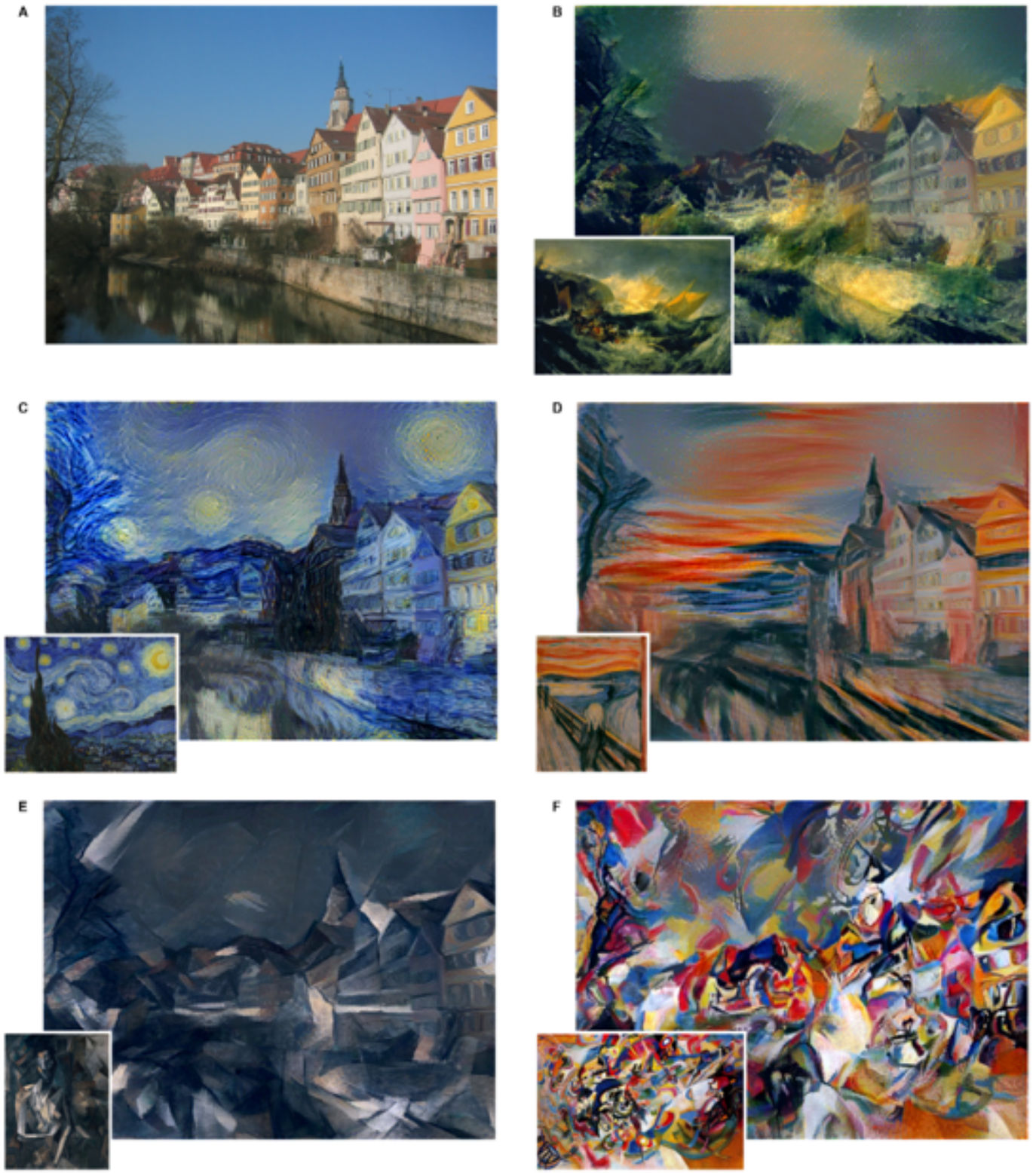
```
var imgData = getPixelData(sourceImage);

var iterate = function(){
  for (var x = 0; x < 300; x+=1)
  {
    for(var y = 0; y < 300; y+=1)
    {
      var dynamicRate =  .01/(1+.0005*iteration);
      perceptron.activate([x/300,y/300]);
      perceptron.propagate(dynamicRate, pixel(imgData,x,y));
    }
  }
  preview();
};
```

```
var imgData = getPixelData(sourceImage);

var iterate = function(){
  for (var x = 0; x < 300; x+=1)
  {
    for(var y = 0; y < 300; y+=1)
    {
      var dynamicRate =  .01/(1+.0005*iteration);
      perceptron.activate([x/300,y/300]);
      perceptron.propagate(dynamicRate, pixel(imgData,x,y));
    }
  }
  preview();
};
```

```
var preview = function(){
  imageData = context.getImageData(0, 0, 300, 300);
  for (var x = 0; x < 300; x++)
  {
    for(var y = 0; y < 300; y++)
    {
      var rgb = perceptron.activate([x/300, y/300]);
      imageData.data[((300 * y) + x) * 4] = (rgb[0] )* 255;
      imageData.data[((300 * y) + x) * 4 + 1] = (rgb[1] ) * 255;
      imageData.data[((300 * y) + x) * 4 + 2] = (rgb[2] ) * 255;
    }
  }

  context.putImageData(imageData,0,0);
  requestAnimationFrame(iterate);
};
```



Source:
A Neural Algorithm of Artistic Style
Leon A. Gatys, Alexander S. Ecker, Matthias Bethge



Source:
A Neural Algorithm of Artistic Style

Leon A. Gatys, Alexander S. Ecker,
Matthias Bethge

Code is an art medium but it can also be an artist collaborator.

Collab

JavaScript is an artistic medium that can do:

- procedural generation
- parameterization
- painting with algorithms
- creating and running systems
- self-learning (?)

But why JavaScript?

JavaScript is an artistic medium that can do:

- procedural generation
- parameterization
- painting with algorithms
- creating and running systems
- self-learning (?)

- be available to anyone with Internet
- cheap iterations

Art is not an app.

Get excited!

Questions?
Email amy@amycheng.info
Submit Issue https://github.com/amycheng/create-art-with-js/issues