<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>snow</title>

<style type="text/css">

canvas {

border:1px solid red;

}

</style>

</head>

<body>

<canvas id="canvas" width="500" height="500"></canvas>

<script type="text/javascript">

var canvas = document.getElementById("canvas");

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

var cw = canvas.width;

var ch = canvas.height;

var particles = [];//存储所有粒子实例对象的

function rnd(min, max) {

return parseInt(Math.random() \* (max - min)) + min;

}

function drawBg() {

ctx.fillStyle = "black";

ctx.fillRect(0,0,cw,ch);

}

function Particle() {

//粒子

this.radius = rnd(2,11);//半径

this.x = rnd(0,cw); //坐标x

this.y = - this.radius;//y轴坐标

this.color = "#fff";//粒子颜色

this.speed = rnd(2,5);//粒子下落速度

}

Particle.prototype.draw = function() {

//绘制粒子

ctx.fillStyle = this.color;

ctx.beginPath();

ctx.arc(this.x,this.y,this.radius,0,Math.PI\*2,false);

ctx.closePath();

ctx.fill();

}

Particle.prototype.down = function() {

//粒子下落

this.y += this.speed;

}

function snow() {

ctx.clearRect(0,0,cw,ch);//绘制图案之前先擦除

drawBg();

if(particles.length < 150) {

//生成的粒子对象不超过150

particles.push(new Particle());

}

for(var i in particles) {

particles[i].draw();

particles[i].down();

if(particles[i].y > ch) {

//当粒子超出屏幕，让这个粒子返回到顶部

// particles[i].y = - particles[i].radius;

particles[i] = new Particle();

}

}

window.requestAnimationFrame(snow);

}

window.requestAnimationFrame(snow);

</script>

</body>

</html>