

Анимация svg графики при помощи css

Вихляев Дмитрий Романович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье рассматриваются анимационные свойства браузера на изображениях векторной графики формата SVG. Для реализации анимации будут использоваться готовые свойства CSS. В результате будет приведён пример анимированного изображения с подробным описанием реализации.

Ключевые слова: CSS, анимация, svg, веб-дизайн.

Animate svg graphics with css

Vikhlyaev Dmitry Romanovich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article discusses the animation properties of the browser on SVG vector graphics images. To implement the animation, ready-made CSS properties will be used. As a result, an example of an animated image with a detailed description of the implementation will be given.

Keywords: CSS, animation, svg, web design.

1 Введение

1.1 Актуальность

SVG (Scalable Vector Graphics – масштабируемая векторная графика) – стандарт векторной графики, разработанный консорциумом W3C, является языком разметки для описания двухмерных графических приложений и изображений. Входит в подмножество языка разметки XML. Графика в формате SVG создается с использованием математических формул, которые при изменении размера изображения можно скорректировать. Таким образом, векторные изображения масштабируются лучше, чем растровые. Размер векторной картинки обычно меньше, чем у сравнимых по качеству изображений в форматах JPEG, GIF или PNG. SVG графика имеет текстовый формат, который можно править как в текстовых, так и в графических редакторах. Так как в графике SVG является текстом, а не изображением, поэтому он индексируется поисковыми системами.

Сейчас формат SVG часто используют для создания иконок карт, логотипов, интерфейсов. Его использование в веб-дизайне постоянно

увеличивается в связи с возможностью легкого связывания с CSS и JavaScript и возможностями анимирования любых частей изображения.

1.2 Обзор исследований

Д.Н.Тимофеев, В.А.Телегин, А.И.Телегин, С.Г.Пудовкина Описали новые функциональные возможности SVG-моделирование управляемых систем тел [1]. В.А.Антошкин, В.А.Морозова разработали графические примитивы для игры "динозавр t-rex" на основе стандарта SVG [2]. А.С.Маринчук продемонстрировал пример создания анимации волны с помощью CSS [3]. М.Е.Кочитов применил анимацию к элементам на веб-странице с помощью свойства TRANSITION в CSS [4].

1.3 Цель исследования

Цель исследования – используя анимационные свойства каскадной таблицы стилей придать видимое движение изображению формата SVG.

2 Материалы и методы

Для реализации используется готовое векторное изображение формата SVG, текстовый редактор поддерживающий HTML, CSS и SVG.

3 Результаты и обсуждения

Каждое изображение формата SVG имеет внутри себя теги для обрисовки основных примитивов. Для понимания браузером, что эти теги являются частью изображения, их вносят внутрь главного тега «svg». Также как и html тег, тег «svg» имеет свои атрибуты, width – ширина, height – высота, fill – цветное заполнение, viewBox – масштабирование под размер контейнера, xmlns – ссылка на стандарт.

Внутренняя часть изображения, как правило, состоит из контейнеров, хранящих в себе отдельные компоненты изображения или изображение целиком. В основном таким контейнером является тег «path». Этот тег имеет атрибуты, в которых хранятся сложные образы. Для группировки основных примитивов и схожих образов используется тег «g» (рис. 1,2).



Рис. 1. Изображение формата svg

```

svg width="625" height="564" viewBox="0 0 625 564" fill="none" xmlns="http://www.w3.org/2000/svg">
  <path d="M298.662 66.4863C288.222 68.3832 236.222 82.707 185.791 131.287C188.722 152.707 159.222 156.707 161.222 202.207C159.96 204.987
  <path d="M193.532 444C175.032 441.5 133.032 441 135.032 431.5C192.032 369 222.532 326.5 232.032 324.5C236.032 331.5 224.032 349.5 218.5
  <path d="M208.86 242.421C210.532 247.5 222.651 288.5 223.943 295C223.943 295.798 222.032 297 220.532 297.5C220.532 298.449 220.963 300.
  <path d="M244.947 463.162C241.947 458.662 141.593 447.431 139.031 446.5C139.532 456 151.032 458 151.032 463.5C157.532 479 126.532 487.5
  <path d="M324.032 379.5C324.946 378.789 404.139 281.281 405.032 280.5C405.032 277.5 398.169 278.427 396.761 278.695C393.032 281.5 249.5
  <path d="M251.648 460C251.032 431.146 274.532 431 274.532 431C278.403 432.5 276.629 436.767 276.391 436.812C249.032 442 257.155 471.625
  <path d="M284.598 443.469C284.532 445.5 274.032 449.5 269.532 442.5C269.512 442.538C268.87 442.345 267.532 442.5 267.032 442.5C266.532
  <path d="M106.5 492.5C109 496.5 110.5 500.5 115 503.5C115.5 505 113 505.5 116 505.5C119 505.5 123.878 503.351 124 502.5C124.214 501.905
  <g>
    <line y1="-0.5" x2="207.535" y2="-0.5" transform="matrix(0.614182 0.789164 -0.803012 0.595963 60.2834 357.654)" stroke="#1576AD"/>
    <line y1="-0.5" x2="210.231" y2="-0.5" transform="matrix(0.859083 0.511836 -0.529866 0.848082 33.4488 385.496)" stroke="#1576AD"/>
    <line y1="-0.5" x2="206.473" y2="-0.5" transform="matrix(0.344824 0.938667 -0.943767 0.330612 87.1181 344.551)" stroke="#1576AD"/>
    <line y1="-0.5" x2="211.22" y2="-0.5" transform="matrix(0.968727 0.248128 -0.259406 0.965769 20.0315 414.976)" stroke="#1576AD"/>
    <line y1="-0.5" x2="207.167" y2="-0.5" transform="matrix(0.291446 -0.956587 0.960291 0.279 92.1496 541.087)" stroke="#1576AD"/>
    <line y1="-0.5" x2="208.773" y2="-0.5" transform="matrix(0.666774 -0.74526 0.760746 0.64905 50.2205 518.157)" stroke="#1576AD"/>
    <line y1="-0.5" x2="210.84" y2="-0.5" transform="matrix(0.914788 -0.403933 0.420184 0.907439 25.063 483.764)" stroke="#1576AD"/>
    <line x1="15" y1="440.5" x2="228" y2="440.5" stroke="#1576AD"/>
    <ellipse cx="121.5" cy="442" rx="17.5" ry="17" fill="#1576AD"/>
    <circle cx="122.5" cy="441.5" r="112.5" stroke="#1576AD" stroke-width="20"/>
  </g>
  <g>
    <line y1="-0.5" x2="207.535" y2="-0.5" transform="matrix(0.614182 0.789164 -0.803012 0.595963 440.283 357.654)" stroke="#1576AD"/>
    <line y1="-0.5" x2="210.231" y2="-0.5" transform="matrix(0.859083 0.511836 -0.529866 0.848082 413.449 385.496)" stroke="#1576AD"/>
    <line y1="-0.5" x2="206.473" y2="-0.5" transform="matrix(0.344824 0.938667 -0.943767 0.330612 467.118 344.551)" stroke="#1576AD"/>
    <line y1="-0.5" x2="211.22" y2="-0.5" transform="matrix(0.968727 0.248128 -0.259406 0.965769 400.031 414.976)" stroke="#1576AD"/>
    <line y1="-0.5" x2="207.167" y2="-0.5" transform="matrix(0.291446 -0.956587 0.960291 0.279 472.15 541.087)" stroke="#1576AD"/>
    <line y1="-0.5" x2="208.773" y2="-0.5" transform="matrix(0.666774 -0.74526 0.760746 0.64905 430.22 518.157)" stroke="#1576AD"/>
    <line y1="-0.5" x2="210.84" y2="-0.5" transform="matrix(0.914788 -0.403933 0.420184 0.907439 405.063 483.764)" stroke="#1576AD"/>
    <line x1="501.5" y1="546" x2="501.5" y2="338" stroke="#1576AD"/>
    <line x1="395" y1="440.5" x2="608" y2="440.5" stroke="#1576AD"/>
    <ellipse cx="501.5" cy="442" rx="17.5" ry="17" fill="#1576AD"/>
    <circle cx="502.5" cy="441.5" r="112.5" stroke="#1576AD" stroke-width="20"/>
  </g>

```

Рис. 2. Текстовый вид svg изображения

Помимо собственных атрибутов тег «svg» и внутренние теги имеют все те же html атрибуты. Одной из главных достоинств SVG графики в веб-дизайне – возможность взаимодействия с CSS. Создавая классы, для каждого образа можно добавлять собственные свойства.

Для создания анимации, в CSS существует свойство animation, которое разделяется на несколько подсвойств. animation-name – устанавливает одну или несколько анимаций для данного элемента. Animation-delay – устанавливает время в секундах или миллисекундах ожидания перед запуском цикла анимации. Animation-duration – задаёт время в секундах или миллисекундах, сколько должен длиться один цикл анимации. Animation-iteration-count – определяет, сколько раз повторять цикл анимации. Animation-timing-function – устанавливает функцию времени для анимации каждого кадра.

В CSS имеется правило «@keyframes», которое устанавливает ключевые кадры при анимации элемента.

Для того чтоб изображение с велосипедом вызывало вид езды вперёд, создаётся анимация, которой необходимо дать имя с помощью свойства animation-name. Данное свойство необходимо для определения анимации. Далее используя свойство animation-duration, задаётся время в секундах, которое определит, как долго анимация будет происходить, притом, чем больше значение времени, тем меньше скорость. Чтобы анимация была непрерывной свойству animation-iteration-count, задаётся значение infinite, в таком случае цикл повторов будет неограничен. Значение linear у свойства animation-timing-function, означает, что функция скорости будет линейной, то есть одинаковой весь период.

Велосипед будет двигаться с лева направо по прямой линии. Правило «@keyframes» позволяет задавать CSS свойства на любом участке анимации. Так как велосипед движется по прямой от начала и до конца можно воспользоваться заготовленными значениями «from» «to», что означает начальный и заключительный момент соответственно. В начальный момент

времени велосипед находится за пределами окна, его крайняя правая точка расположена в нулевой отметке по горизонтали. Таким образом, будет складываться впечатление, что велосипед будто выезжает из невидимой зоны. Для этого в структуре «from» нужно прописать наружный левый отступ со значением противоположным ширине изображения. Чтобы велосипед переместился с левого края окна в правый край. Независимо от заданного размера браузера, необходимо в структуру «to» указать левый отступ, равный 100 процентам (рис. 3).

```
svg{
  margin: 0px;
  width: 300px;
  animation-name: go;
  animation-duration: 5s;
  animation-iteration-count: infinite;
  animation-timing-function: linear;
}
@keyframes go {
  from {
    margin-left: -300px;
  }
  to {
    margin-left: 100%;
  }
}
```

Рис. 3. Анимация движения велосипеда по горизонтали

Придав вращение колёсам велосипеда, можно получить более реалистичную анимацию. Объединив оба контейнера, в которых лежат примитивы частей колёс в один класс, можно легко придать анимацию сразу двум скоплениям примитивов. За вращение объектов в CSS, отвечает свойство «transform». В данном случае вращение колёс необходимо задать относительно оси z. Функция «rotateZ()» отвечает заданным требованиям, поворачивая объект на заданный угол в градусах. Так как колёса крутятся постоянно по часовой стрелке то в начальный момент «to» в качестве параметров указывается значение ноль. Для полного оборота колёс в заключительный момент «from» задаётся значений равное 360 градусам. Если запустить программу, то вращение осуществиться, однако центром вращения по умолчанию служит верхний левый угол контейнера, в данном случае всего изображения. Чтобы это исправить, свойству «transform-box» необходимо указать значение «fill-box» что приводит к аналогичному вращению, но уже относительно собственного контейнера. Затем изменив центр вращения с левого верхнего угла в центр с помощью свойства «transform-origin», указав значение «center», после чего вращения колёс станет подобным настоящим (рис. 4,5).

```
.koleso{
  animation-name: wheel;
  animation-duration: 2s;
  animation-iteration-count: infinite;
  animation-timing-function: linear;
  transform-origin: center;
  transform-box: fill-box;
}
@keyframes wheel{
  from {transform: rotateZ(0deg);}
  to { transform: rotateZ(360deg); }
}
```

Рис. 4. Анимация вращения колёс велосипеда



Рис. 5. Результат анимации

Таким образом, была реализована анимация графического изображения формата SVG с полным описанием ключевых действий.

Исходный код данной анимации можно скачать по ссылке [URL:https://drive.google.com/file/d/1BKSP6yR0CsMwrvXfxs4ctgGxEDvekr9/view?usp=sharing](https://drive.google.com/file/d/1BKSP6yR0CsMwrvXfxs4ctgGxEDvekr9/view?usp=sharing).

Библиографический список

1. Тимофеев Д.Н., Телегин В.А., Телегин А.И., Пудовкина С.Г. SVG-моделирование управляемых систем тел// Вестник Южно-Уральского государственного университета. Серия: Компьютерные технологии, управление, радиоэлектроника. 2019. Т. 19. № 4. С. 156-165.
2. Антошкин В.А., Морозова С.И. Разработка графических примитивов для игры "динозавр T-REX" на основе стандарта SVG// Информатика и прикладная математика. 2021. № 27. С. 10-15.

3. Маринчук А.С. Создание анимации волны с помощью CSS// Постулат. 2021. № 1 (63).
4. Кочитов М.Е. Применение анимации к элементам на веб-странице с помощью свойства TRANSITION в CSS// Постулат. 2019. № 8 (46). С. 31.
5. Гербек А.Н. Анимация кнопки SVG в адаптивном меню// В сборнике: наука и общество: проблемы и перспективы. сборник научных трудов III Международная научно-практическая конференция. 2018. С. 6-9.
6. Богомолов Н.А., Ковалев А.Д., Сеницын М.Н. Использование стандарта SVG для визуализации данных в интернет// В сборнике: Научный сервис в сети Интернет. Труды Всероссийской научной конференции . Московский государственный университет им. М.В.Ломоносова, Ростовский государственный университет, Институт вычислительной математики РАН. 2003. С. 193-195.