ИКТ в НОС

Квадрати и кубове

Тема №10

Квадрат и правоъгълник

Квадрат в СУИКА

Квадрат

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

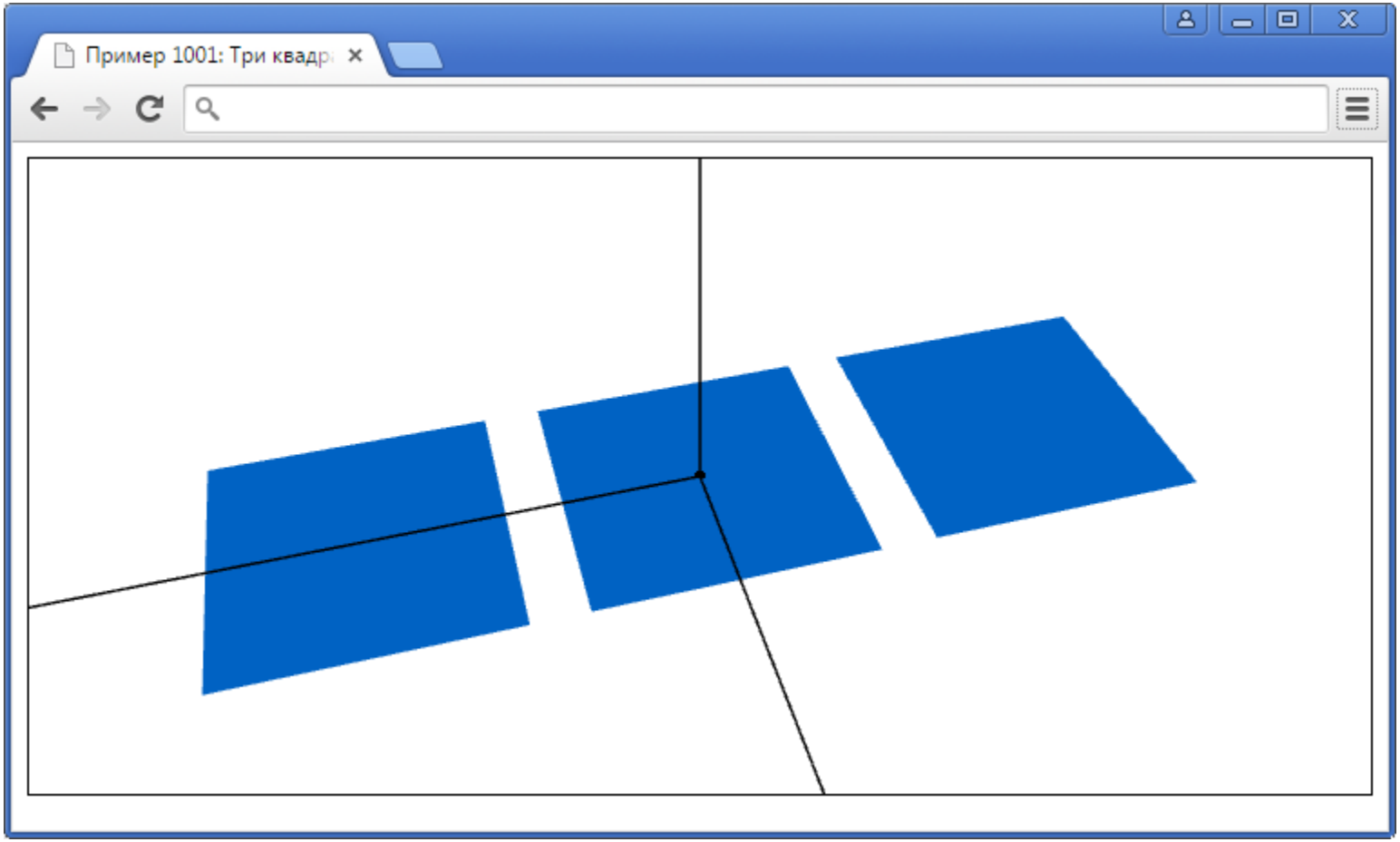
Създаване на квадрат

- Чрез клас new Suica.Square (*център, размер*)
- Чрез функция square (*център, размер*)
- Центърът е координати на точка, масив от три числа
- Размерът е число и е дължината на страната

Пример

- Да се създадат 3 квадрата един до друг
- Да са еднакви по размер
- Да има малко разстояние между тях

```
square([-6,0,0],5);
square([ 0,0,0],5);
square([+6,0,0],5);
```



ПРОБА

Свойства

Режим на рисуване

- Незадължително свойство
- Записва се в променливата mode
- Определя как ще се нарисува квадрата
  - Suica.POINT рисуват се върховете
  - Suica.LINE рисуват се ръбовете
  - Suica.SOLID рисуват се стените (по подразбиране)

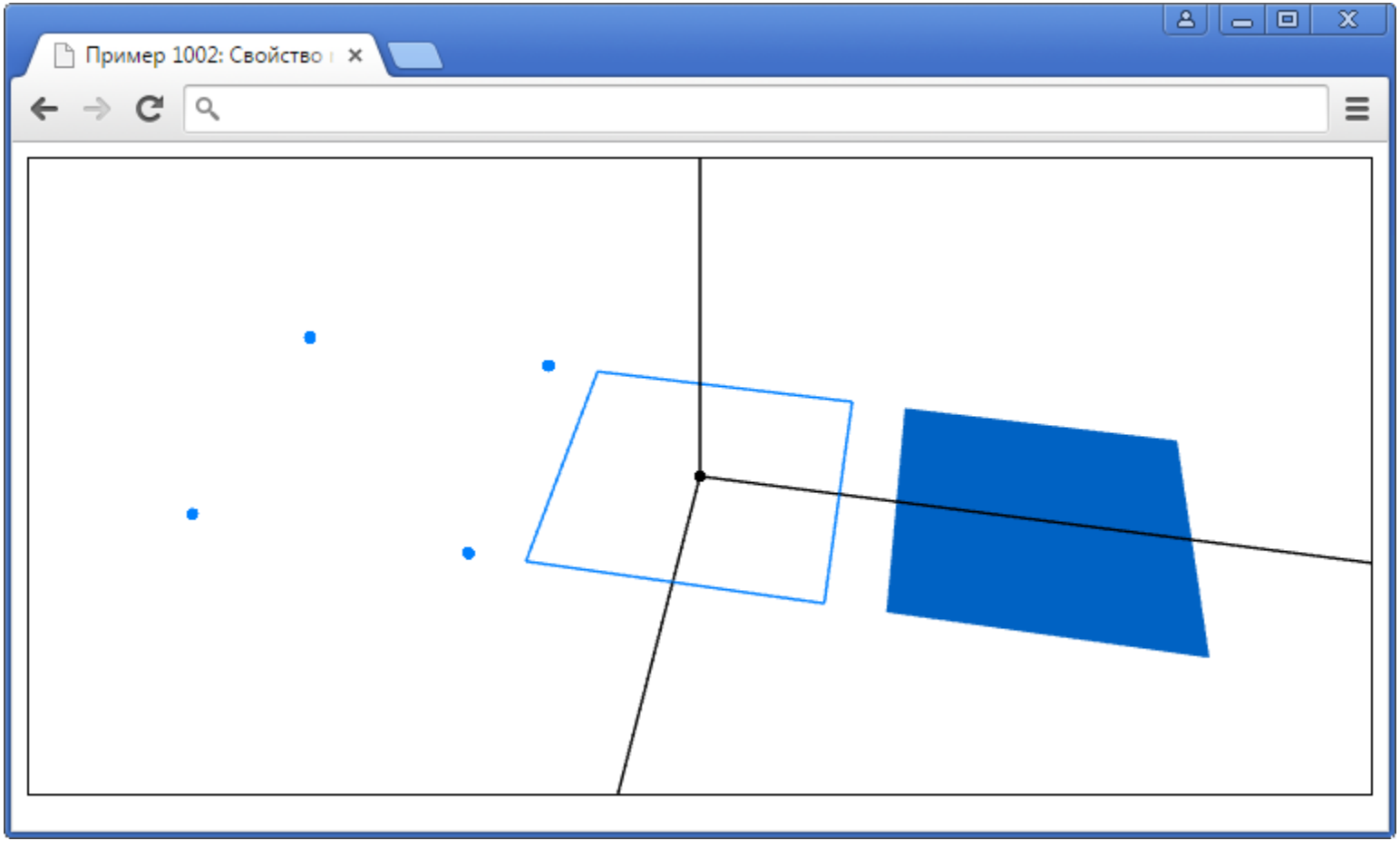
Пример

- Три квадрата в трите режима на рисуване

```
a = square([0,-6,0],5);
a.mode = Suica.POINT;
a.pointSize = 7;

a = square([0,0,0],5);
a.mode = Suica.LINE;

a = square([0,6,0],5);
a.mode = Suica.SOLID;
```



ПРОБА

Начало

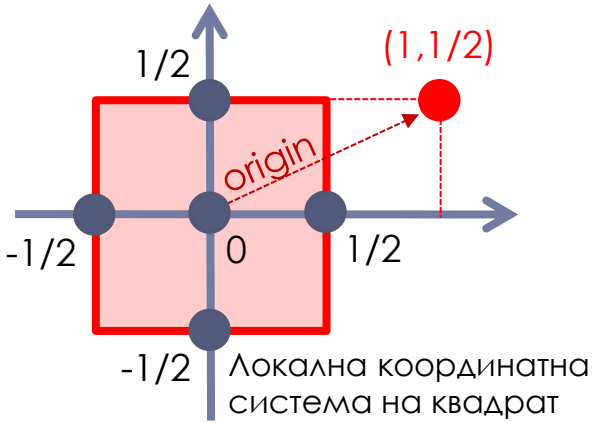
- Незадължително свойство
- Записва се в променливата origin
- Определя коя точка е „центърът“

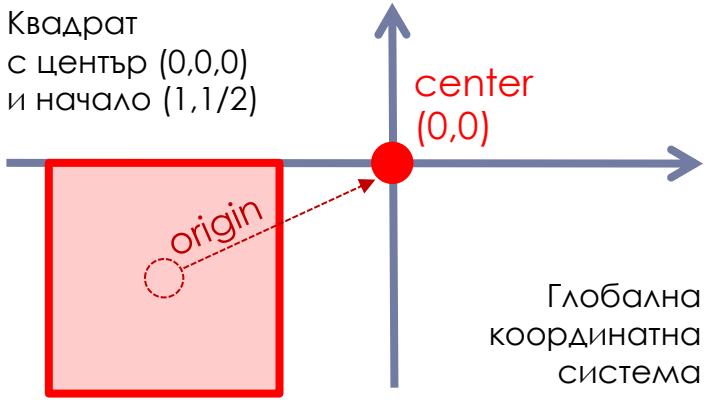
Роля на origin

- В някои случаи е по-удобно да се задава положение на квадрат чрез координати на негов връх или среда на страна
- Със свойството origin се постига тази гъвкавост
- По подразбиране origin е (0,0,0) и това съвпада с геометричния център на квадрата

Координати на origin

- Измерват се в локална координатна система на квадрата
  - Начало = геометричният център на квадрата
  - Единица разстояние = дължината на страната на квадрата
- При рисуване на квадрат, той се позиционира с неговото начало origin да съвпада с подадените координати за център





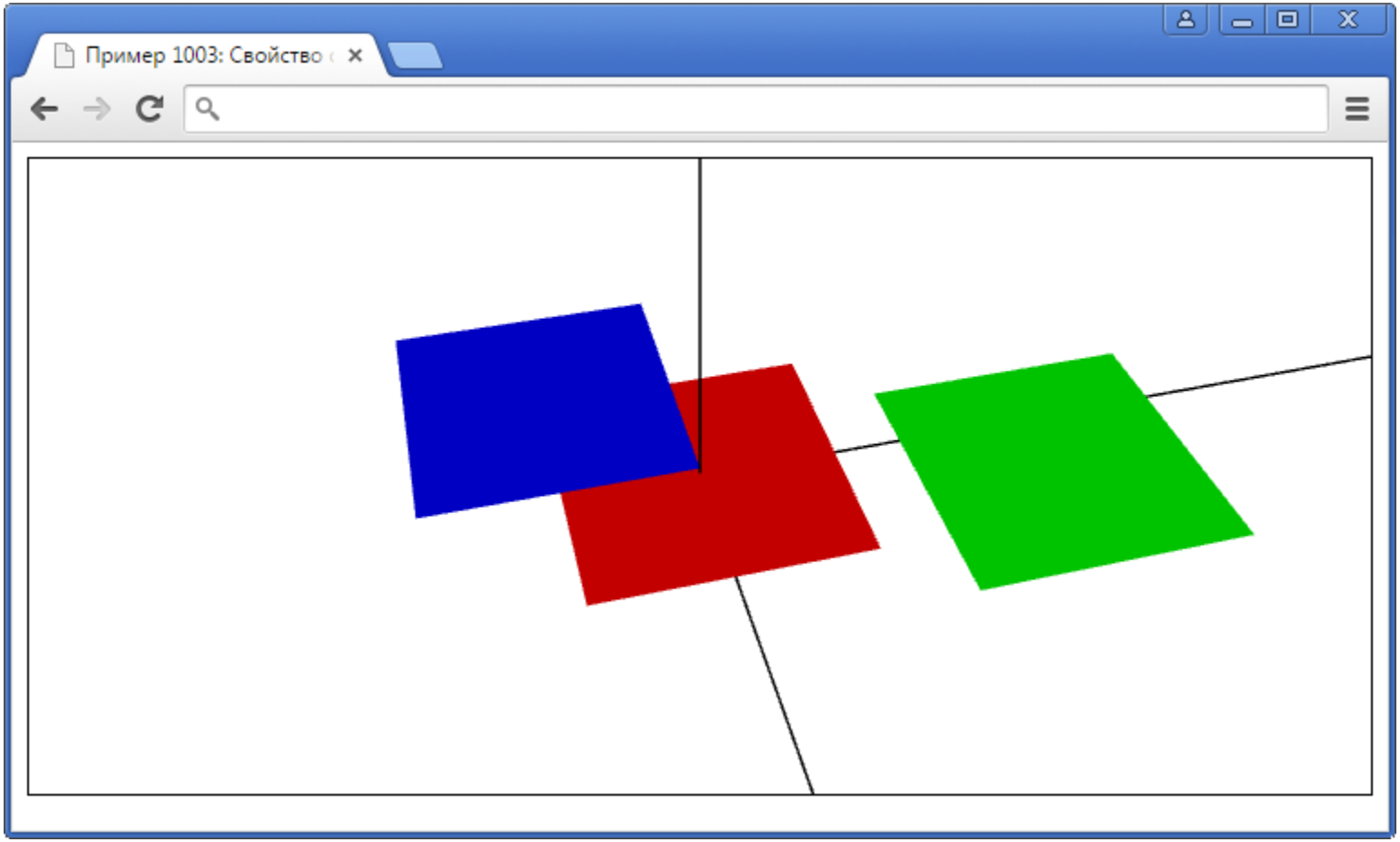
Пример

- Квадрати с origin (0,0,0), (1/2,1/2,0) и (-1/4,5/4,0)

```
a = square([0,0,0.1],5);
a.color = [1,0,0];

a = square([0,0,0.2],5);
a.origin = [0.5,0.5,0];
a.color = [0,0,1];

a = square([0,0,0.1],5);
a.origin = [-0.25,-1.25,0];
a.color = [0,1,0];
```



ПРОБА

Правоъгълник в СУИКА

Правоъгълник

- Графичен обект със свойства подобни на квадрата
- Използва се за рисуване на квадрати и правоъгълници

Създаване на правоъгълник

- Чрез клас new Suica.Rectangle (*център, размери*)
- Чрез функция rectangle (*център, размери*)
- Центърът е координати на точка, масив от три числа
- Размерите са масив от 2 числа – дължини на страните

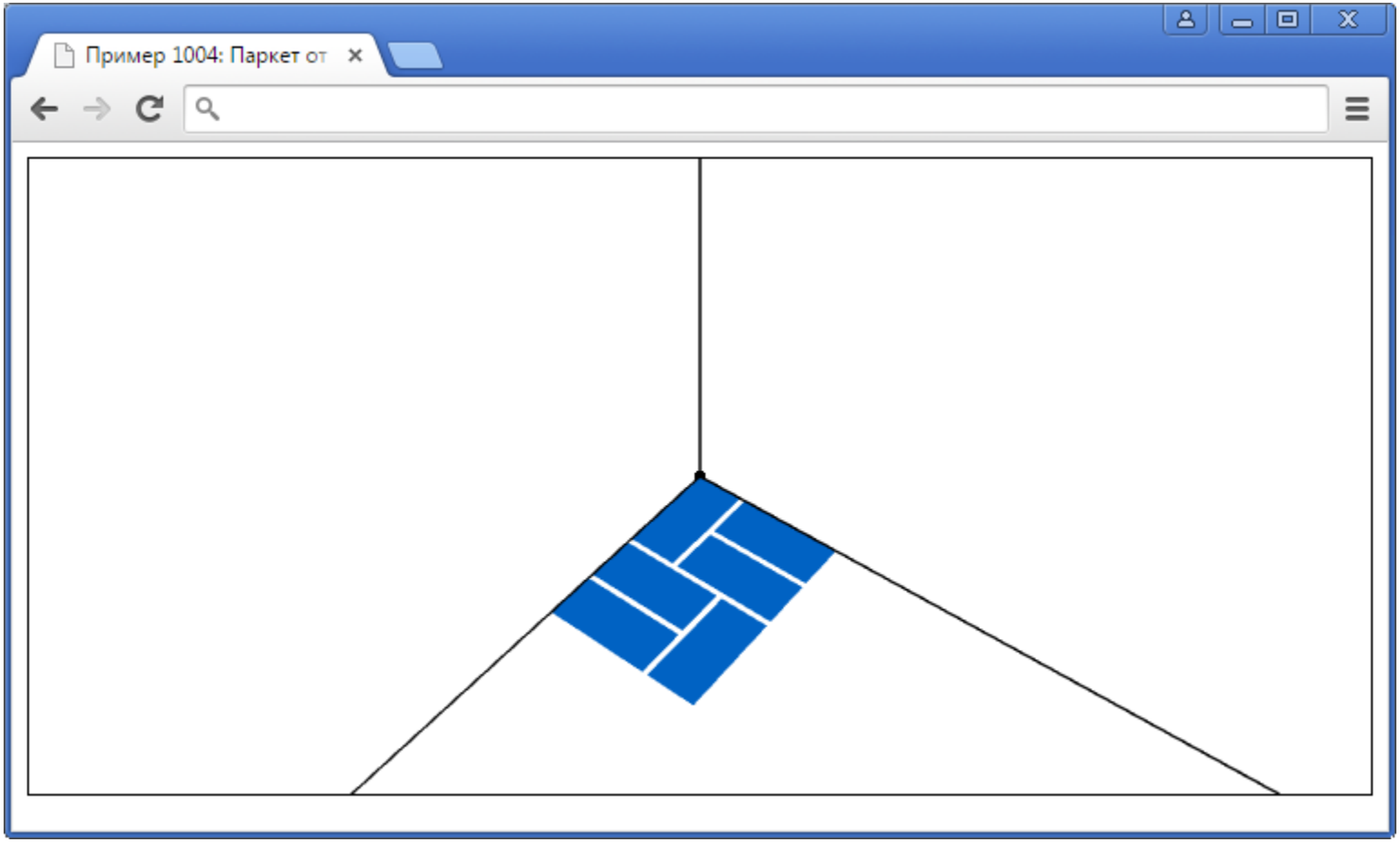
Пример

- Фрагмент от паркет от правоъгълници 2x1
- За по-лесни координати, центровете са в единия им връх
- Реалните размери са 1.9x0.9, за да има фуги

```
a = rectangle([0,0,0],[1.9,0.9]);
a.origin = [-0.5,-0.5,0];

a = rectangle([0,1,0],[0.9,1.9]);
a.origin = [-0.5,-0.5,0];

a = rectangle([1,1,0],[0.9,1.9]);
a.origin = [-0.5,-0.5,0];
:
```



ПРОБА

Куб и паралелепипед

Куб в СУИКА

Куб

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

Създаване на куб

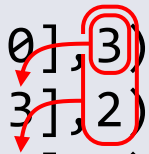
- Чрез клас new Suica.Cube (*център, размер*)
- Чрез функция cube (*център, размер*)
- Центърът е координати на точка, масив от три числа
- Размерът е число и е дължината на страната

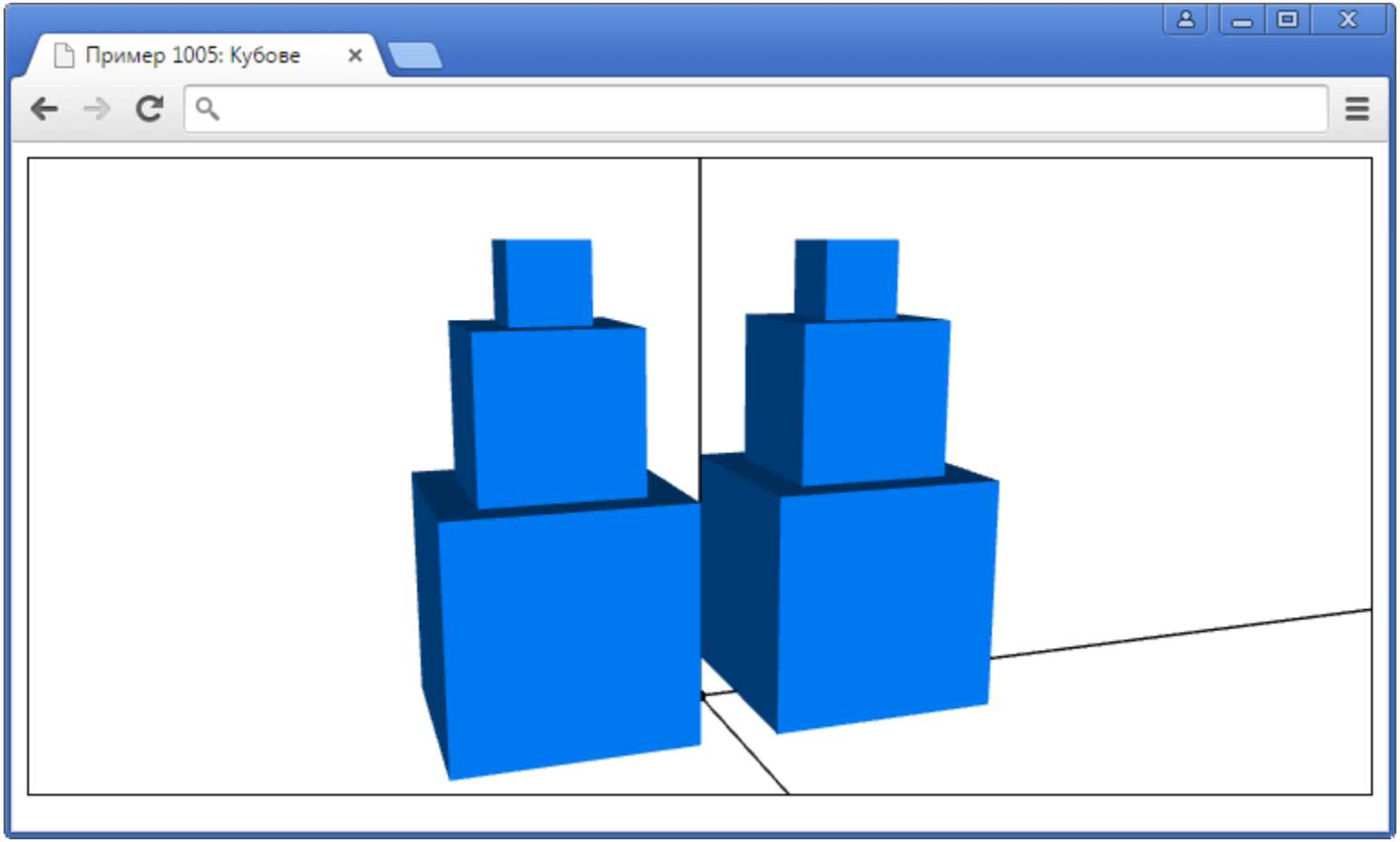
Пример

- Две пирамиди от кубове с размери 3, 2 и 1
- В едната пирамида кубовете са без отместване
- В другата пирамида те са с отместване

```
cube([0,-2,1.5],3);
cube([0,-2,4.0],2);
cube([0,-2,5.5],1);

a = cube([0,2,0],3); a.origin=[0,0,-1/2];
a = cube([0,2,3],2); a.origin=[0,0,-1/2];
a = cube([0,2,5],1); a.origin=[0,0,-1/2];
```





ПРОБА

Паралелепипед в СУИКА

Паралелепипед

- Графичен обект със свойства
- Използва се за рисуване на кубове и правоъгълни паралелепипеди

Създаване на паралелепипед

- Чрез клас new Suica.Cuboid (*център, размери*)
- Чрез функция cuboid (*център, размери*)
- Центърът е координати на точка, масив от три числа
- Размерите са масив от 3 числа – дължини на страните

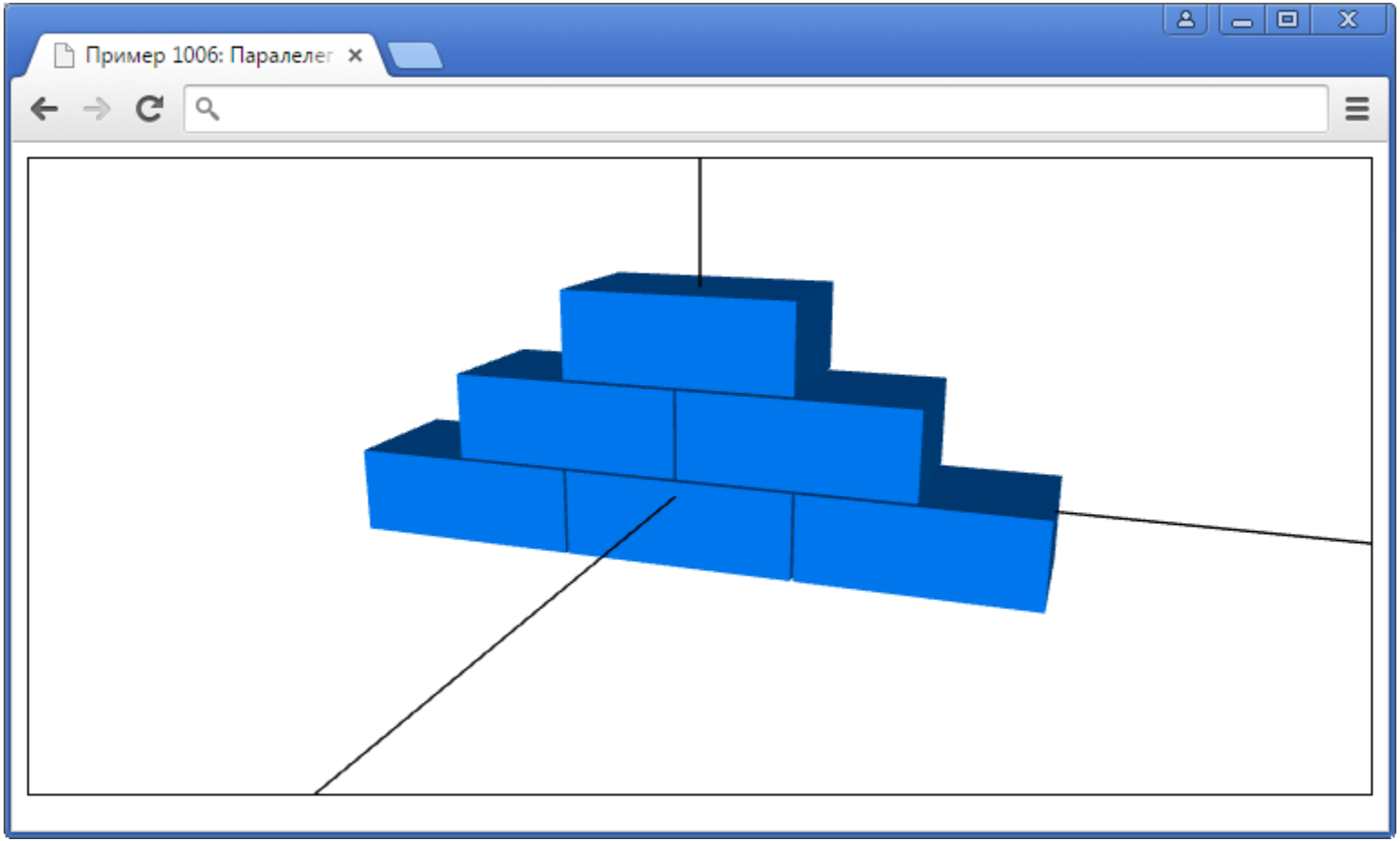
Пример

- Малка стена от 6 тухли с размери 8x5x3

```
cuboid([0,0,-1],[5,8,3]);
cuboid([0,-8.1,-1],[5,8,3]);
cuboid([0,+8.1,-1],[5,8,3]);

cuboid([0,-4.05,2.1],[5,8,3]);
cuboid([0,4.05,2.1],[5,8,3]);

cuboid([0,0,5.2],[5,8,3]);
```



ПРОБА

Ориентация

Ориентация на обект

Положение на обект

- Визуалното положение на 2D/3D обект се определя от свойствата center и size/sizes

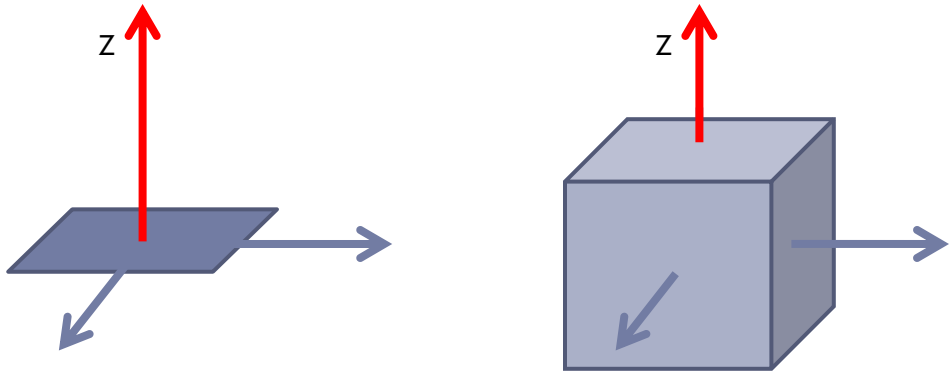
Проблем

- Те не са достатъчни
- Няма как да се завърти обект

Реализация на ориентацията

Невидими елементи

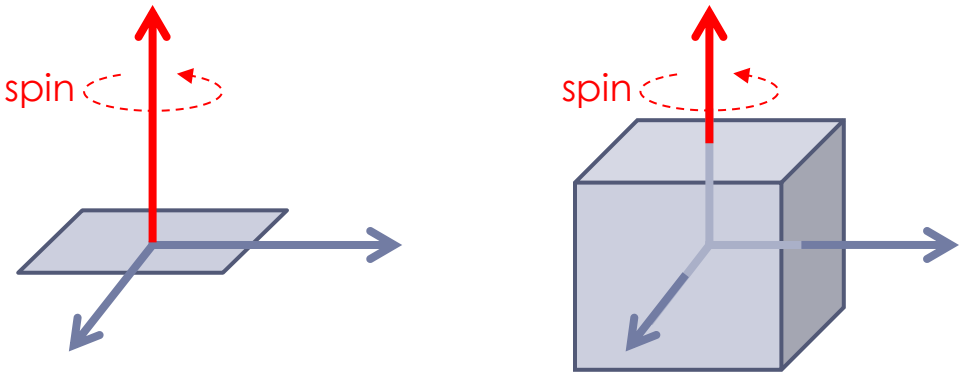
- Всеки обект си има локална координатна система (тя се използва при свойството origin)
- Локалната ос Z се използва за ориентиране на обекта



Свойства

Свойство spin

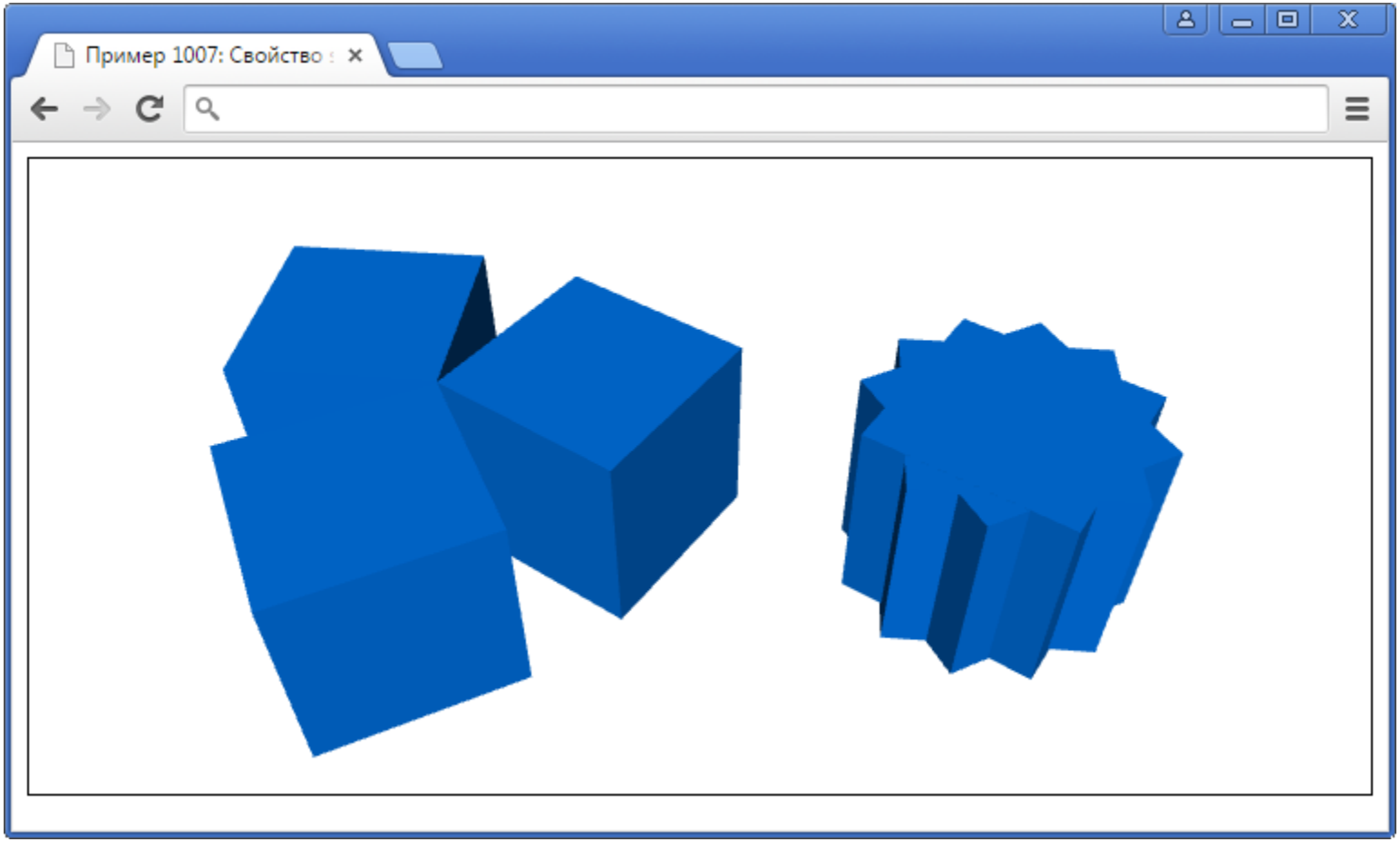
- Завърта обекта около локалната Z ос
- Стойността на spin е в радиани
- По подразбиране spin е 0



Пример

- Кубове, завъртени около централната си ос
- Кубове, завъртени около околен ръб чрез преместване на центъра върху ръба

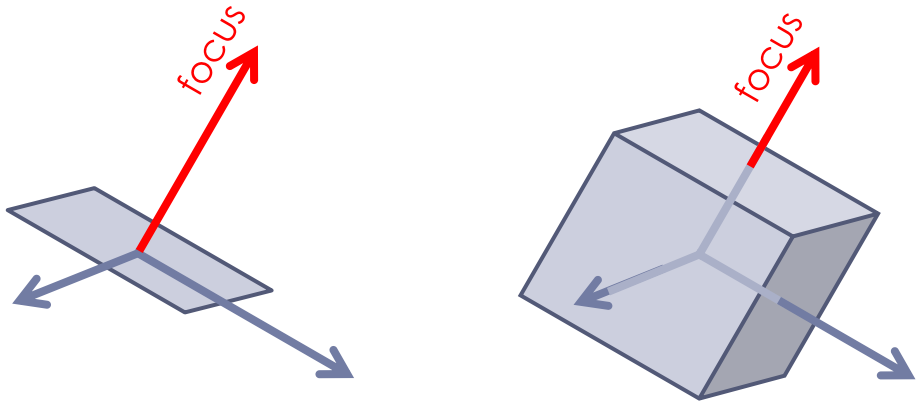
```
a = cube([0,7,0],5);
a.spin = Math.PI/6;
:
a = cube([0,-6,0],5);
a.origin = [0.5,0.5,0];
a.spin = 2*Math.PI/3;
```



ПРОБА

Свойство focus

- Променя посоката на локалната Z ос
- Променя и другите оси, за да запази координатната система
- Стойността на focus е вектор
- По подразбиране е (0,0,1) и съвпада с глобалната ос Z



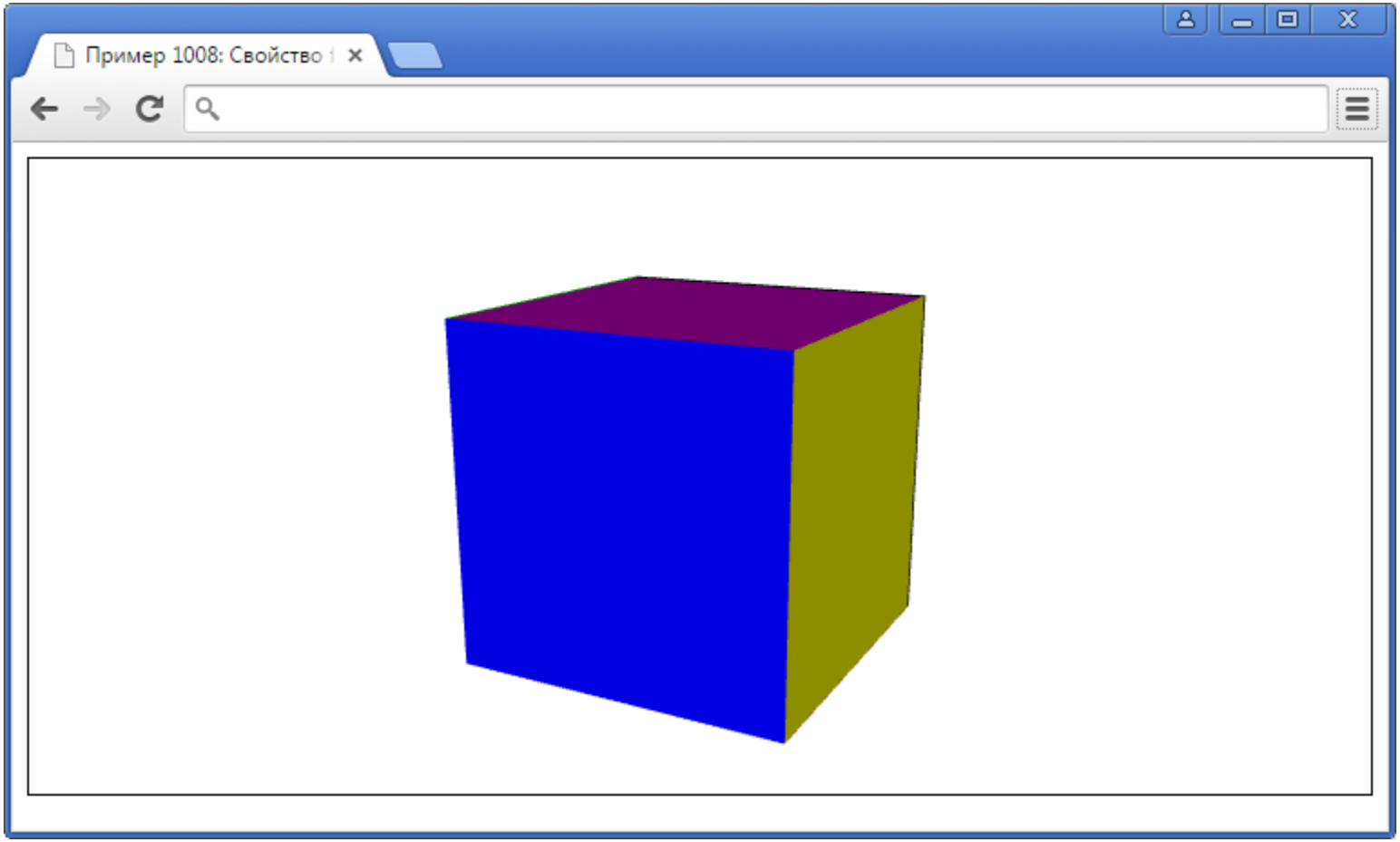
Пример

- Куб със стени с различни цветове – сглобен от квадрати

```
a = square([-3,0,0],6); a.color = [0,0,0];
a.focus = [-1,0,0];
a = square([3,0,0],6); a.color = [0,0,1];
a.focus = [1,0,0];

a = square([0,-3,0],6); a.color = [0,1,0];
a.focus = [0,-1,0];
a = square([0,3,0],6); a.color = [1,1,0];
a.focus = [0,1,0];

a = square([0,0,-3],6); a.color = [1,0,0];
a = square([0,0,3],6); a.color = [1,0,1];
```



ПРОБА

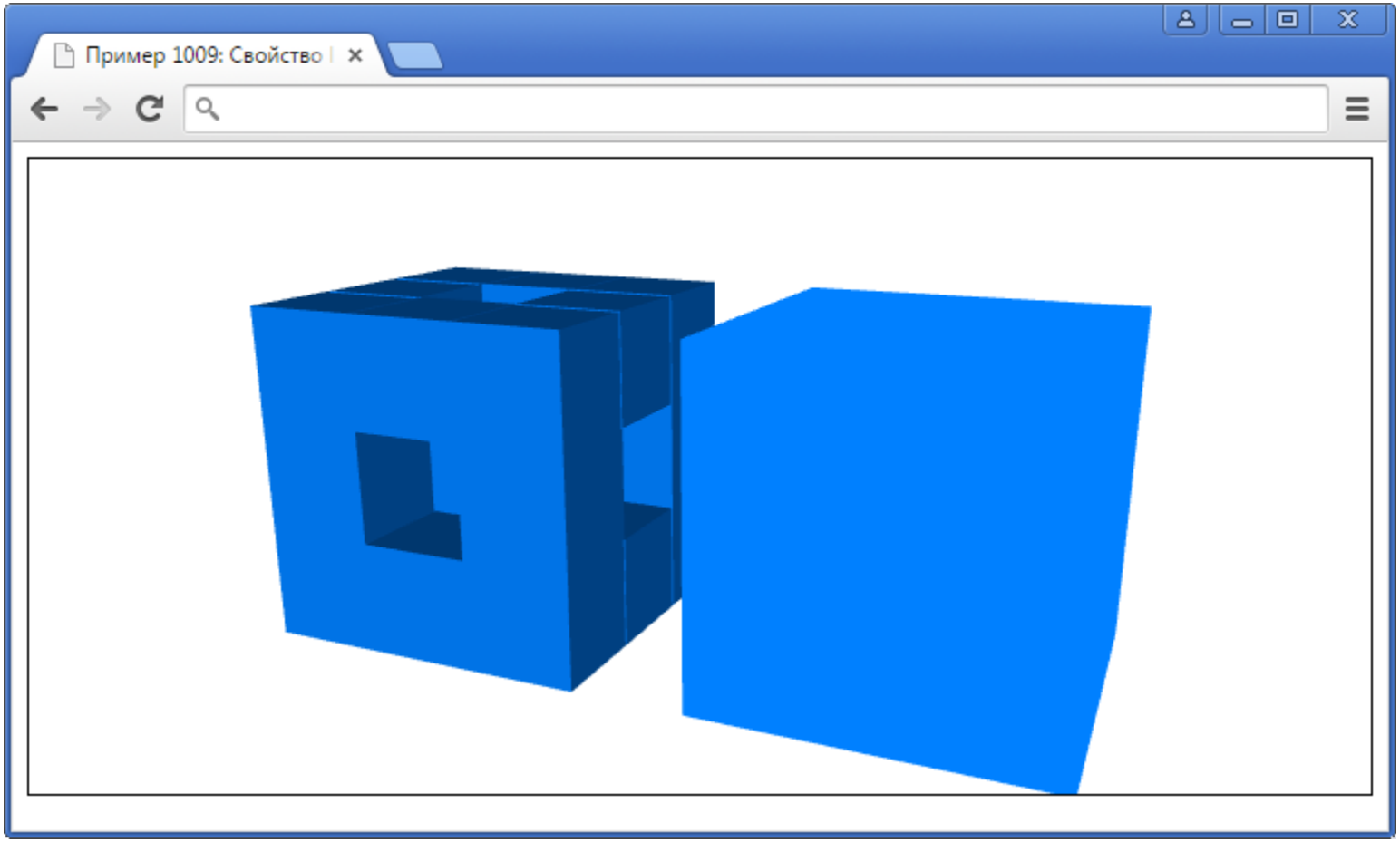
Взаимодействие на свойствата

- Ако се промени ориентацията с focus, чрез spin се прави завъртане около новата ос
- Векторът focus е с начало center на обекта, който може да бъде изместен с origin
- Не всяка ориентация е възможно да се направи само със focus, понякога е нужно да се ползват и focus, и spin

Свойство light

- Използва или игнорира светлинен източник
- При използването му страничните стени са по-тъмни
- При игнорирането му всички стени са с еднакъв цвят

```
for (var x=-1; x<=1; x++)
   for (var y=-1; y<=1; y++)
      for (var z=-1; z<=1; z++)
         if (Math.abs(x)+Math.abs(y)+Math.abs(z)!=1)
         {
               a = cube([x-2,y,z],1);
               a = cube([x+2,y,z],1);
               a.light = false;
         }
```



ПРОБА

Примери

Пример №1

Пирамида от квадрати

- Концентрични квадрати един над друг
- Размерът им се смалява до достигане на 1x1

Идея за решение

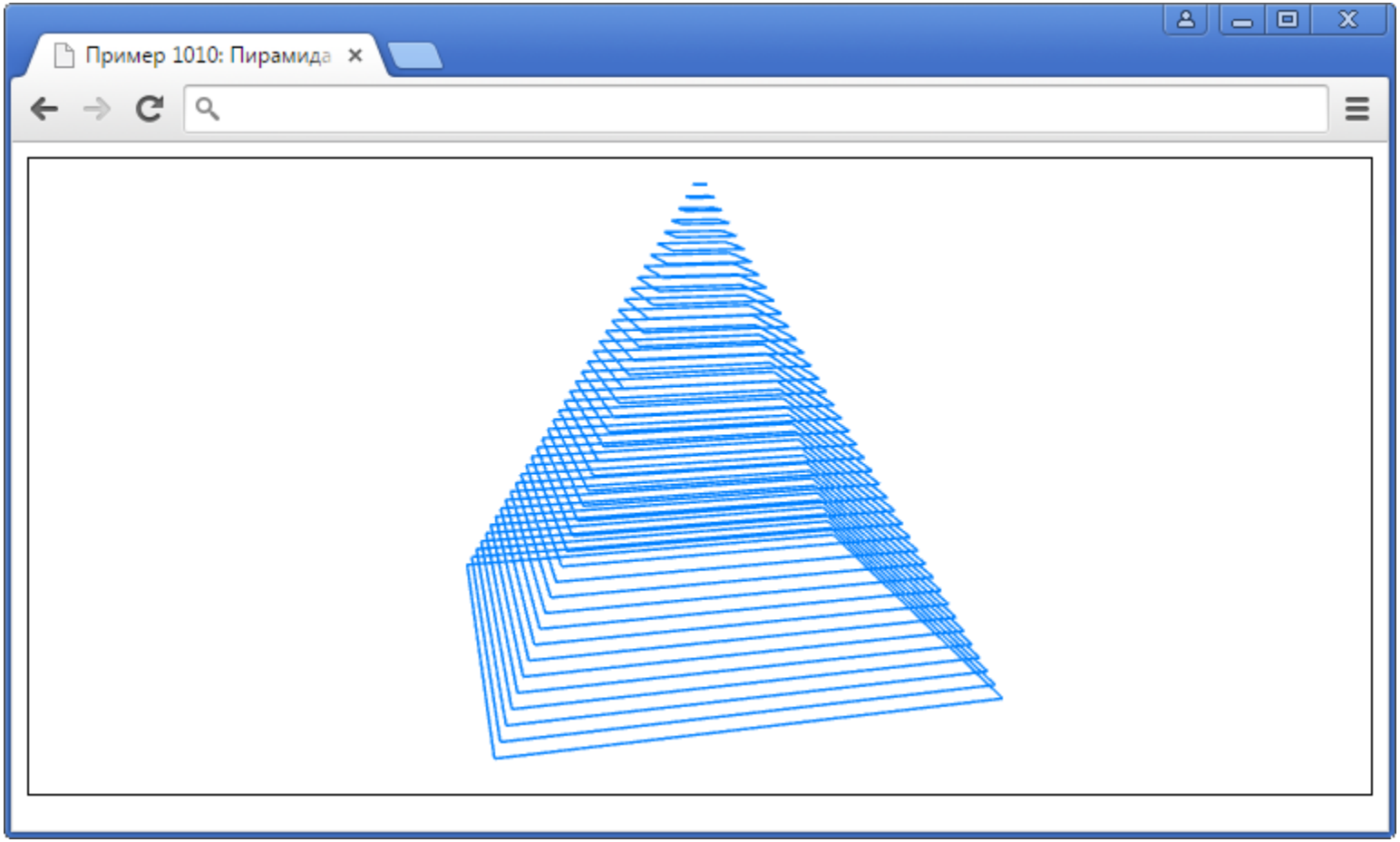
- Избираме си начало – върхът на пирамидата
- Всеки следващ квадрат е с размер, увеличен с 1
- Координатите на центърът му по X и Y са същите, но по Z са намалени с 1

Решение

- Генерираме квадратите от върха към основата
- Вертикалната позиция на квадрат е в z, което намалява
- Размерът е в i, което се увеличава

```
n = 40;
z = 25;

for (var i=1; i<=n; i++)
{
   a = square([0,0,z],i);
   a.mode = Suica.LINE;
   z--;
}
```



ПРОБА

Пример №2

Долепени правоъгълници

- Две перпендикулярни прави
- Случайни правоъгълници, долепени до тях
- Центровете на всички да са по самите прави

Идея за решение

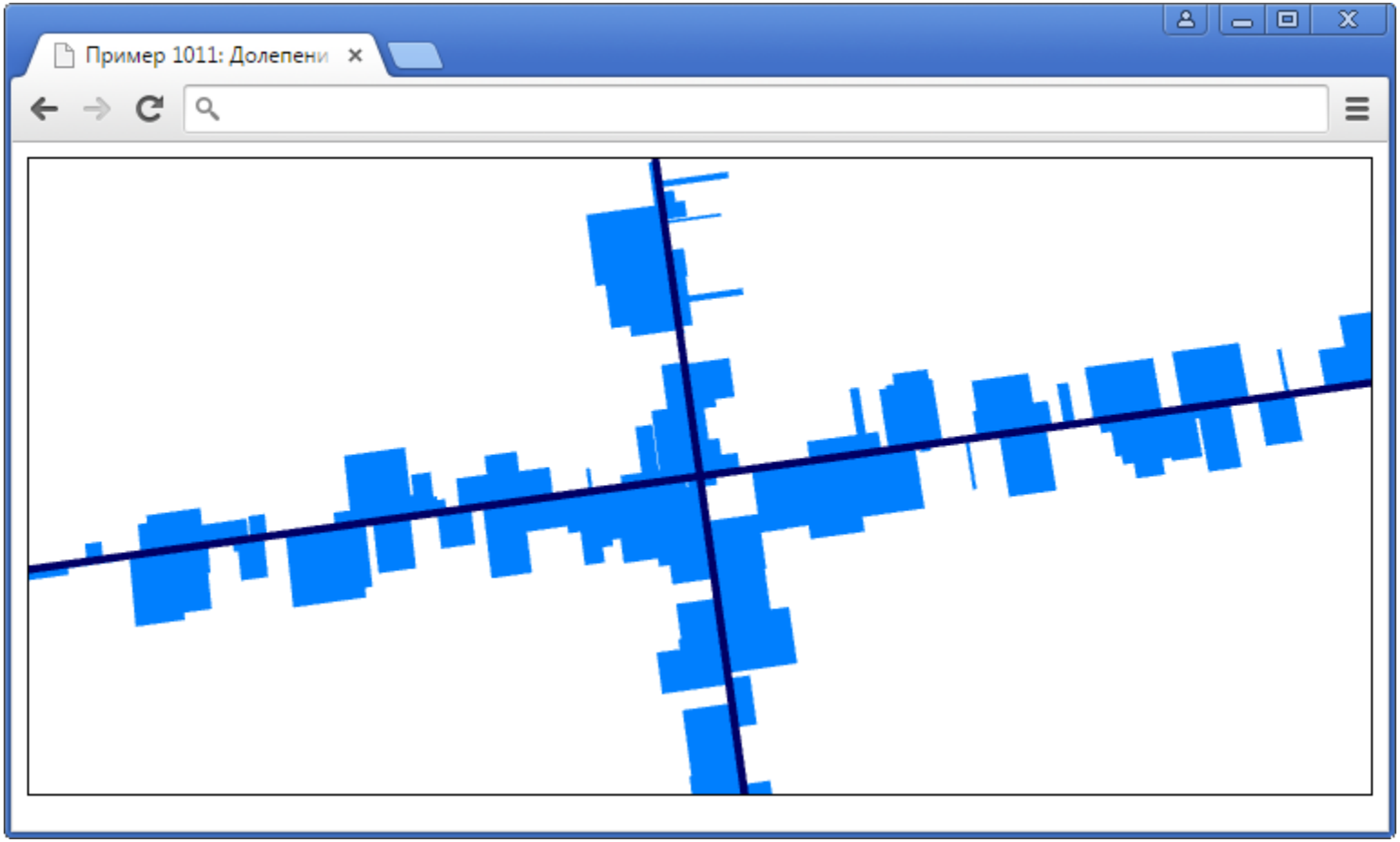
- За удобство правите са по осите X и Y
- Правоъгълниците, долепени по X, отместваме с $(0,\pm 1/2,0)$
- Другите правоъгълници отместваме с $(\pm 1/2,0,0)$

Решение

- Правоъгълници с центрове (x,0,0) и (0,y,0)
- За отместване използваме знака на случайно число от -1 до 1
- На правоъгълници с център (x,0,0) отместването е (0,±1/2,0)
- На правоъгълници с център (0,y,0) отместването е (±1/2,0,0)

```
for (var i=0; i<n; i++)
{
   a = rectangle([random(-10,10),0,0],[…]);
   a.origin = [0,Math.sign(random(-1,1))/2,0];

   a = rectangle([0,random(-10,10),0],[…]);
   a.origin = [Math.sign(random(-1,1))/2,0,0];
}
```



ПРОБА

Пример №3

Вита стълба

- Плоскости са разположени спираловидно

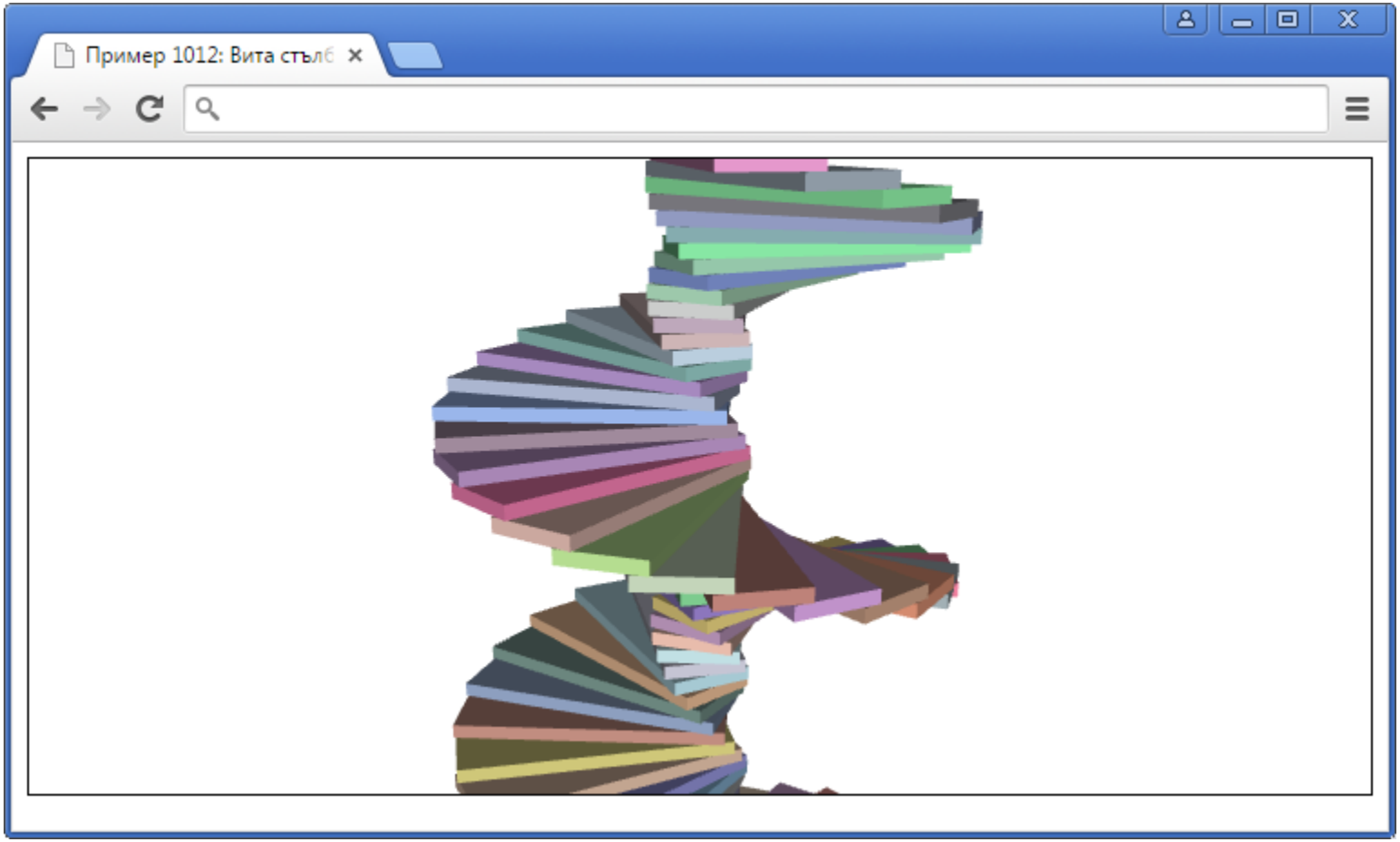
Идея за решение

- Всяко стъпало е плосък правоъгълен паралелепипед
- Центровете им са разположени по вертикална линия
- Центровете на стъпалата са изместени в края им
- Всяка стъпало е завъртяно на съответния ъгъл

Решение

- Стъпалата са разположени по вертикала от -40 до 24
- Отместването на центъра е почти до края – 0.4 вместо 0.5
- Всяко стъпало е завъртяно на 15 градуса спрямо предходното

```
for (var z=-40; z<25; z++)
{
   a = cuboid([0,0,z],[20,6,1]);
   a.origin = [-0.4,0,0];
   a.color = [random(0.5,1),random(0.5,1),
                                    random(0.5,1)];
   a.spin = z*radians(15);
}
```



ПРОБА

Пример №4

Сфера от кубове

- Пръснати по сфера, случайно завъртени кубове

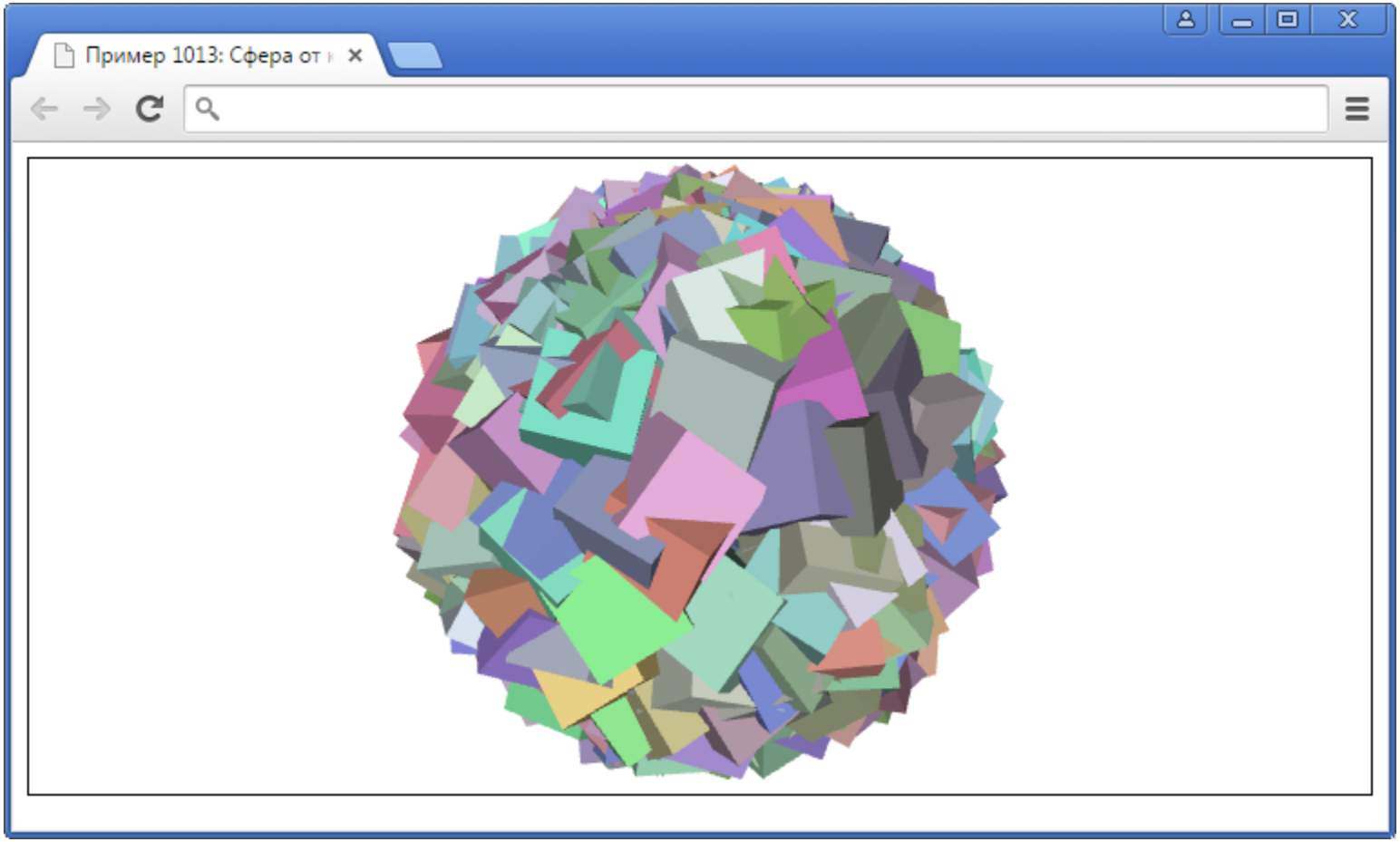
Идея за решение

- Избираме два случайни ъгъла и фиксиран радиус
- Те определят точка от сфера – там е центъра на куб
- Ориентацията на куба е зададена случайно, за да изглеждат кубовете хаотични

Решение

- Трансформация от сферични към декартови координати
- Случаен вектор за focus, случаен ъгъл за spin

```
for (var i=0; i<300; i++)
{
   a = random(0,2*Math.PI);
   b = random(0,2*Math.PI);
   c = cube([10*Math.cos(a)*Math.cos(b),
             10*Math.sin(a)*Math.cos(b),
             10*Math.sin(b)],4);
   c.color = [random(0.5,1),random(0.5,1),random(0.5,1)];
   c.spin  = random(0,2*Math.PI);
   c.focus = [random(-1,1),random(-1,1),random(-1,1)];
}
```



ПРОБА

Обобщение

Графични обекти

Квадрат

- Конструира се с new Suica.Square или square
- Има център center и дължина на страната size
- Поддържа свойства mode, origin, spin, focus и light

Правоъгълник

- Конструира се с new Suica.Rectangle или rectangle
- Същите свойства като квадрата, но със sizes вместо size

Куб

- Конструира се с new Suica.Cube или cube
- Има център center и дължина на страната size
- Същите свойства като квадрата

Правоъгълен паралелепипед

- Конструира се с new Suica.Cuboid или cuboid
- Същите свойства като куба, но със sizes вместо size

Общи свойства

- mode – режим на рисуване на обект (POINT, LINE, SOLID)
- size – дължина на ръб на квадрат и куб
- sizes – дължини на ръбове на правоъгълник и правоъгълен паралелепипед
- origin – отместване на центъра на обект
- spin – завъртане около локалната Z ос
- focus – посока на локалната Z ос
- light – осветяване на обект

ИКТ в НОС

Край

Коментари, въпроси