

ИКТ В НОС

Библиотека СУИКА

Тема №8

Графика в браузер



Какво е WebGL?

- Технология за 3D графика директно в браузър

Защо WebGL?

- Платформено независим – от смартфони до компютри
- Тръгва в браузър и не иска специализирана работна среда
- Използва графичния хардуер



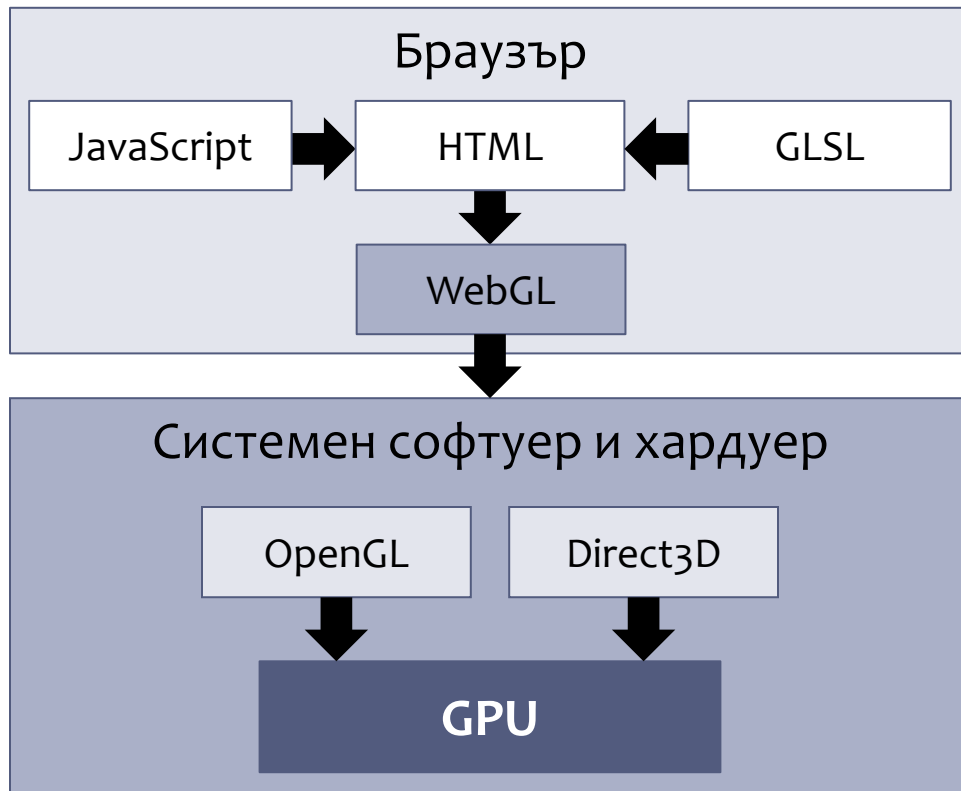
Какво е СУИКА?

- Библиотека за 3D графика, използваща WebGL

Защо СУИКА?

- Лесни за използване обекти
- Скрива сложността на WebGL
- Реализира някои от елементите в типичната архитектура на интерактивно графично онлайн приложение

Архитектура на WebGL приложение



Приложение на СУИКА

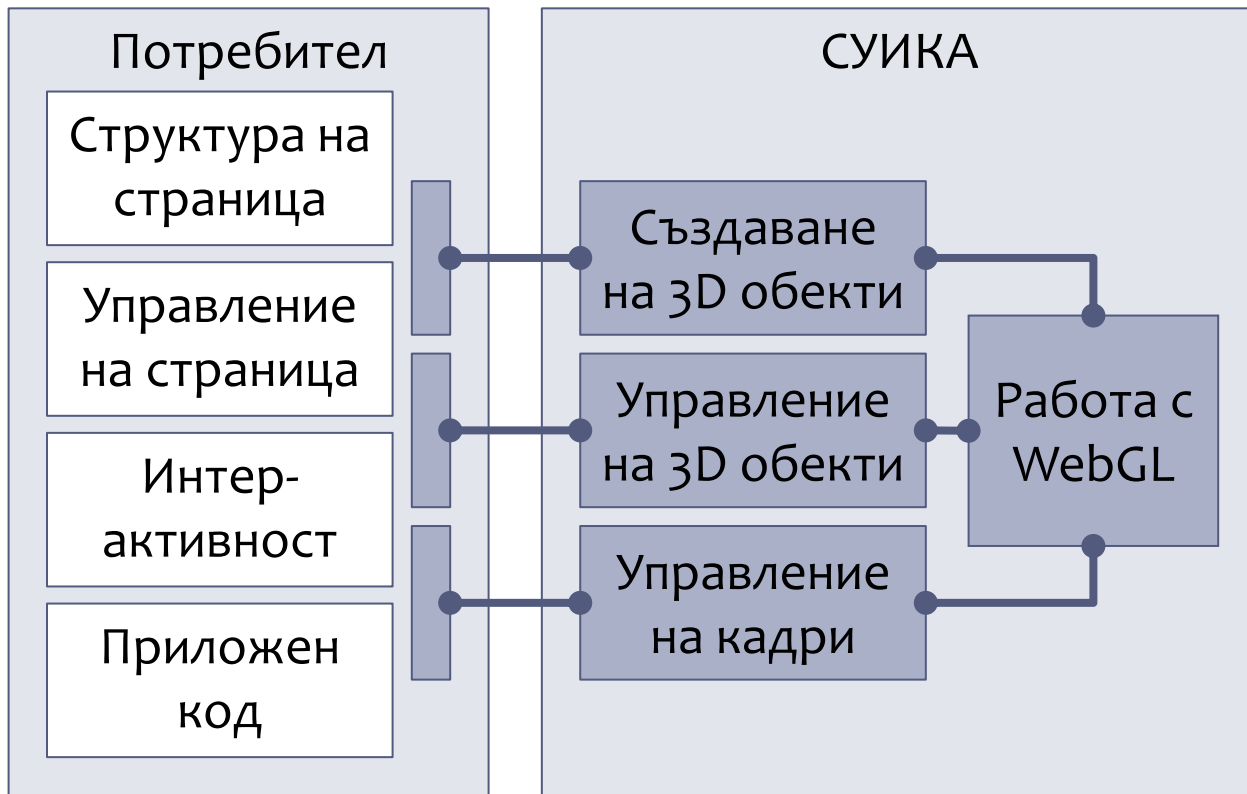


Библиотеката СУИКА поема

- Създаване на базисни 3D обекти
- Управление на техните свойства
- Генериране на изображения и сменяне на кадри

За потребителя остава

- Структура и управление на страницата
- Интерактивност и обработване на събития
- Същинската логика на приложението



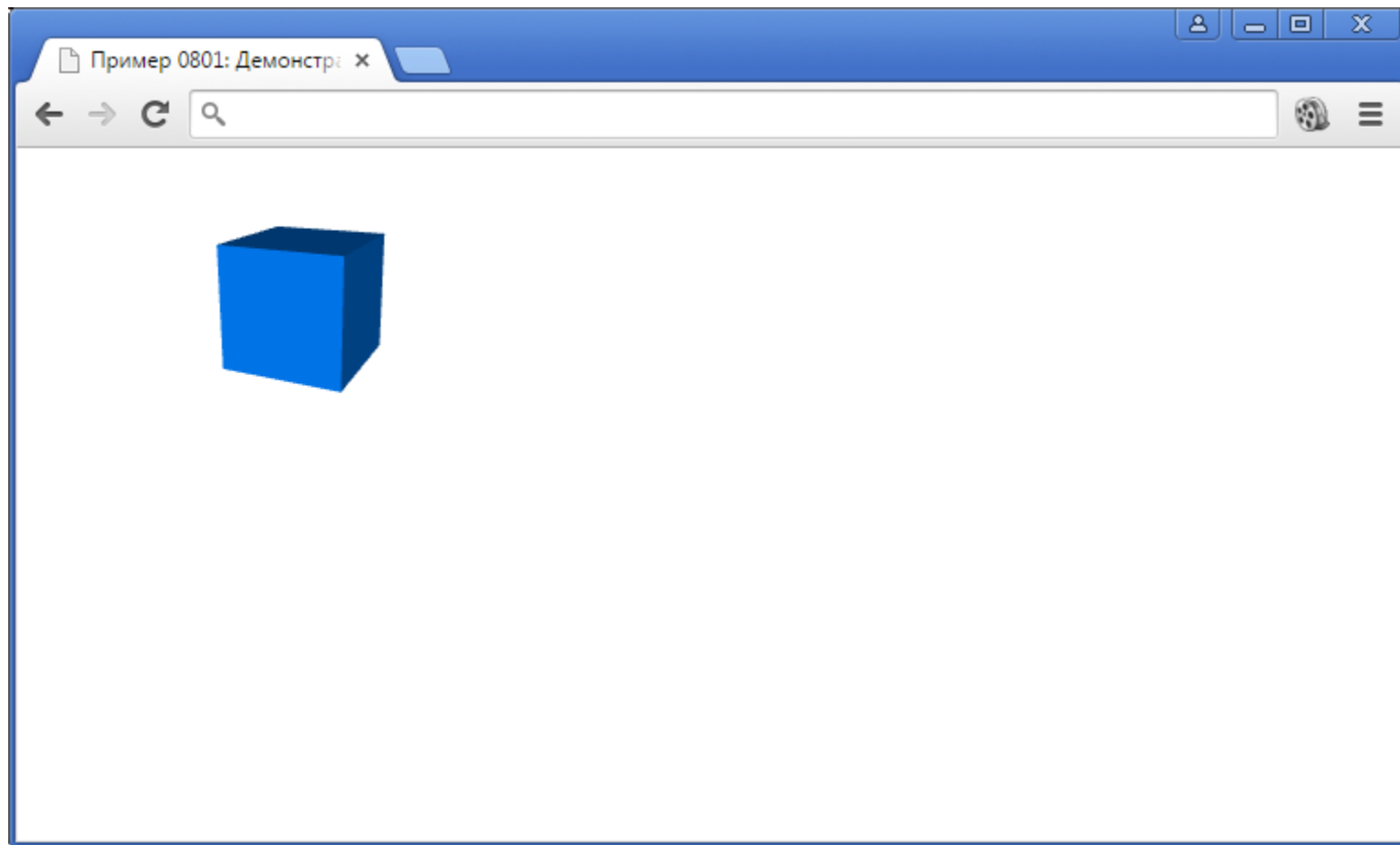
Само за демонстрация



Минимална програма

- Създава и върти син куб

```
function main()  
{  
    new Suica();  
    cube([0,0,0],1);  
    demo(4);  
}
```



ПРОБА

Само за илюстрация

Код на СУИКА приложение

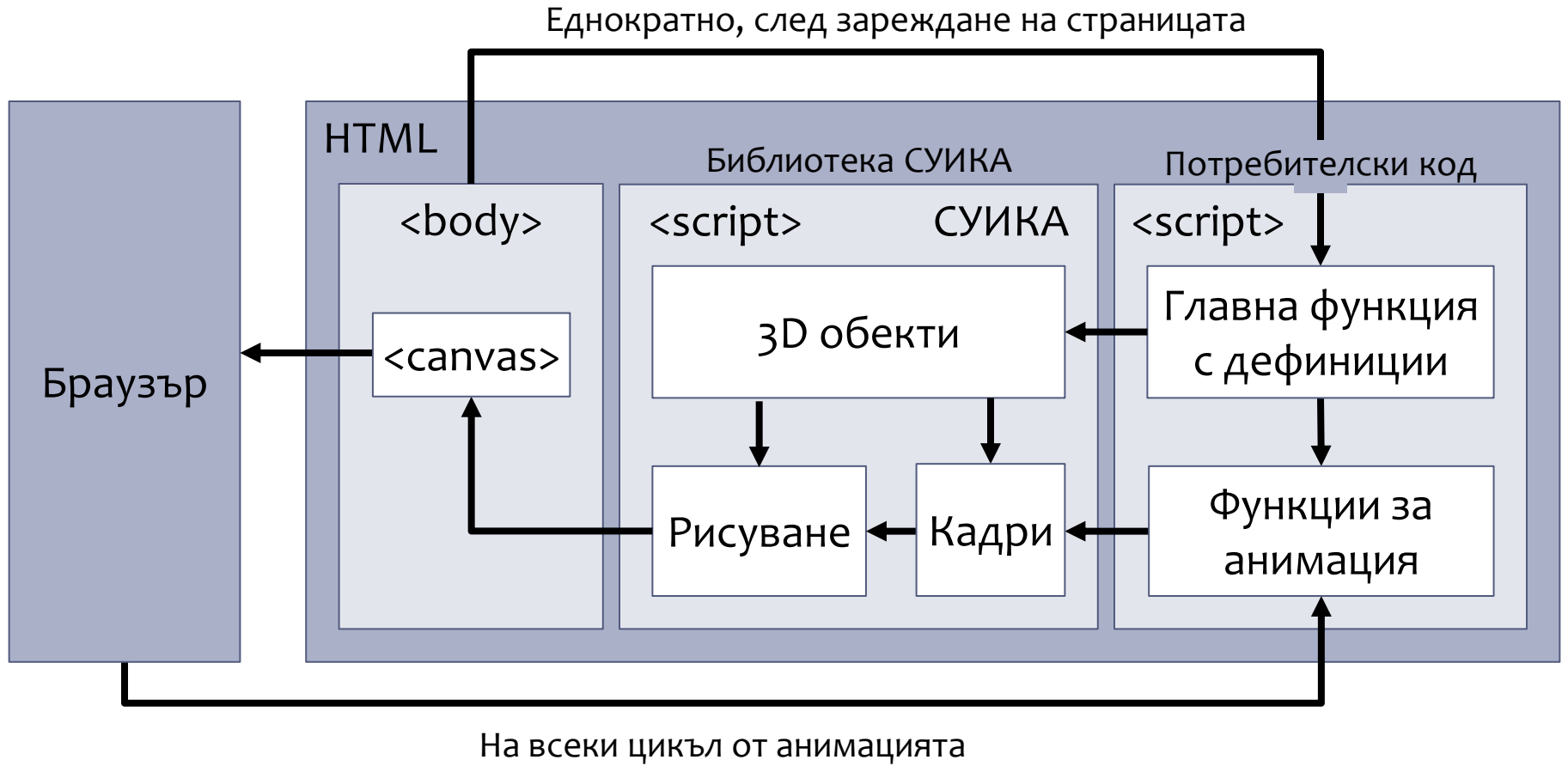


Основни елементи

- Библиотека СУИКА
- Главна функция
- Функции за анимация (незадължителни)
- Активиране на главната функция
- Елемент <canvas> за показване на графика

Детайли по основните елементи

- Библиотеката се включва с елемент `<script>`
- Главната и допълнителните функции са в друг `<script>`
- Елементът `<body>` съдържа елемент `<canvas>` и единствено в него може да се рисува 3D съдържание
- След зареждането на `<body>` се активира събитие `onload`, което извиква еднократно главната функция
- Тя създава графичните обекти и описва техните свойства
- Ако има функции за анимация, те работят непрекъснато



Инициализация



Скриптове в елемента <head>

- Включване на библиотеката
- Минимизиран вариант **suica.min.js**
- Пълен вариант (с коментари) **suica.js**

```
<head>  
  <script src="suica.min.js"></script>  
</head>
```

Програма

- Включена в отделен скрипт в <head>
- Трябва да има главна функция
- Ние избираме името ѝ, но по традиция е **main**

```
<head>
  <script src="suica.min.js"></script>
  <script>
    function main()
    {
      ...
    }
  </script>
</head>
```


В елемента <body>

- Инициализацията на СУИКА очаква заредена страница
- Улавяне на събитието **onload** след зареждане на страницата
- Извикване на главната функция
- С елемента <noscript> се проверява и предупреждава при невъзможност за ползване на JavaScript

```
<body onload="main()">
```

```
  <noscript>
```

```
    Искаме JavaScript, но няма!
```

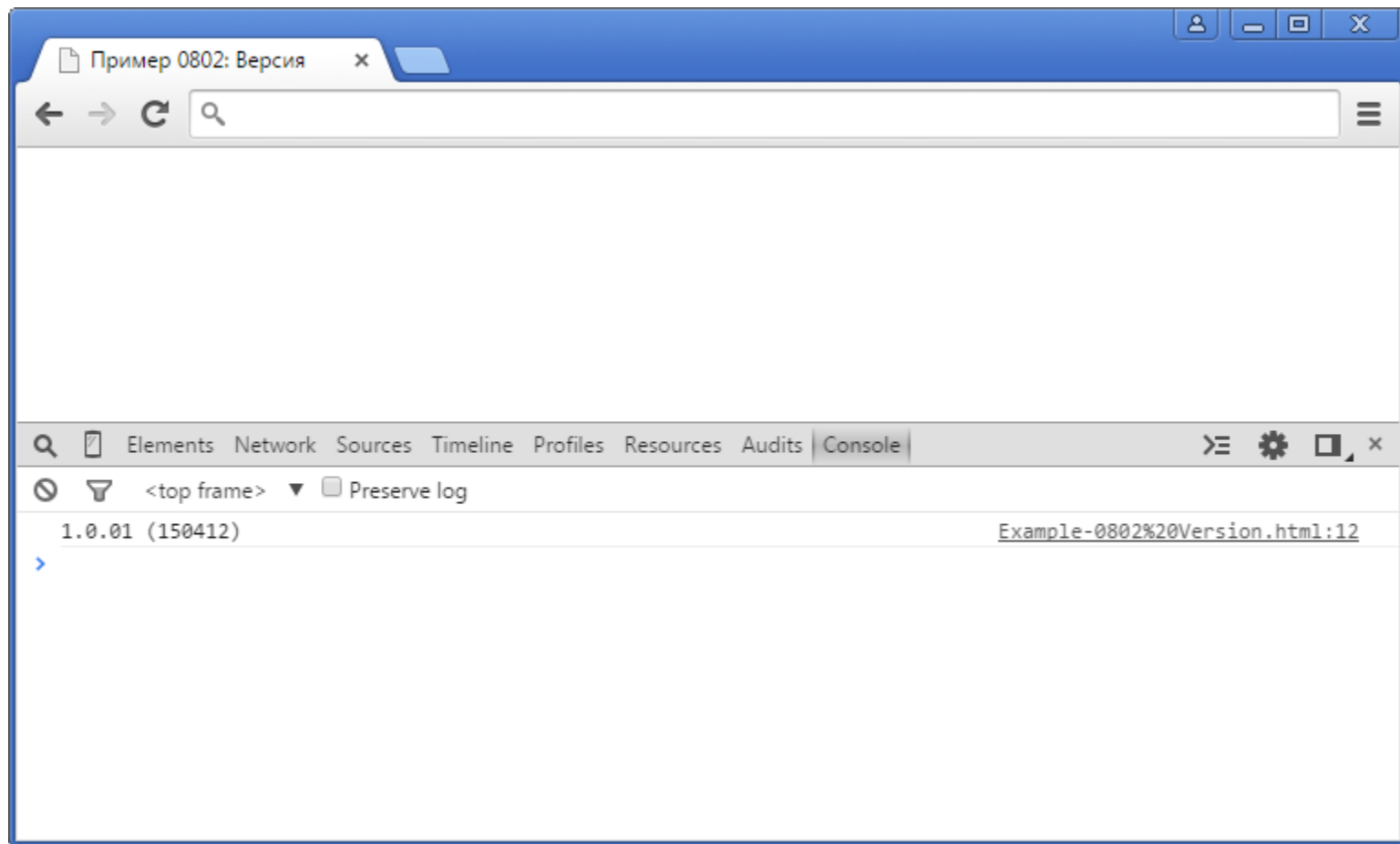
```
  </noscript>
```

```
</body>
```

В главната функция

- Ще покажем версията на библиотеката

```
function main()  
{  
  console.log(Suica.version);  
}
```



ПРОБА

Графичен прозорец



Елемент <canvas>

- WebGL рисува само в елемент <canvas>
- Всеки <canvas> е в ролята на графичен прозорец
- Създаваме и настройваме елемента (размер, рамка, ...)
- Проверка и предупреждение, ако не се поддържа <canvas>

```
<canvas width="740" height="350"  
        style="border: solid 1px Black;">
```

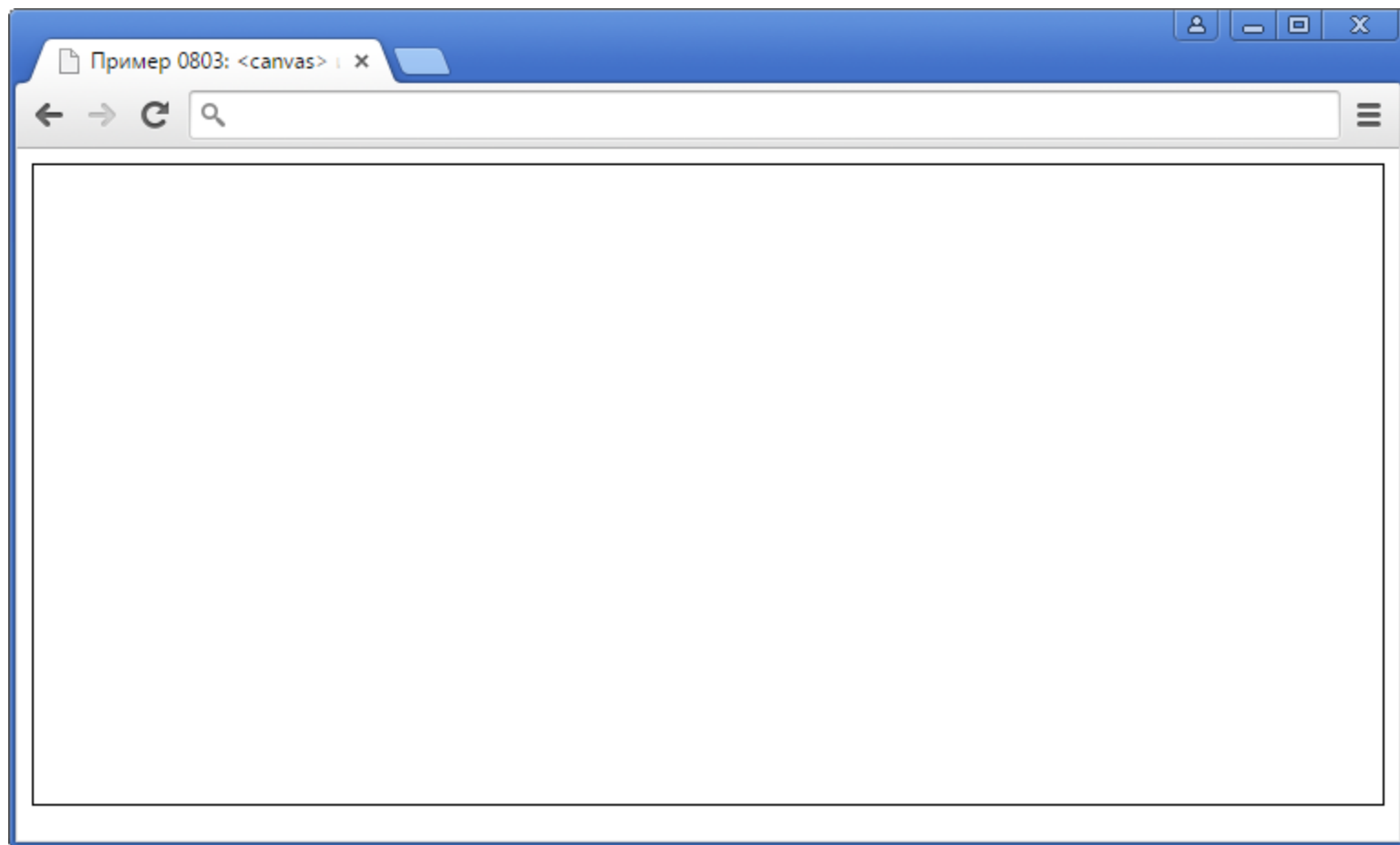
Искаме canvas, но няма!

```
</canvas>
```

Suica инстанция

- В библиотеката СУИКА има дефиниция на обект **Suica**
- За всеки <canvas> трябва да има инстанция на Suica
- Създаването на инстанция е с new

```
function main()
{
    new Suica();
}
```



ПРОБА

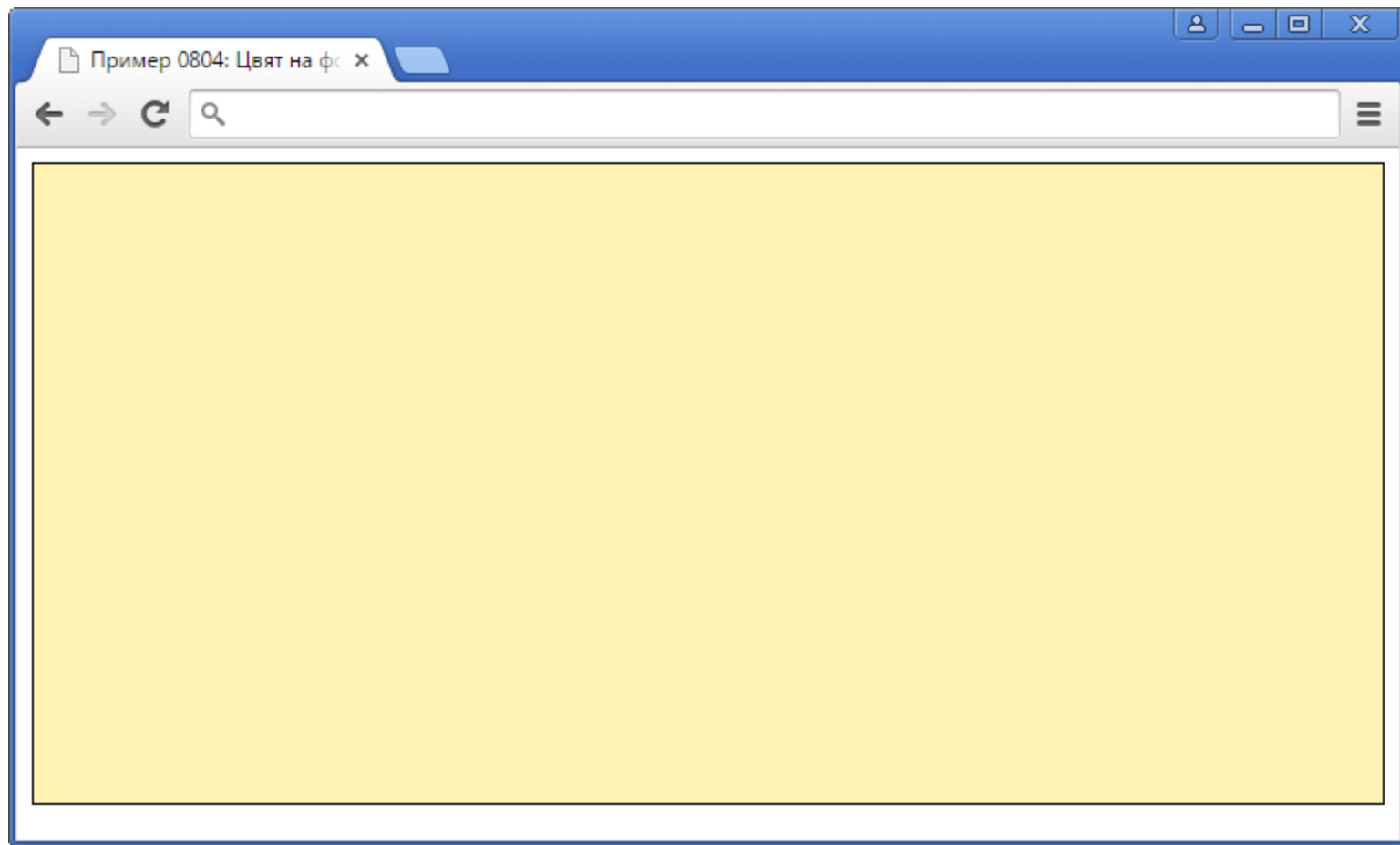
Съдържание на <canvas>

- Не може да се контролира с HTML и CSS
- Контролира се само през съответната инстанция на `Suica`

Цвят на фона

- Цветовете са масиви с три числа – червено, зелено и синьо
- Стойностите им са дробни числа от 0 до 1
- Цветът дефинираме с `background (цвят)`

```
background([1,0.95,0.7]);
```

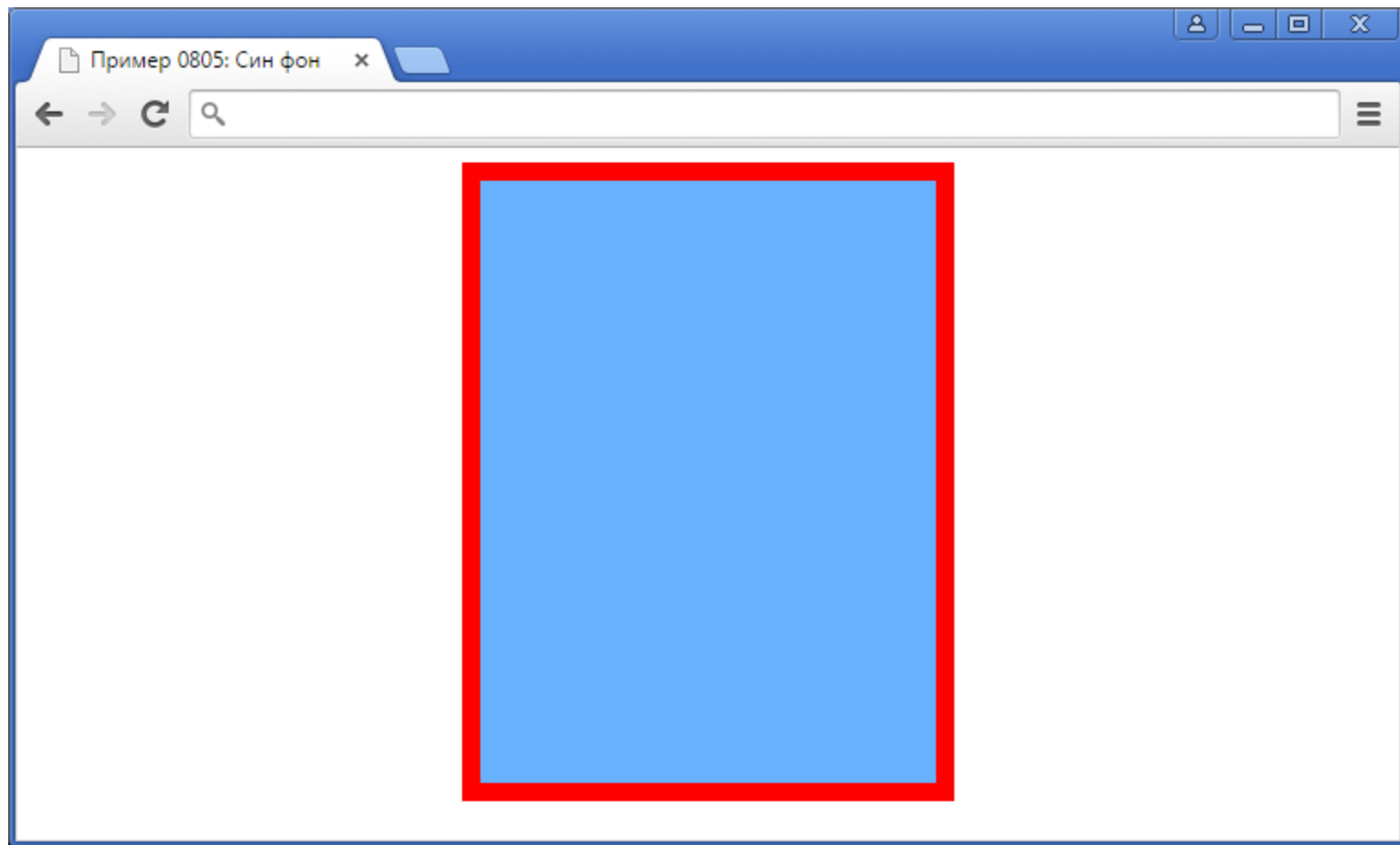


ПРОБА

Пример с други параметри

- Синкав фон
- Центрирано графично поле
- Дебела червена рамка

```
<style>
  body    { text-align: center; }
  canvas  { border: solid 10px Red; }
</style>
:
background([0.4,0.7,1.0]);
:
<canvas width="250" height="330">...</canvas>
```



ПРОБА



Цветове



Цветове в СУИКА

- Използват цветовия модел RGB
- Масив от три компоненти: **червена (R)**, **зелена (G)** и **синя (B)**
- Всяка компонента е число от 0 до 1
- Минимален интензитет при 0, максимален – при 1



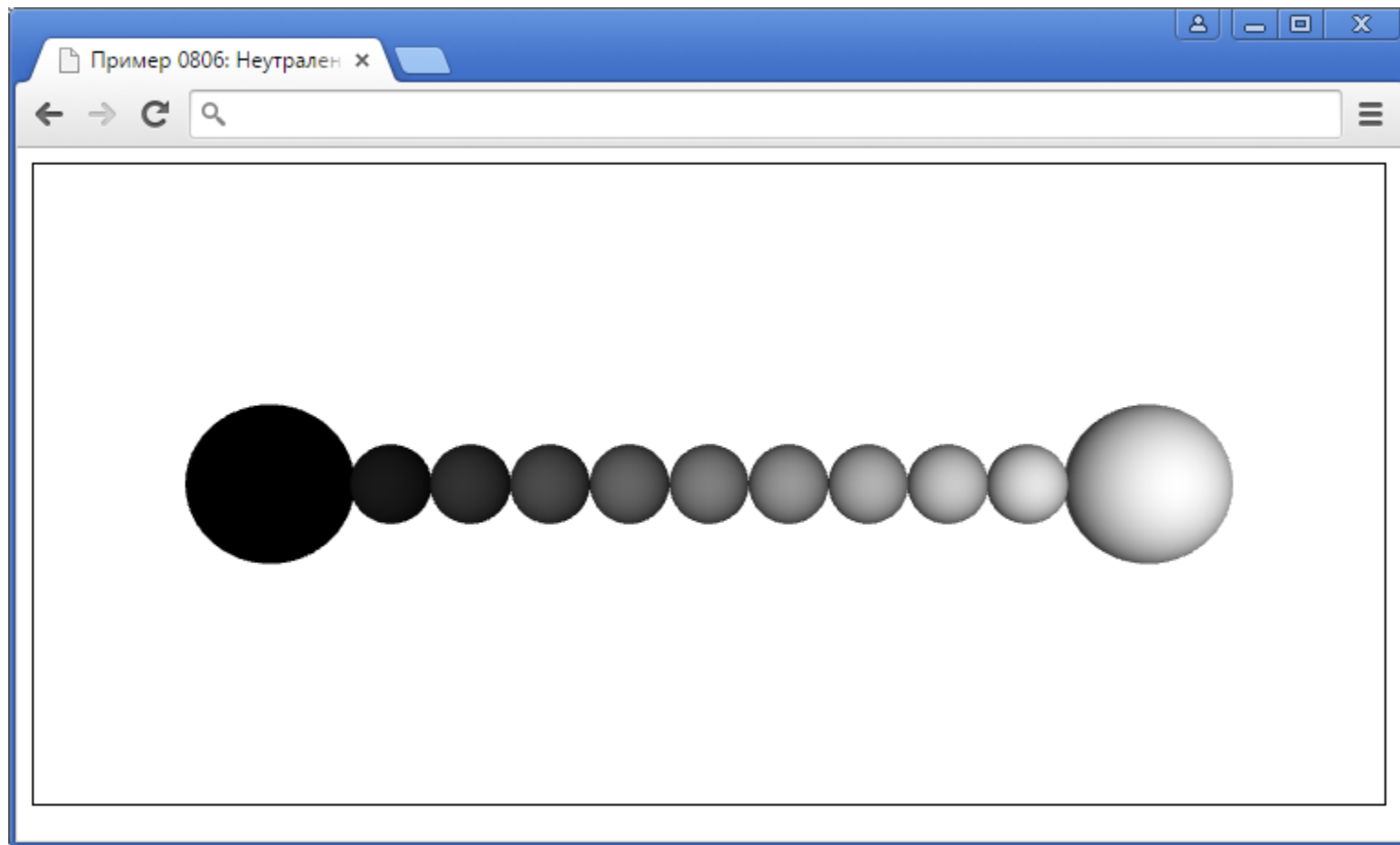
Неутрални цветове

Неутрални цветове $R=G=B$

- Бяло $(1,1,1)$ и черно $(0,0,0)$
- Всички степени на сивото (x,x,x)

Калибриране

- Ако мониторът не е калибриран цветово, неутралните цветове могат да имат оттенък



ПРОБА

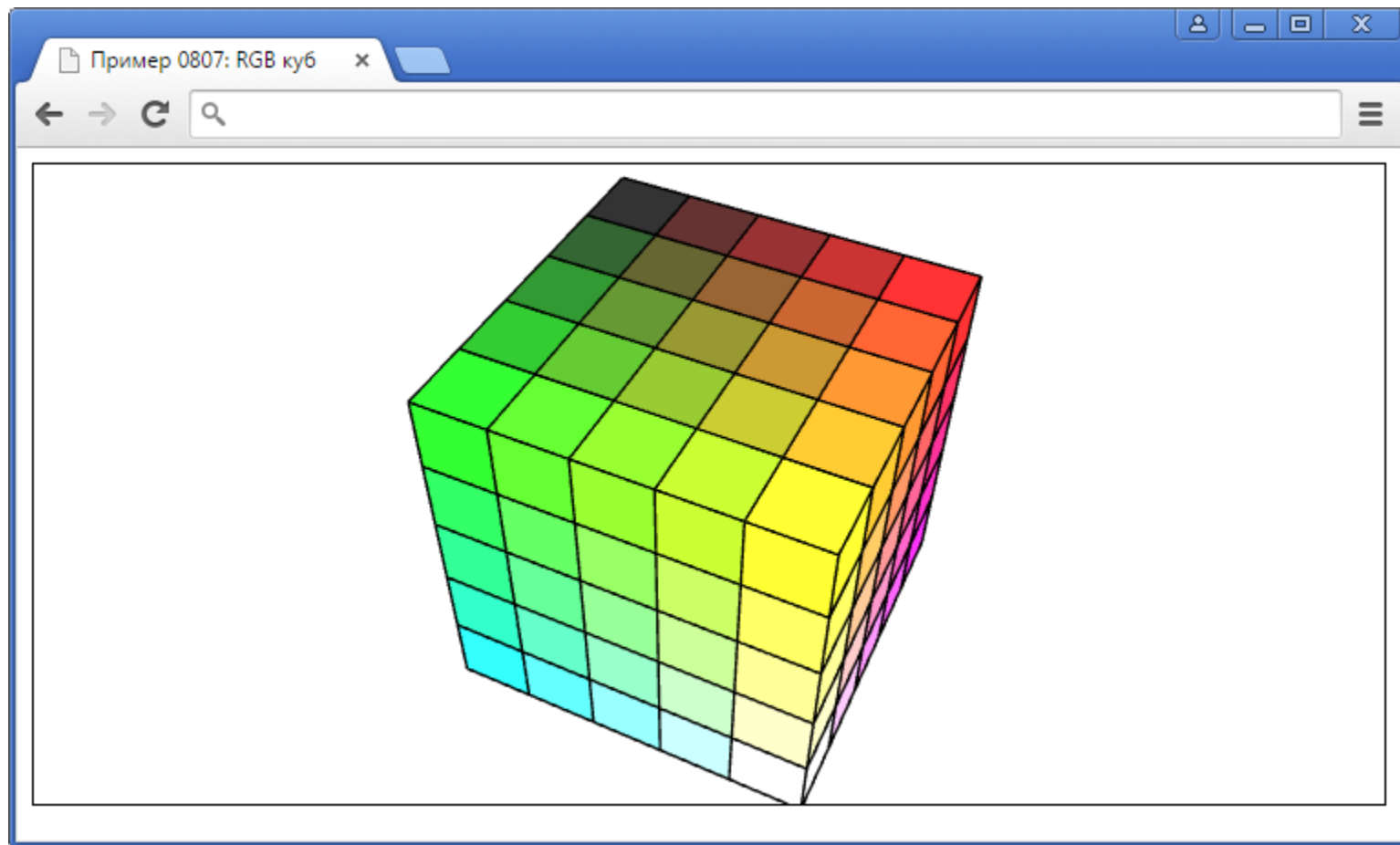
Само за илюстрация

Цветова аритметика



Основни цветове

- 0 0 0 = черно
- 1 0 0 = червено
- 0 1 0 = зелено
- 0 0 1 = синьо
- 1 1 0 = червено+зелено = жълто
- 1 0 1 = червено+синьо = лилаво
- 0 1 1 = зелено+синьо = синьозелено
- 1 1 1 = червено+зелено+синьо = бяло



ПРОБА

Само за илюстрация

Помощни команди

Координатна система



Координатна система на СУИКА

- Тримерна и декартова
- Винаги съществува
- Посоките на осите са относителни

Показване на координатната система

- Не може, тя е абстрактно понятие
- Но можем да нарисуваме подходящи обекти

Помощна команда oxuz

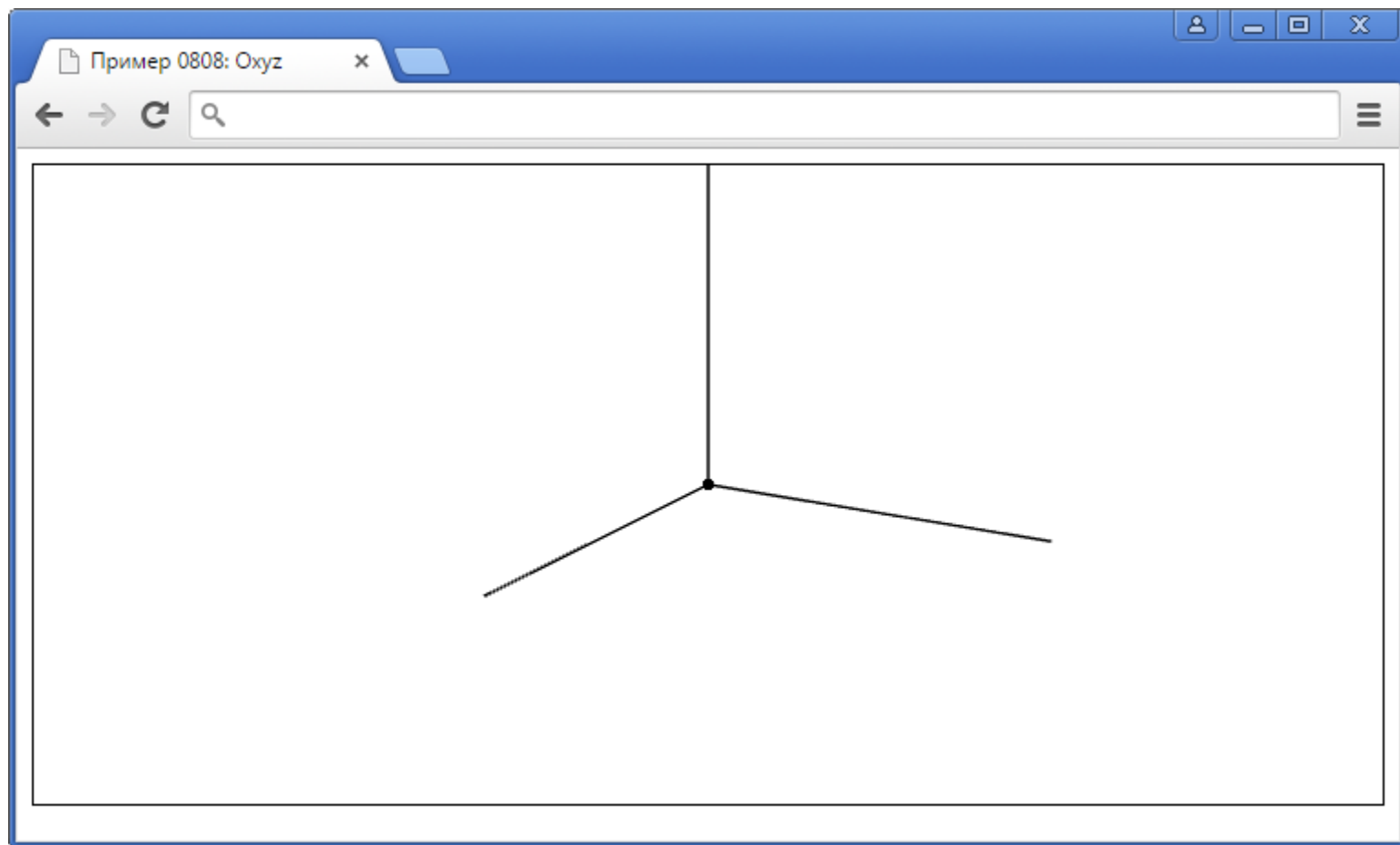
- Създава точка в началото на координатната система и три отсечки по трите оси
- Параметър е дължината на осите, ако липсва е 30

```
oxuz();
```

```
oxuz(100);
```

Ориентация

- Началото е в средата на <canvas>
- X е надолу и наляво, Y е надясно, Z е нагоре



ПРОБА



Помощна команда demo

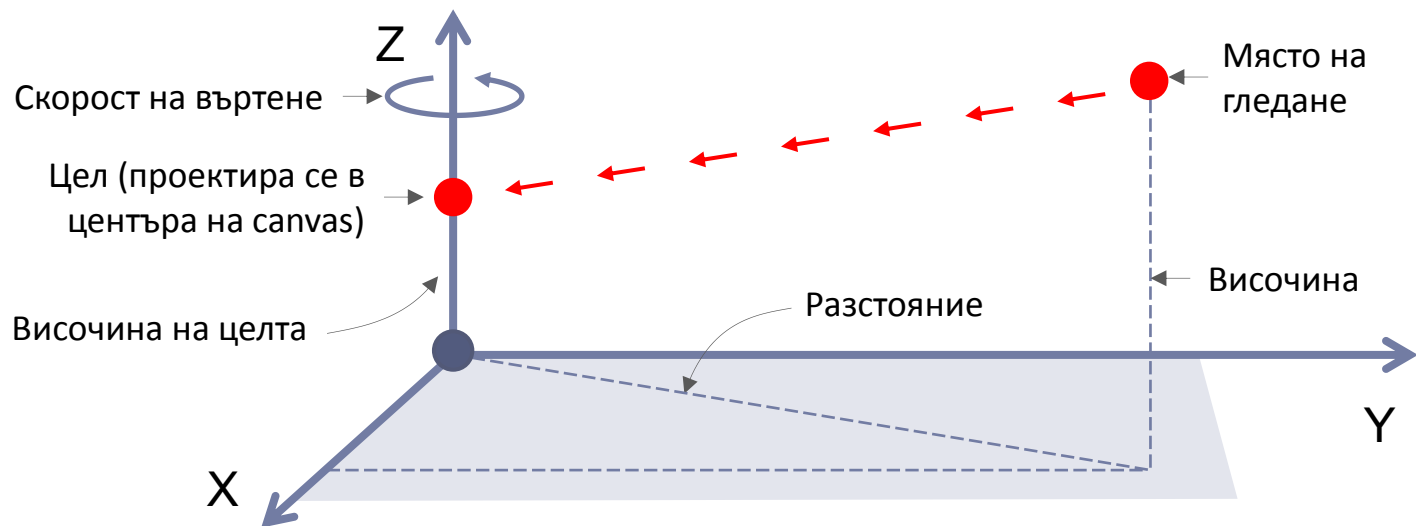
- Включва режим на автоматично въртене на гледната точка
- Алтернативна представа – сцената се върти, а не ние

Параметри

- Разстояние (по хоризонтала)
- Скорост на въртене (18° в секунда, пълен кръг за 20 секунди)
- Височина на гледната точка (като % от разстоянието)
- Височина на целта (като % от разстоянието)

Допълнение

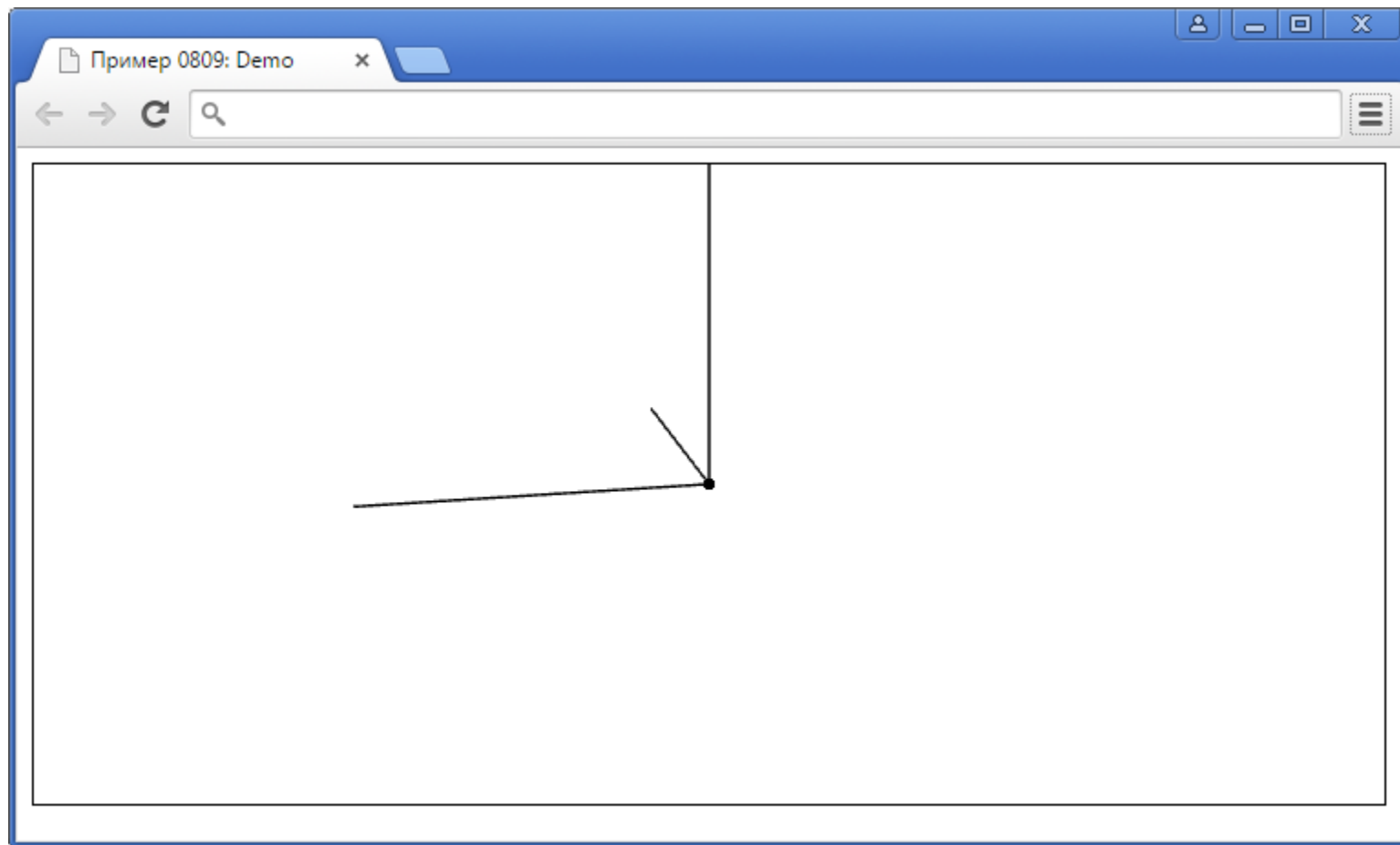
- Всички параметри са незадължителни
- Могат да се пропускат само последни параметри



Пример

- Команда без параметри
- По подразбиране разстоянието е 100
- Височината е 0.3
- Скоростта е 1
- Целта е на височина 0.1

```
function main()  
{  
    new Suica();  
    oxyz();  
    demo();  
}
```

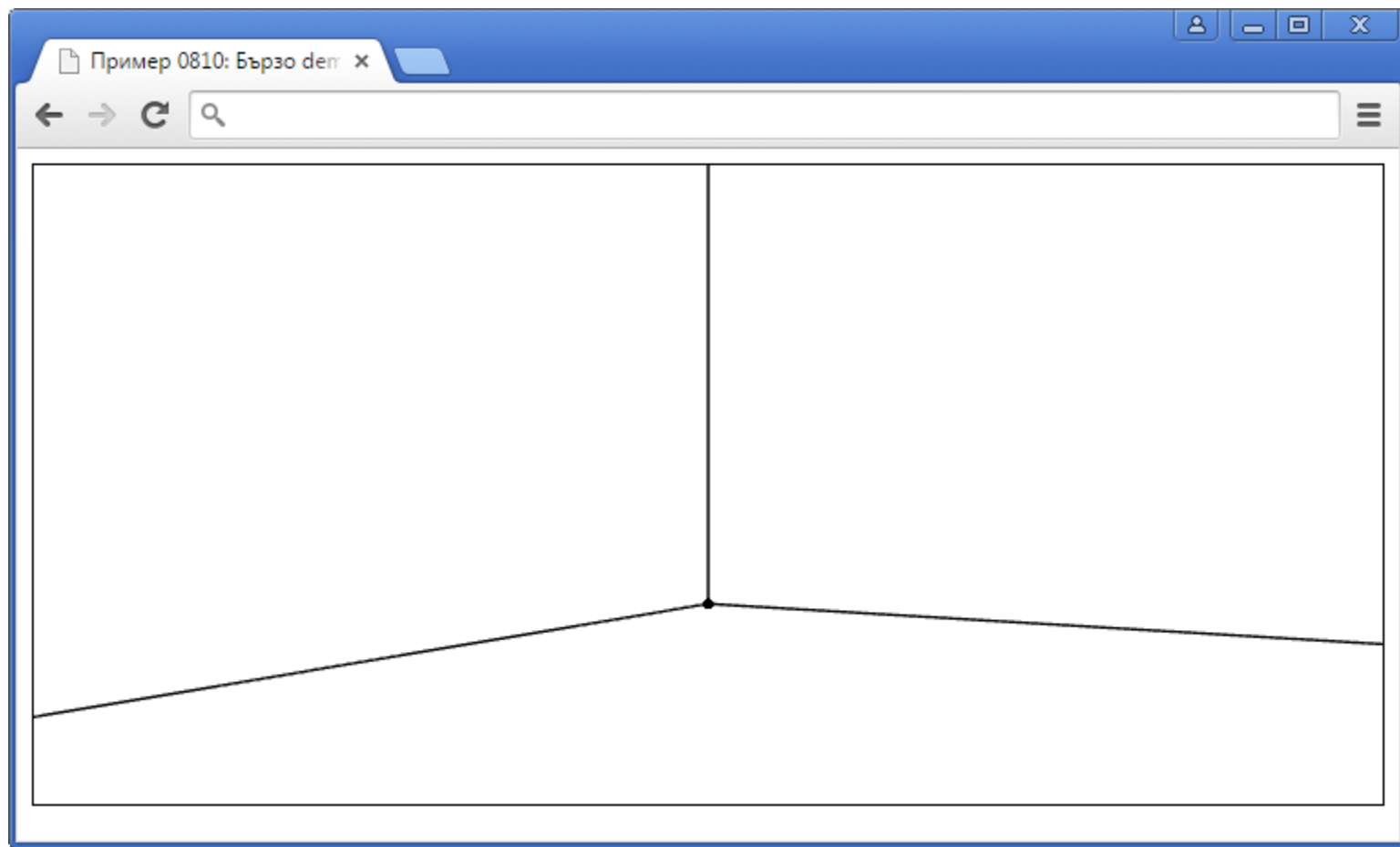


ПРОБА

Пример 2

- Близко разстояние 50 единици
- 9 пъти по-бързо въртене
- Ниско над равнината XY, на 5 единици
- Хоризонтално гледане към цел на височина също 5 единици

```
function main()
{
    new Suica();
    oxyz();
    demo(50,9,0.1,0.1);
}
```



ПРОБА



Помощни функции и проекти

- Функциите oхуз и demo са само за удобство
 - Визуализира се координатната система
 - Възможност за оглеждане на сцената
- Не се очаква да се ползват в проектите
 - Визуализациите ще са по-сложни
 - Движенията ще са по-сложни
- Използването им в проект не се оценява

Обобщение

Онлайн графика



WebGL

- Технология за 3D графика в браузър
- Мултиплатформена

Библиотека SUICA

- Използва WebGL
- Олекотено създаване на графика

За използване на СУИКА

- Включва се **suica.min.js** или **suica.js**
- В onload на <body> се активира главната функция
- Рисуването става само в <canvas>
- За всеки <canvas> се прави отделна инстанция на Suica

Библиотека СУИКА

- **Suica** – клас Suica
- **version** – текуща версия на СУИКА
- **background** – цвят на фона
- **oxyz** – образ на координатната система
- **demo** – автоматично въртене на гледната точка



ИКТ в НОС

Край

Коментари, въпроси