let nama, val;

const url\_string = document.URL;

const url = new URL(url\_string);

let sender;

if (url.searchParams.get('by') != null) {

sender = url.searchParams.get('by');

} else {

sender = "Silvia";

}

document.querySelector(".tombol").addEventListener('click', function () {

Swal.fire("Hallo Baby", "Aku ada pertanyaan nih buat kamu By", "question").then(function () {

Swal.fire("Jawab yang jujur ya!").then(function () {

Swal.fire("Awas aja kalau kamu bohong", "", "error").then(function () {

const {

value: name

} = Swal.fire({

title: 'Masukin nama kamu dulu',

input: 'text',

inputLabel: '',

showCancelButton: true,

inputValidator: (value) => {

if (!value) {

return 'Isi dulu dong by'

} else {

nama = value;

}

}

}).then(function () {

const pertanyaan = Swal.fire({

title: `${nama} kamu sayang sama ${sender}?`,

showDenyButton: true,

showCancelButton: false,

confirmButtonText: `Sayang`,

denyButtonText: `Enggak`,

}).then((result) => {

/\* Read more about isConfirmed, isDenied below \*/

if (result.isConfirmed) {

Swal.fire(`${sender} juga sayang banget sama ${nama}`).then(function () {

Swal.fire({

title: 'Seberapa sayang emangnya?',

icon: 'question',

input: 'range',

inputLabel: 'Antara 1 - 100 ya',

inputAttributes: {

min: 1,

max: 100,

step: 1

},

inputValue: 50

}).then((e) => {

val = e.value

Swal.fire(`Makasih ya udah sayang sama ${sender} ${val}%`).then(function () {

Swal.fire({

title: `Sekarang ${nama} kangen ga sama ${sender}?`,

showDenyButton: true,

showCancelButton: false,

confirmButtonText: `Kangen :(`,

denyButtonText: `Enggak`,

}).then((result) => {

/\* Read more about isConfirmed, isDenied below \*/

if (result.isConfirmed) {

Swal.fire(`Huhu iya ${sender} juga kangen ${nama} , makasihhh yaa`).then(function () {

Swal.fire('Terakhir deh sayang').then(function () {

Swal.fire('Coba klik ikon hati di paling bawah dong')

})

})

} else if (result.isDenied) {

Swal.fire('yahhh , emang ga kangen sama pacar sendiri', '', 'error').then(function () {

Swal.fire('Yaudah deh okee sayangg :((')

})

}

})

})

})

})

} else if (result.isDenied) {

Swal.fire(`Yakin ga suka sama ${sender}?`, '', 'error').then(function () {

Swal.fire('Yaudah dehh okee sayang :((')

})

}

})

})

});

});

});

});

document.querySelector('.hati').addEventListener('click', function () {

confetti();

const teks = document.getElementById('teks');

const btn = document.querySelector('.tombol');

teks.classList.remove('d-none')

btn.classList.add('d-none')

console.log(teks);

console.log(btn);

})

'use strict';

// If set to true, the user must press

// UP UP DOWN ODWN LEFT RIGHT LEFT RIGHT A B

// to trigger the confetti with a random color theme.

// Otherwise the confetti constantly falls.

var onlyOnKonami = false;

function confetti() {

// Globals

var $window = $(window),

random = Math.random,

cos = Math.cos,

sin = Math.sin,

PI = Math.PI,

PI2 = PI \* 2,

timer = undefined,

frame = undefined,

confetti = [];

var runFor = 2000

var isRunning = true

setTimeout(() => {

isRunning = false

}, runFor);

// Settings

var konami = [38, 38, 40, 40, 37, 39, 37, 39, 66, 65],

pointer = 0;

var particles = 150,

spread = 20,

sizeMin = 5,

sizeMax = 12 - sizeMin,

eccentricity = 10,

deviation = 100,

dxThetaMin = -.1,

dxThetaMax = -dxThetaMin - dxThetaMin,

dyMin = .13,

dyMax = .18,

dThetaMin = .4,

dThetaMax = .7 - dThetaMin;

var colorThemes = [

function () {

return color(200 \* random() | 0, 200 \* random() | 0, 200 \* random() | 0);

},

function () {

var black = 200 \* random() | 0;

return color(200, black, black);

},

function () {

var black = 200 \* random() | 0;

return color(black, 200, black);

},

function () {

var black = 200 \* random() | 0;

return color(black, black, 200);

},

function () {

return color(200, 100, 200 \* random() | 0);

},

function () {

return color(200 \* random() | 0, 200, 200);

},

function () {

var black = 256 \* random() | 0;

return color(black, black, black);

},

function () {

return colorThemes[random() < .5 ? 1 : 2]();

},

function () {

return colorThemes[random() < .5 ? 3 : 5]();

},

function () {

return colorThemes[random() < .5 ? 2 : 4]();

}

];

function color(r, g, b) {

return 'rgb(' + r + ',' + g + ',' + b + ')';

}

// Cosine interpolation

function interpolation(a, b, t) {

return (1 - cos(PI \* t)) / 2 \* (b - a) + a;

}

// Create a 1D Maximal Poisson Disc over [0, 1]

var radius = 1 / eccentricity,

radius2 = radius + radius;

function createPoisson() {

// domain is the set of points which are still available to pick from

// D = union{ [d\_i, d\_i+1] | i is even }

var domain = [radius, 1 - radius],

measure = 1 - radius2,

spline = [0, 1];

while (measure) {

var dart = measure \* random(),

i, l, interval, a, b, c, d;

// Find where dart lies

for (i = 0, l = domain.length, measure = 0; i < l; i += 2) {

a = domain[i], b = domain[i + 1], interval = b - a;

if (dart < measure + interval) {

spline.push(dart += a - measure);

break;

}

measure += interval;

}

c = dart - radius, d = dart + radius;

// Update the domain

for (i = domain.length - 1; i > 0; i -= 2) {

l = i - 1, a = domain[l], b = domain[i];

// c---d c---d Do nothing

// c-----d c-----d Move interior

// c--------------d Delete interval

// c--d Split interval

// a------b

if (a >= c && a < d)

if (b > d) domain[l] = d; // Move interior (Left case)

else domain.splice(l, 2); // Delete interval

else if (a < c && b > c)

if (b <= d) domain[i] = c; // Move interior (Right case)

else domain.splice(i, 0, c, d); // Split interval

}

// Re-measure the domain

for (i = 0, l = domain.length, measure = 0; i < l; i += 2)

measure += domain[i + 1] - domain[i];

}

return spline.sort();

}

// Create the overarching container

var container = document.createElement('div');

container.style.position = 'fixed';

container.style.top = '0';

container.style.left = '0';

container.style.width = '100%';

container.style.height = '0';

container.style.overflow = 'visible';

container.style.zIndex = '9999';

// Confetto constructor

function Confetto(theme) {

this.frame = 0;

this.outer = document.createElement('div');

this.inner = document.createElement('div');

this.outer.appendChild(this.inner);

var outerStyle = this.outer.style,

innerStyle = this.inner.style;

outerStyle.position = 'absolute';

outerStyle.width = (sizeMin + sizeMax \* random()) + 'px';

outerStyle.height = (sizeMin + sizeMax \* random()) + 'px';

innerStyle.width = '100%';

innerStyle.height = '100%';

innerStyle.backgroundColor = theme();

outerStyle.perspective = '50px';

outerStyle.transform = 'rotate(' + (360 \* random()) + 'deg)';

this.axis = 'rotate3D(' +

cos(360 \* random()) + ',' +

cos(360 \* random()) + ',0,';

this.theta = 360 \* random();

this.dTheta = dThetaMin + dThetaMax \* random();

innerStyle.transform = this.axis + this.theta + 'deg)';

this.x = $window.width() \* random();

this.y = -deviation;

this.dx = sin(dxThetaMin + dxThetaMax \* random());

this.dy = dyMin + dyMax \* random();

outerStyle.left = this.x + 'px';

outerStyle.top = this.y + 'px';

// Create the periodic spline

this.splineX = createPoisson();

this.splineY = [];

for (var i = 1, l = this.splineX.length - 1; i < l; ++i)

this.splineY[i] = deviation \* random();

this.splineY[0] = this.splineY[l] = deviation \* random();

this.update = function (height, delta) {

this.frame += delta;

this.x += this.dx \* delta;

this.y += this.dy \* delta;

this.theta += this.dTheta \* delta;

// Compute spline and convert to polar

var phi = this.frame % 7777 / 7777,

i = 0,

j = 1;

while (phi >= this.splineX[j]) i = j++;

var rho = interpolation(

this.splineY[i],

this.splineY[j],

(phi - this.splineX[i]) / (this.splineX[j] - this.splineX[i])

);

phi \*= PI2;

outerStyle.left = this.x + rho \* cos(phi) + 'px';

outerStyle.top = this.y + rho \* sin(phi) + 'px';

innerStyle.transform = this.axis + this.theta + 'deg)';

return this.y > height + deviation;

};

}

function poof() {

if (!frame) {

// Append the container

document.body.appendChild(container);

// Add confetti

var theme = colorThemes[onlyOnKonami ? colorThemes.length \* random() | 0 : 0],

count = 0;

(function addConfetto() {

if (onlyOnKonami && ++count > particles)

return timer = undefined;

if (isRunning) {

var confetto = new Confetto(theme);

confetti.push(confetto);

container.appendChild(confetto.outer);

timer = setTimeout(addConfetto, spread \* random());

}

})(0);

// Start the loop

var prev = undefined;

requestAnimationFrame(function loop(timestamp) {

var delta = prev ? timestamp - prev : 0;

prev = timestamp;

var height = $window.height();

for (var i = confetti.length - 1; i >= 0; --i) {

if (confetti[i].update(height, delta)) {

container.removeChild(confetti[i].outer);

confetti.splice(i, 1);

}

}

if (timer || confetti.length)

return frame = requestAnimationFrame(loop);

// Cleanup

document.body.removeChild(container);

frame = undefined;

});

}

}

$window.keydown(function (event) {

pointer = konami[pointer] === event.which ?

pointer + 1 :

+(event.which === konami[0]);

if (pointer === konami.length) {

pointer = 0;

poof();

}

});

if (!onlyOnKonami) poof();

};