ИКТ в НОС

# Геометрично визуализиране

Тема №20

# Проблеми и цели

# Цели

**Визуализация на геометрични обекти**

- Координатна система
- Отсечка и вектор
- Сфера, конус, цилиндър
- Тор

# Проблеми

## Геометрия и компютърна графика

- Имат много общи обекти
- Има разминаване във визуализирането им
- В образователен контекст не винаги е удачно да се ползват традиционните обекти от компютърната графика

## В тази лекция

- Някои примери за визуализация (ама само някои!)
- Примери за гледане, а не за copy-and-paste

# Координатна система

# Координатна система

## Стандартно показване

- С командата oxyz

## Липсват му

- Оси, които са достатъчно дълги
- Стрелки в края им
- Деления по осите
- Оси в отрицателни посоки
- Надписи и т.н.

# Стрелки

## Стрелки на координатните оси

- За удобство ще използваме три стила:

  black – дефинира черен цвят

  ox – ориентира обект по посока на оста X

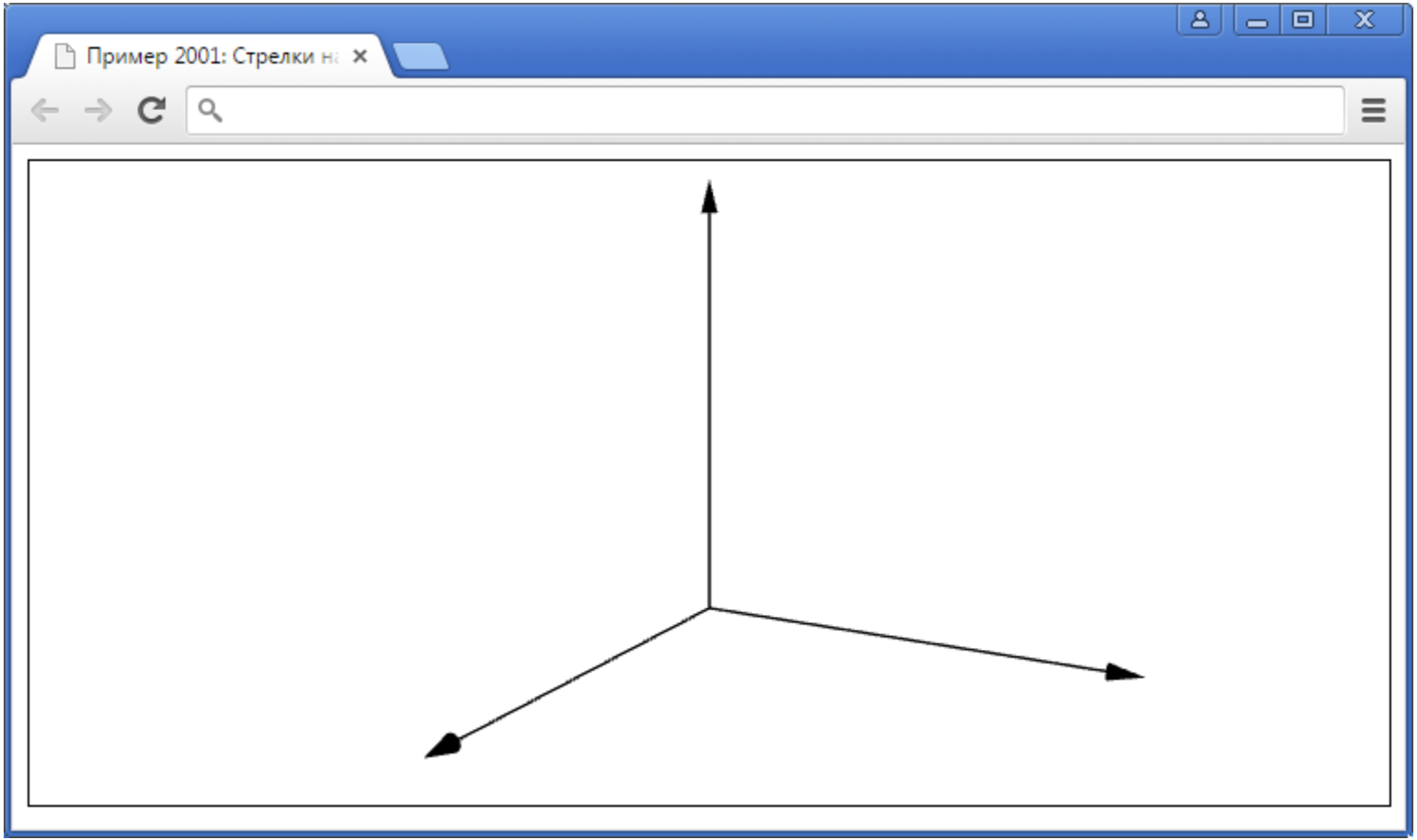
  oy – ориентира обект по посока на оста Y

```
black = {color:[0,0,0]};

ox = {focus:[1,0,0]};
oy = {focus:[0,1,0]};
```

# Обекти

- Трите оси са отсечки с нужната дължина
- Стрелките са черни неосветени конуси
- Ориентацията на осите и стрелките е със стиловете ox и oy

```
s = segment([0,0,0],[0,0,50]).custom(black);
sameAs(s).custom(ox);
sameAs(s).custom(oy);

cone([0,0,50],1,4).custom(black);
cone([50,0,0],1,4).custom(black).custom(ox);
cone([0,50,0],1,4).custom(black).custom(oy);
```




ПРОБА

# Всички оси

## Положителни и отрицателни оси

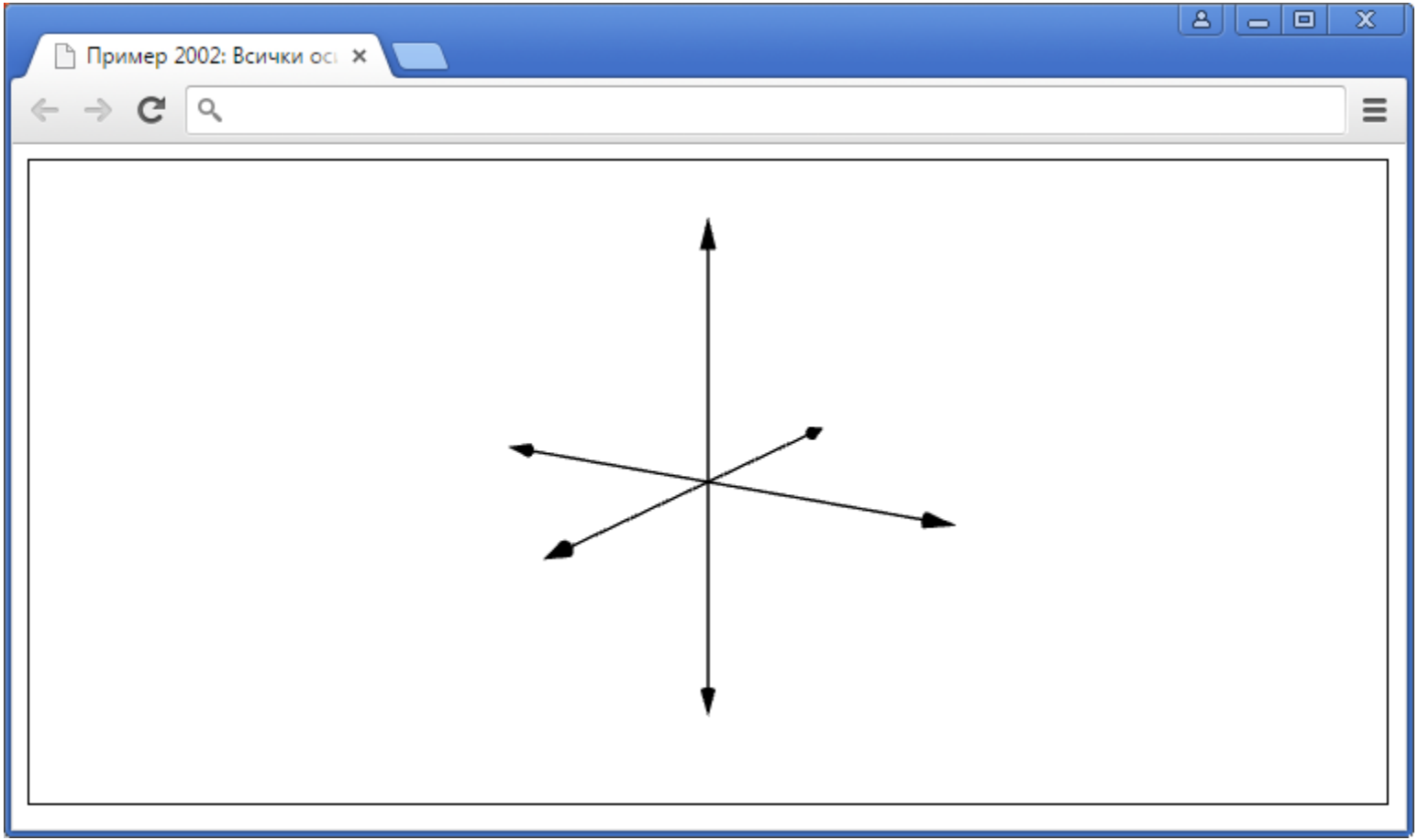
- Алтернативно решение поради няколкото еднакви обекта
- Създаваме групов обект
- Определяме му общ черен цвят

```
g = group([
      segment([0,0,0],[0,0,30]),
      cone([0,0,30],1,4)
   ]);
g.color = [0,0,0];
g.mergeColor();
```

# Останалите оси

- Клонираме петкратно вертикалната ос
- Променяме ориентацията на груповия обект да сочи последователно в останалите пет посоки

```
sameAs(g).custom({focus:[1,0,0]});
sameAs(g).custom({focus:[0,1,0]});

sameAs(g).custom({focus:[-1,0,0]});
sameAs(g).custom({focus:[0,-1,0]});
sameAs(g).custom({focus:[0,0,-1]});
```




ПРОБА

# Оси с имена

## Етикет с името на всяка ос

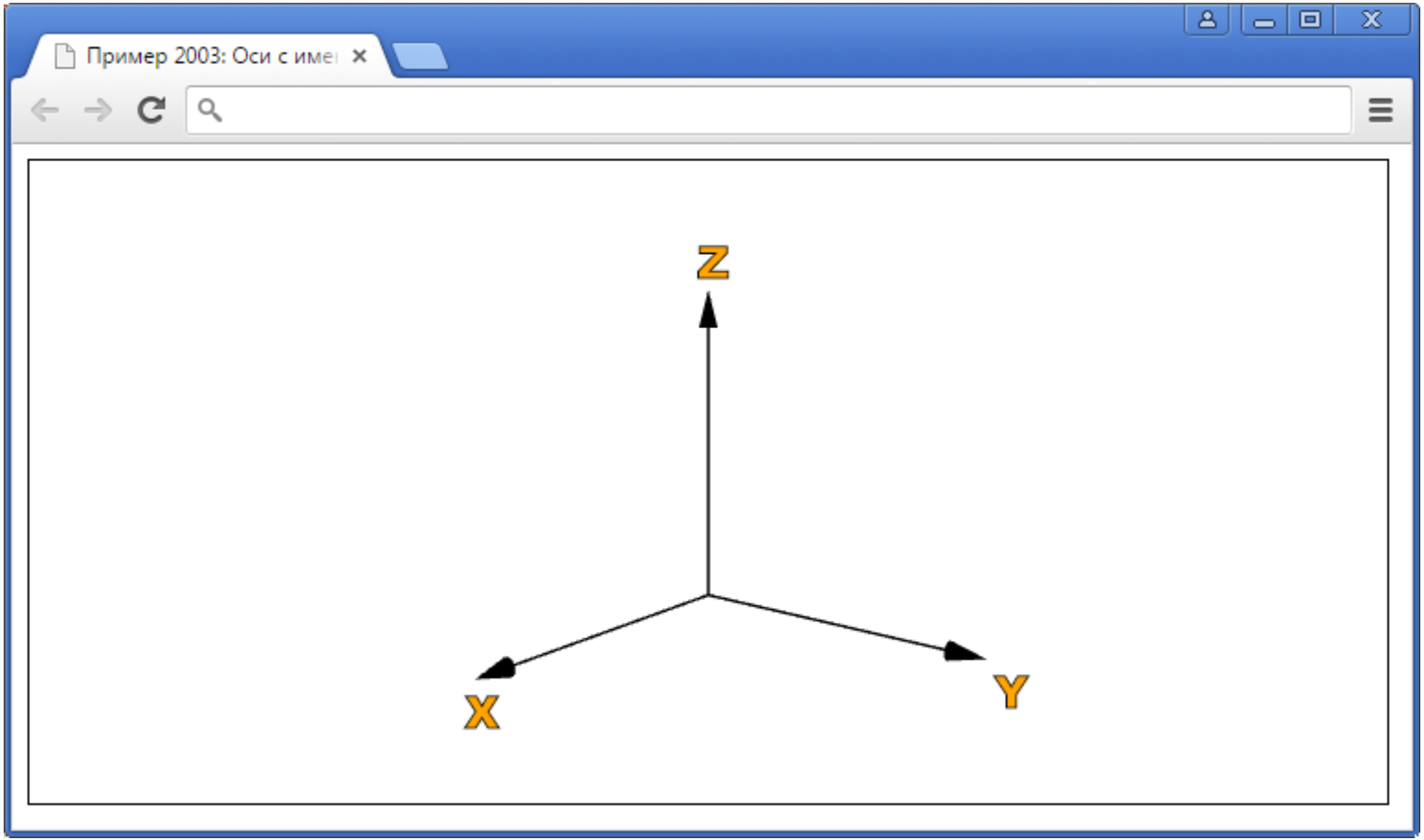
- Етикетът е HTML елемент и позволява форматиране с CSS
- Форматирането изисква да е с абсолютна позиция и с вертикален индекс над нивото на графичното поле

```
.label {
   position: absolute;
   z-index: 10;
   background-color: transparent;
   :
}
```

# Позициониране на етикет

- За оста X – чрез функцията getPosition намиране къде по екрана се намира 3D позиция с координати [35,0,0]
- Изчислената позиция е в пиксели
- Записваме я в style.left и style.top с добавено px отзад
- Аналогично се слагат етикети и на другите две оси

```
var e = document.getElementById('x');

var pos = getPosition([35,0,0]);

e.style.left = pos[0]+"px";
e.style.top = pos[1]+"px";
```



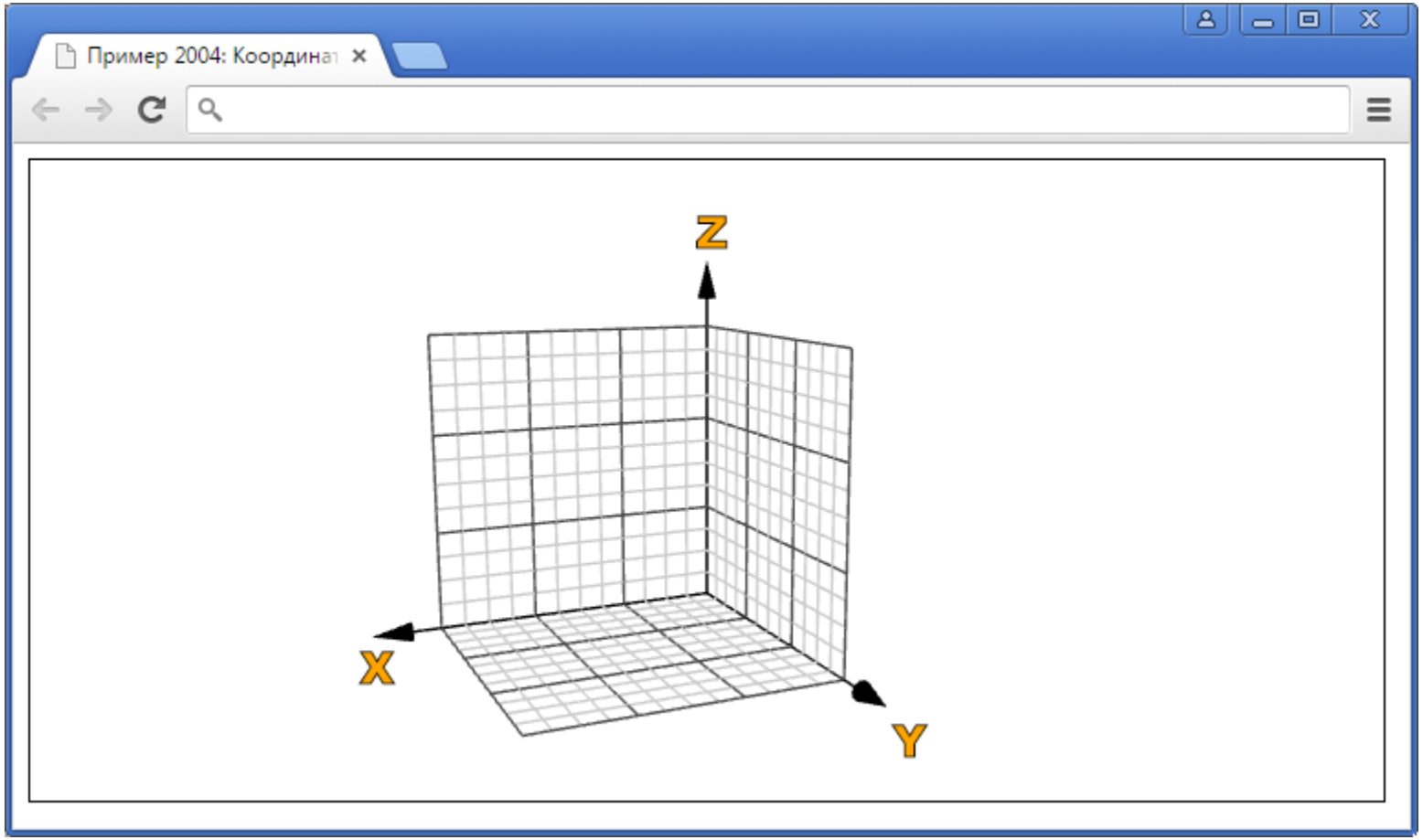

ПРОБА

# Координатна мрежа

## Мрежа от отсечки

- В равнината XY се генерират две групи отсечки: едната с отсечки, успоредни на оста X, а другата – успоредни на Y
- Аналогично и за останалите равнини

```
for (var i=0; i<=30; i+=2.5)
{
   segment( [i,0,0], [i,30,0]);
   segment( [0,i,0], [30,i,0]);
   :
}
```




ПРОБА

# Отсечка и вектор

# Отсечка и вектор

**Стандартно показване**

- С командата segment

**Липсват им**

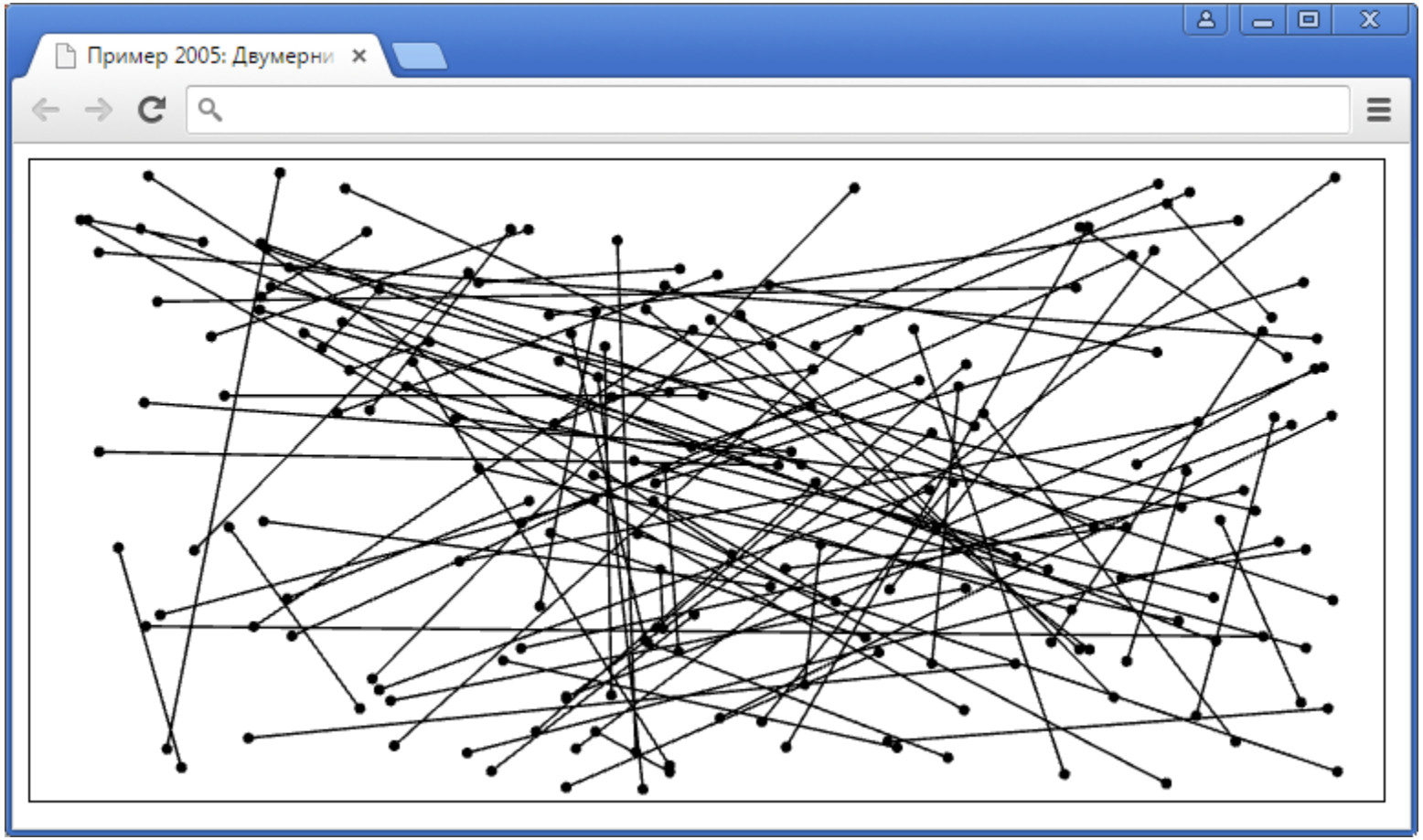
- Начални и крайни точки
- Стрелка в края на вектор
- Обемност (дебелина)
- Наклонен надпис

# Случайни отсечки

## Двумерни отсечки

- Линиите са стандартни отсечки
- Краищата са дебели точки (с pointSize) или окръжности

```
var black = {color:[0,0,0]};
for (var i=0; i<100; i++)
{
   p=circle([random(-350,350),...],3).custom(black);
   q=circle([random(-350,350),...],3).custom(black);
   segment(p.center,q.center).custom(black);
}
```

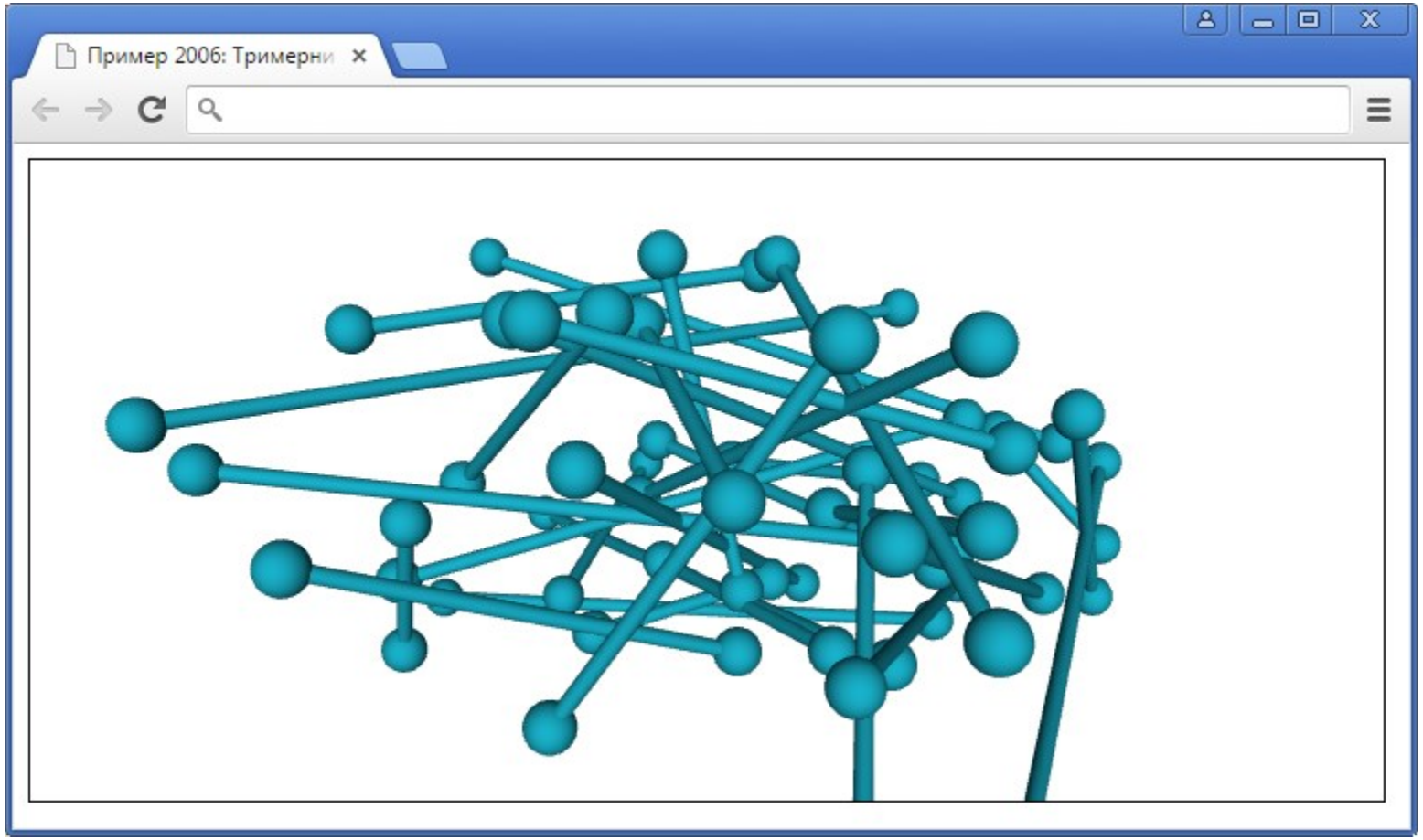



ПРОБА

# Тримерни отсечки

- Линиите са цилиндри с нужната дължина
- Краищата са сфери

```
var color = {color:[0.1,0.7,0.8]};
for (var i=0; i<30; i++)
{
   p = sphere([random(-50,50),...],3).custom(color);
   q = sphere([random(-50,50),...],3).custom(color);

   v = vectorPoints(q.center,p.center);
   cylinder(p.center,1,Math.sqrt(scalarProduct(v,v)))
                        .custom(color).custom({focus:v});
}
```




ПРОБА

# Случайни вектори

## Особеност

- При отсечката крайните обекти са в края ѝ
- При вектора стрелката трябва да е изместена, за да може върхът да достигне крайната точка

## 2D и 3D вектори

- Стрелката е нарисувана като конус
- Отместването на конуса така, че центърът му да е във върха, се прави като свойството origin = [0,0,1]

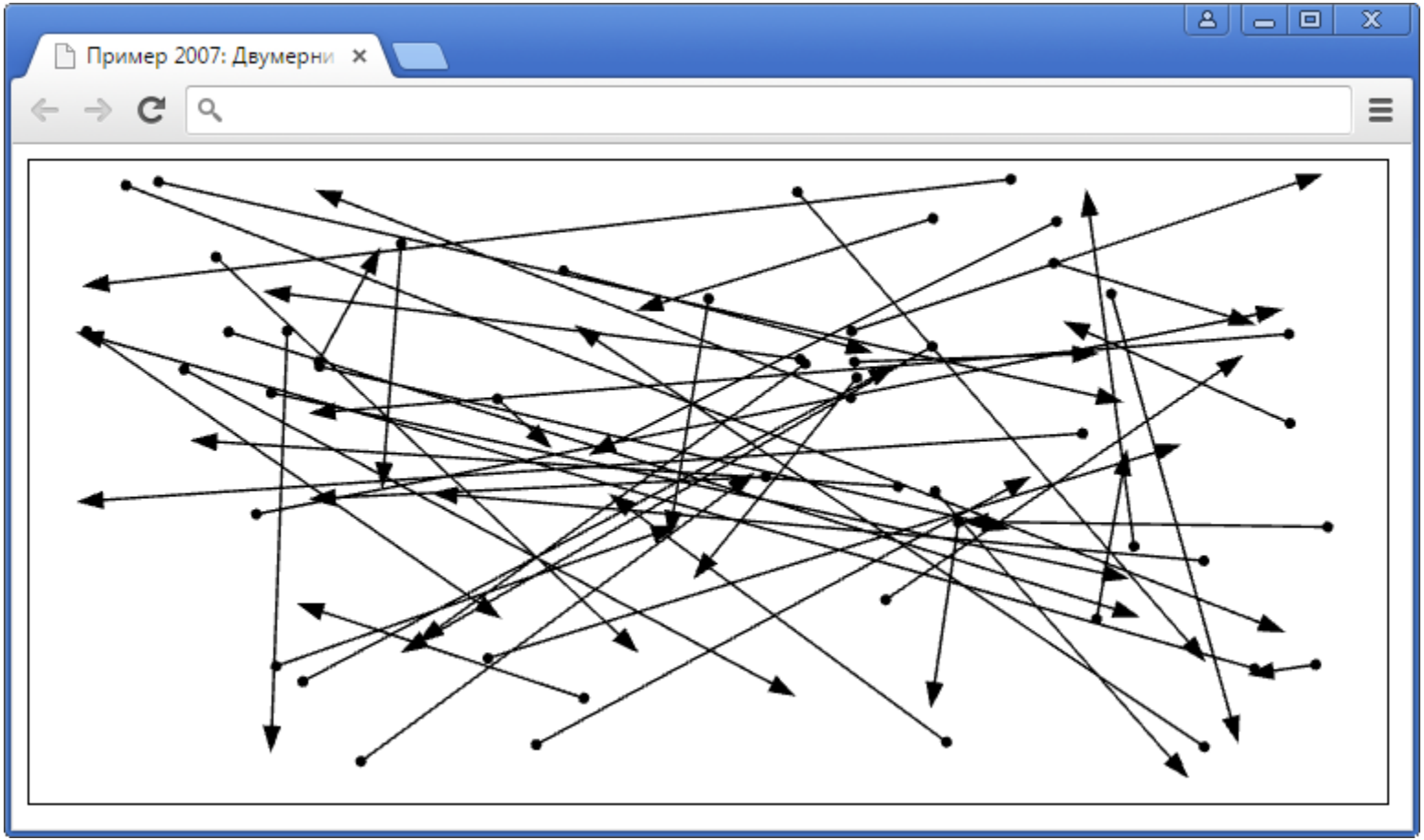
## Тяло на вектора

- Ако тялото на вектора е широко, това ще доведе до проблем във върха – ще се вижда заедно със стрелката

# Реализация

- Начална окръжност p и край на вектора q
- Стрелката е конус с променен origin и с ориентация по вектора v между центъра на p и q
- Коефициентът l е скъсяването на тялото с 15 единици, толкова, колкото е дълга стрелката

```
p = circle([...],3);
q = [...,0];
v = vectorPoints(q,p.center);
cone(q,5,15).custom({origin:[0,0,1], focus:v});
l = 15/Math.sqrt(scalarProduct(v,v));
q = [q[0]-l*v[0],q[1]-l*v[1],0];
segment(p.center,q);
```

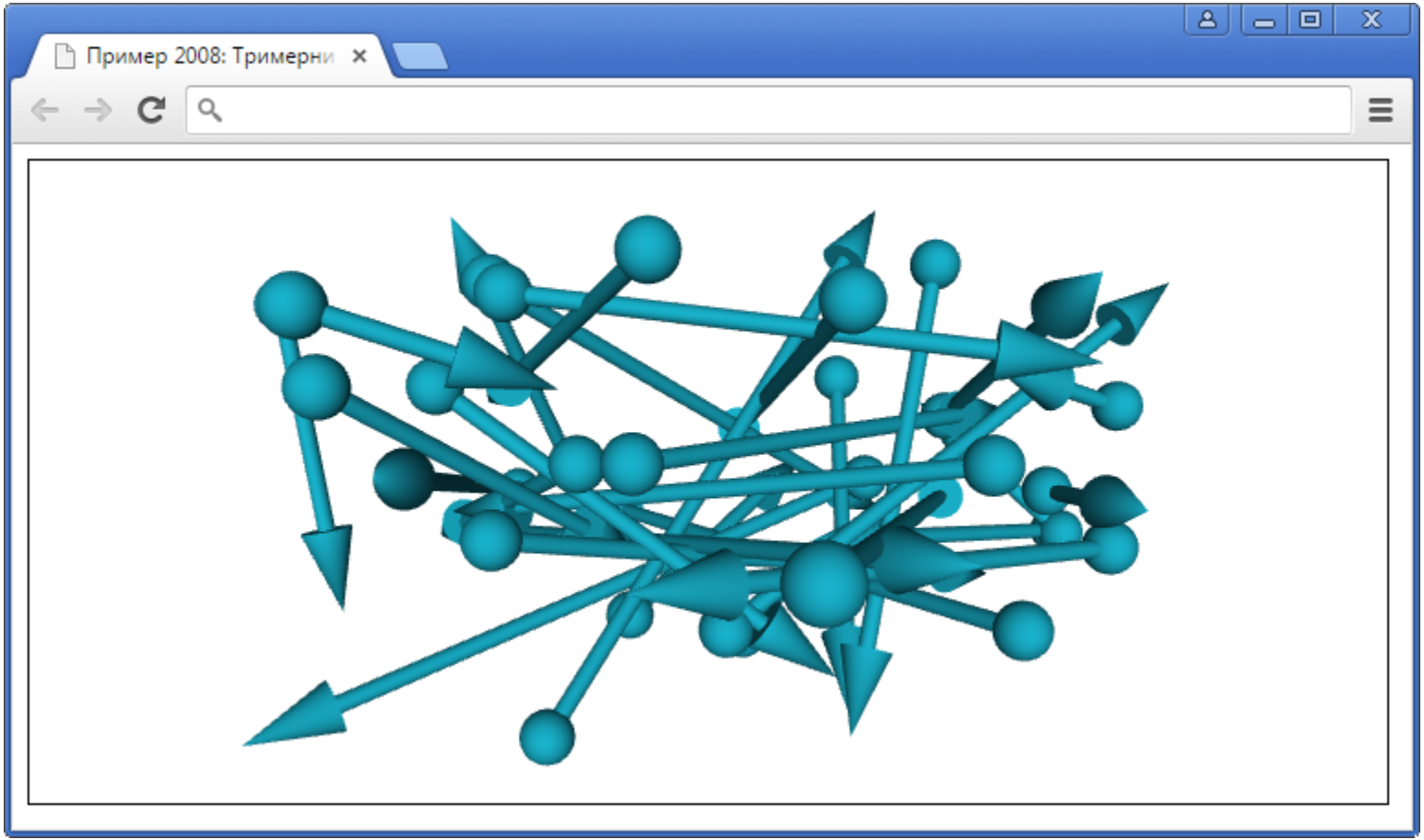



ПРОБА

# Алтернативно решение (с 3D вектор)

- Директно създаваме началната окръжност p и крайния, правилно отместен, но неориентиран конус q
- Едва сега оправяме ориентацията и на конуса (чрез v)
- Директно създаваме тялото на вектора с правилната дължина и ориентация

```
p = sphere([...],3);
q = cone([...],3,10).custom({origin:[0,0,1]});
v = vectorPoints(q.center,p.center);
q.focus = v;
cylinder( p.center, 1,
          Math.sqrt(scalarProduct(v,v))-10
        ).custom(color).custom({focus:v});
```




ПРОБА

# Върхове на обекти

## Триъгълна призма с фигури по стените

- Модел на няколко обекта, по-сложни от отсечка
- Те са нарисувани с линии, но имат точки по върховете

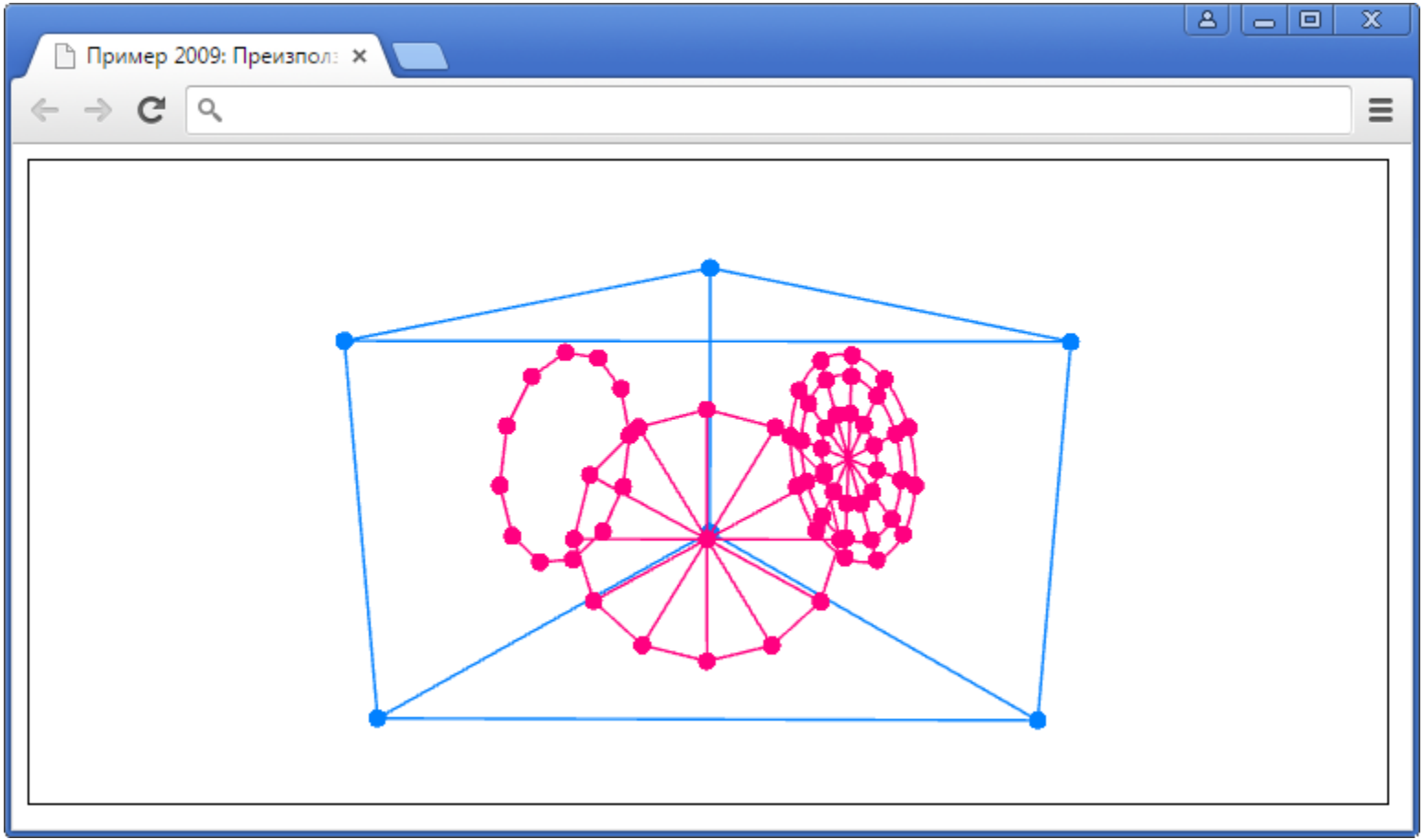
## Идея за реализация №1

- Аналогично на отсечката с окръжности в края
- Лесна имплементация
- Прекалено много обекти

# Идея за реализация №2

- Преизползване на обект
- Обект се създава в режим на рисуване с линии
- После  се клонира, но в режим на рисуване с точки

```
a = prism(...).custom({mode:Suica.LINE,...});
sameAs(a).custom({mode:Suica.POINT,pointSize:10});

a = cylinder(...).custom({mode:Suica.LINE,...});
sameAs(a).custom({mode:Suica.POINT,pointSize:10});
```




ПРОБА

# Сфера, цилиндър и конус

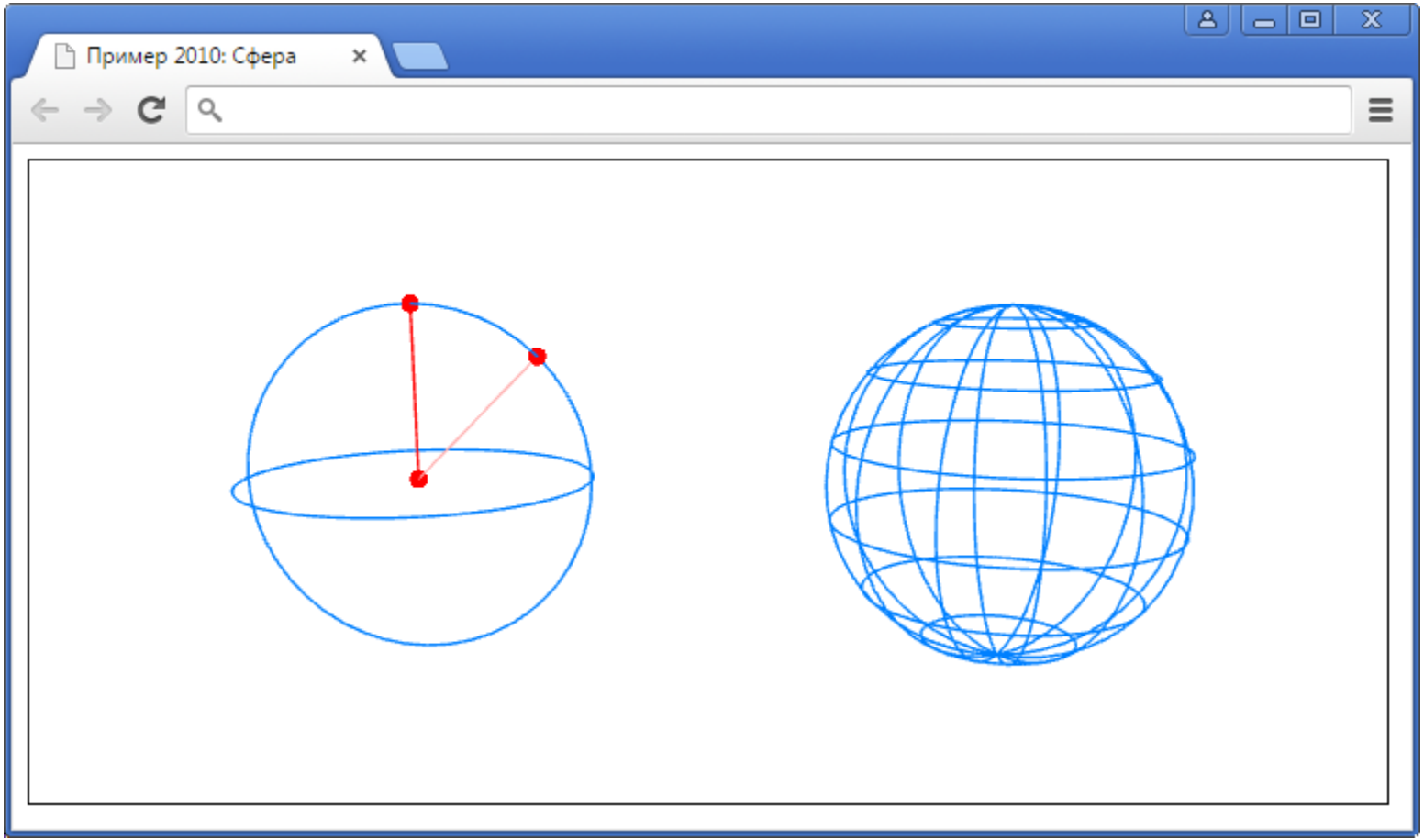
# Сфера, цилиндър и конус

## Стандартно показване

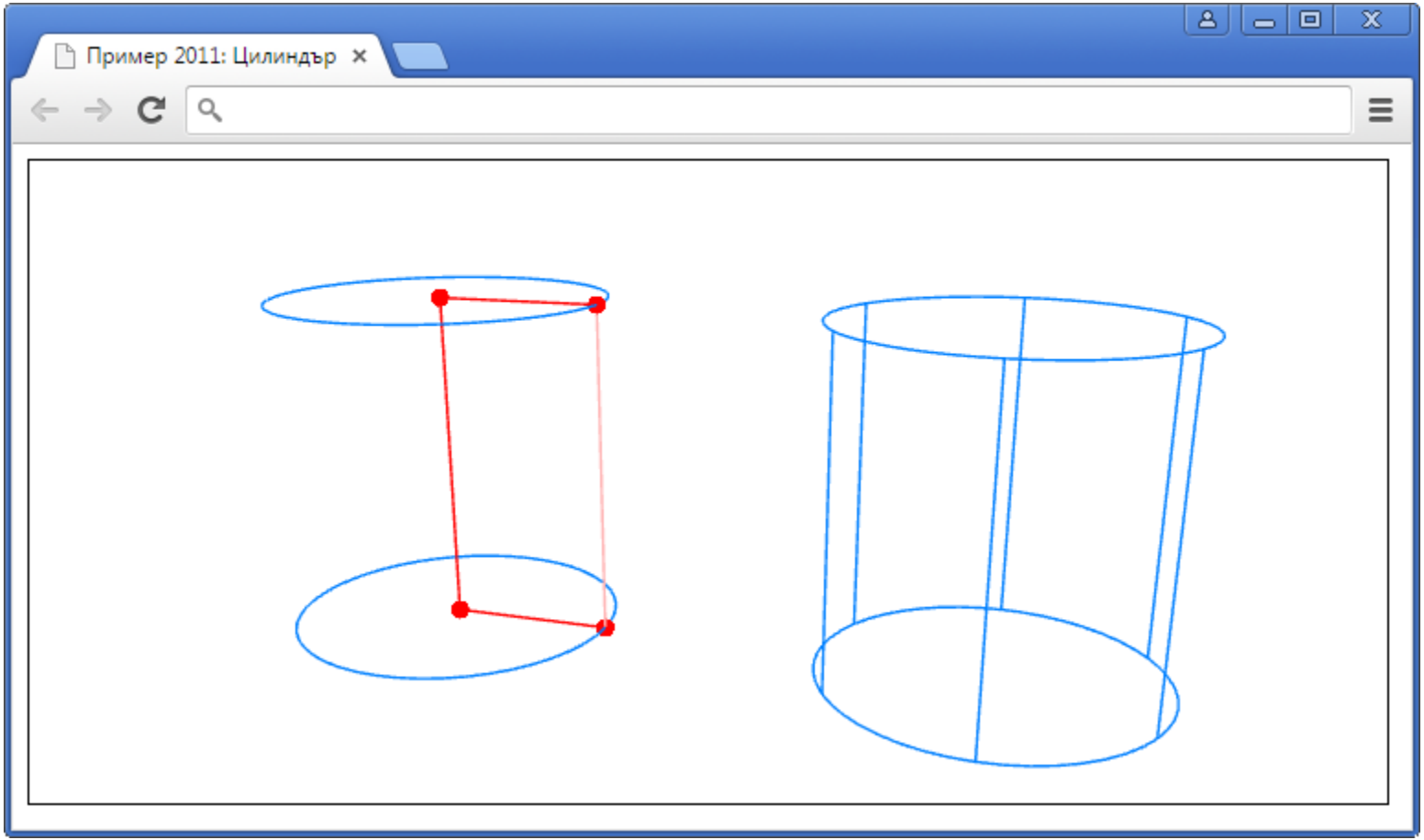
- С командите sphere, cylinder, cone
- С командата segment

## Липсва им

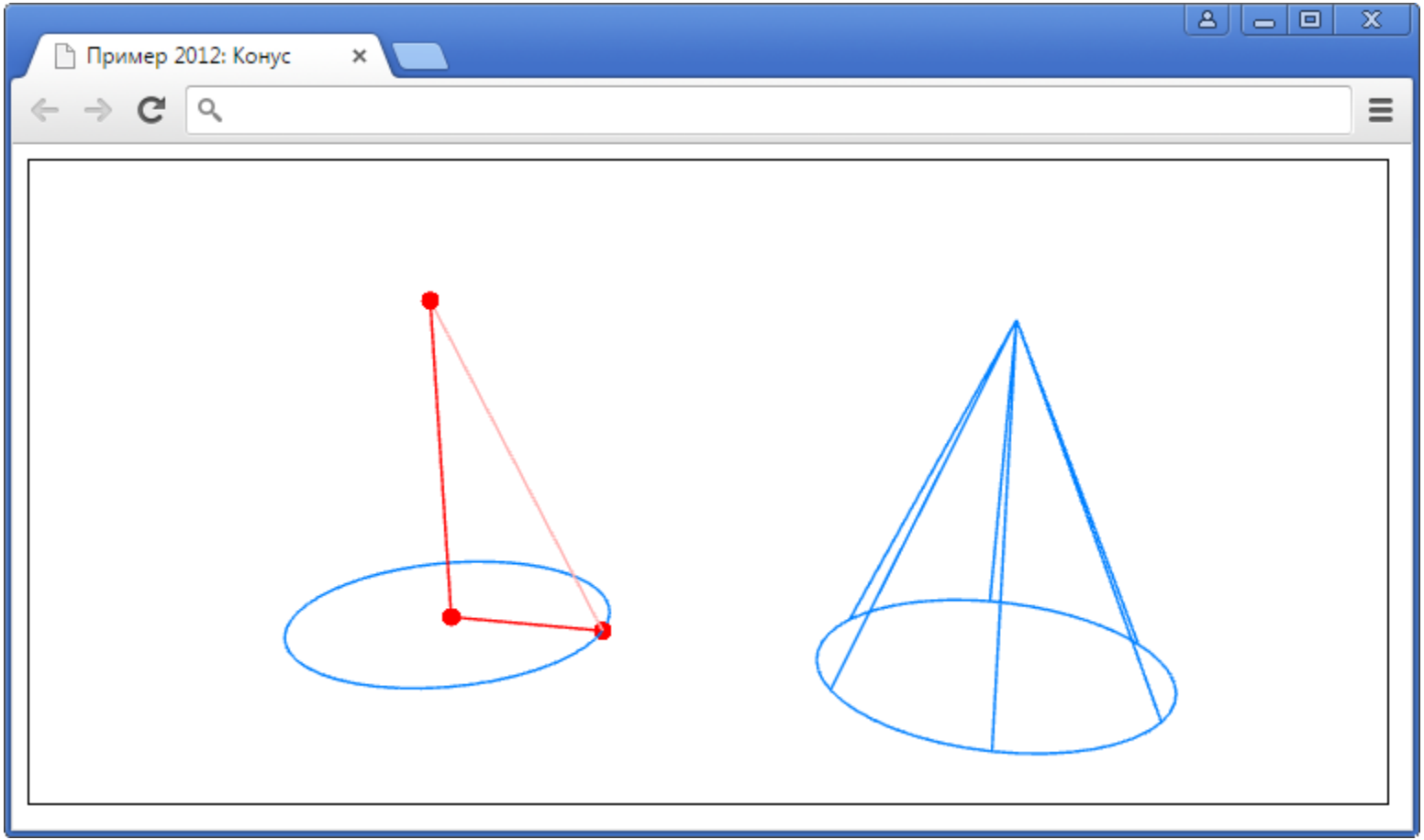
- Нищо не им липсва, а имат излишество
- Рисуват се „ръчно“ само тези елементи, които са нужни в конкретния случай
- Feci quod potui, faciant meliora potentes




ПРОБА




ПРОБА




ПРОБА

# Top

# Визуализиране на тор

## Основни проблеми

- Няма такъв вграден обект в СУИКА
- Няма друг обект, който след промяна на параметрите да стане с формата на тор
- Трябва да се конструира ръчно

# Създаване на тор

## Вариант №1 – вертикални окръжности

- Торът е окръжност, завъртяна около ос извън нея
- Дефинираме отместена окръжност и я въртим

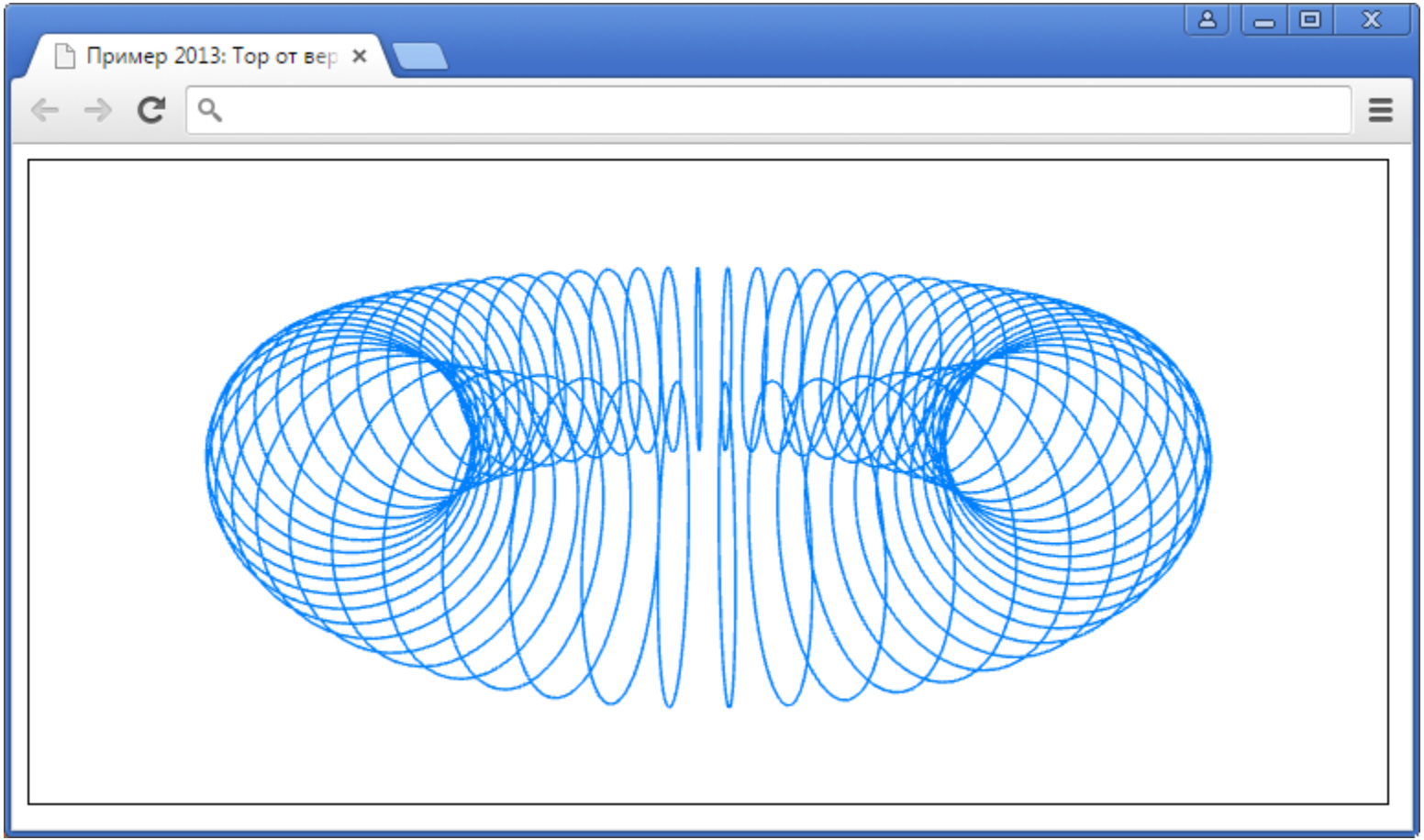
## Проблем

- Ако я изместим с center, то въртенето със spin ще я завърти около новия ѝ център, а не около външна ос
- Трябва spin да не знае, че има променен center
- Ако я изместим с origin, вероятно ще има проблем с focus

# Решение

- Скриваме правилно ориентирана окръжност в групов обект
- Така и окръжността, и групата си имат свои собствени focus, spin и center, независещи от тези на другия обект
- Клонираме с въртене групата

```
a = group([circle([0,30,5],10).custom({
              mode:Suica.LINE,
              color:[0,0.5,1],
              focus:[1,0,0]})
        ]);

for (var i=0; i<60; i++)
   sameAs(a).custom({spin:i*2*Math.PI/60});
```

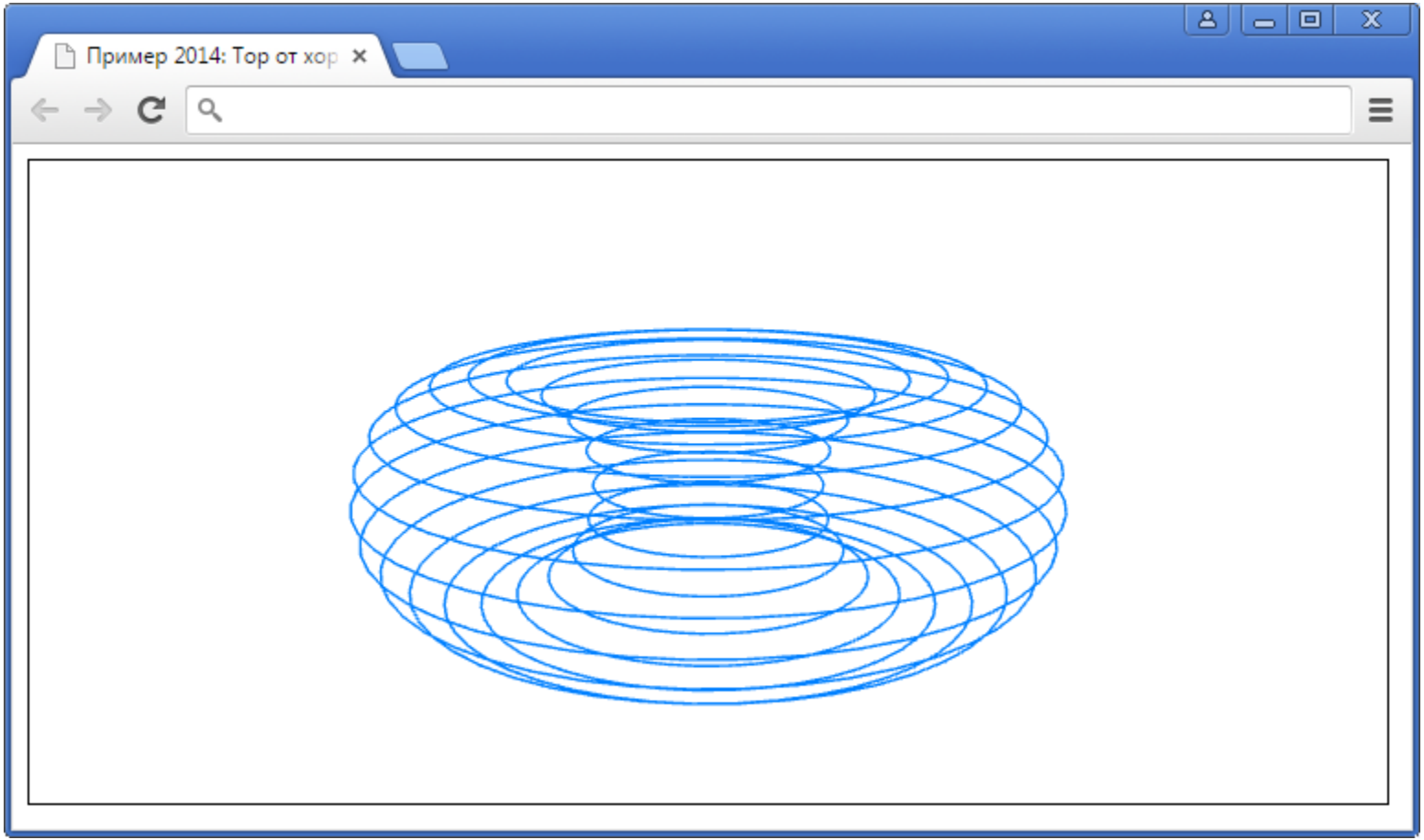



ПРОБА

# Вариант №2 – хоризонтални окръжности

- Създаваме хоризонтални окръжности, като техните z координати и радиуси описват … окръжност

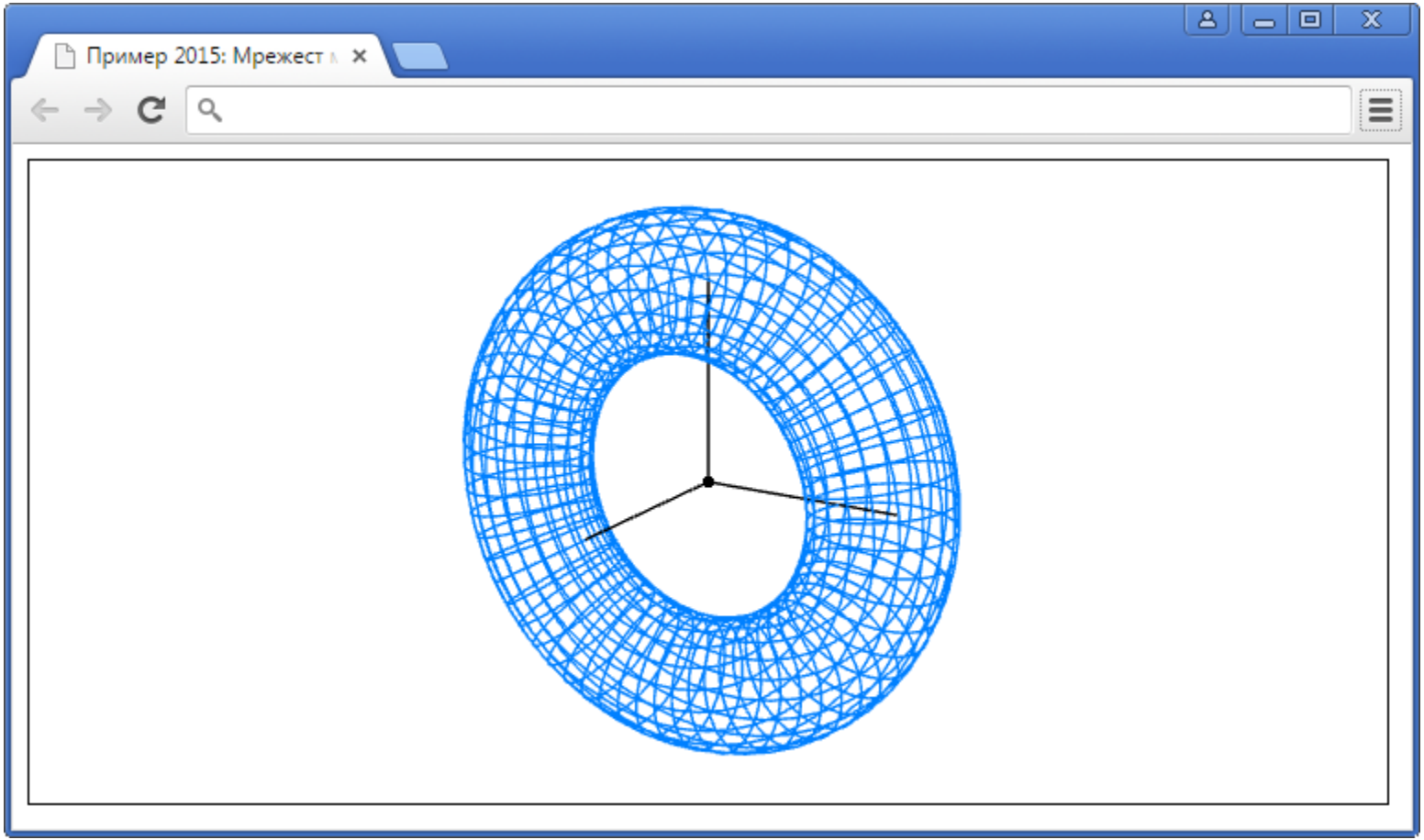
```
for (var i=0; i<20; i++)
{
   var a = 2*Math.PI*i/20;
   circle( [0,0,10*Math.sin(a)],
           20+10*Math.cos(a) ).custom({
        mode:Suica.LINE,
        color:[0,0.5,1]});
}
```




ПРОБА

# Вариант №3 – комбинирани окръжности

- Торът изглежда като сглобен от плочки
  - Създаваме вертикални окръжности
  - Създаваме хоризонтални окръжности
- Всичките ги слагаме в нова група – така можем да ориентираме целият тор и да го изправим вертикално




ПРОБА

# Обобщение

# Визуализиране

## Визуализиране на геометрични обекти

- Образите на геометричните  обекти невинаги съвпадат с образите на графичните обекти
- Композират се образи от няколко обекта, за да се създаде желания образ
- Понякога образът се изгражда наново
- В СУИКА са дефинирани само някои обекти; всички останали се генерират от потребителя

# И най-важното

- Всеки обект може да бъде представен графично по няколко различни начина
- Изборът е въпрос на дизайн, стил или възможност

ИКТ в НОС

# Край

Коментари, въпроси