Creating Generative Art - Basics

General concepts

1. Create the Canvas: size
2. Create the Dot: the foundational unit
3. Undersand P5js two main functions: setup() & draw()

1. Single Dot: Art's foundation

The creation and understanding of this class is the foundation of html canvas art. We define what it 'feels' to be a dot through a class. The creation of this abstract template is the foundation for creating art. Once this object is 'grasped' we can start experimenting and exploring the canvas.

> Concepction of the Dot Class

```
// Dot object
class Dot {
  constructor(x,y){
    this.x = x;
    this.y = y;
  }
  on(){
    noStroke();
    fill(color('white'))
    ellipse(this.x, this.y,size,size)
  }
}
```

> Creation of a physical Dot

```
// variables
let w = window.innerHeight;
let h = window.innerWidth;
let size = 2;
let spacing = 5;
let dot = {};

function setup() {
  createCanvas(w, h);
  background('black');
  dot = new Dot(w/2, h/2);
}

function draw() {
  dot.on();
}
```

```
// object
class Dot {
  constructor(x,y){
    this.x = x;
    this.y = y;
  }
  on(){
    noStroke();
    fill(color('white'))
    ellipse(this.x, this.y,size,size)
  }
}
```

2. Single Dot Animation

Here we crete a function which allows us to move the dot in different positions. Using **framCount** as a variable that can change thing over time.

```
// variables
let w = window.innerHeight;
let h = window.innerWidth;
let size = 2;
let spacing = 5;
let dot = {};

function setup() {
  createCanvas(w, h);
  background('black');
  dot = new Dot(w/2, h/2);
}

function draw() {
  dot.on();
  dot.move();
}

// object
class Dot {
  constructor(x,y){
    this.x = x;
    this.y = y;
  }
  on(){
    noStroke();
    fill(color('white'))
    ellipse(this.x, this.y,size,size)
  }
  move(){
        this.x += random(-3,3)
    this.y += random(-3,3)
```

```
  }
}
```

3. Single row generation

Here we create a list of single row dots with a for loop. Once the list of dots is created we experiment with random walks and trasparency. We start feeling the power of computation instantly!

```
// variables
let w = window.innerHeight;
let h = window.innerWidth;
let size = 2;
let spacing = 10;
let dots = [];

function setup() {
  createCanvas(w, h);
  background('black');
  // dot matrix generation
  for(let x = spacing/2; x < w; x += spacing){
    dots.push(new Dot(x, h/2));
  }
}

console.log(dots)

function draw() {
        dots.map((dot)=>{
    dot.on();
    // dot.move();
  })
}

// object
class Dot {
  constructor(x,y){
    this.x = x;
    this.y = y;
  }
  on(){
    noStroke();
    fill(color(255,255,255))
    // fill(color(random(255),21))
    ellipse(this.x, this.y,size,size)
  }
  move(){
       this.x += random(-3,3)
    this.y += random(-3,3)
  }
```

4. Single row Animation

Here we create a list of single row dots with a for loop. Once the list of dots is created we experiment with random walks and trasparency. We start feeling the power of computation instantly!

```
// variables
let w = window.innerHeight;
let h = window.innerWidth;
let size = 2;
let spacing = 10;
let dots = [];

function setup() {
  createCanvas(w, h);
  background('black');
  // dot matrix generation
  for(let x = spacing/2; x < w; x += spacing){
    dots.push(new Dot(x, h/2));
  }
}

console.log(dots)

function draw() {
        dots.map((dot)=>{
    dot.on();
    dot.move();
  })
}

// object
class Dot {
  constructor(x,y){
    this.x = x;
    this.y = y;
  }
  on(){
    noStroke();
    // fill(color(255,255,255))
    fill(color(random(255),21))
    ellipse(this.x, this.y,size,size)
  }
  move(){
        this.x += random(-3,3)
    this.y += random(-3,3)
  }
```

5. Full Matrix generation

Here we create a list of single row dots with a for loop. Once the list of dots is created we experiment with random walks and trasparency. We start feeling the power of computation instantly!

```
// variables
let w = window.innerWidth;
let h = window.innerHeight;
let size = 2;
let spacing = 10;
let dots = [];

function setup() {
  createCanvas(w, h);
  background('black');
  // dot matrix generation
  for(let x = spacing/2; x < w; x += spacing){
    for(let y = spacing/2; y < h; y += spacing) {
      dots.push(new Dot(x, y));
    }
  }
}

console.log(dots)

function draw() {
  dots.map((dot)=>{
    dot.on();
    // dot.move();
  })
}

// object
class Dot {
  constructor(x,y){
    this.x = x;
    this.y = y;
  }
  on(){
    noStroke();
    fill(color('white'))
    // fill(color(random(255),21))
    ellipse(this.x, this.y,size,size)
  }
  move(){
    this.x += random(-3,3)
    this.y += random(-3,3)
  }
}
```

5. Full Matrix animation

Here we create a list of single row dots with a for loop. Once the list of dots is created we experiment with random walks and trasparency. We start feeling the power of computation instantly!

```
}// variables
let w = window.innerWidth;
let h = window.innerHeight;
let size = 2;
let spacing = 10;
let dots = [];

function setup() {
  createCanvas(w, h);
  background('black');
  // dot matrix generation
  for(let x = spacing/2; x < w; x += spacing){
    for(let y = spacing/2; y < h; y += spacing) {
      dots.push(new Dot(x, y));
    }
  }
}

console.log(dots)

function draw() {
  dots.map((dot)=>{
    dot.on();
    dot.move();
  })
}

// object
class Dot {
  constructor(x,y){
    this.x = x;
    this.y = y;
  }
  on(){
    noStroke();
    fill(color('white'))
    // fill(color(random(255),21))
    ellipse(this.x, this.y,size,size)
  }
  move(){
    this.x += random(-3,3)
    this.y += random(-3,3)
  }
}
```