

Реализация анимации хаотичного перемещения частиц на JavaScript

Стрельцова Марина Николаевна

Приамурский государственный университет им. Шолом-Алейхема

Студент

Аннотация

В данной статье реализована анимация хаотичного перемещения частиц на языке программирования JavaScript, а также, в качестве вспомогательных инструментов, использованы html и CSS. На выходе получена анимация, состоящая из точек, которые перемещаются в случайном порядке, соединяясь при этом линиями и создающие привлекательный эффект.

Ключевые слова: хаотичное перемещение, JavaScript, Html, CSS

Implementation of the animation of chaotic movement of particles in JavaScript

Streltsova Marina Nikolaevna

Sholom-Aleichem Priamursky State University

Student

Abstract

This article implements the animation of chaotic movements in the JavaScript programming language, as well as the auxiliary tools used by html and CSS. At the output, the animation of the dots moves in a random order, connecting at this line and creating an attractive effect.

Keywords: chaotic movement, JavaScript, Html, CSS

Пользователей на сегодняшний день трудно удивить строгим и не ярким дизайном сайта. Каждый пытается выделиться на фоне других, применяя различные анимации, привлекающие пользователей своей уникальностью. Именно поэтому важно в современных условиях уметь придумать и реализовать разного рода анимации, которые помогут увеличить количество посетителей на сайте.

В статье В.Е. Селькина рассматриваются алгоритмы анимации элементов интерфейса веб-приложений. Так же описанные результаты в исследование могут помочь в разработке собственной библиотеки анимации на Javascript [1]. Научная работа А.А. Леванова и А. Ю. Винокурова посвящена созданию универсальной программной платформы для разработки мультимедийных электронных учебников на базе Javascript и HTML5 [2]. D. H. Mosley и J. M. André описывают использование JavaScript в примерах квантово-химических приложений, предназначенных для образовательных целей [3]. Создание игрового движка на языках программирования HTML5 и

JavaScript описано в статье Д. Д. Трескина [4]. В работе О. Ю. Лазаревой и Д.Д.Ложкиной описаны графические библиотеки JavaScript, так же рассмотрены возможности библиотек для добавления интерактивности электронным ресурсам [5]. Е. Fritz и Т. Zhao в своей статье представляют семантику стрелочного DSL в JavaScript, который может кодировать асинхронные программы как конечный автомат, где краевые переходы запускаются внешними событиями [6]. Создания эффекта перемещения частиц на языке программирования JavaScript описывается в статье М.Н. Стрельцовой [7]. В работе Л. И. Мустафиной и А. А. Дмитриевской рассмотрены популярные технологии анимирования сайтов, условия их применения и особенности веб-анимации [8].

Целью данной статья является реализация анимации хаотичного перемещения частиц на JavaScript.

JavaScript (JS) — это полноценный динамический язык программирования, который применяется к HTML документу и добавляет интерактивность на веб-сайт [9].

В начале работы создаем файл index.html и подключаем файл стилей style.css и файл анимации script.js (Рис.1).

```
1  <!DOCTYPE html>
2  <html lang="ru">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>Document</title>
8      <link rel="stylesheet" href="style.css">
9  </head>
10 <body>
11     <script src="script.js"></script>
12 </body>
13 </html>
```

Рисунок 1 – Код документа index.html

Файл style.css должен выглядеть следующим образом. (Рис. 2)

```
1  body{
2      margin: 0%;
3      overflow: hidden;
4      background: rgb(17, 17, 19);
5  }
6
```

Рисунок 2 – Код документа style.css

Теперь опишем структуру js файла, который создаст нужную анимацию. Вызовем основную функцию в которой пропишем весь нужный код и объявим все необходимые переменные. Также создадим массив частиц particles и массив свойств, описывающих различные характеристики частиц:

цвет фона, скорость движения, радиус окружности частицы, количество частиц, цвет частиц и другие (Рис. 3).

```
(function() {

    var canvas = document.createElement('canvas'),
        ctx = canvas.getContext('2d'),
        w = canvas.width = innerWidth,
        h = canvas.height = innerHeight,
        particles = [],
        properties = {
            bgColor          : 'rgba(17, 17, 19, 1)',
            particleColor    : 'rgba(255, 40, 40, 1)',
            particleRadius   : 3,
            particleCount    : 60,
            particleMaxVelocity : 0.5,
            lineLength       : 150,
            particleLife     : 6,
        };
};
```

Рисунок 3 – Объявление переменных

Далее напомним функцию, отвечающую за динамическое изменение canvas при увеличении или уменьшении размера окна (Рис. 4).

```
document.querySelector('body').appendChild(canvas);

window.onresize = function() {
    w = canvas.width = innerWidth,
    h = canvas.height = innerHeight;
}
```

Рисунок 4 – Динамическое изменение canvas

Напомним класс particles, который будет отвечать за движение частиц, их начальное положение и время жизни а также отрисовку (Рис. 5).

```
25 class Particle {
26     constructor() {
27         this.x = Math.random()*w;
28         this.y = Math.random()*h;
29         this.velocityX = Math.random()*(properties.particleMaxVelocity*2)-properties.particleMaxVelocity;
30         this.velocityY = Math.random()*(properties.particleMaxVelocity*2)-properties.particleMaxVelocity;
31         this.life = Math.random()*properties.particleLife*60;
32     }
33     position() {
34         this.x + this.velocityX > w && this.velocityX > 0 || this.x + this.velocityX < 0 && this.velocityX < 0 ? this.velocityX*-1 : this.velocityX;
35         this.y + this.velocityY > h && this.velocityY > 0 || this.y + this.velocityY < 0 && this.velocityY < 0 ? this.velocityY*-1 : this.velocityY;
36         this.x += this.velocityX;
37         this.y += this.velocityY;
38     }
39     reDraw() {
40         ctx.beginPath();
41         ctx.arc(this.x, this.y, properties.particleRadius, 0, Math.PI*2);
42         ctx.closePath();
43         ctx.fillStyle = properties.particleColor;
44         ctx.fill();
45     }
46     reCalculateLife() {
47         if(this.life < 1) {
48             this.x = Math.random()*w;
49             this.y = Math.random()*h;
50             this.velocityX = Math.random()*(properties.particleMaxVelocity*2)-properties.particleMaxVelocity;
51             this.velocityY = Math.random()*(properties.particleMaxVelocity*2)-properties.particleMaxVelocity;
52             this.life = Math.random()*properties.particleLife*60;
53         }
54         this.life--;
55     }
56 }
```

Рисунок 5 – Класс частиц

Следующим шагом будет написание функции отрисовки фона с цветом из переменной bgColor массива properties (Рис. 6).

```

58 function reDrawBackground() {
59     ctx.fillStyle = properties.bgColor;
60     ctx.fillRect(0, 0, w, h);
61 }

```

Рисунок 6 – Функция reDrawBackground

Функция drawLines отвечает за отрисовку линий, которые будут в случайном порядке соединять различные точки (Рис. 7).

```

63 function drawLines() {
64     var x1, y1, x2, y2, length, opacity;
65     for(var i in particles) {
66         for(var j in particles) {
67             x1 = particles[i].x;
68             y1 = particles[i].y;
69             x2 = particles[j].x;
70             y2 = particles[j].y;
71             length = Math.sqrt(Math.pow(x2 - x1, 2) + Math.pow(y2 - y1, 2));
72             if(length < properties.lineLength) {
73                 opacity = 1-length/properties.lineLength;
74                 ctx.lineWidth = '0.5';
75                 ctx.strokeStyle = 'rgba(255, 40, 40, '+opacity+')';
76                 ctx.beginPath();
77                 ctx.moveTo(x1, y1);
78                 ctx.lineTo(x2, y2);
79                 ctx.closePath();
80                 ctx.stroke();
81             }
82         }
83     }
84 }

```

Рисунок 7 – Функция drawLines

Каждой частице присвоим методы из класса particles через цикл (Рис. 8).

```

86 function reDrawParticles() {
87     for(var i in particles) {
88         particles[i].reCalculateLife();
89         particles[i].position();
90         particles[i].reDraw();
91     }
92 }
93

```

Рисунок 8 – Функция reDrawParticles

Далее опишем функцию loop, отвечающую за вызов всех функций, написанных ранее. Также будем создавать новые частицы, если их количество меньше заданного в переменной particleCount. В конце вызовем функцию init для запуска анимации (Рис. 9).

```

94  function loop() {
95      redrawBackground();
96      redrawParticles();
97      drawLines();
98      requestAnimationFrame(loop);
99  }
00
01  function init() {
02      for(var i = 0 ; i < properties.particleCount ; i++){
03          particles.push(new Particle);
04      }
05      loop();
06  }
07
08  init();
09
10  }())

```

Рисунок 9 – Функции init и loop

Запустим файл index.html и посмотрим как выглядит анимация (Рис. 10).

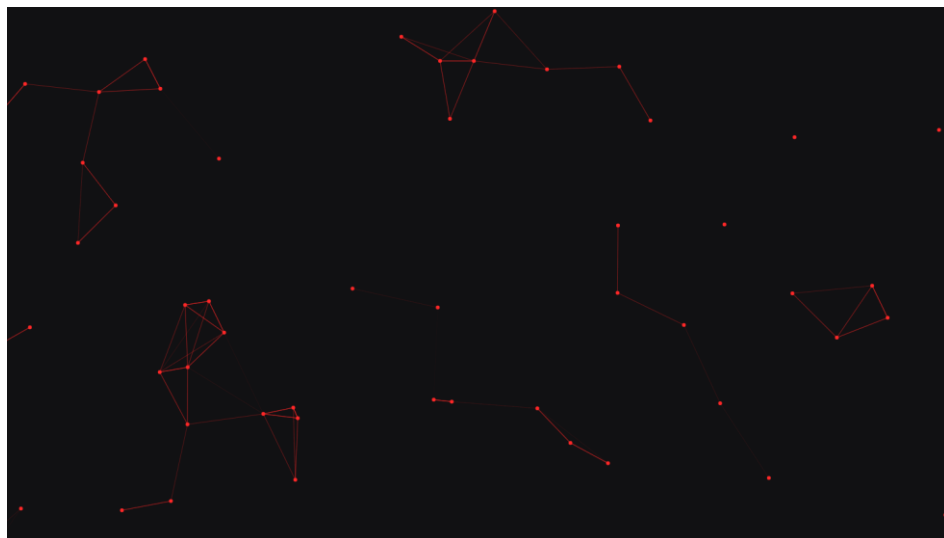


Рисунок 10 – Готовый вид анимации

Изменяя параметры массива properties можно получать анимации разного вида (Рис. 11-12)

```

properties = {
    bgColor          : 'rgba(17, 17, 19, 1)',
    particleColor     : 'rgba(255, 40, 40, 1)',
    particleRadius    : 2,
    particleCount     : 100,
    particleMaxVelocity : 0.3,
    lineLength        : 300,
    particleLife       : 10,
};

```

Рисунок 11 – Массив properties

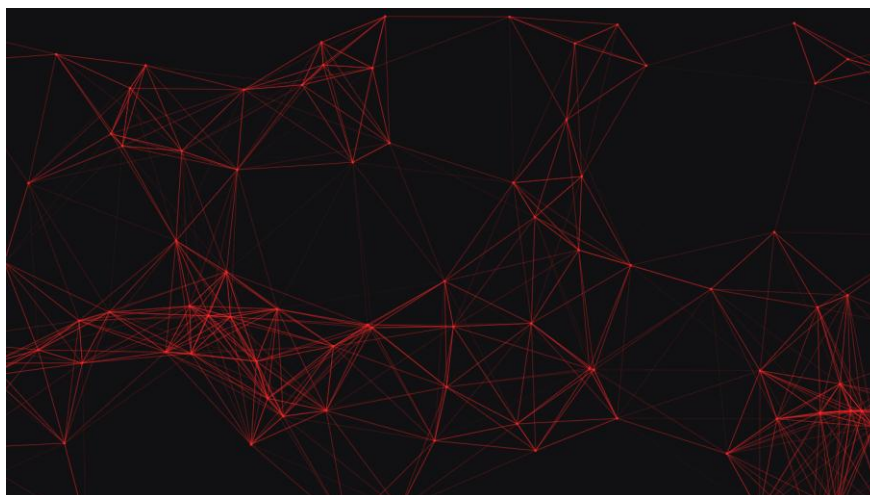


Рисунок 12 – Измененный вид анимации

Таким образом, в рамках данного исследования была реализована анимация хаотичного перемещения частиц с помощью JavaScript, а также рассмотрены особенности языка JS для реализации данной анимации.

Библиографический список

1. Селькин В. Е. Исследование производительности алгоритмов анимации на Javascript // Международный научно-исследовательский журнал. 2014. №. 10 (29) Часть 1. С. 25-27.
2. Леванов А. А., Винокуров А. Ю. Платформа для разработки мультимедийных электронных учебников на базе Javascript и HTML5 // Электронное обучение в непрерывном образовании. 2016. №. 1. С. 279-286.
3. Mosley D. H., André J. M. Use of JavaScript in simple quantum-chemical applications // Journal of Molecular Structure: THEOCHEM. 1997. Т. 419. №. 1-3. С. 57-62.
4. Трескин Д. Д. Создание игрового движка средствами html5 и javascript // Интеллектуальный потенциал XXI века инновационной России. 2019. С. 74-77.
5. Лазарева О. Ю., Ложкина Д. Д. Обзор графических javascript-библиотек для создания интерактивных электронных изданий // Вестник Московского государственного университета печати. 2016. №. 1.
6. Fritz E., Zhao T. Typing and semantics of asynchronous arrows in JavaScript // Science of Computer Programming. 2017. Т. 141. С. 1-39.
7. Стрельцова М. Н. Создание эффекта перемещения частиц на JavaScript // Постулат. 2021. №. 1.
8. Мустафина Л. И., Дмитриевская А. А. Анимации в веб-дизайне: Современное состояние, возможности применения // Альманах научных работ молодых ученых Университета ИТМО. 2017. С. 170-172.
9. JavaScript.com URL: <https://www.javascript.com/> (дата обращения: 19.01.2021).