<!DOCTYPE html>

<html><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>💗</title>

<style>

canvas {

position: absolute;

left: 0;

top: 0;

width: 100%;

height: 100%;

background-color: rgba(0, 0, 0, .2);

}

</style>

</head>

<body>

<canvas id="heart" width="1920" height="947"></canvas>

<script>

window.requestAnimationFrame =

window.\_\_requestAnimationFrame ||

window.requestAnimationFrame ||

window.webkitRequestAnimationFrame ||

window.mozRequestAnimationFrame ||

window.oRequestAnimationFrame ||

window.msRequestAnimationFrame ||

(function () {

return function (callback, element) {

var lastTime = element.\_\_lastTime;

if (lastTime === undefined) {

lastTime = 0;

}

var currTime = Date.now();

var timeToCall = Math.max(1, 33 - (currTime - lastTime));

window.setTimeout(callback, timeToCall);

element.\_\_lastTime = currTime + timeToCall;

};

})();

window.isDevice = (/android|webos|iphone|ipad|ipod|blackberry|iemobile|opera mini/i.test(((navigator.userAgent || navigator.vendor || window.opera)).toLowerCase()));

var loaded = false;

var init = function () {

if (loaded) return;

loaded = true;

var mobile = window.isDevice;

var koef = mobile ? 0.5 : 1;

var canvas = document.getElementById('heart');

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

var width = canvas.width = koef \* innerWidth;

var height = canvas.height = koef \* innerHeight;

var rand = Math.random;

ctx.fillStyle = "rgba(0,0,0,1)";

ctx.fillRect(0, 0, width, height);

var heartPosition = function (rad) {

//return [Math.sin(rad), Math.cos(rad)];

return [Math.pow(Math.sin(rad), 3), -(15 \* Math.cos(rad) - 5 \* Math.cos(2 \* rad) - 2 \* Math.cos(3 \* rad) - Math.cos(4 \* rad))];

};

var scaleAndTranslate = function (pos, sx, sy, dx, dy) {

return [dx + pos[0] \* sx, dy + pos[1] \* sy];

};

window.addEventListener('resize', function () {

width = canvas.width = koef \* innerWidth;

height = canvas.height = koef \* innerHeight;

ctx.fillStyle = "rgba(0,0,0,1)";

ctx.fillRect(0, 0, width, height);

});

var traceCount = mobile ? 20 : 50;

var pointsOrigin = [];

var i;

var dr = mobile ? 0.3 : 0.1;

for (i = 0; i < Math.PI \* 2; i += dr) pointsOrigin.push(scaleAndTranslate(heartPosition(i), 210, 13, 0, 0));

for (i = 0; i < Math.PI \* 2; i += dr) pointsOrigin.push(scaleAndTranslate(heartPosition(i), 150, 9, 0, 0));

for (i = 0; i < Math.PI \* 2; i += dr) pointsOrigin.push(scaleAndTranslate(heartPosition(i), 90, 5, 0, 0));

var heartPointsCount = pointsOrigin.length;

var targetPoints = [];

var pulse = function (kx, ky) {

for (i = 0; i < pointsOrigin.length; i++) {

targetPoints[i] = [];

targetPoints[i][0] = kx \* pointsOrigin[i][0] + width / 2;

targetPoints[i][1] = ky \* pointsOrigin[i][1] + height / 2;

}

};

var e = [];

for (i = 0; i < heartPointsCount; i++) {

var x = rand() \* width;

var y = rand() \* height;

e[i] = {

vx: 0,

vy: 0,

R: 2,

speed: rand() + 5,

q: ~~(rand() \* heartPointsCount),

D: 2 \* (i % 2) - 1,

force: 0.2 \* rand() + 0.7,

f: "hsla(0," + ~~(40 \* rand() + 60) + "%," + ~~(60 \* rand() + 20) + "%,.3)",

trace: []

};

for (var k = 0; k < traceCount; k++) e[i].trace[k] = {x: x, y: y};

}

var config = {

traceK: 0.4,

timeDelta: 0.01

};

var time = 0;

var loop = function () {

var n = -Math.cos(time);

pulse((1 + n) \* .5, (1 + n) \* .5);

time += ((Math.sin(time)) < 0 ? 9 : (n > 0.8) ? .2 : 1) \* config.timeDelta;

ctx.fillStyle = "rgba(0,0,0,.1)";

ctx.fillRect(0, 0, width, height);

for (i = e.length; i--;) {

var u = e[i];

var q = targetPoints[u.q];

var dx = u.trace[0].x - q[0];

var dy = u.trace[0].y - q[1];

var length = Math.sqrt(dx \* dx + dy \* dy);

if (10 > length) {

if (0.95 < rand()) {

u.q = ~~(rand() \* heartPointsCount);

} else {

if (0.99 < rand()) {

u.D \*= -1;

}

u.q += u.D;

u.q %= heartPointsCount;

if (0 > u.q) {

u.q += heartPointsCount;

}

}

}

u.vx += -dx / length \* u.speed;

u.vy += -dy / length \* u.speed;

u.trace[0].x += u.vx;

u.trace[0].y += u.vy;

u.vx \*= u.force;

u.vy \*= u.force;

for (k = 0; k < u.trace.length - 1;) {

var T = u.trace[k];

var N = u.trace[++k];

N.x -= config.traceK \* (N.x - T.x);

N.y -= config.traceK \* (N.y - T.y);

}

ctx.fillStyle = u.f;

for (k = 0; k < u.trace.length; k++) {

ctx.fillRect(u.trace[k].x, u.trace[k].y, 1, 1);

}

}

ctx.fillStyle = "rgba(255,255,255,1)";

for (i = u.trace.length + 13; i--;) ctx.fillRect(targetPoints[i][0], targetPoints[i][1], 2, 2);

window.requestAnimationFrame(loop, canvas);

};

loop();

};

var s = document.readyState;

if (s === 'complete' || s === 'loaded' || s === 'interactive') init();

else document.addEventListener('DOMContentLoaded', init, false);

</script>

</body></html>