ICT in SES

Squares and cubes

Lesson №10

Square and rectangle

Square in Suica

Square

- Graphical object with properties
- Used to draw a square

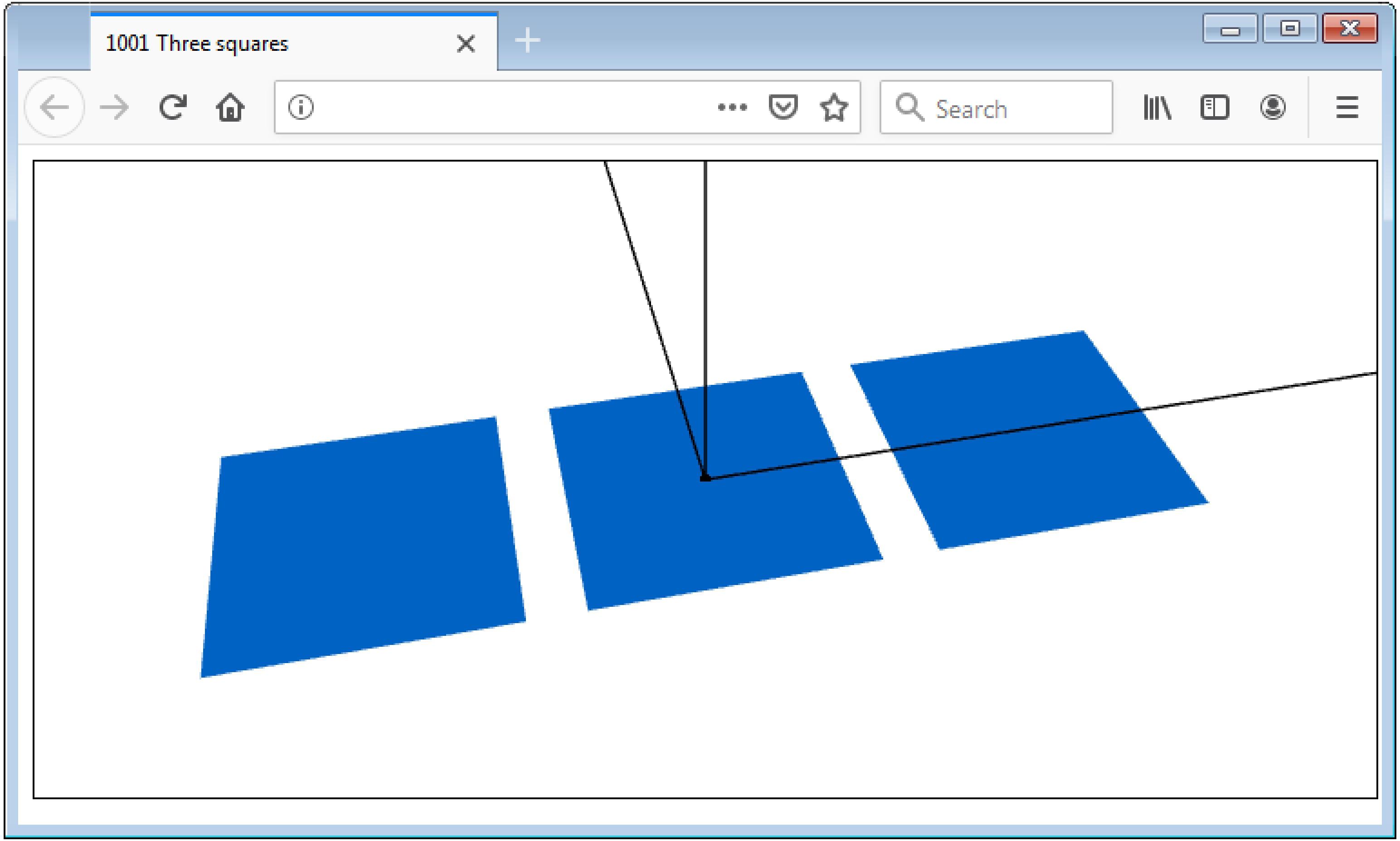
Creating a square

- With class new Suica.Square (*center, size*)
- With function square (*center, size*)
- Center is coordinates of a point, an array of three numbers
- Size is a number, the length of the cube edge

Example

- Create three squares in a row
- They have equal sizes
- There is some distance between them

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



TRY IT

Properties

Drawing mode

- Optional property
- Stored in mode
- Defines how the square is draw

Suica.POINT	drawn are the vertices
Suica.LINE	drawn are the edges
Suica.SOLID	drawn is a full square (default mode)

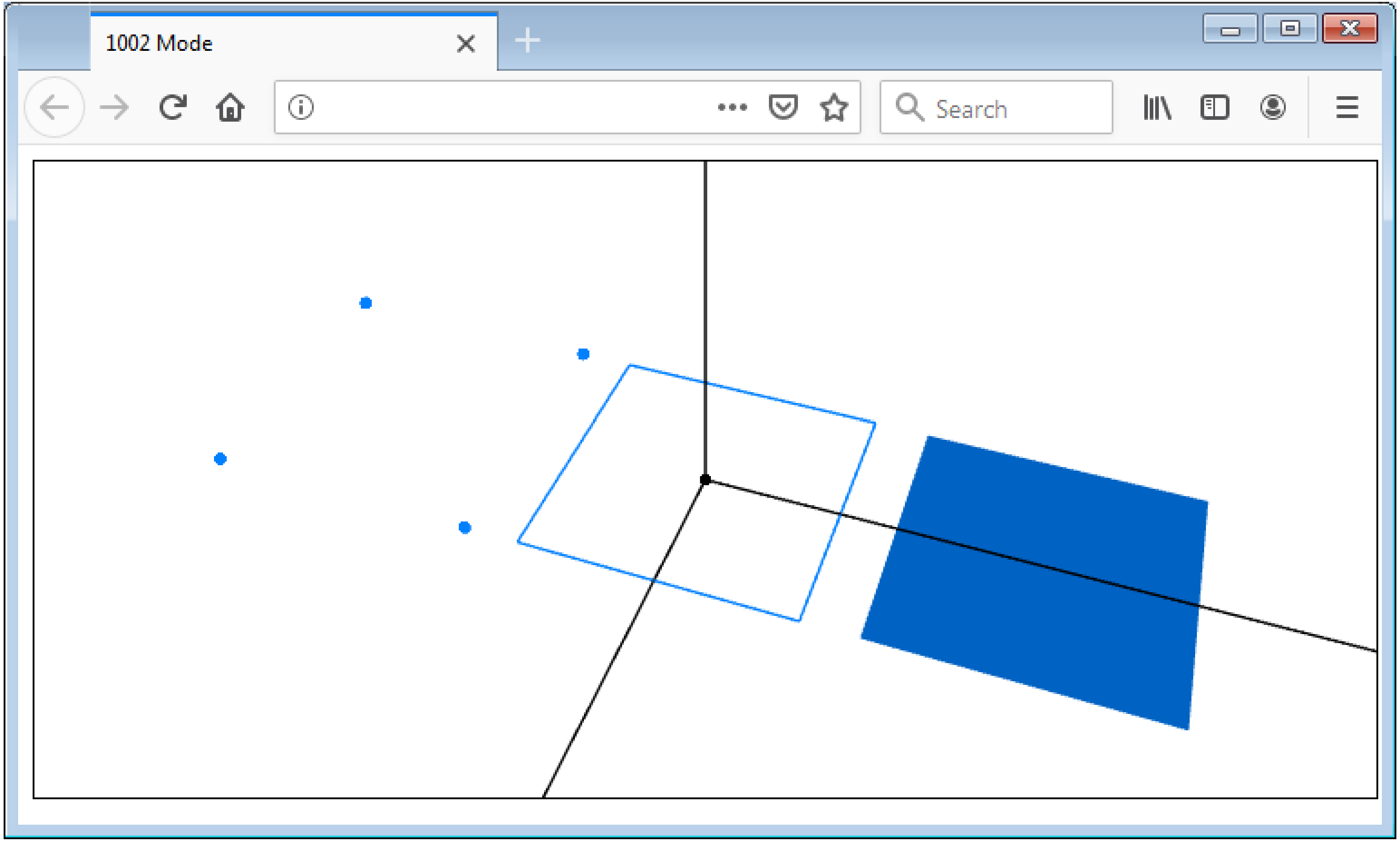
Example

- Three squares using three drawing modes

```
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;
```



TRY IT

Origin

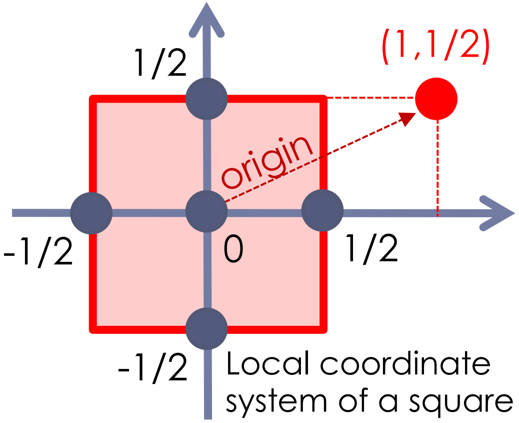
- Optional property
- Store in origin
- Defines the “center” of an object

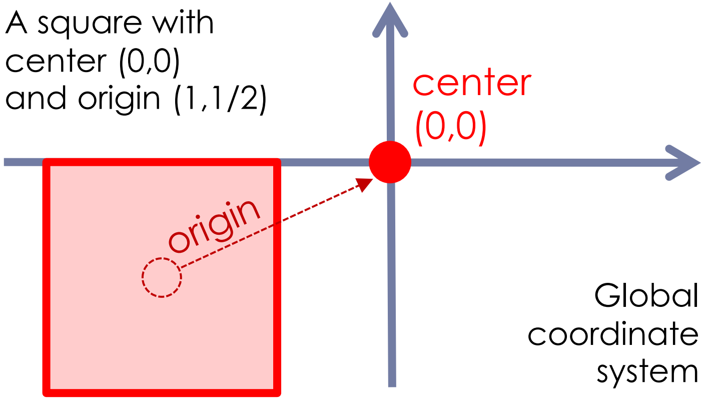
Purpose of origin

- In some cases it is easier to define location of a square via its vertex or edge middle point
- The property origin provides this functionality
- By default origin is (0,0,0) – thus the center of a square is the same as its geometrical middle point

Coordinates in origin

- Measured in the local coordinate system of the object
  - Origin = the geometrical center of the square
  - Unit = the size of the square
- When a square is drawn, its origin is places at the point, specified in the parameters for the square center





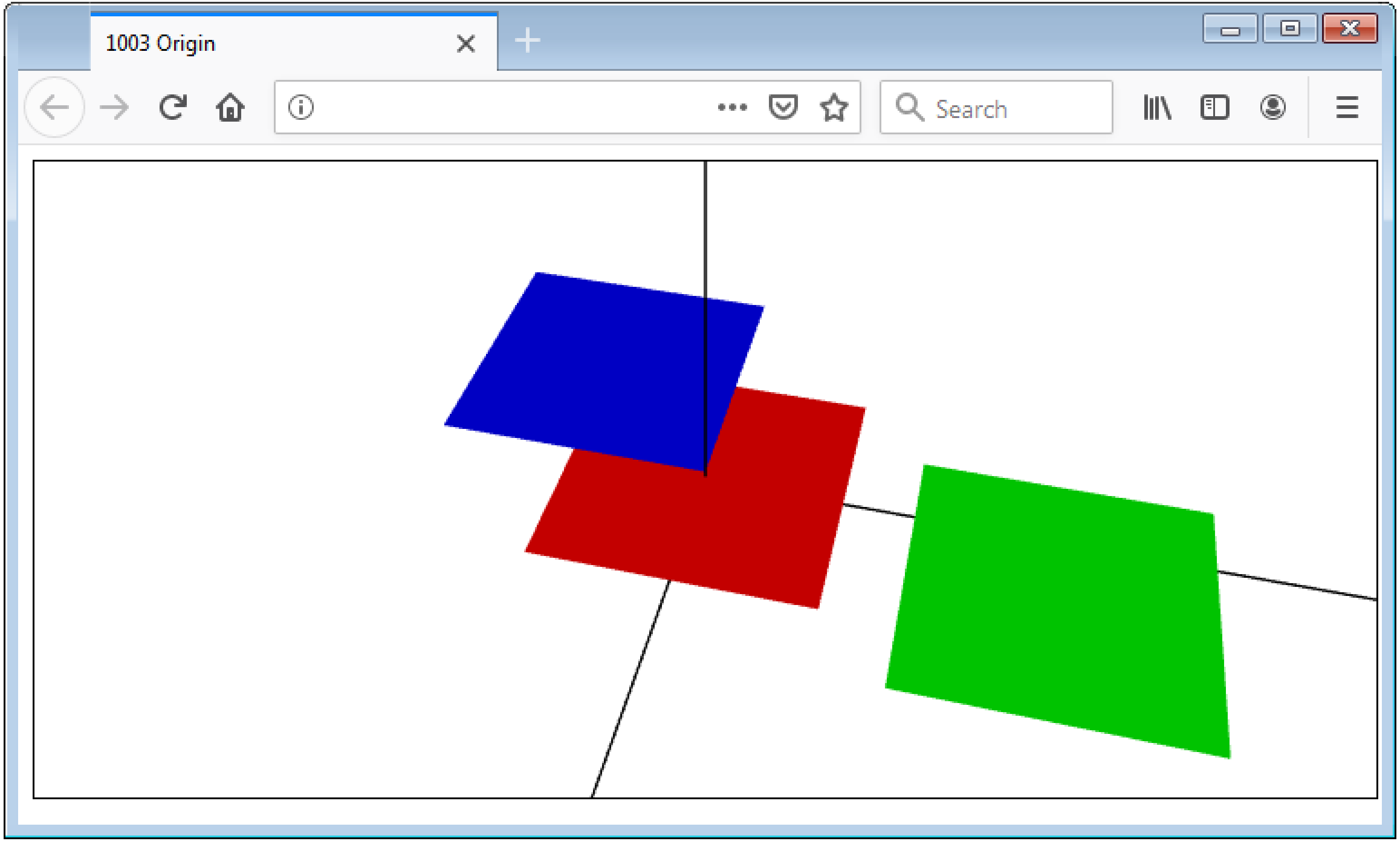
Example

- Squares with origins (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];
```



TRY IT

Rectangle in Suica

Rectangle

- Graphical object with properties similar to the square
- Used to draw squares and rectangles

Creating a rectangle

- With class new Suica.Rectangle (*center, sizes*)
- With function rectangle (*center, sizes*)
- Center is coordinates of a point, an array of three numbers
- Sizes is an array of 2 numbers – lengths of rectangle’s sides

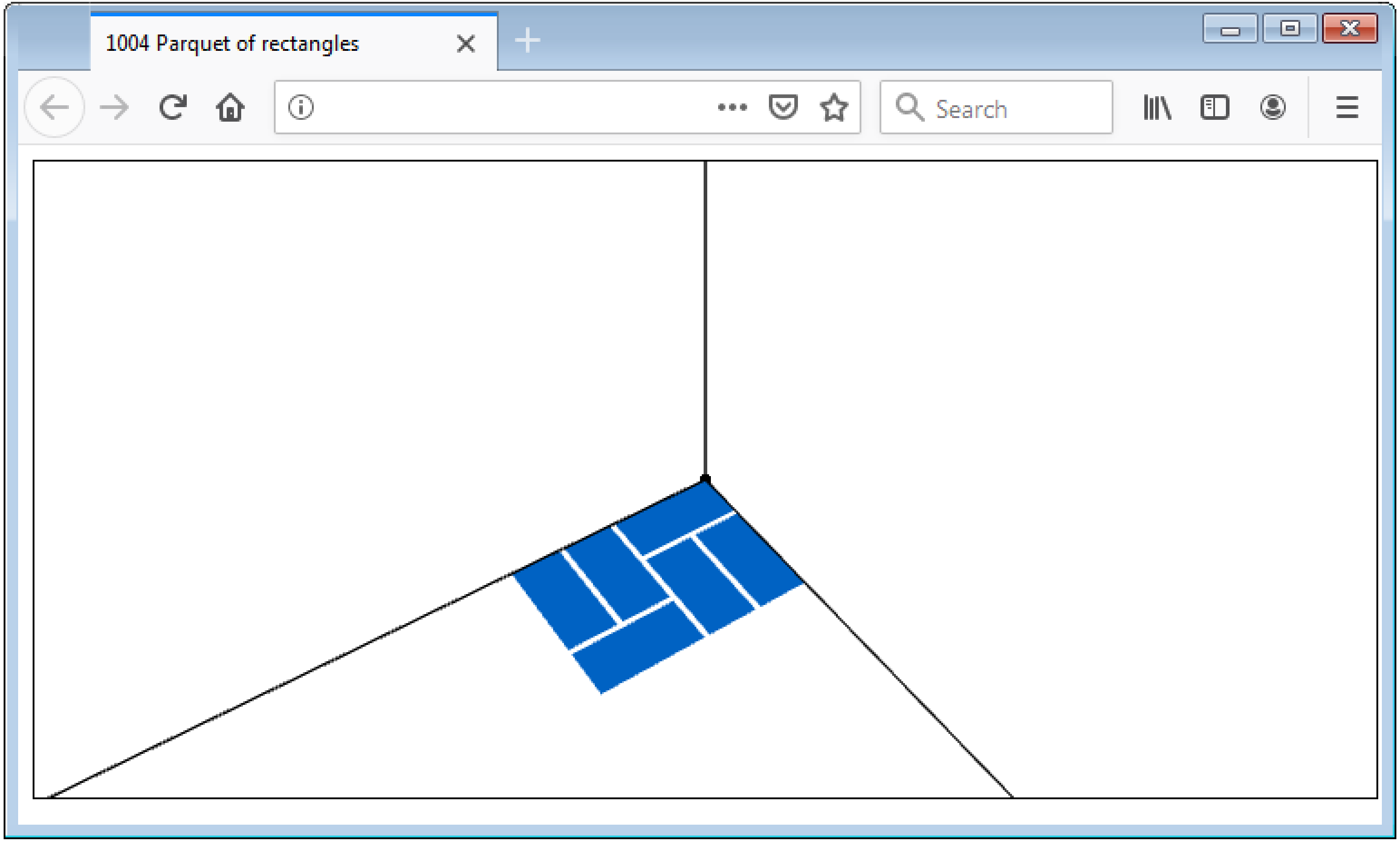
Example

- A parquet of rectangles 2x1
- For easier positioning the centers are at the vertices
- The actual sizes are 1.9x0.9 to leave a tiny gap

```
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];
:
```



TRY IT

Cube and cuboid

Cube in Suica

Cube

- Graphical objects with properties
- Used to draw a cube

Creating a cube

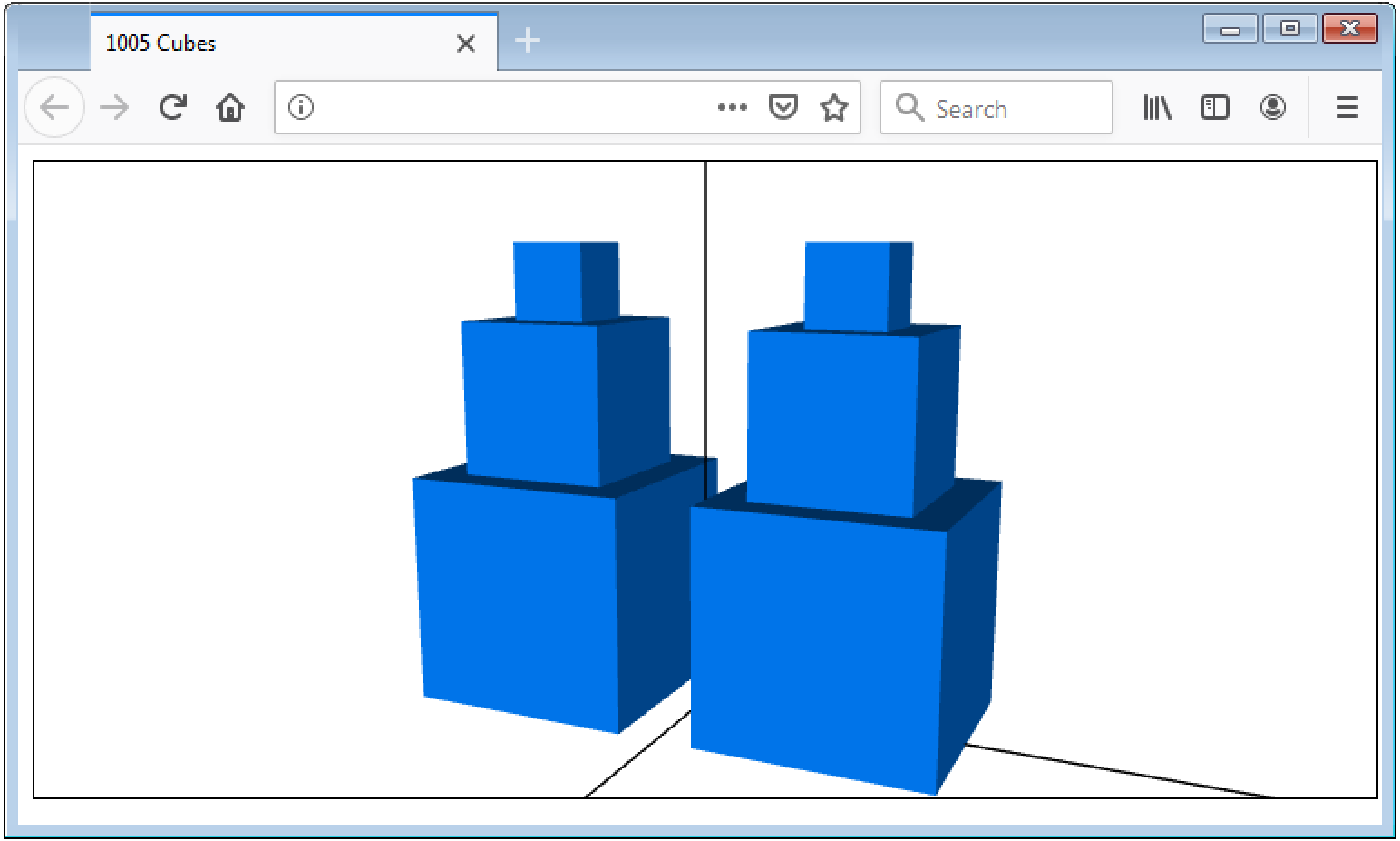
- With class new Suica.Cube (*center, size*)
- With function cube (*center, size*)
- Center is coordinates of a point, an array of three numbers
- Size is a number for the length of the cube edge

Example

- Two pyramids of cubes with sizes 3, 2 and 1
- One of the pyramids is with cubes with original origins
- The other pyramid is with cubes with modified origins

```
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];
```



TRY IT

Cuboid in Suica

Cuboid

- Graphical object with properties
- Used to draw cubes and cuboids

Creating cuboid

- With class new Suica.Cuboid (*center, sizes*)
- With function cuboid (*center, sizes*)
- Center is coordinates of a point, an array of three numbers
- Sizes is an array of three numbers – the lengths of the sides

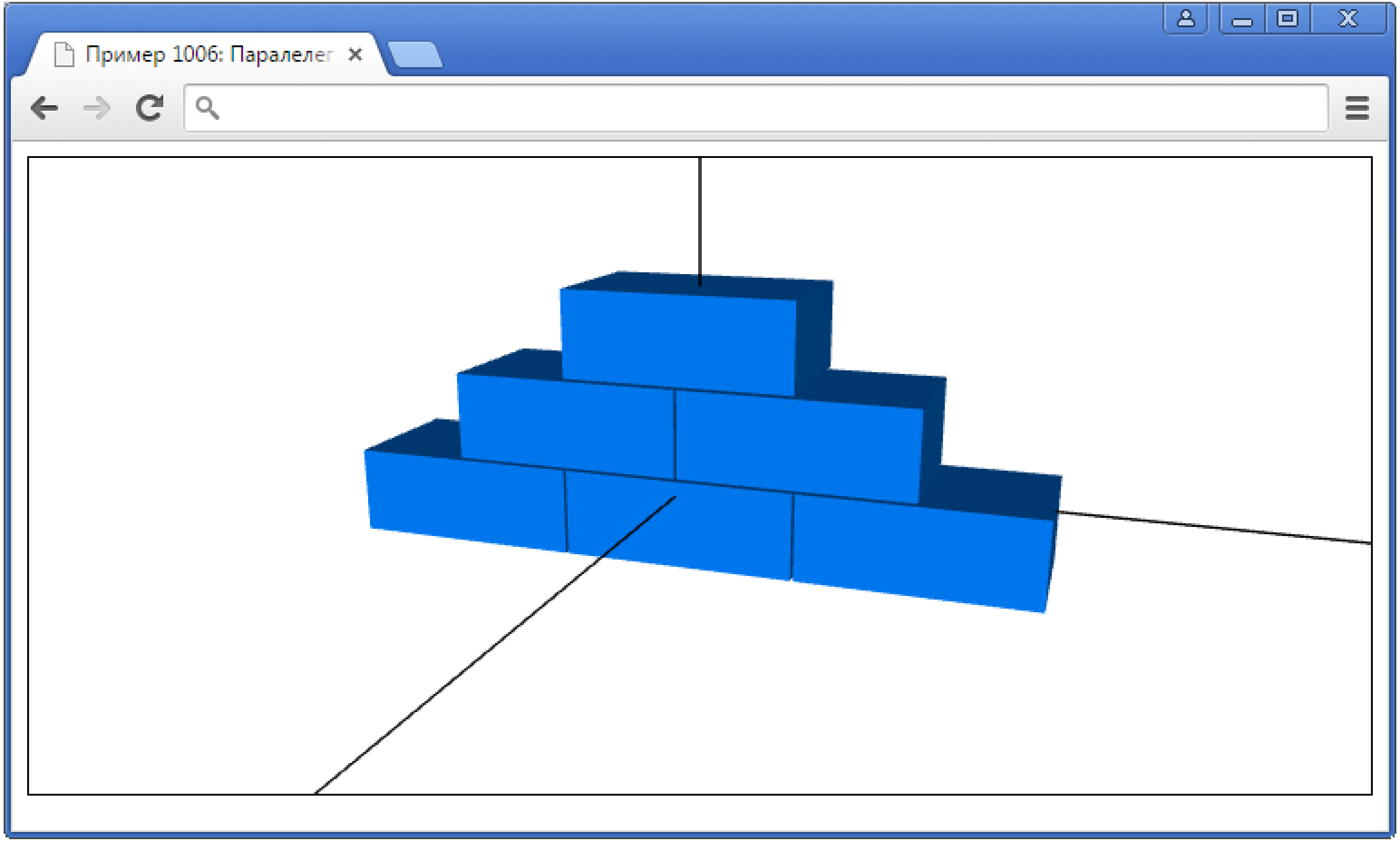
Example

- Small wall with 6 bricks of size 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]);
```



TRY IT

Orientation

Object orientation

Position

- Visual location of 2D/3D object is determined by properties center and size/sizes

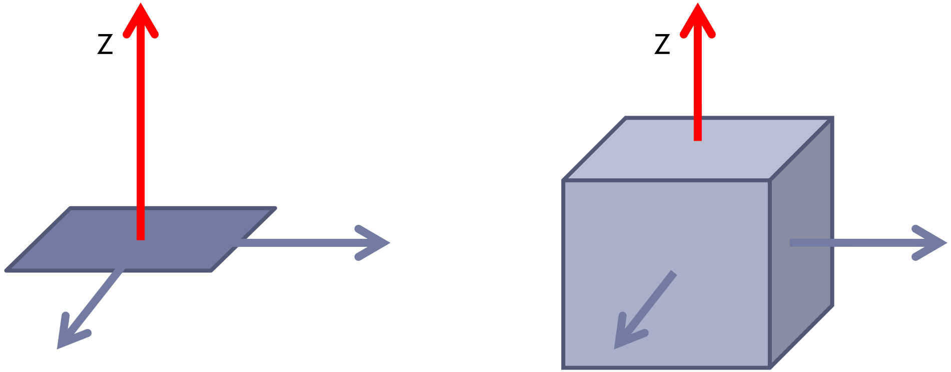
A problem

- They are not enough
- There is no way to rotate the object

Implementation of orientation

Invisible elements

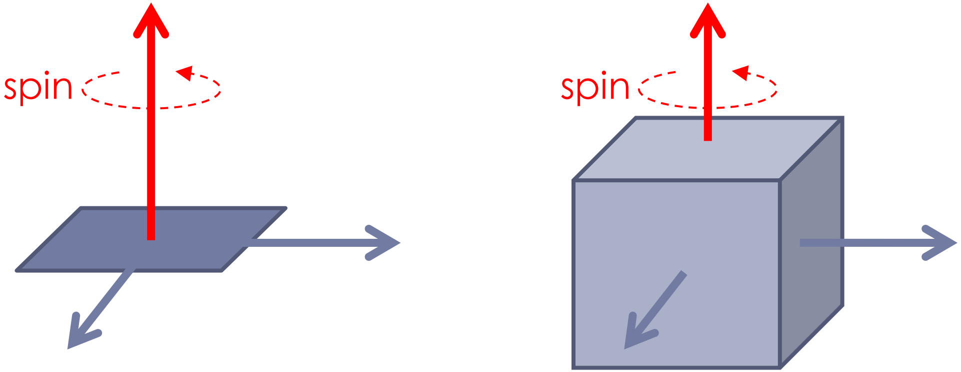
- Every object has a local coordinate system (it is used for the property origin)
- The local Z-axis is used to orient the object



Properties

Property spin

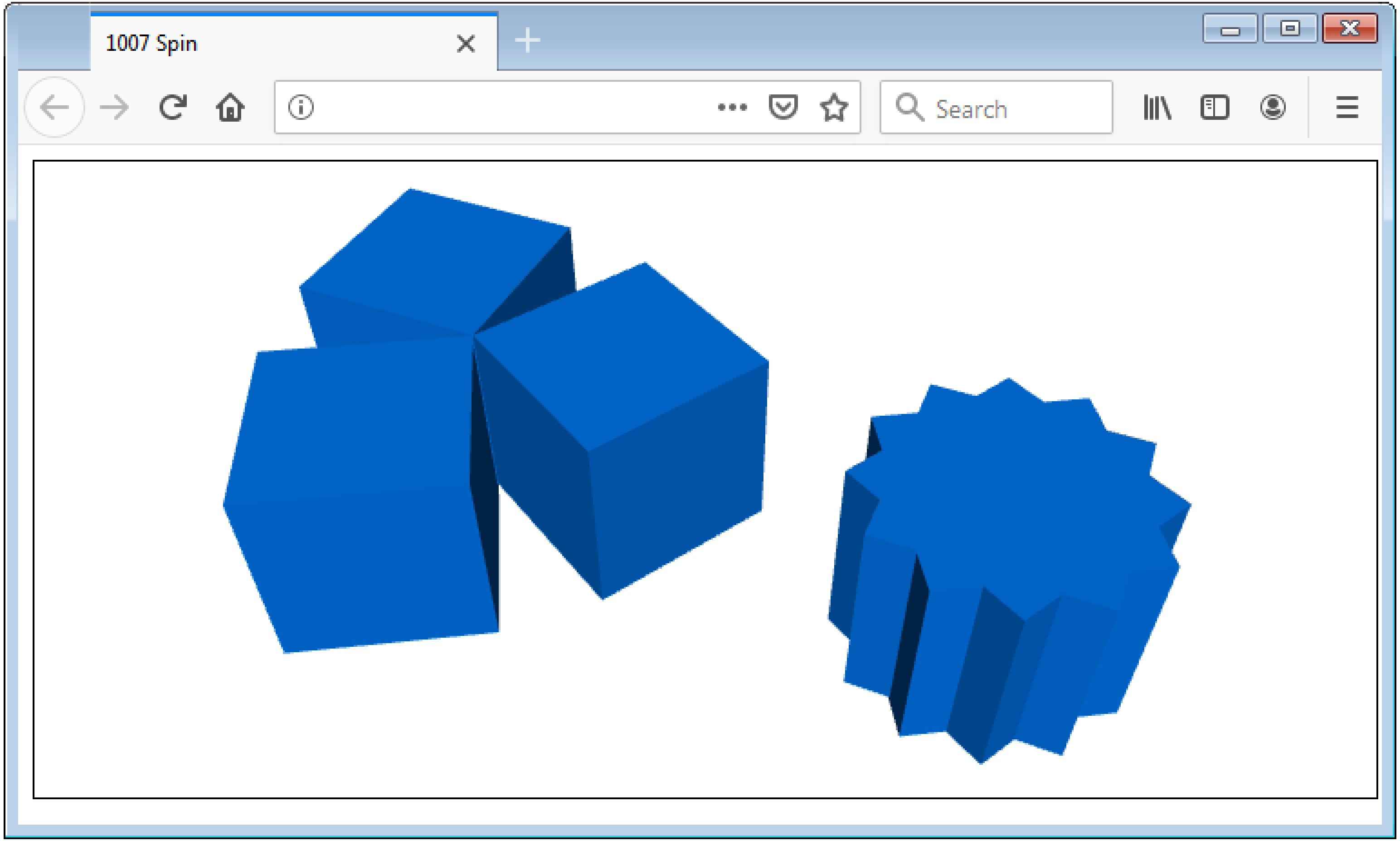
- Rotates the object around the local Z-axis
- The value of spin is in radians
- By default it is 0



Example

- Cubes, rotated around their central axis
- Cubes, rotated around their side edges (done by moving the center to the edge)

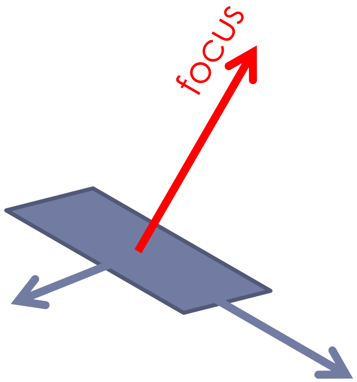
```
a = cube([0,7,0],5);
a.spin = Math.PI/6;
:
a = cube([0,-6,0],5);
a.origin = [0.5,0.5,0];
a.spin = 2*Math.PI/3;
```

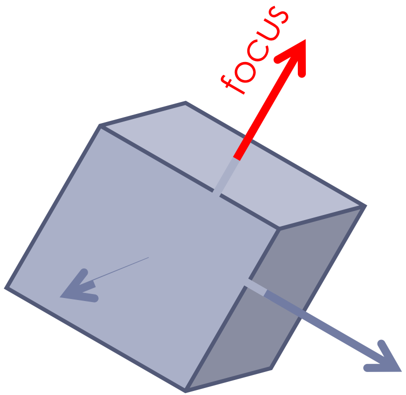


TRY IT

Property focus

- Changes the direction of the local Z-axis
- Also changes the other local axes to maintain orthogonality
- The value of focus a vector
- By default it is (0,0,1) and coincides with the global Z-axis





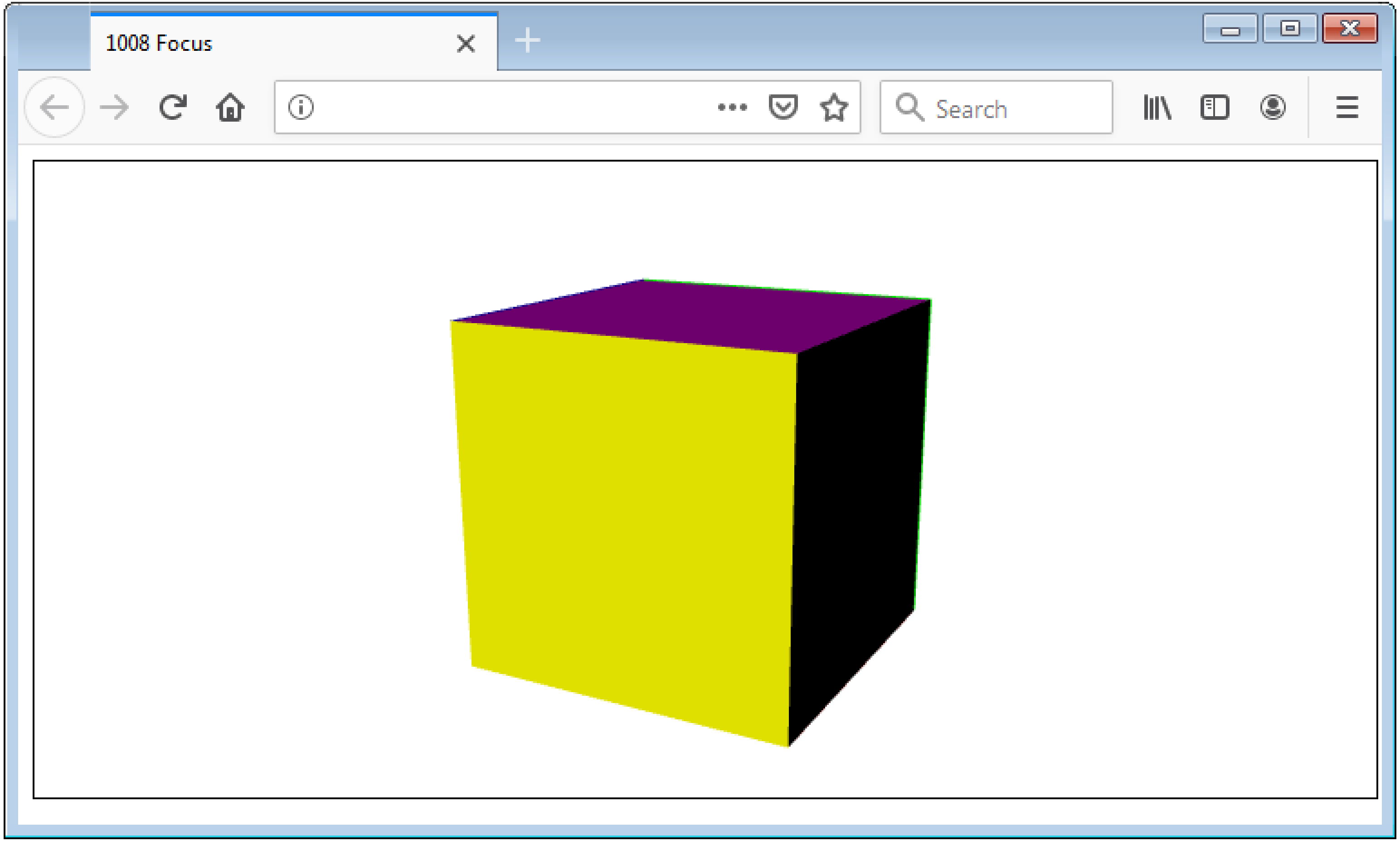
Example

- Cube wit multicolour sides – constructed from squares

```
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];
```



TRY IT

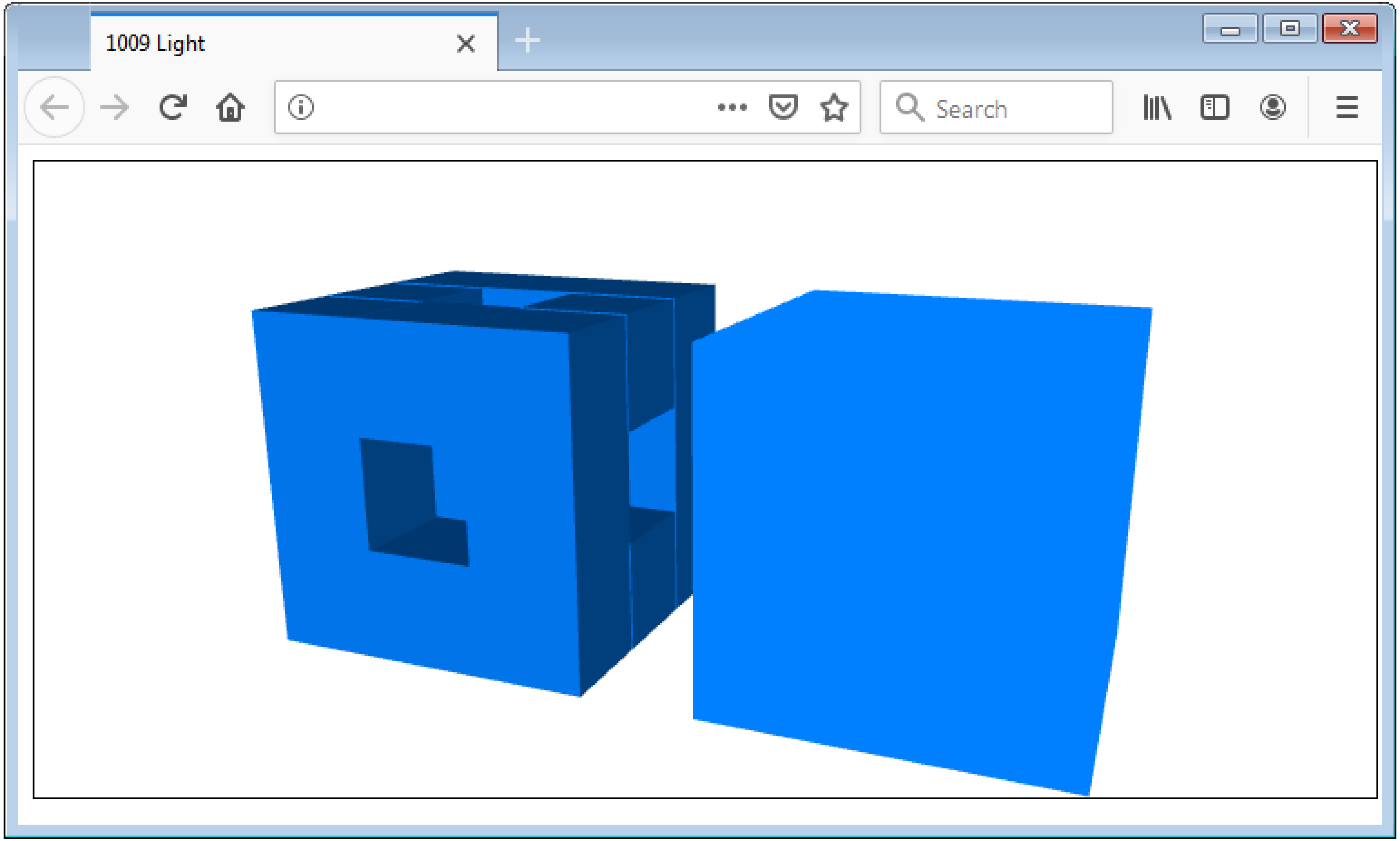
Interaction of properties

- When orientation is changed by focus, then spin rotates around the new axis
- The vector focus starts from the center of the object, which can be offset by origin
- Not every orientation can be achieved by using only focus, sometimes spin is also needed

Property light

- Determines whether light shading is used or not used
- When light shading is on, the side surfaces become darker
- When light shading is off, all surfaces use their original colours

```
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;
       }
```



TRY IT

Examples

Example №1

A pyramid or squares

- Concentric squares one over another
- Their sizes get smaller until they reach 1x1

Idea

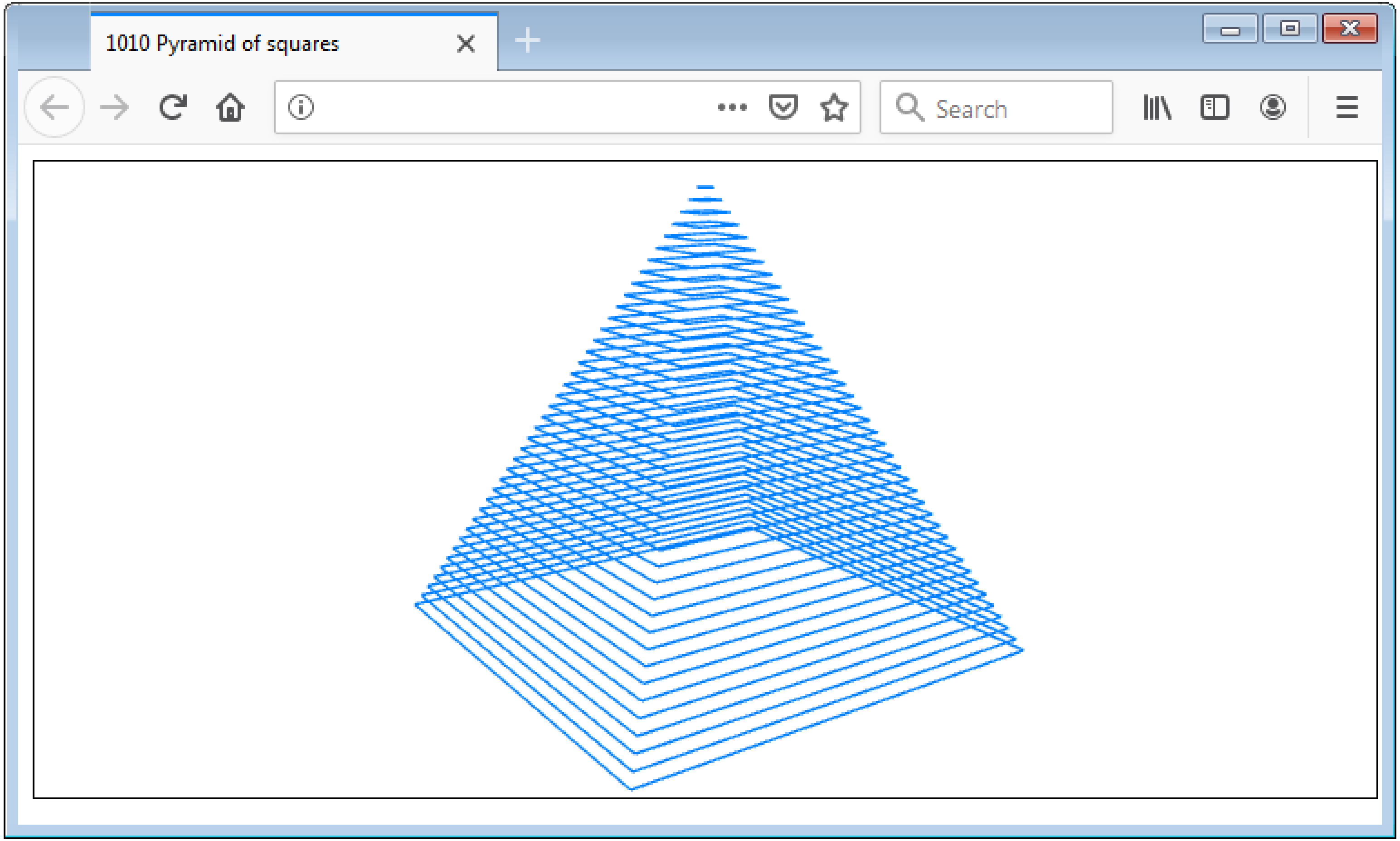
- Initial square – the one at the top of the pyramid
- Every next square is increased by 1
- The X and Y coordinates of the center is the same, only Z is decreased by 1

Solution

- Generating squares top to bottom
- The vertical position of a square is decreasing in z
- The size in increasing in i

```
n = 40;
z = 25;

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



TRY IT

Example №2

Glued rectangles

- Two perpendicular lines
- Random rectangles glued to them

Idea

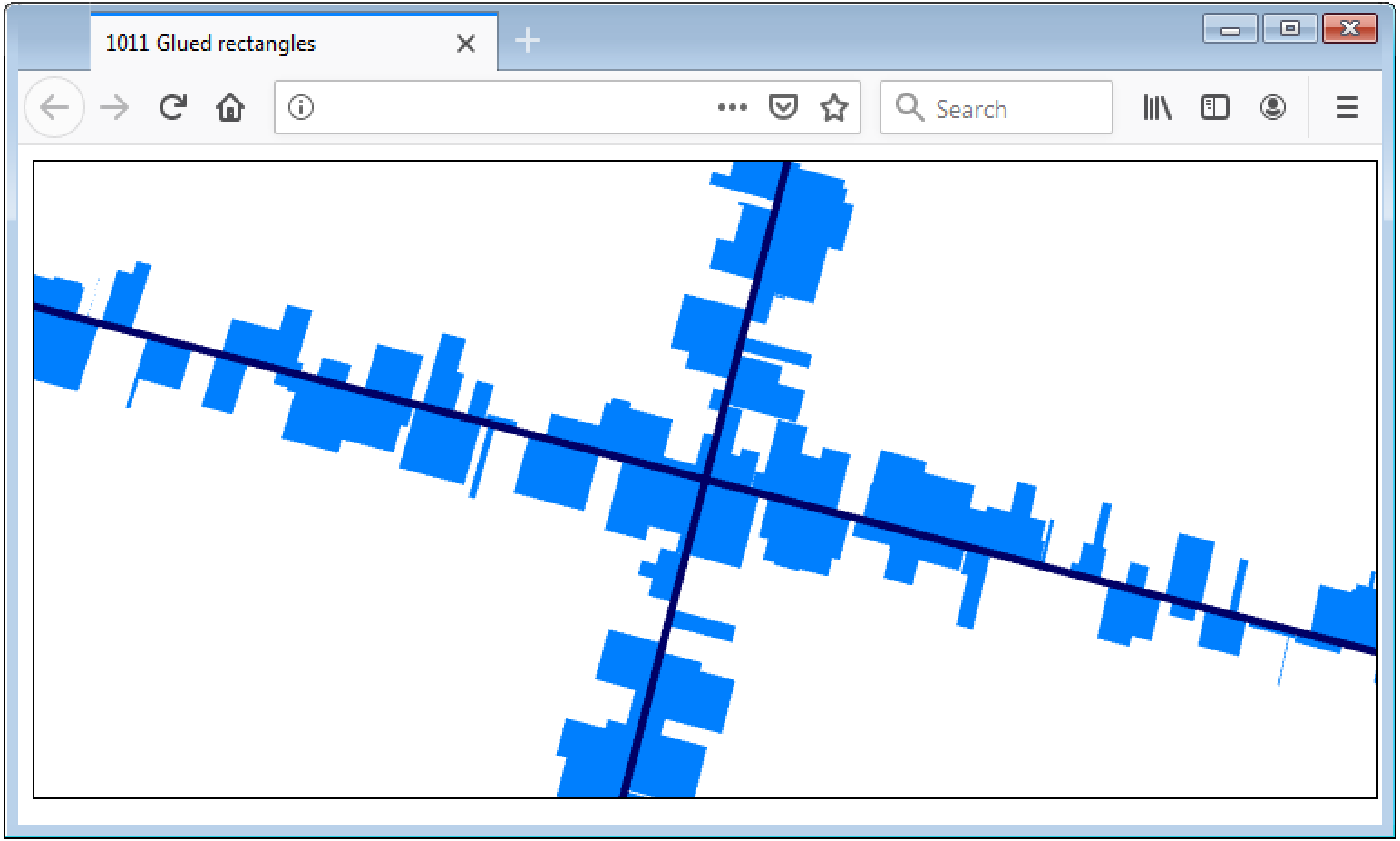
- For convenience the lines are along X and Y axes
- The centers of rectangles are on the lines if origin is properly set

Solution

- Rectangles have centers (x,0,0) and (0,y,0)
- For offset use the sign if a random number from -1 to +1
- Rectangles with centers (x,0,0) have origins (0,±1/2,0)
- Rectangles with centers (0,y,0) have origins (±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];
}
```



TRY IT

Example №3

Spiral staircase

- Steps are positioned in a spiral

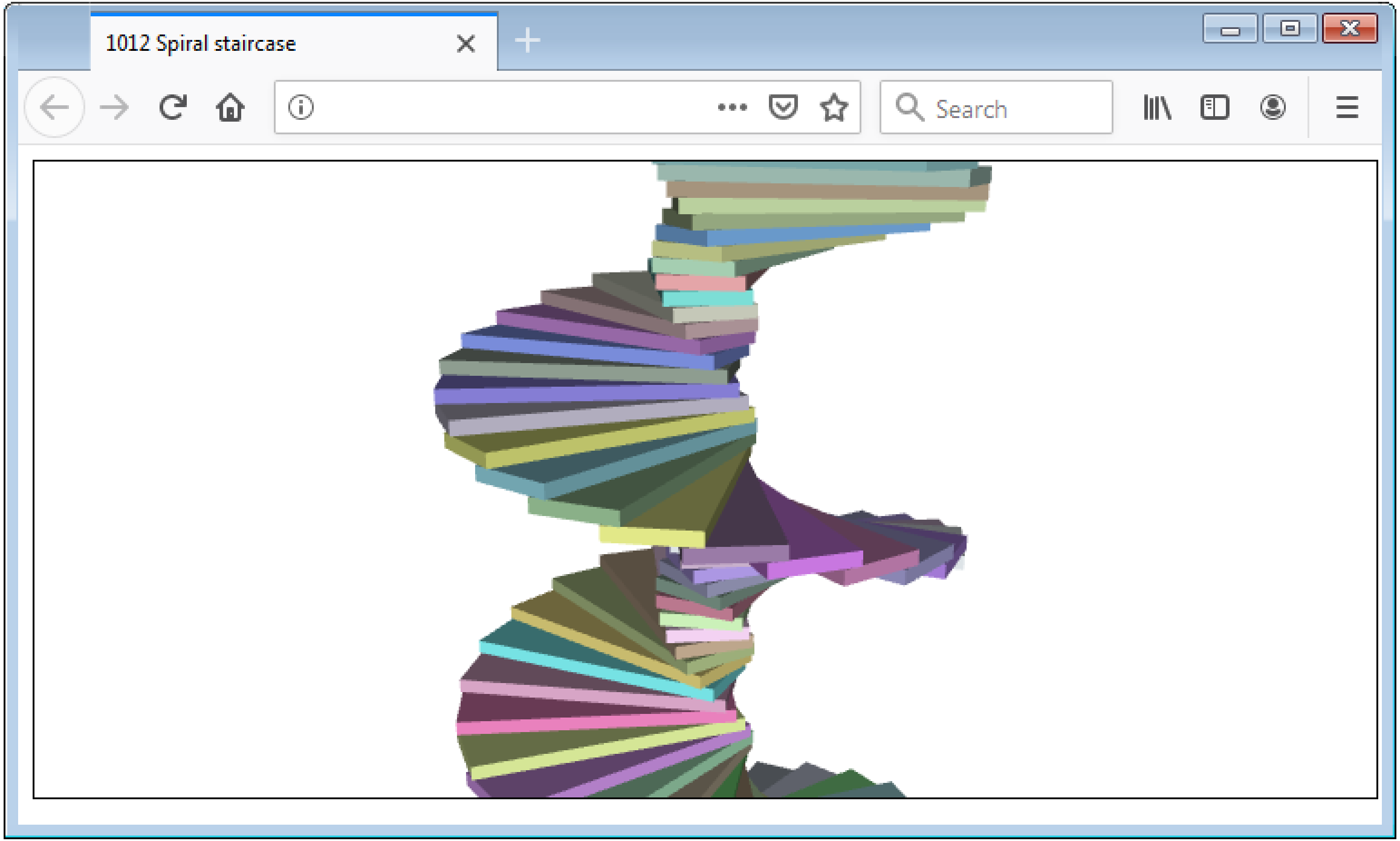
Idea

- Each step is a flat cuboid
- Their centers are on a vertical line
- Their centers are shifted to make this possible
- Each step is rotated depending on its number

Solution

- Steps are positioned vertically from -40 do 24
- Their centers are shifted almost to the end of the steps (to 0.4 instead of 0.5)
- Each step is rotated 15 degrees relative to the next step

```
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);
}
```



TRY IT

Example №4

Sphere of cubes

- Randomly oriented cubes on the surface of a sphere

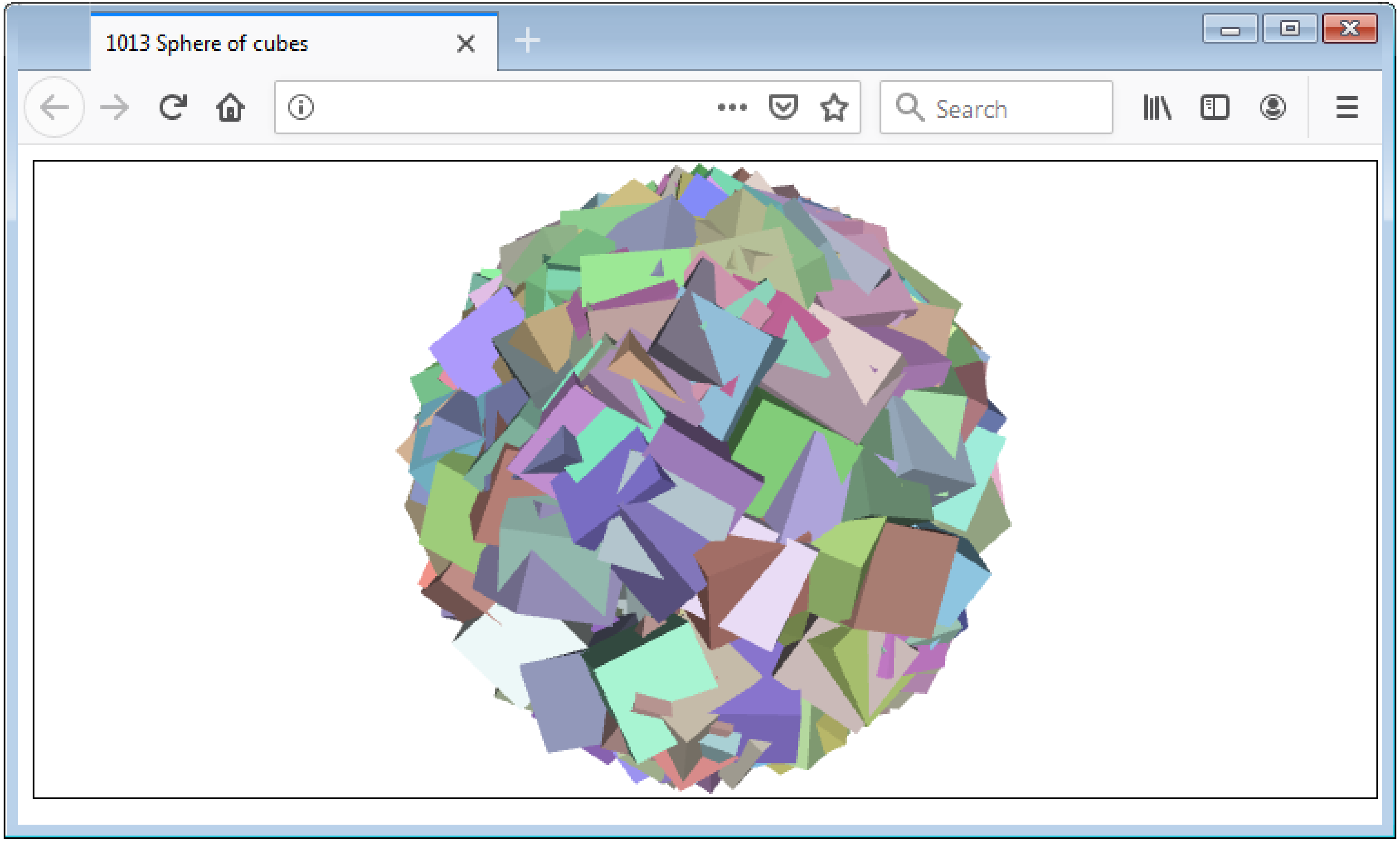
Idea

- Get two random angles and a fixed radius
- They define a point on the sphere – the position of a cube
- Cube’s orientation is random

Solution

- Transforming spherical coordinates to Cartesian
- Random vector for focus, random angle for 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)];
}
```



TRY IT

Summary

Graphical objects

Square

- Created with new Suica.Square or square
- Has center and size
- Supports mode, origin, spin, focus and light

Rectangle

- Created with new Suica.Rectangle or rectangle
- Same properties as the square, except for sizes instead of size

Cube

- Created with new Suica.Cube or cube
- Has center and size
- Same properties as the square

Cuboid

- Created with new Suica.Cuboid or cuboid
- Same properties as the cube, except for sizes instead of size

Common properties

- mode – drawing mode (POINT, LINE, SOLID)
- size – length of the edge of a square or a cube
- sizes – lengths of the edges of a rectangle or a cuboid
- origin – offset of the object’s center
- spin – rotation around the local Z-axis
- focus – direction of the local Z-axis
- light – using or ignoring light shading

ICT in SES

End

Comments, questions