

**ИКТ В НОС**

## **Точки и линии**

Тема №9

# Общи свойства

# Координати

---



## Координати в СУИКА

- Фиксирана декартова координатна система
- Координати се представят чрез масив от числа
- Редът на координатите е фиксиран:  $[x, y, z]$

## Употреба

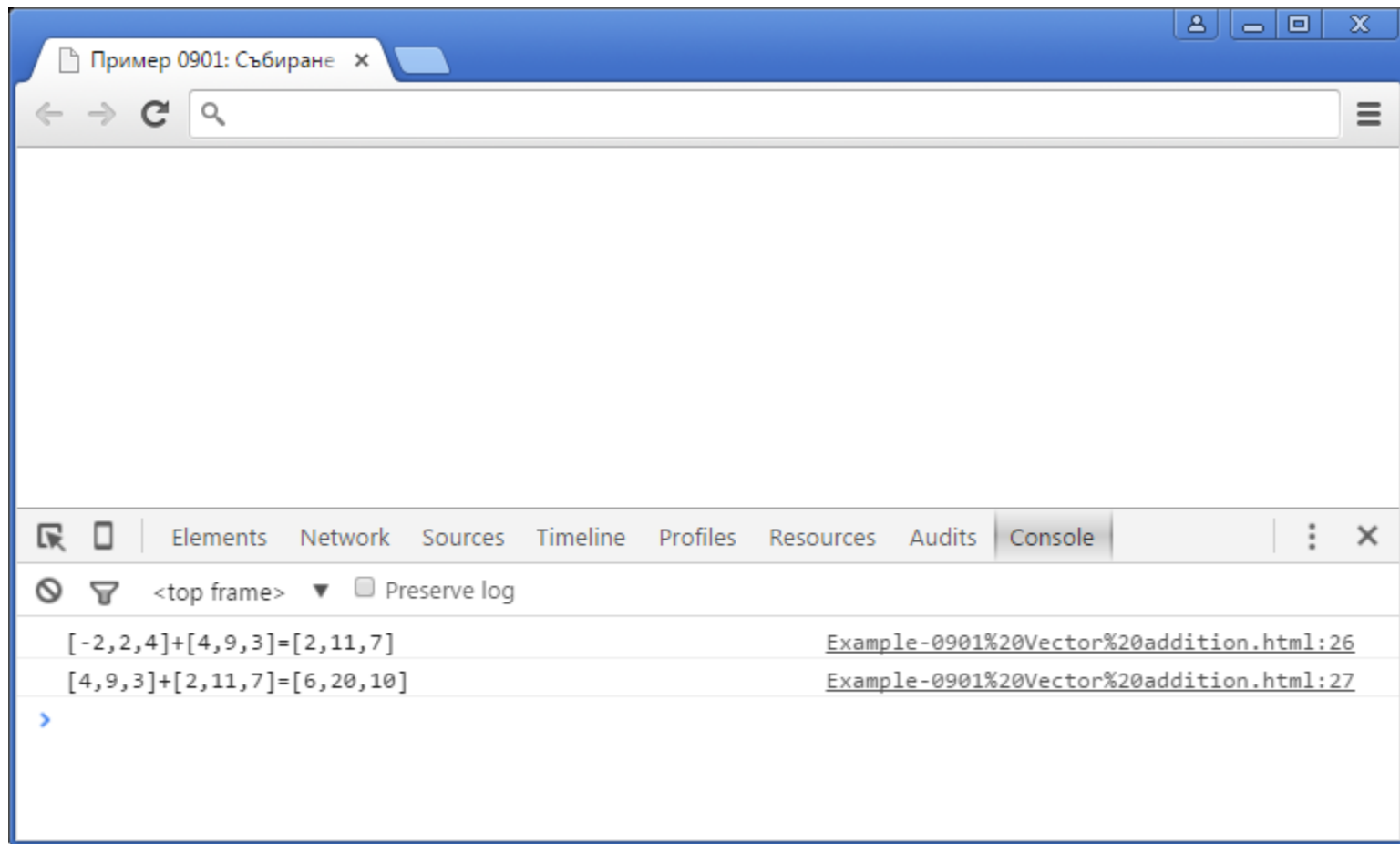
- Координати на точки
- Координати на обекти
- Вектори



## Събиране на вектори

- Покомпонентно събиране

```
function vAdd(a,b)
{
  var x = a[0]+b[0];
  var y = a[1]+b[1];
  var z = a[2]+b[2];
  return [x,y,z];
}
```

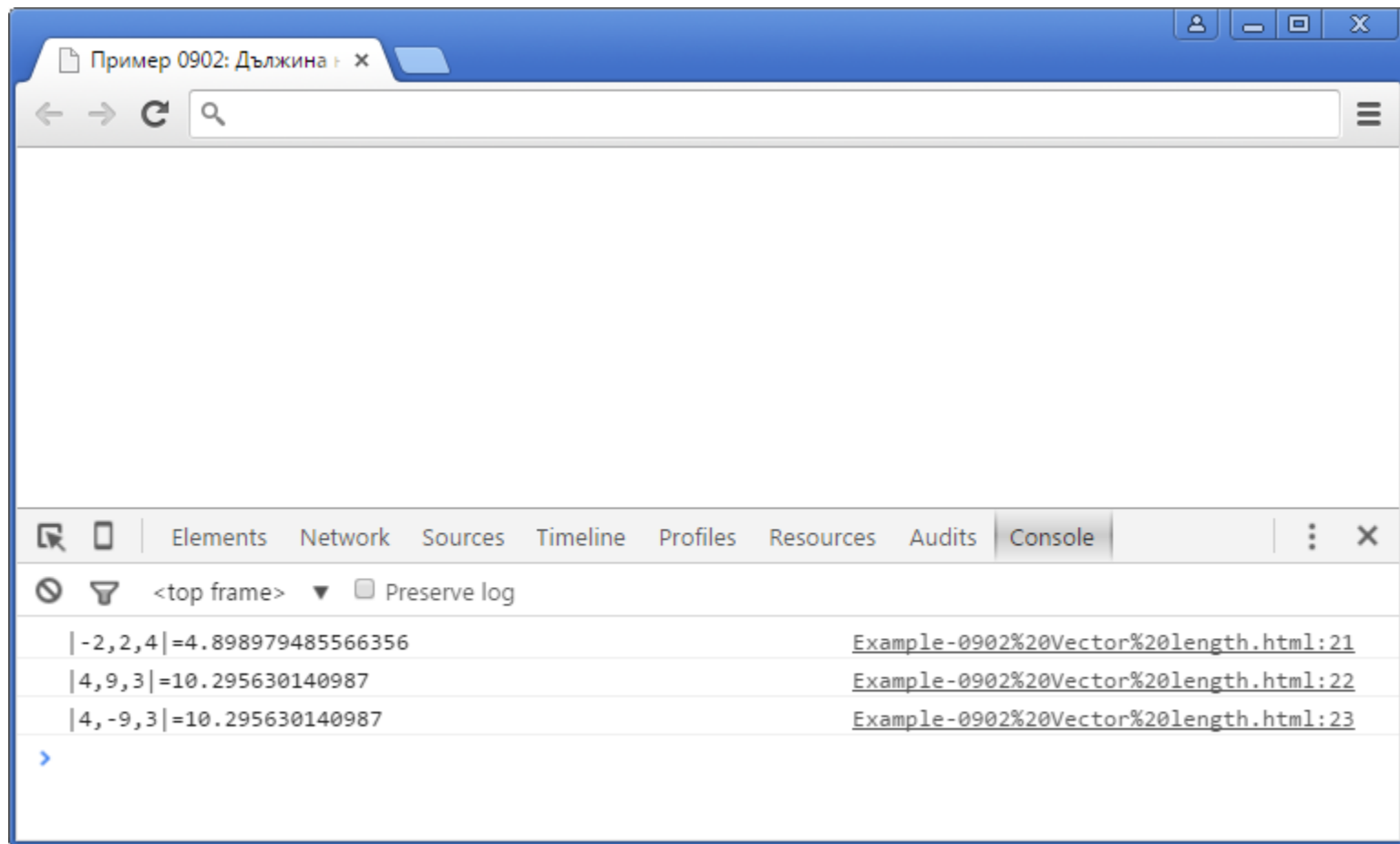


ПРОБА

# Дължина на вектор

- Координатите са дължините на вектора по осите

```
function vLen(a)
{
    return Math.sqrt(a[0]*a[0]+a[1]*a[1]+a[2]*a[2]);
}
```



ПРОБА

## Вградени функции за вектори

- Разлика на вектори (вектор между точки) **vectorPoints**
- Единичен вектор **unitVector**
- Скаларно произведение **scalarProduct**
- Векторно произведение **vectorProduct**

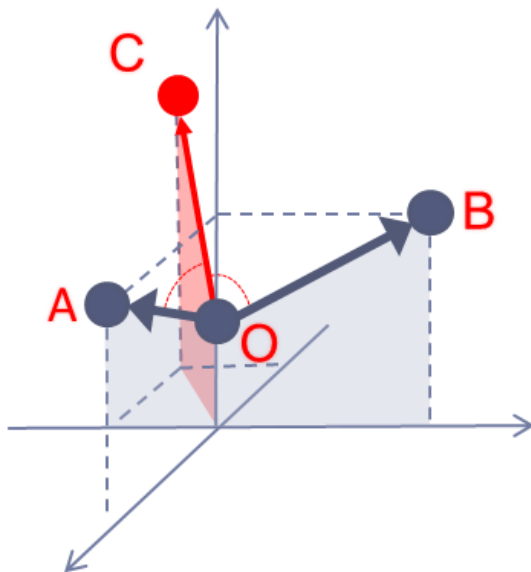
## Задача

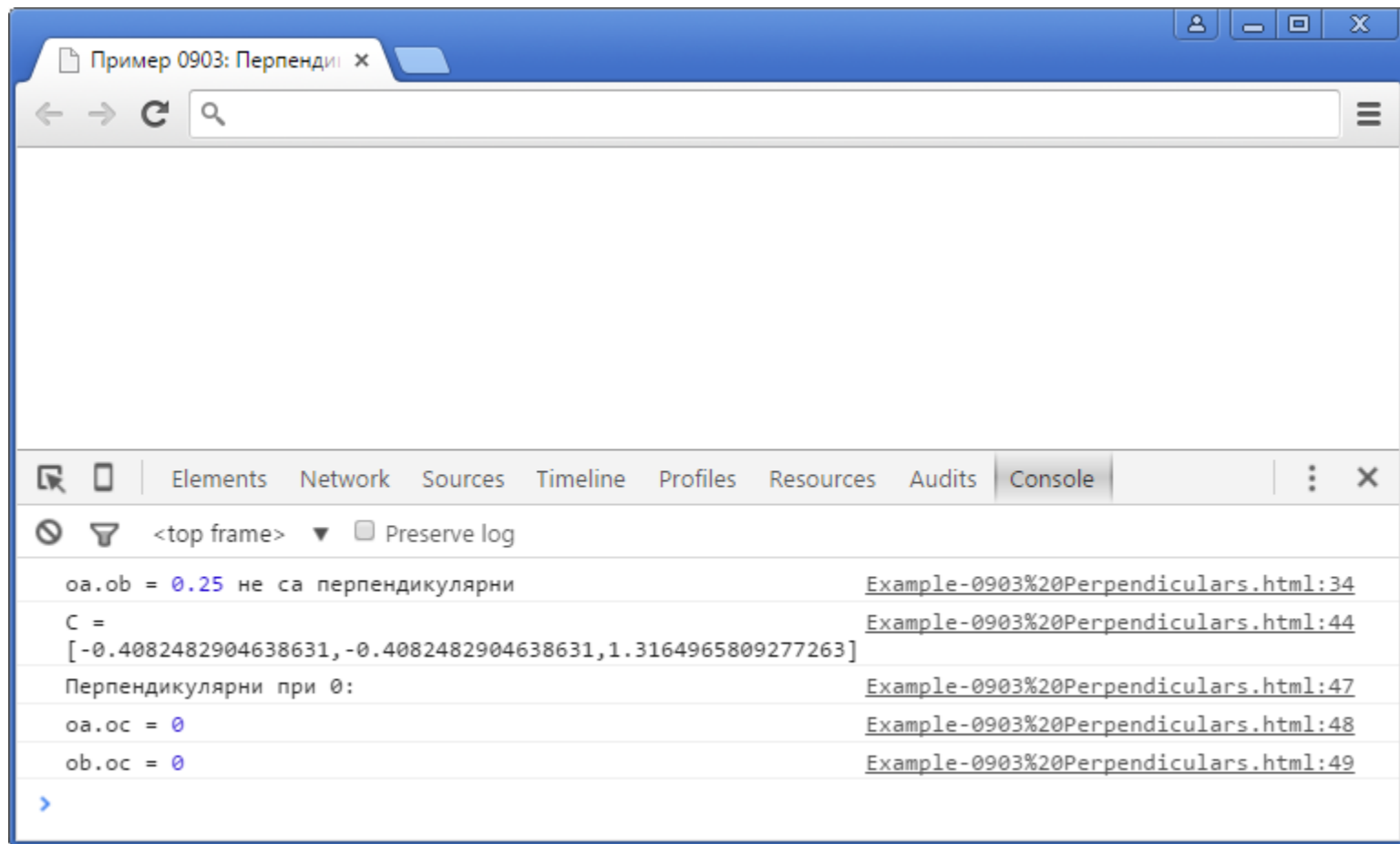
- Точки  $A(1,0,1)$ ,  $B(0,1,1)$  и  $O(0,0,1/2)$
- Перпендикулярни ли са векторите  $\overrightarrow{OA}$  и  $\overrightarrow{OB}$
- Намерете точка  $C$ , такава че векторът  $\overrightarrow{OC}$  да е перпендикулярен едновременно на  $\overrightarrow{OA}$  и  $\overrightarrow{OB}$  и е с дължина 1



## Решение

- $\overrightarrow{OA}$  и  $\overrightarrow{OB}$  са перпендикулярни при скалярно произведение 0
- Векторното им произведение дава вектор, перпендикулярен едновременно и на двата





ПРОБА

**Точка**

# Точка в СУИКА

---



## Точка

- Графичен обект със свойства
- Използва се за рисуване на точка

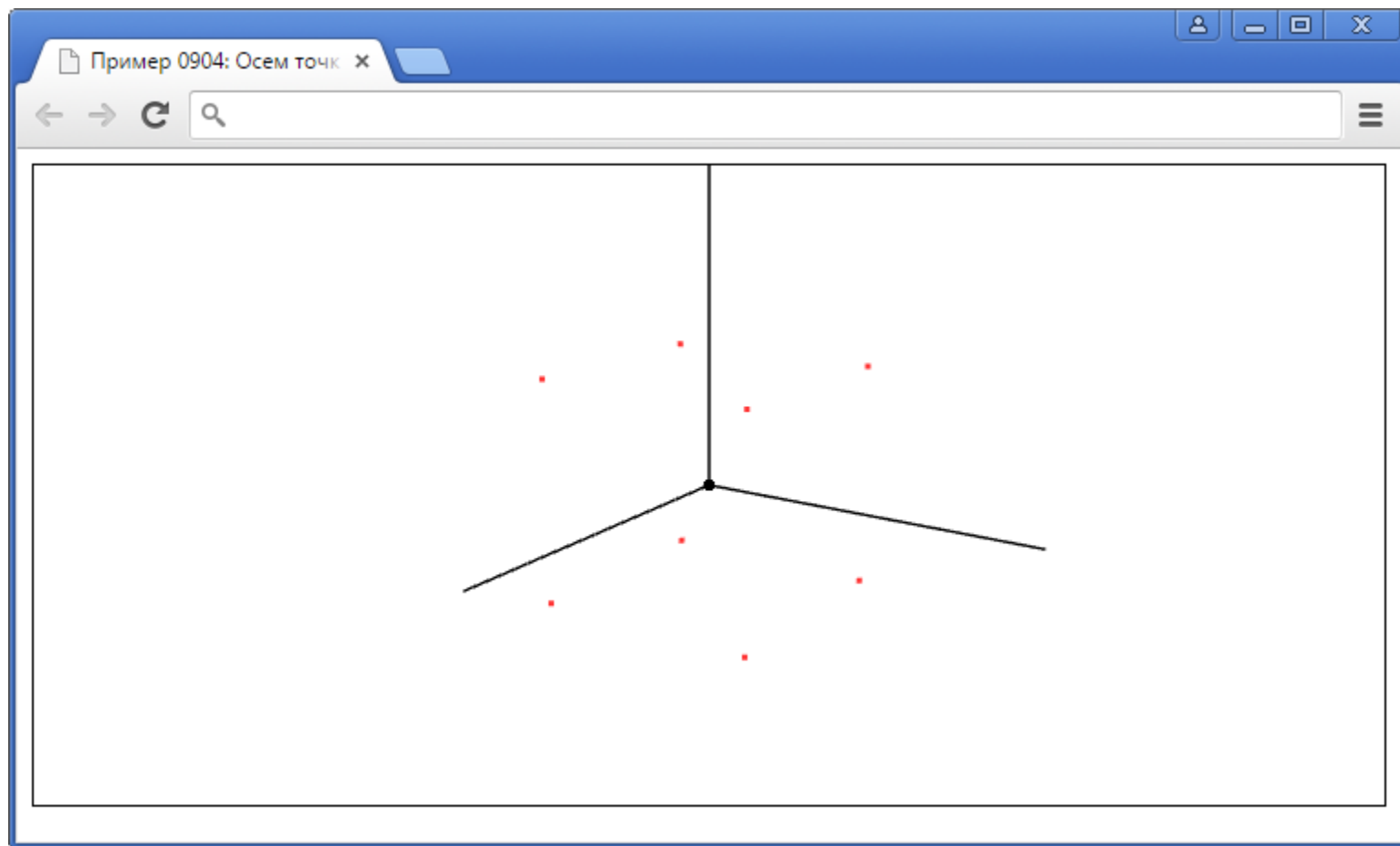
## Създаване на точка

- Чрез клас `new Suica.Point ( координати )`
- Чрез функция `point ( координати )`
- Координатите са вектор (масив от три числа)

## Пример

- Да се създадат 8 точки във върховете на куб
- При страна 20, върховете са с координати  $\pm 10$
- Осемте комбинации генерират осемте върха

```
point([+10,+10,+10]);  
point([+10,+10,-10]);  
point([+10,-10,+10]);  
point([+10,-10,-10]);  
point([-10,+10,+10]);  
point([-10,+10,-10]);  
point([-10,-10,+10]);  
point([-10,-10,-10]);
```



ПРОБА



## Център на точка

- Задава се при конструирането на точка
- Записва се в променливата **center**

## Размер на точка

- Определя само видимия размер
- Не зависи от разстоянието
- По подразбиране е 3
- Записва се в променлива **pointSize**

## Пример

- Точки от върховете на куб да се направят големи
- Последователно запомняме точките в променлива
- Чрез нея дефинираме свойството

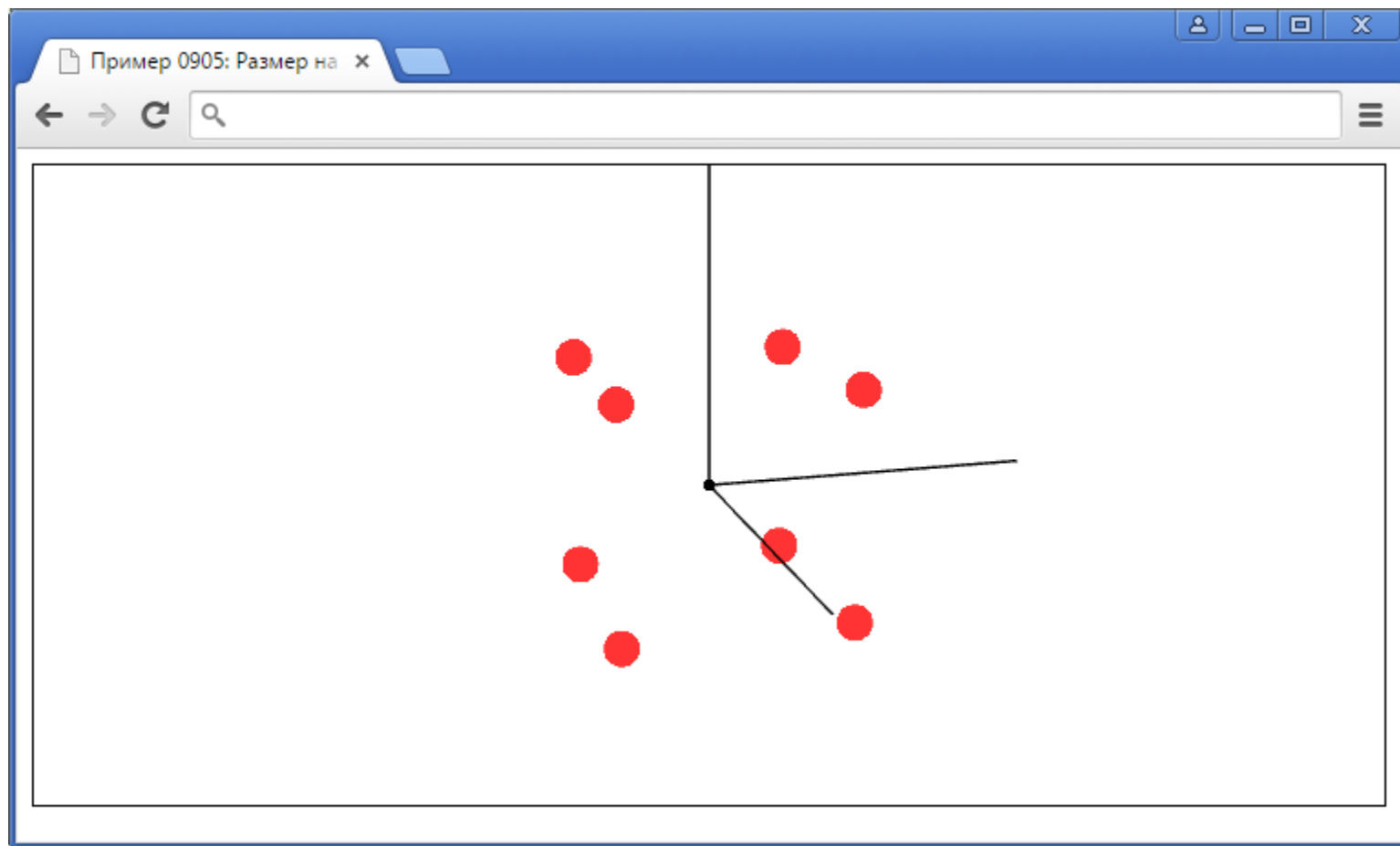
```
a = point([+10,+10,+10]);
```

```
a.pointSize = 20;
```

```
a = point([+10,+10,-10]);
```

```
a.pointSize = 20;
```





ПРОБА

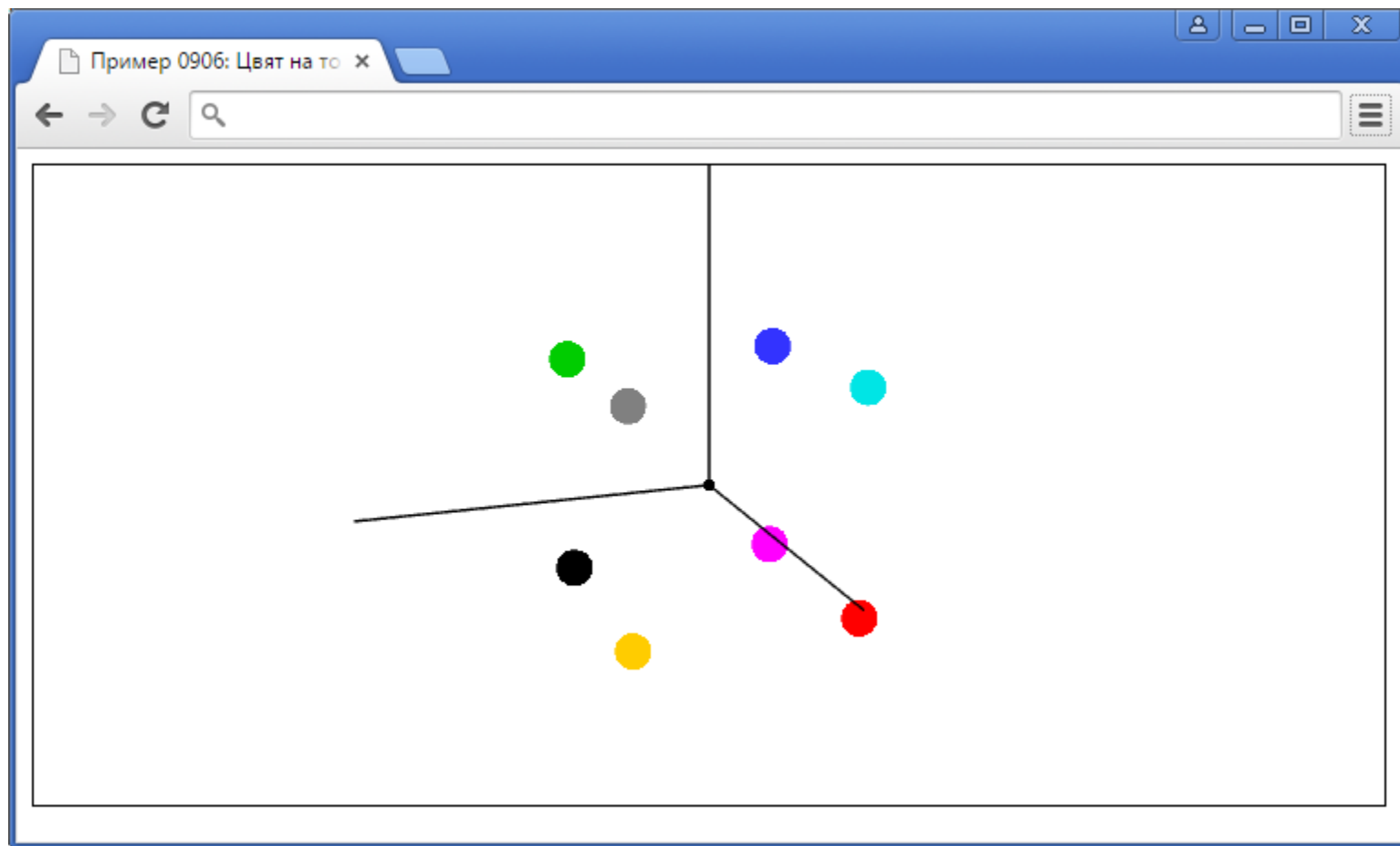
## Цвят на точка

- Определя цвета на точката
- Масив от три числа от 0 до 1
- По подразбиране е червен
- Записва се в променлива **color**

## Пример

- Точките да са големи и цветни

```
a = point([+10,+10,+10]);  
a.pointSize = 20;  
a.color = [0.5,0.5,0.5];
```



ПРОБА

## Видимост на точка

- Определя дали точката да се рисува
- По подразбиране е **true**
- Записва се в променлива **visible**

## **Примери с точки**

# Пример №1

---



## Случайни точки по линия

- Две точки в пространството
- Да нарисуваме  $n$  случайни точки между тях

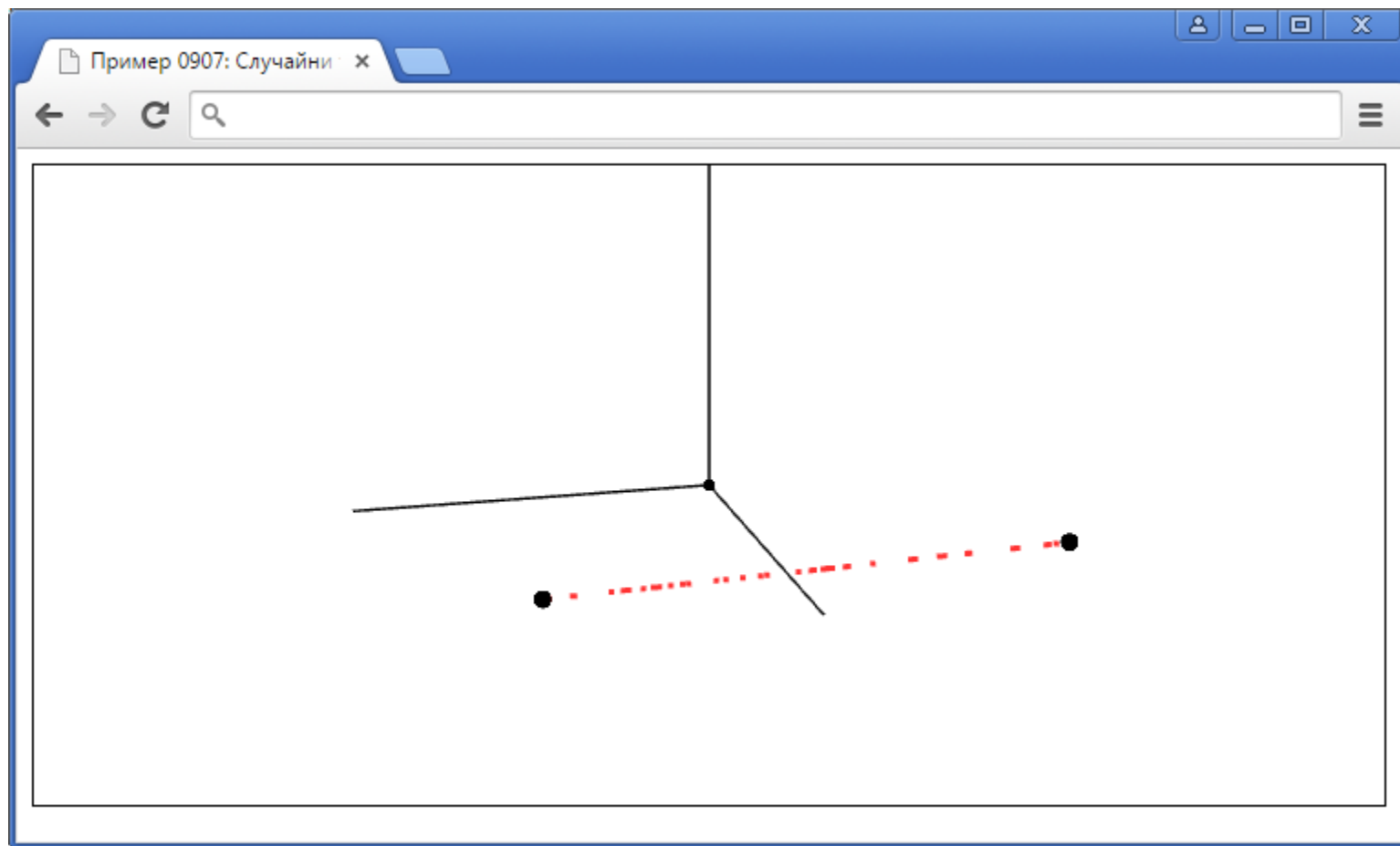
## Идея за решение

- Линейна комбинация от двете точки
- Коефициентът на комбинация е между 0 и 1
- Коефициентът е случайно избран

## Решение

- Достъп до координатите на точките чрез **center**
- Случайно число чрез функцията **random(om,do)**

```
for (var i=0; i<n; i++)  
{  
    var k = random(0,1);  
  
    var x = a.center[0]*(1-k)+k*b.center[0];  
    var y = a.center[1]*(1-k)+k*b.center[1];  
    var z = a.center[2]*(1-k)+k*b.center[2];  
  
    point([x,y,z]);  
}
```



ПРОБА



# Пример №2

---



## Случайни точки по окръжност

- Невидима окръжност с фиксиран радиус
- Да нарисуваме  $n$  случайни точки по окръжността

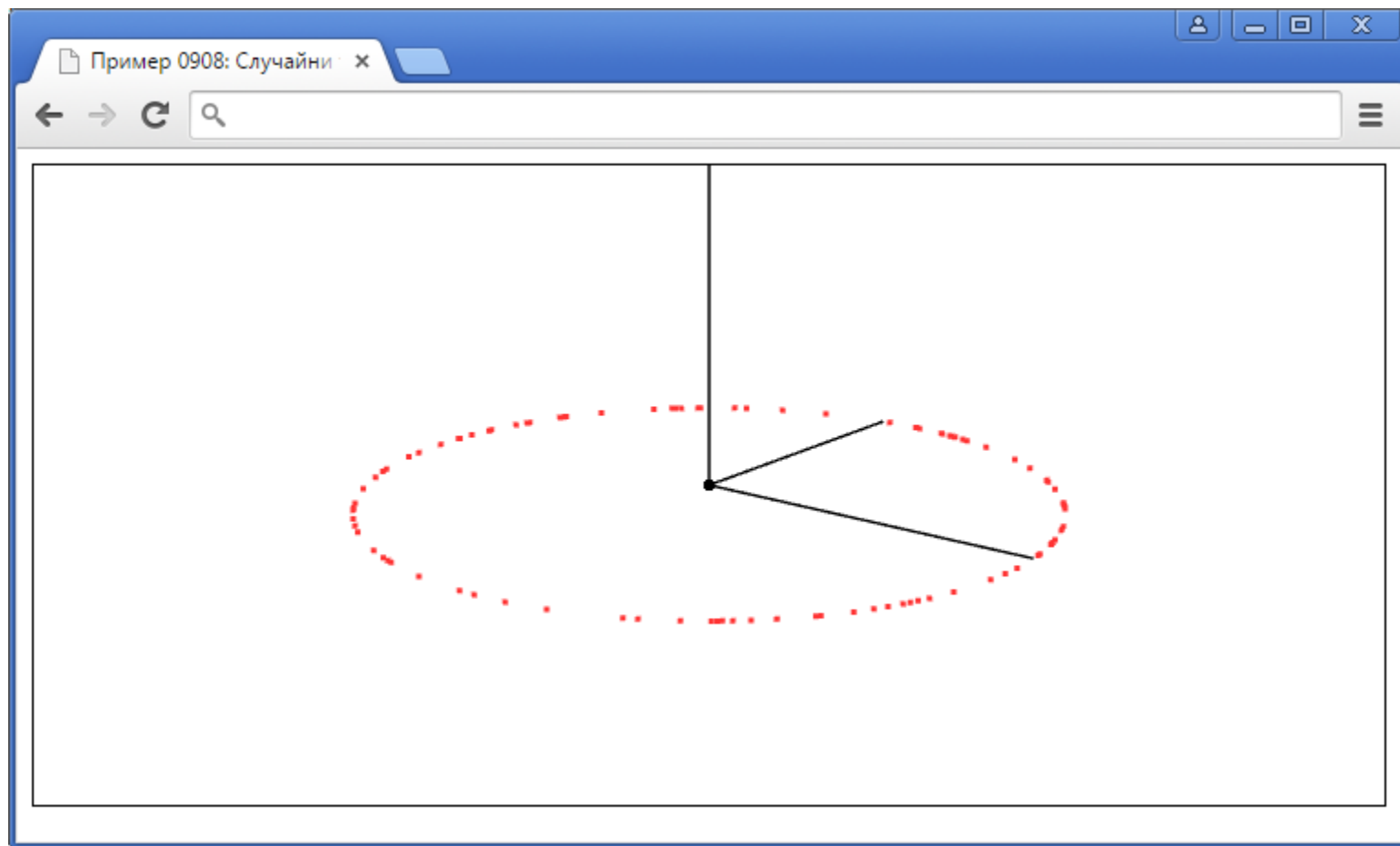
## Идея за решение

- За удобство центърът на окръжността е  $(0,0,0)$
- В полярни координати точките имат еднакъв радиус и случаен ъгъл

## Решение

- Ще използваме **radians(ъгъл)** за получаване на радиани
- От полярни координати ще получим декартови

```
for (var i=0; i<n; i++)  
{  
    var alpha = random(0,radians(360));  
  
    var x = r*Math.cos(alpha);  
    var y = r*Math.sin(alpha);  
  
    point([x,y,0]);  
}
```

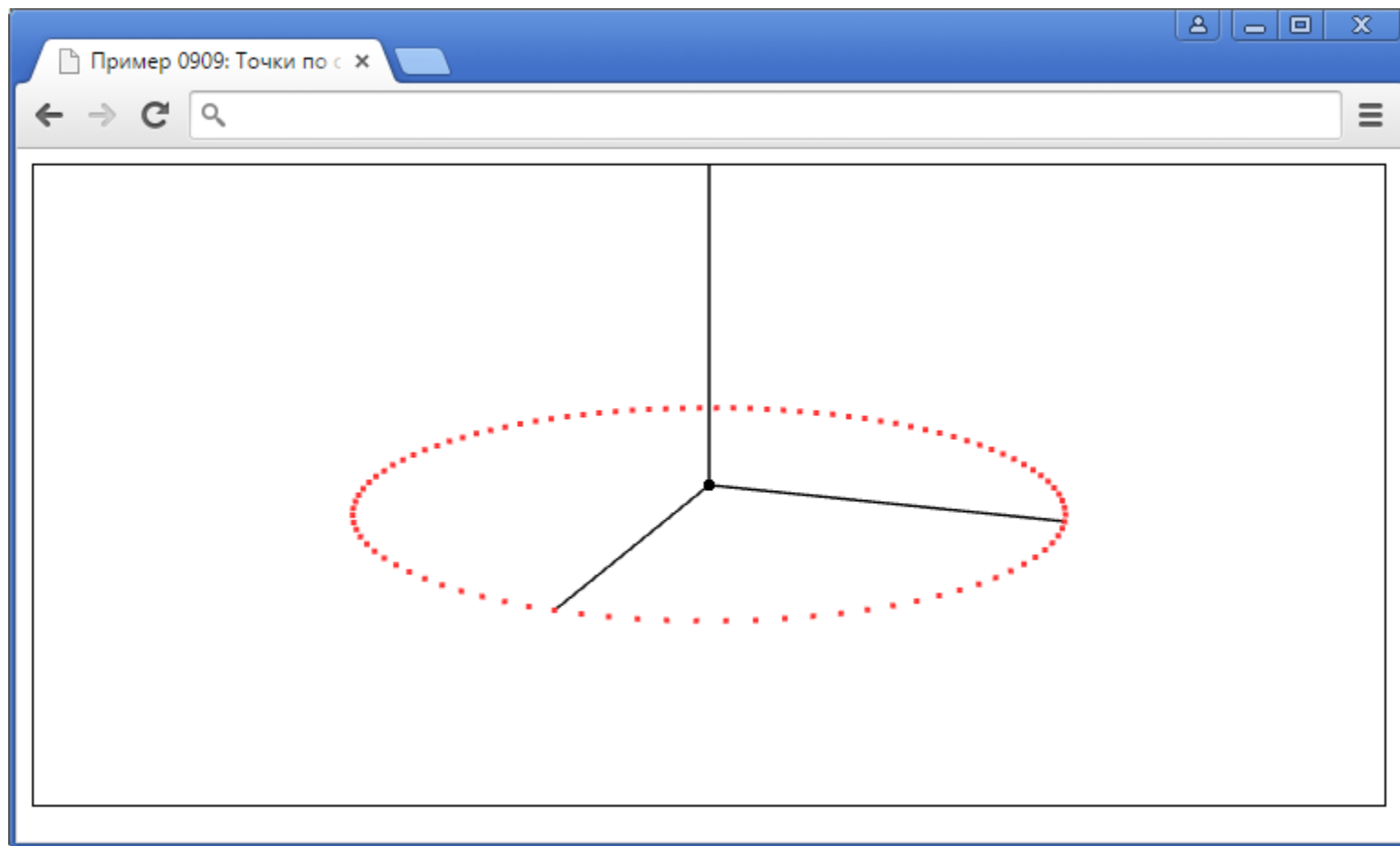


ПРОБА

## Вариант

- Равномерно разположени  $n$  точки
- Точка  $N^{\circ}i$  е на ъгъл  $360.i/n$

```
for (var i=0; i<n; i++)  
{  
    var alpha = radians(360*i/n);  
  
    var x = r*Math.cos(alpha);  
    var y = r*Math.sin(alpha);  
  
    point([x,y,0]);  
}
```



ПРОБА

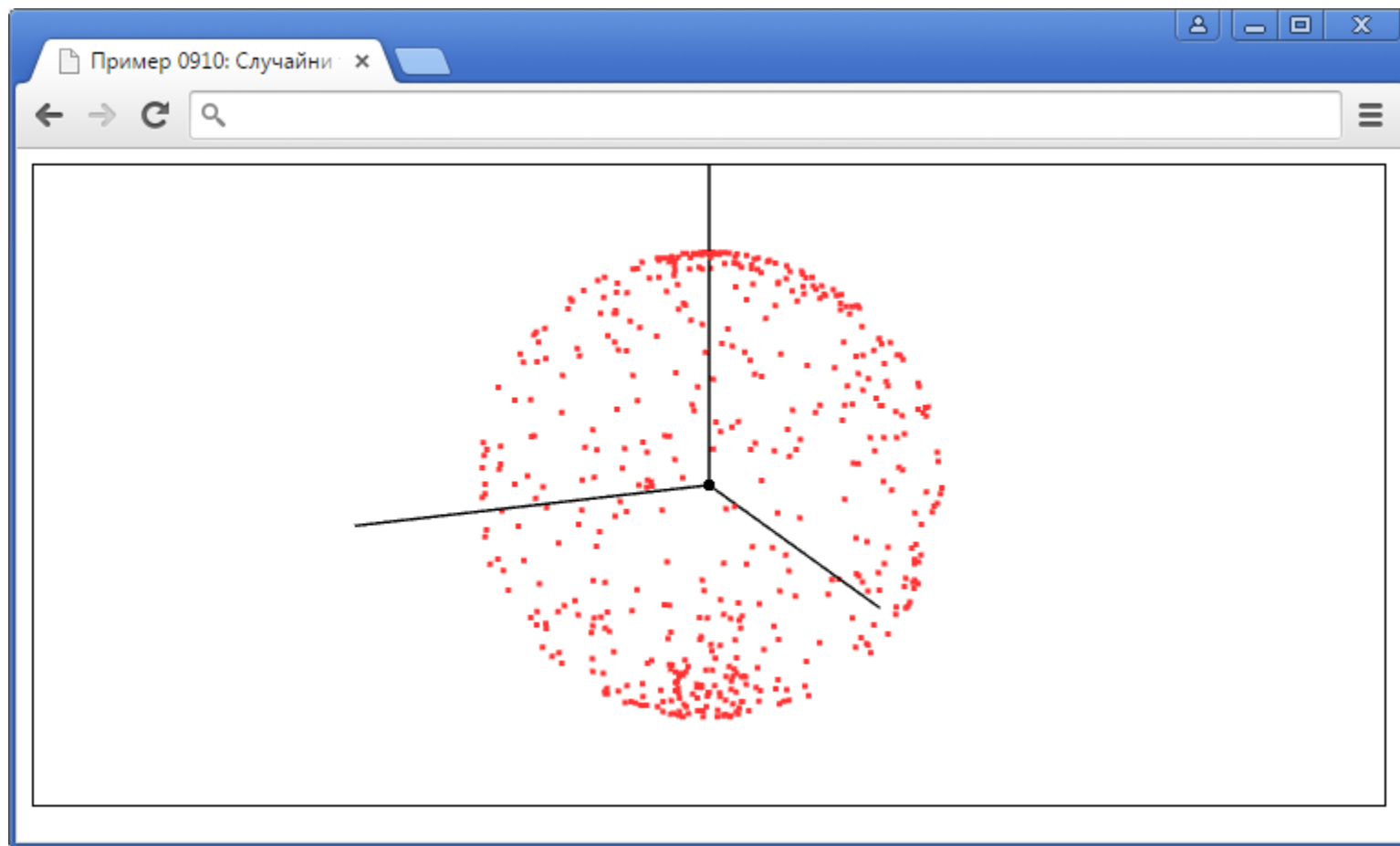
# Пример №3



## Случайни точки по сфера

- Използваме аналогично решение като при окръжност
- Преобразуваме сферични координати в декартови

```
var alpha = radians(random(0,360));  
var beta  = radians(random(-90,90));  
  
var x = r*Math.cos(alpha)*Math.cos(beta);  
var y = r*Math.sin(alpha)*Math.cos(beta);  
var z = r*Math.sin(beta);  
  
point([x,y,z]);
```



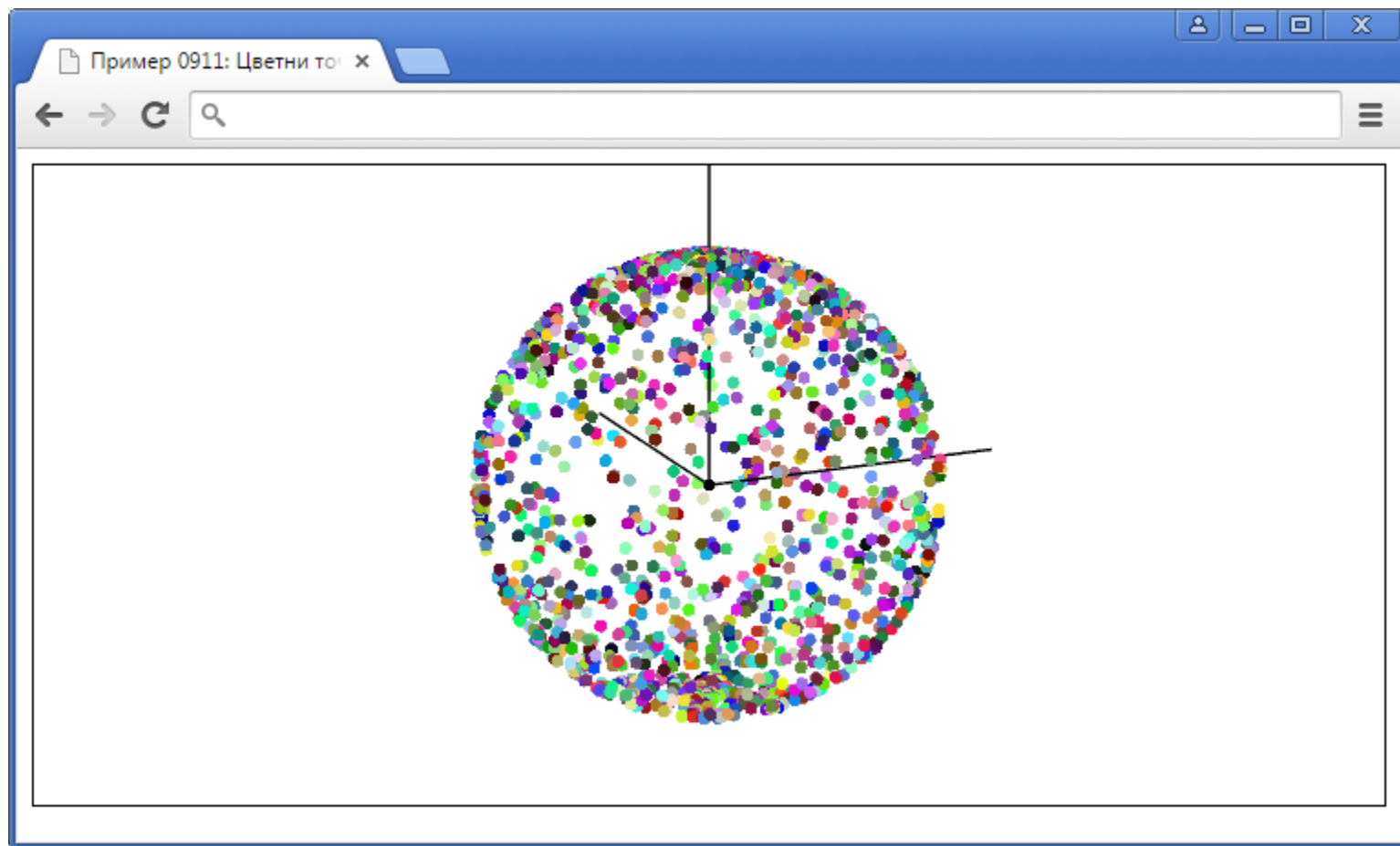
ПРОБА

# Вариант

- Големи цветни точки
- Случаен избор на цвят с random

```
var a = point([x,y,z]);  
a.color = [random(0,1),random(0,1),random(0,1)];  
a.pointSize = 7;
```





ПРОБА

**Линия, лъч, отсечка**

# Линия в СУИКА



## Линия

- Графичен обект със свойства
- Използва се за рисуване на линии през две точки

## Създаване на линия

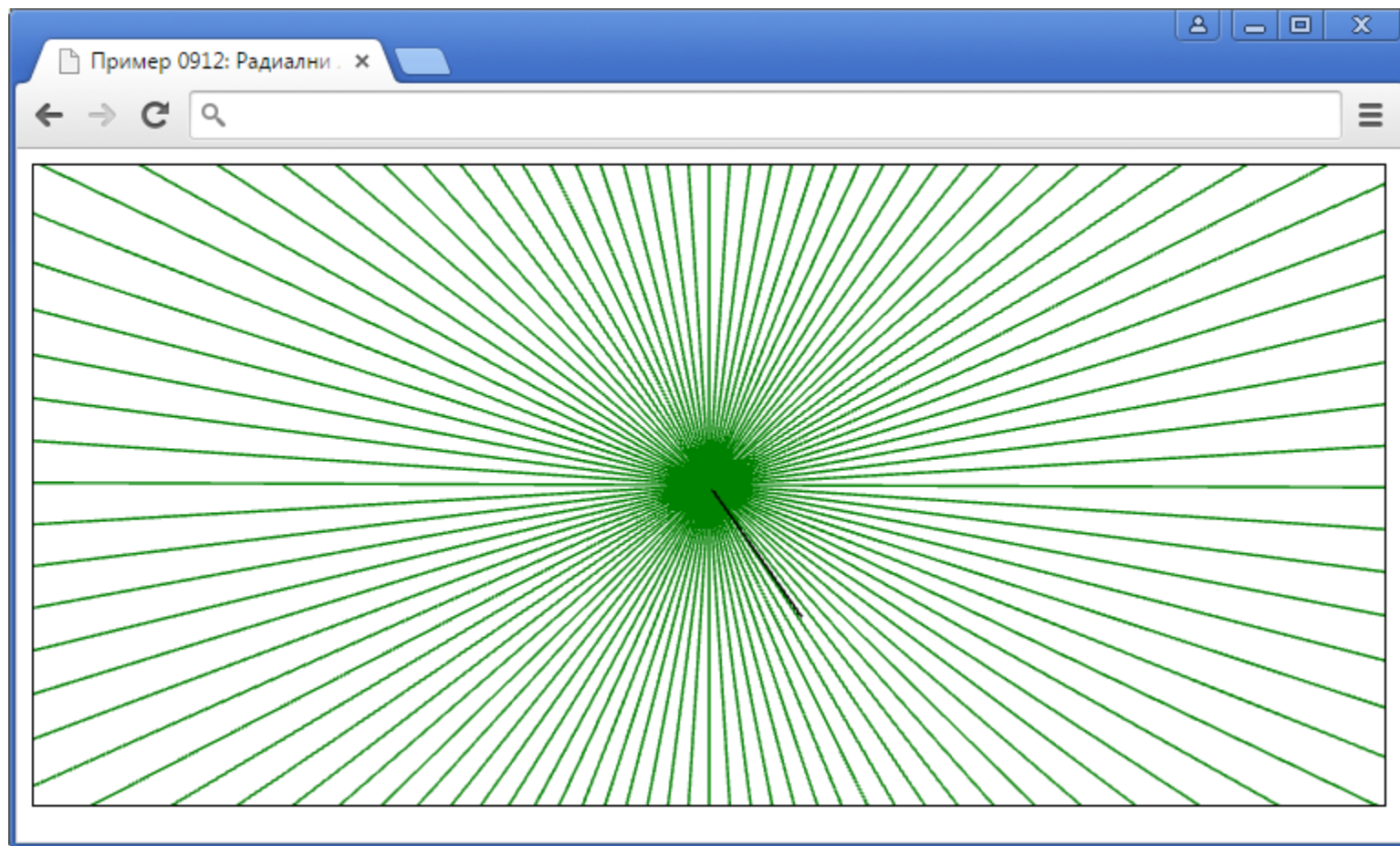
- Чрез клас `new Suica.Line ( точка, точка )`
- Чрез функция `line ( точка, точка )`



## Пример

- Да се създадат радиално и равномерно разположени линии
- Едната точка на всички линии е фиксирана, а другата е от окръжност около първата

```
for (var i=0; i<n; i++)  
{  
    var alpha = radians(180*i/n);  
  
    var x = Math.cos(alpha);  
    var z = Math.sin(alpha);  
  
    line([0,0,0],[x,0,z]);  
}
```



ПРОБА



## Свойства на линия

- Координатите на точките, през които минава линия, се записват в свойствата **to** и **from**
- Цветът е в **color**, зелен по подразбиране
- Видимостта е във **visible**



## Лъч

- Използва се за рисуване на лъчи от точка през друга точка
- Същите свойства като при линията

## Създаване на лъч

- Чрез клас `new Suica.Ray ( точка, точка )`
- Чрез функция `ray ( точка, точка )`

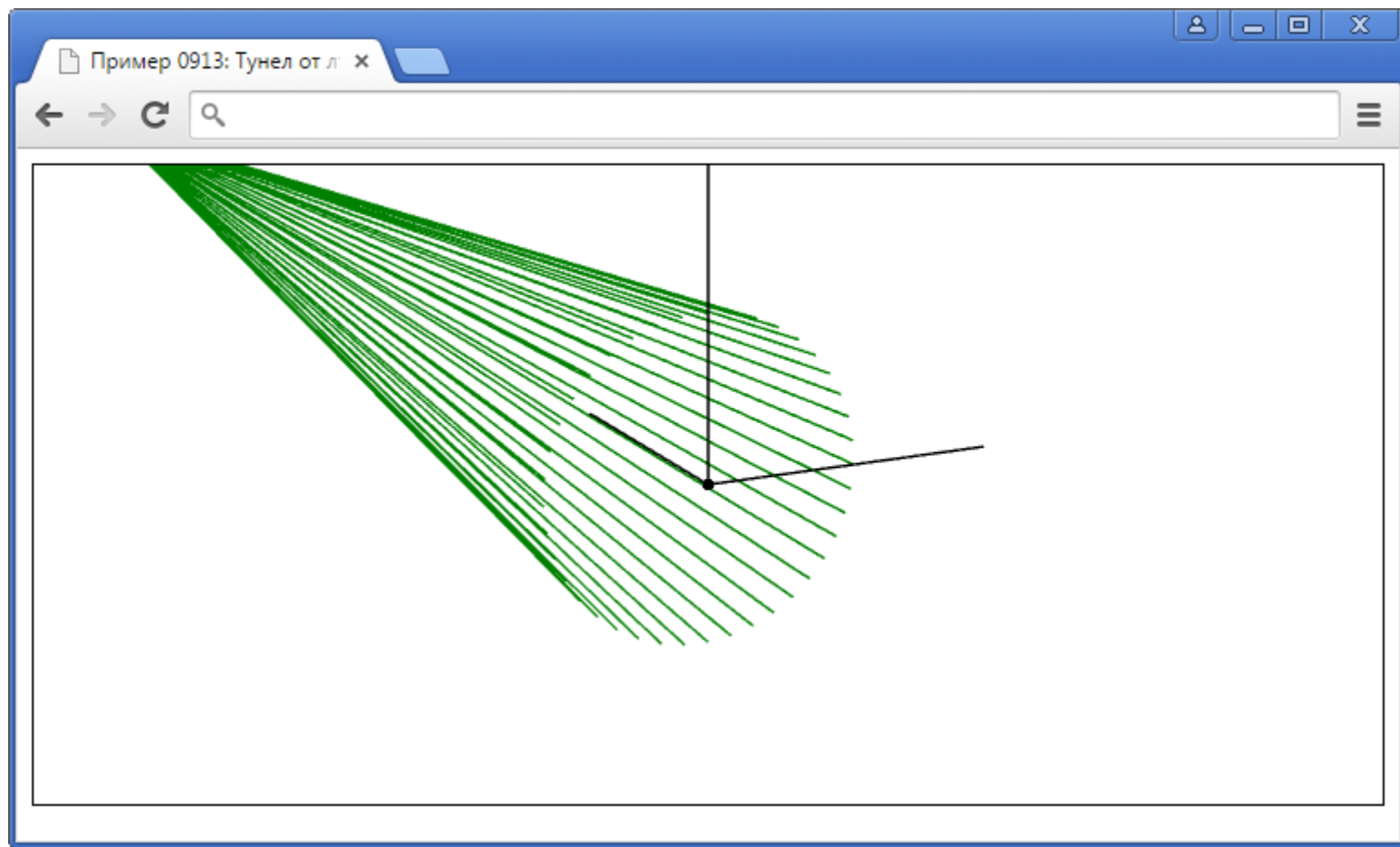


## Пример

- Да се създаде тунел от успоредни лъчи
- Началната точка на всеки лъч е от окръжност, а втората е отместена от нея на 1

```
for (var i=0; i<n; i++)  
{  
    var alpha = radians(360*i/n);  
  
    var x = 15*Math.cos(alpha);  
    var z = 15*Math.sin(alpha);  
  
    ray([x,0,z],[x,1,z]);  
}
```





ПРОБА

# Отсечка в СУИКА



## Отсечка

- Използва се за рисуване на отсечки между две точки
- Същите свойства като при линията

## Създаване на отсечка

- Чрез клас `new Suica.Segment ( точка, точка )`
- Чрез функция `segment ( точка, точка )`

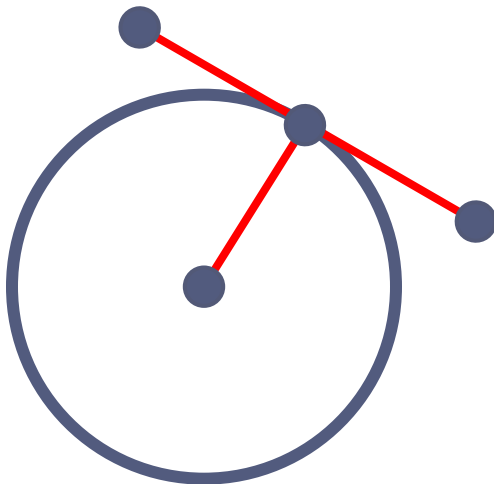


## Пример

- Да се създадат отсечки-допирателни на окръжност

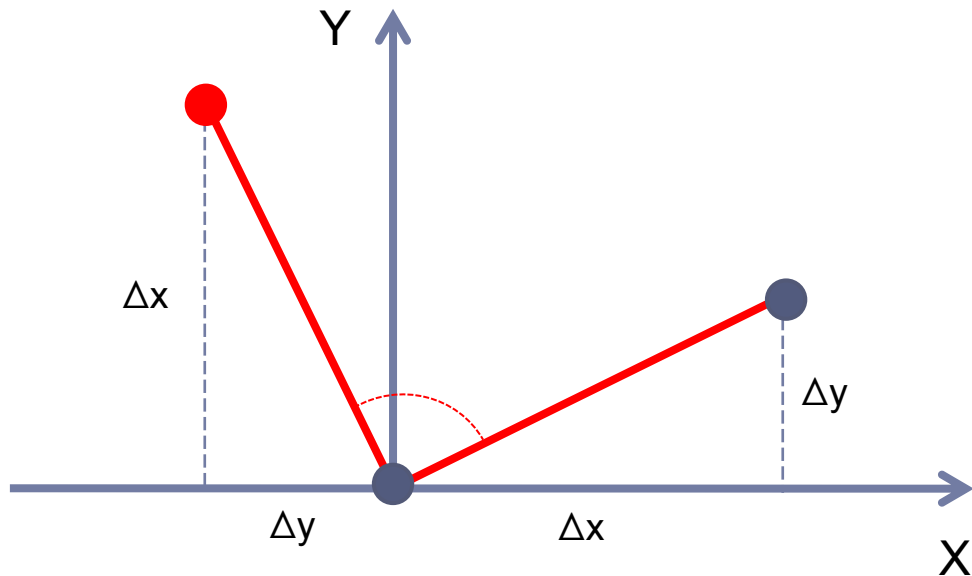
## Идея за решение

- Имаме точка на допиране
- Векторът от центъра на окръжността до тази точка е перпендикулярен на векторите от тази точка до краищата на отсечката



# Завъртане на $90^\circ$

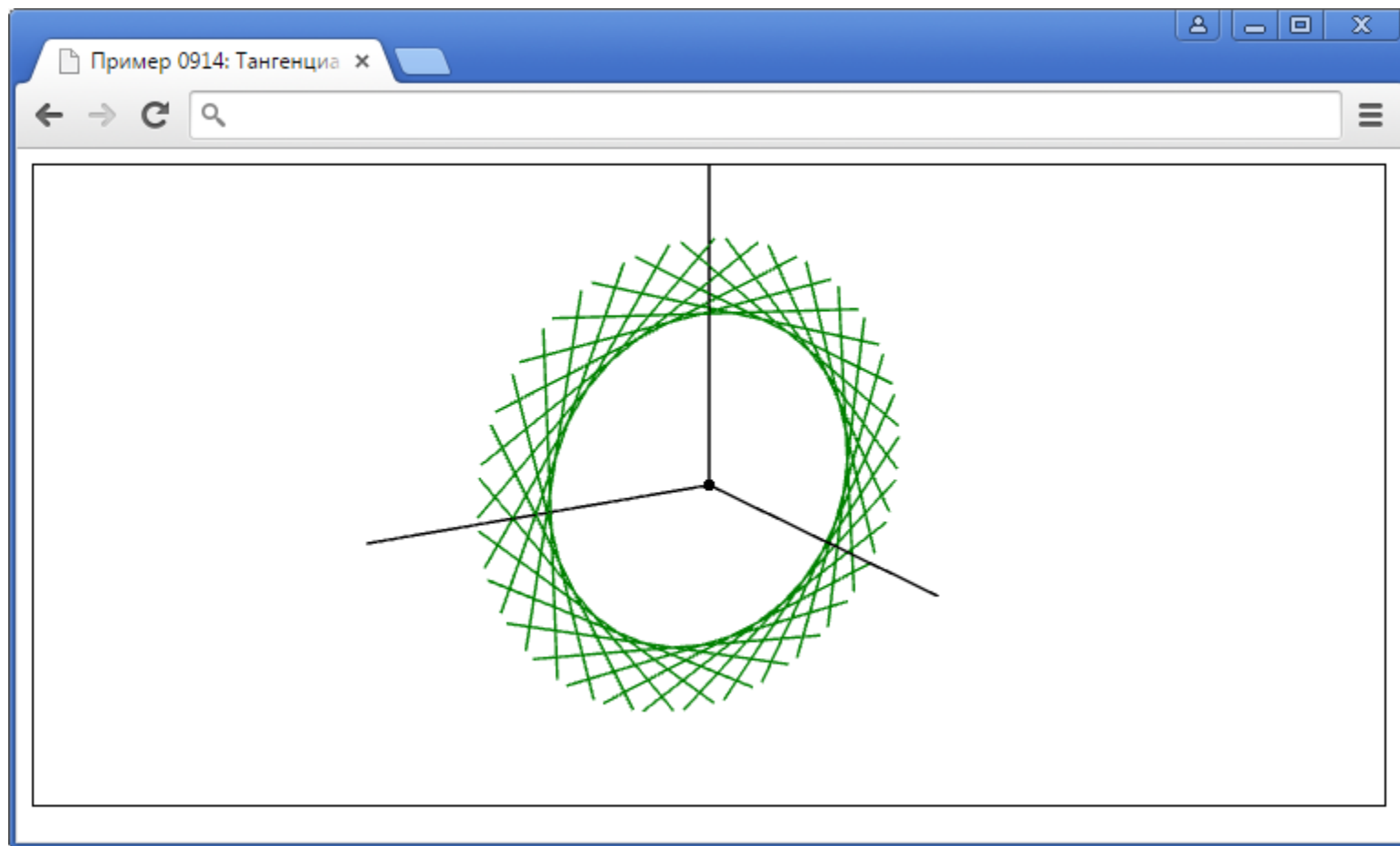
- Използваме еднакви триъгълници
- Ако единият вектор е  $(\Delta x, \Delta y)$ , то перпендикулярни на него са  $(-\Delta y, \Delta x)$  и  $(\Delta y, -\Delta x)$



## Решение

- Координатите на точка по окръжността са координати на вектора до нея
- Използваме ги за намиране на крайните точки на отсечките

```
for (var i=0; i<n; i++)  
{  
    var alpha = radians(360*i/n);  
  
    var dX = 15*Math.cos(alpha);  
    var dZ = 15*Math.sin(alpha);  
  
    segment([dX-dZ,0,dZ+dX],[dX+dZ,0,dZ-dX]);  
}
```



ПРОБА

# Обобщение

# Графични обекти

---



## Стойности

- Координатите и векторите са масиви от три числа
- Цветовете са масиви от три числа между 0 и 1

## Помощни функции

- **random** – случайно дробно число в интервал
- **radians** – преобразува градуси в радиани



## Общи свойства

- **center** – координати на обект
- **color** – цвят на обект
- **visible** – видимост или невидимост на обект
- **pointSize** – размер на точка (за образи на обекти, представени чрез точки)

# Графични обекти

---



## Точка

- Конструира се с `new Suica.Point` или `point`
- Поддържа общите свойства `center`, `color`, `visible` и `pointSize`

## Линия

- Конструира се с `new Suica.Line` или `line`
- Точките на създаване са в свойствата `from` и `to`
- Поддържа общите свойства `color` и `visible`

## Лъч

- Конструира се с `new Suica.Ray` или `ray`
- Свойствата са същите като при линията

## Отсечка

- Конструира се с `new Suica.Segment` или `segment`
- Свойствата са същите като при линията



# ИКТ в НОС

**Край**

Коментари, въпроси