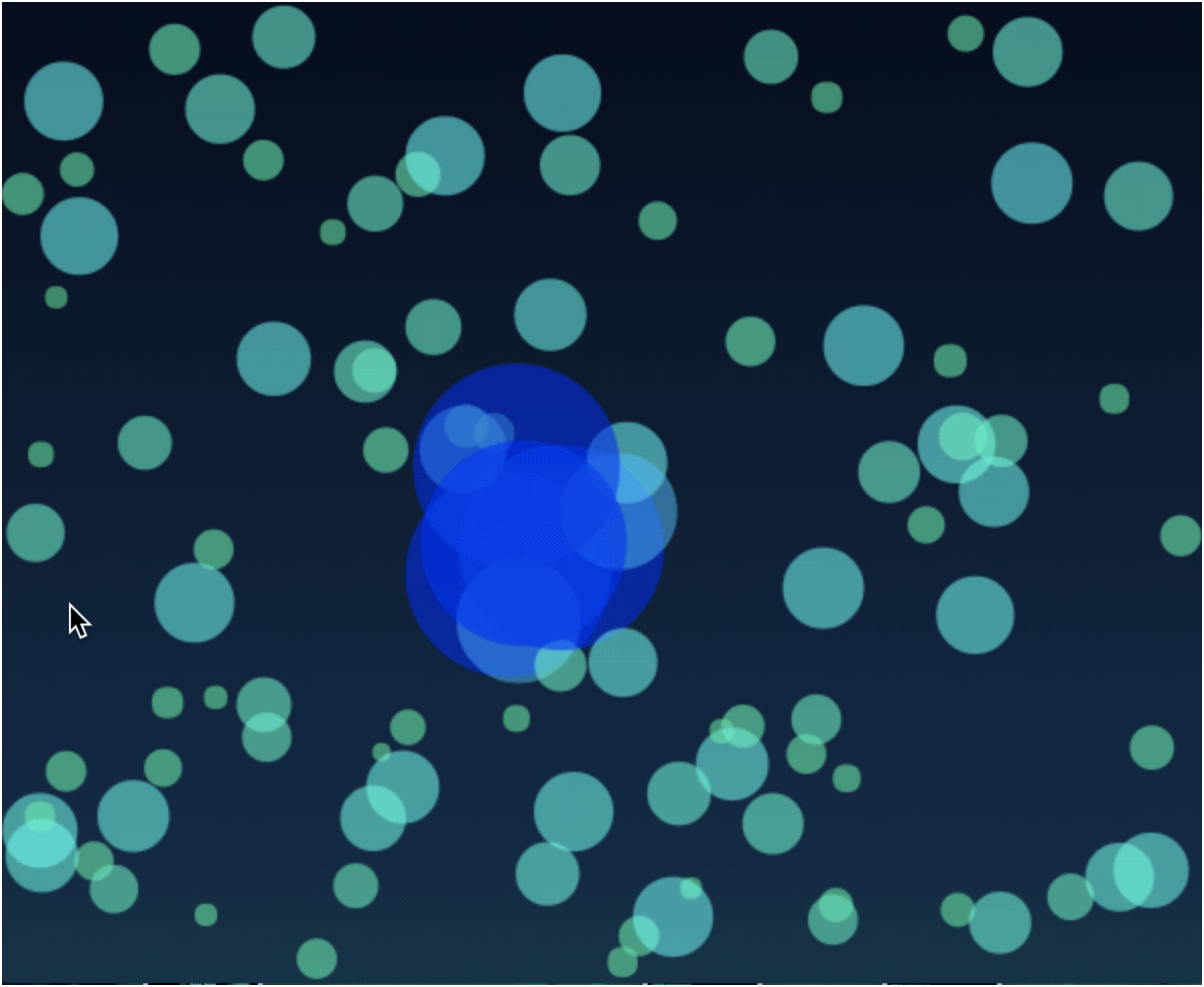
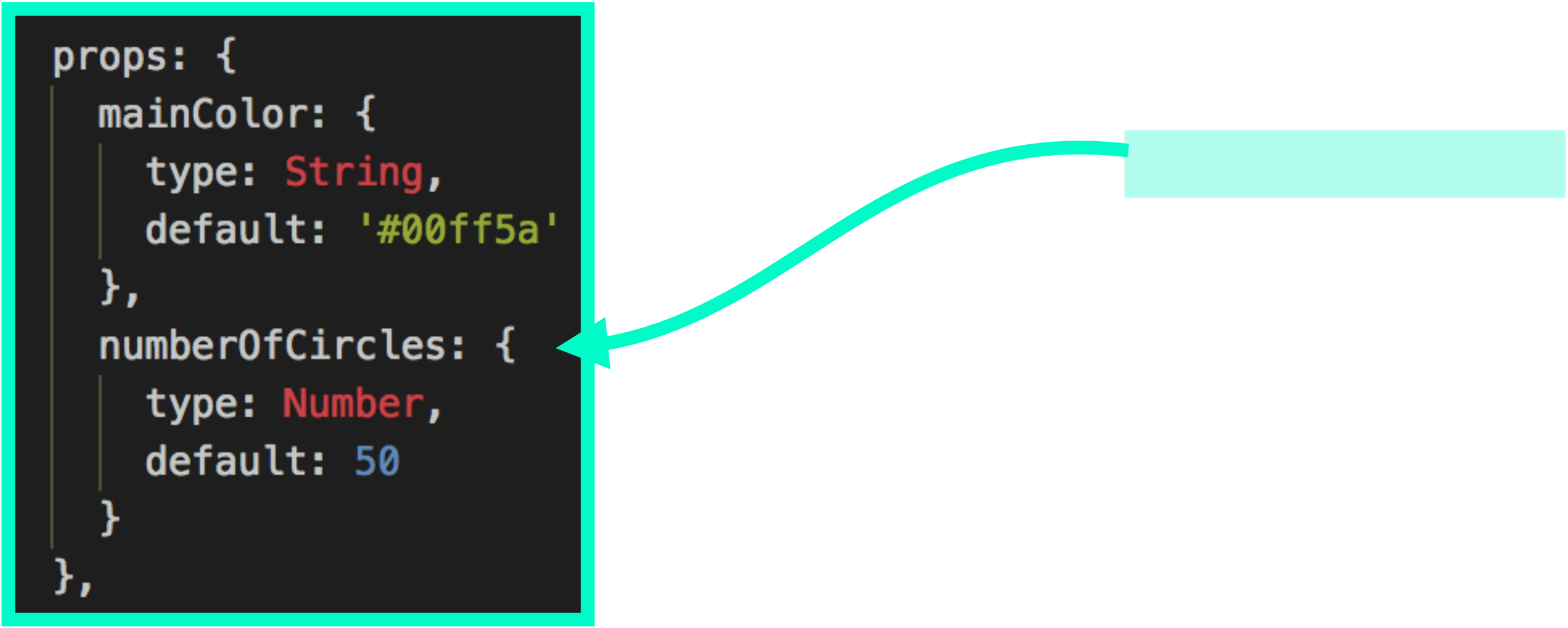
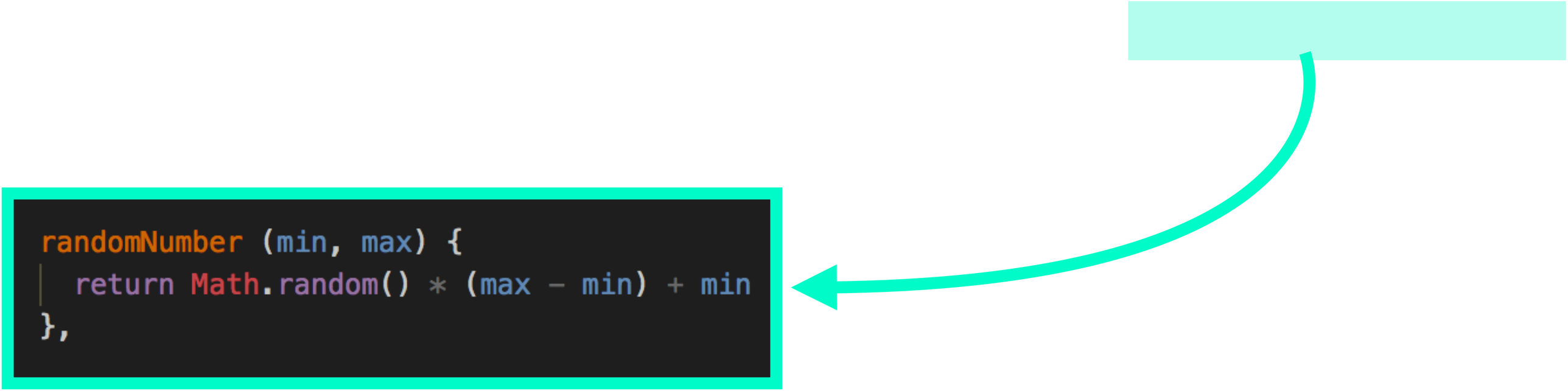
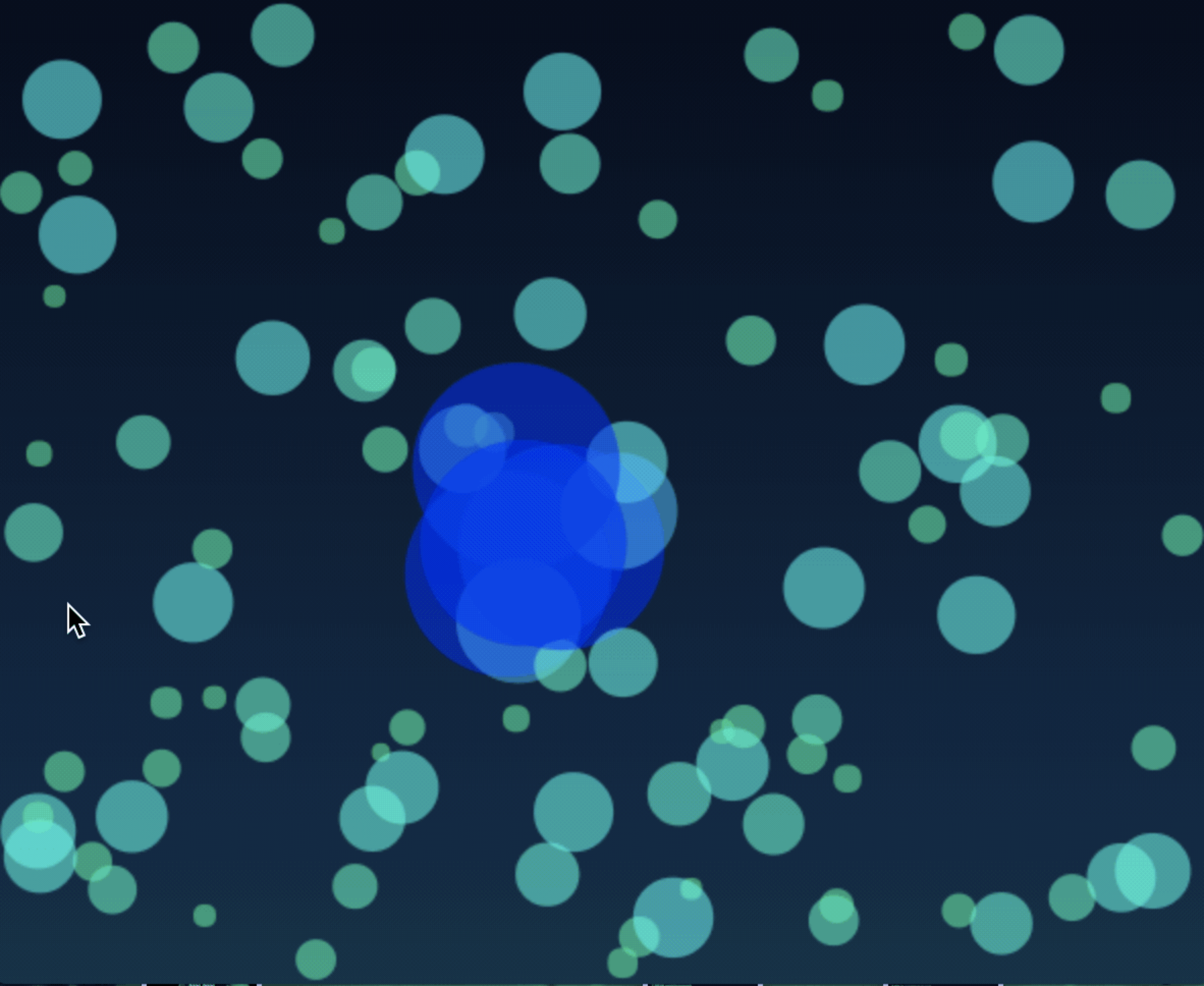
INTEGRATING WITH CANVAS



```
mounted () {
  this.$refs.canvas.width = window.innerWidth
  this.$refs.canvas.height = window.innerHeight

  for (let i = 0; i < this.numberOfCircles; i++) {
    let radius = this.randomNumber(5, 20)
    let diameter = radius * 2;

    let circleObj = {
      radius: radius,
      x: this.randomNumber(diameter, window.innerWidth - diameter),
      y: this.randomNumber(diameter, window.innerHeight - diameter),
      dx: this.randomNumber(-1, 1),
      dy: this.randomNumber(-1, 1),
      color: this.mainColor
    }

    this.circles.push(
      new CreateCircle(circleObj, this.canvasContext, this.mouse)
    )
  }

  this.animate()
}
```


props: {
mainColor: {
type: String,
default: '#00ff5a'
},
numberOfCircles: {
type: Number,
default: 50
}
},


randomNumber (min, max) {
return Math.random() * (max - min) + min
},



```
const CreateCircle = (circleObj, canvasContext, mouse) => {
  return {
    x: circleObj.x,
    y: circleObj.y,
    dx: circleObj.dx,
    dy: circleObj.dy,
    radius: circleObj.radius,
    minRadius: circleObj.radius,
    maxRadius: circleObj.radius * 2,
    draw () {
      canvasContext.beginPath()
      canvasContext.arc(this.x, this.y, this.radius, 0, Math.PI * 2, false)
      canvasContext.fillStyle = Color(circleObj.color).rotate(this.radius * 2).alpha(0.5)
      canvasContext.fill()
    },
    update () {
      if (this.x + this.radius > innerWidth || this.x - this.radius < 0) {
        this.dx = -this.dx
      }

      if (this.y + this.radius > innerHeight || this.y - this.radius < 0) {
        this.dy = -this.dy
      }

      this.x += this.dx
      this.y += this.dy

      this.interactWithMouse()
      this.draw()
    },
    interactWithMouse () {
      if (mouse.x - this.x < 50 && mouse.x - this.x > -50 &&
                mouse.y - this.y < 50 && mouse.y - this.y > -50) {
        if (this.radius < this.maxRadius) {
          this.radius += 1
        }
      } else if (this.radius > this.minRadius) {
        this.radius -= 1
      }
    }
  }
}
```