



ICT in SES

Circular motion

Lesson N°14

Motion on a circle

Circular trajectories



Role in computer graphics

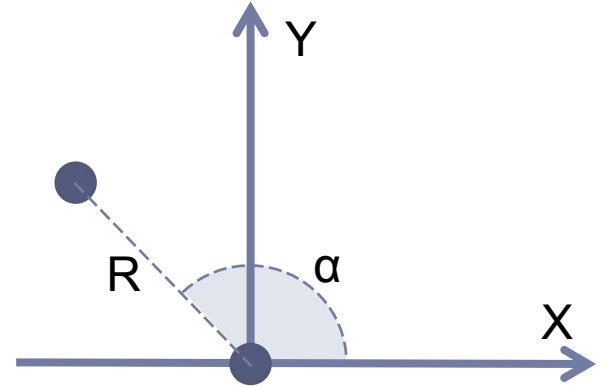
- Rotational motion (hands of a clock)
- Orbiting an object (satellite around a planet)
- Scene rotation (as in the demo command)

Implementation

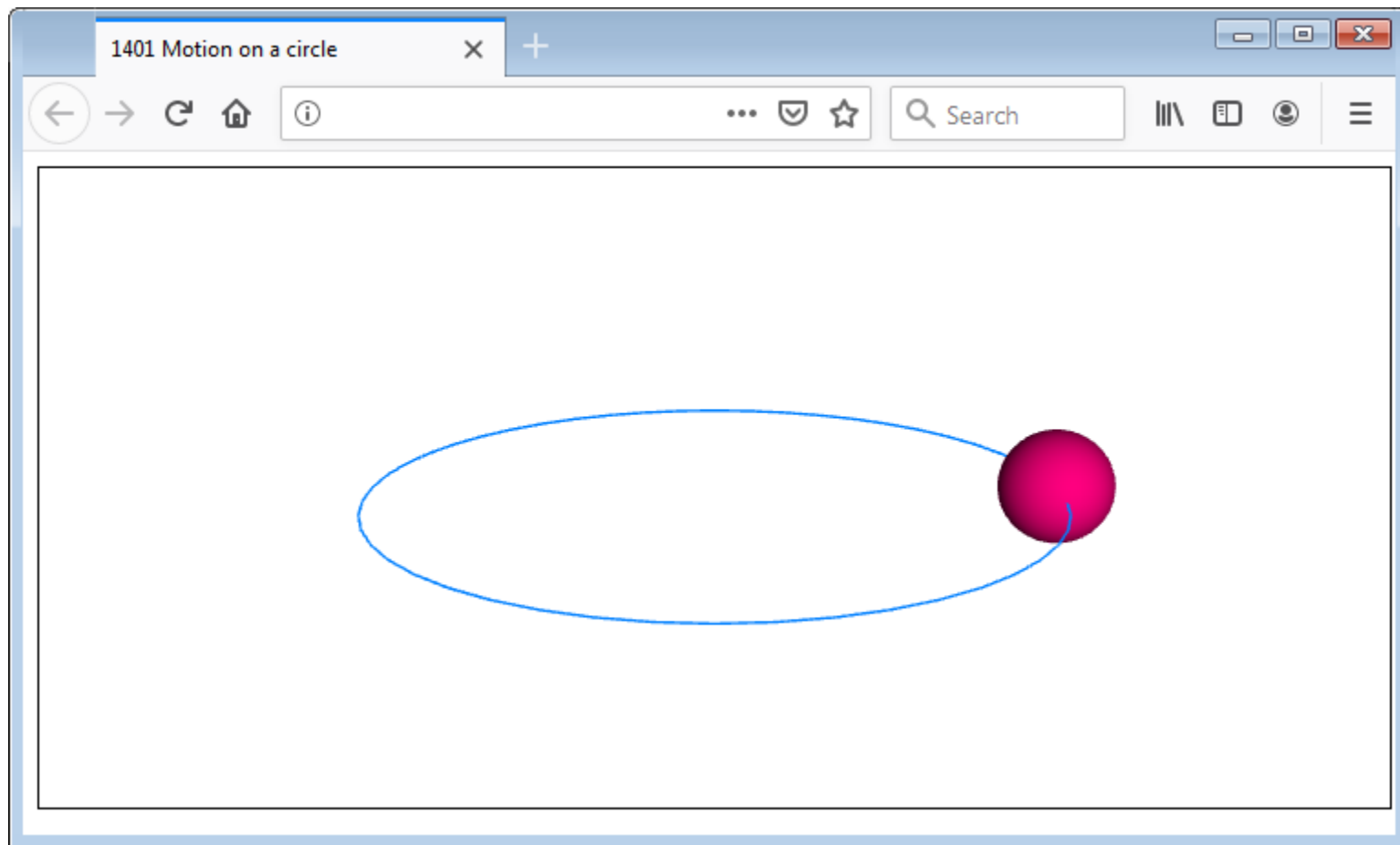
- Polar coordinates transformed to Cartesian

First example

- Sphere moves on a circle in plane XY
- Center at $[R.\cos(\alpha), R.\sin(\alpha), 0]$
- Angle is the time
- Radius is fixed



```
function loop()
{
    t = Suica.time;
    a.center = [30*Math.cos(t), 30*Math.sin(t), 0];
}
```



TRY IT

Direction



Motion direction

- Because of motion on a line there are only two directions

Direction depends on

- Change in the angle: $+\Delta\alpha$ or $-\Delta\alpha$
- Coordinate axes: XY or YX
- Transformation: $\sin(x)$ or $\cos(x)$
- Sign of a radius: $R_x > 0$ or $R_x < 0$

Velocities



Angular velocity

- Change of angle for a frame
- Does not depend on the radius

Linear velocity

- Travelled distance for a frame
- Depends on the angular velocity
- Depends on the radius

Relation between velocities

- Angular velocity φ and radius R
- Linear velocity is $v=R\varphi$ if angles are in radians

Options

- Modifying each of the velocities, keeping the other intact
- Modification of angular velocity only:

$$v = \left(\frac{R}{k}\right) \cdot (k\varphi)$$

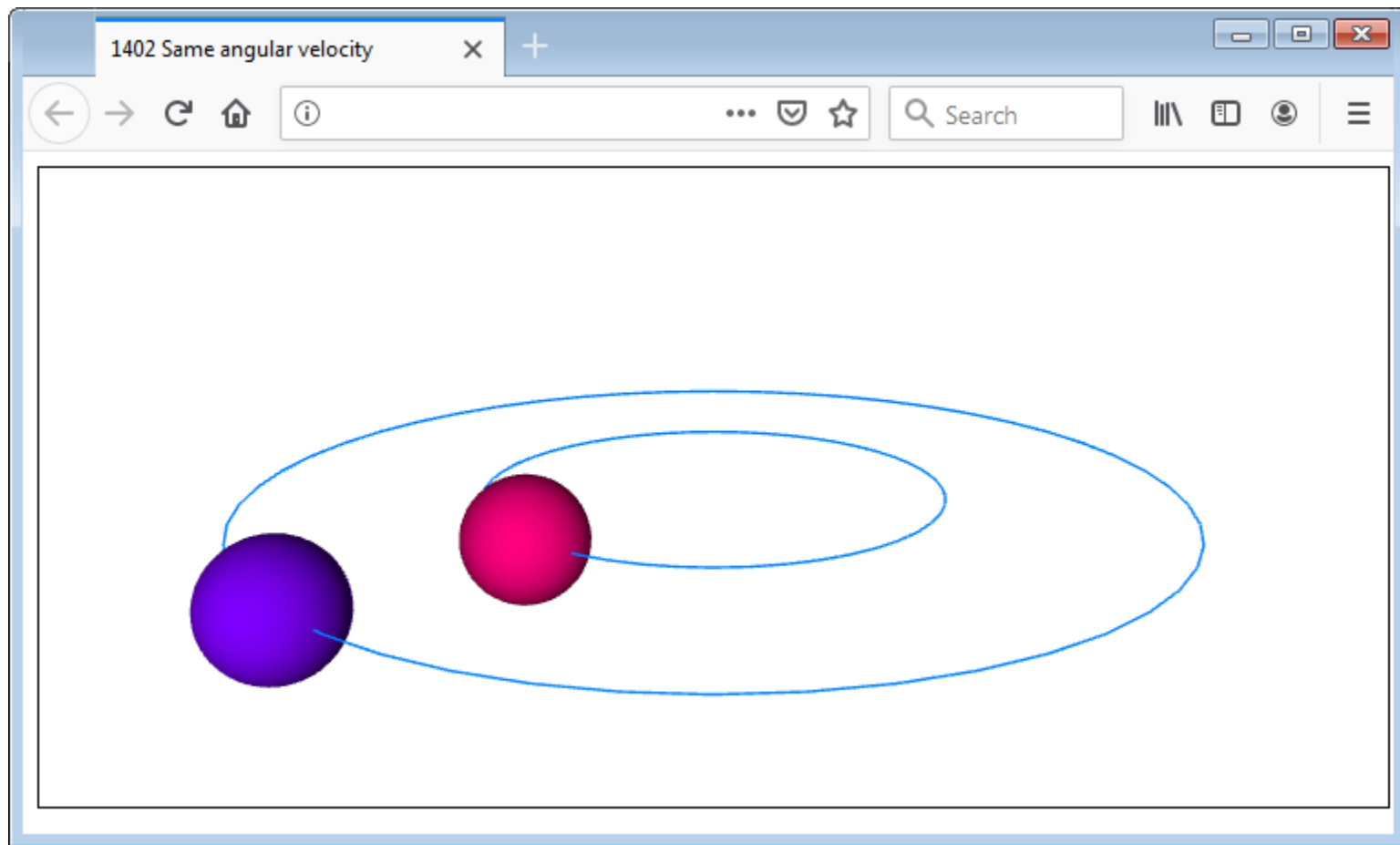
- Modification of linear velocity only:

$$(kv) = (kR) \cdot \varphi$$

Angular velocities

- Two spheres move on concentric circles
- Same angular velocities
- Observation: they make a full round for the same time

```
function loop()
{
    t = Suica.time;
    a.center = [20*Math.cos(t), 20*Math.sin(t), 0];
    b.center = [40*Math.cos(t), 40*Math.sin(t), 0];
}
```

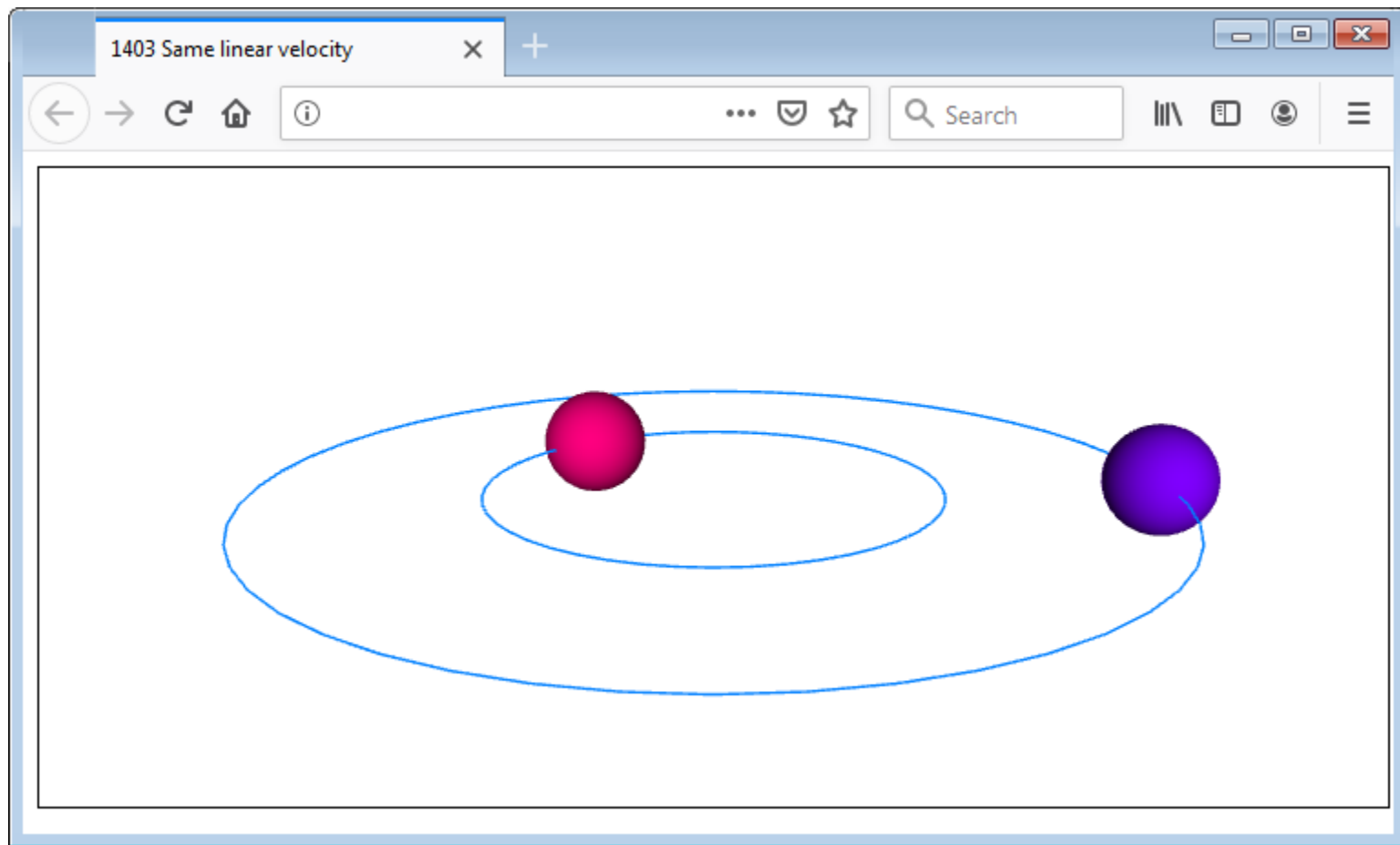


TRY IT

Linear velocities

- Two spheres move on concentric circles
- Same linear velocities
- Observation: the sphere on the small circle makes a full round faster than the other sphere

```
function loop()
{
  t = Suica.time;
  a.center = [20*Math.cos(2*t), 20*Math.sin(2*t), 0];
  b.center = [40*Math.cos(t), 40*Math.sin(t), 0];
}
```



TRY IT

Illusion with balls

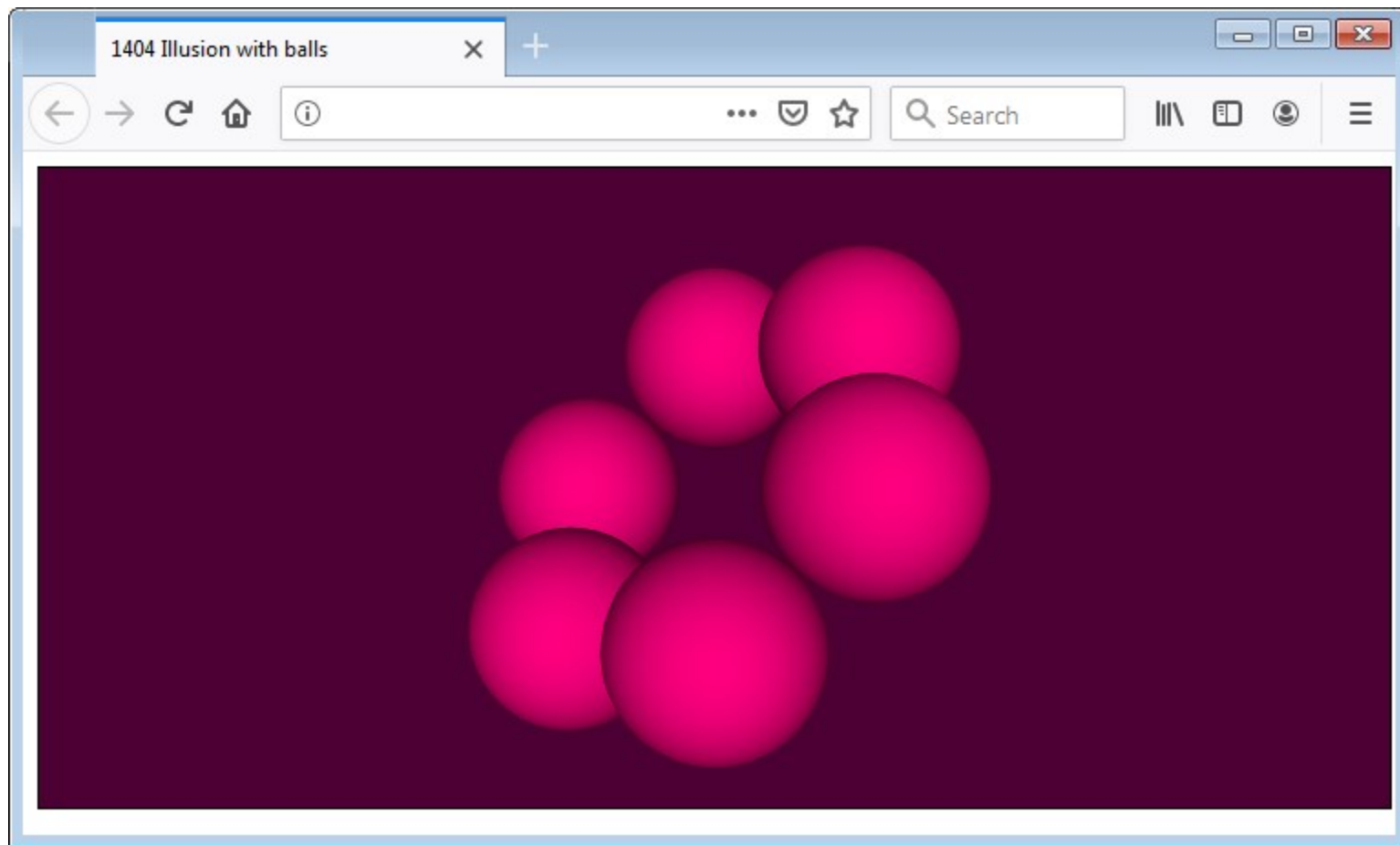


Six balls

- Circular motion without collision
- Each two are in a separate plane

```
x1 = sphere([0,0,0],8);  
x2 = sameAs(x1);  
y1 = sameAs(x1);  
y2 = sameAs(x1);  
z1 = sameAs(x1);  
z2 = sameAs(x1);
```

```
x1.center = [0,c,s];  
x2.center = [0,-c,-s];  
y1.center = [s,0,c];  
y2.center = [-s,0,-c];  
z1.center = [c,s,0];  
z2.center = [-c,-s,0];
```



TRY IT

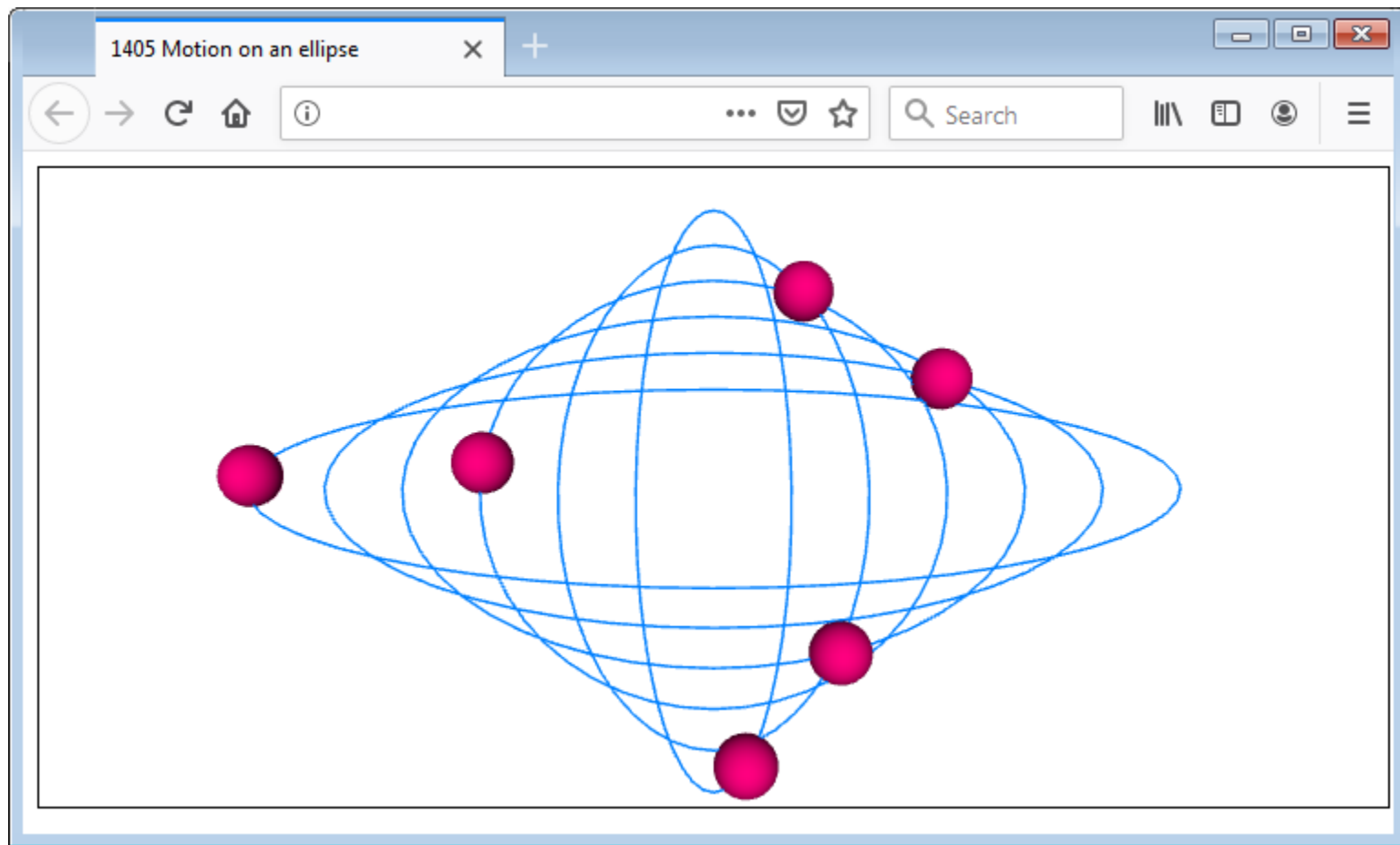
Ellipse



Motion on an ellipse

- Similar to motion on a circle
- Different radii

```
for (var i=0; i<6; i++)  
{  
    c = Math.cos(t+2*Math.PI*i/3);  
    s = Math.sin(t+2*Math.PI*i/3);  
    balls[i].center = [(10+10*i)*c, (38-5*i)*s, 0];  
}
```



TRY IT

Relative motion

Relative motion



Motion center is not $(0,0)$

- Composition of translation and circular motion
- Center can move too (e.g. in a nested motion)

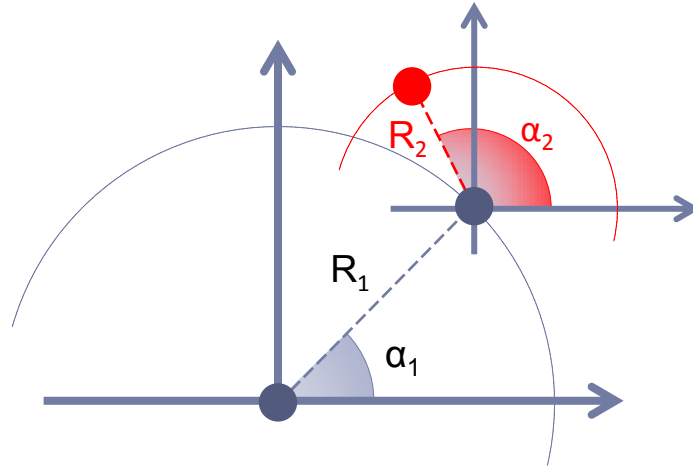
Examples

- Satellite around the Moon around the Earth around the Sun
- Swinging

Implementation of nested circular motion

- Object moves along a circle with radius R_1
- Another object move along a circle with radius R_2 around the first object:

$$\begin{cases} x = R_1 \cos(\alpha_1) + R_2 \cos(\alpha_2) \\ y = R_1 \sin(\alpha_1) + R_2 \sin(\alpha_2) \end{cases}$$



Star system



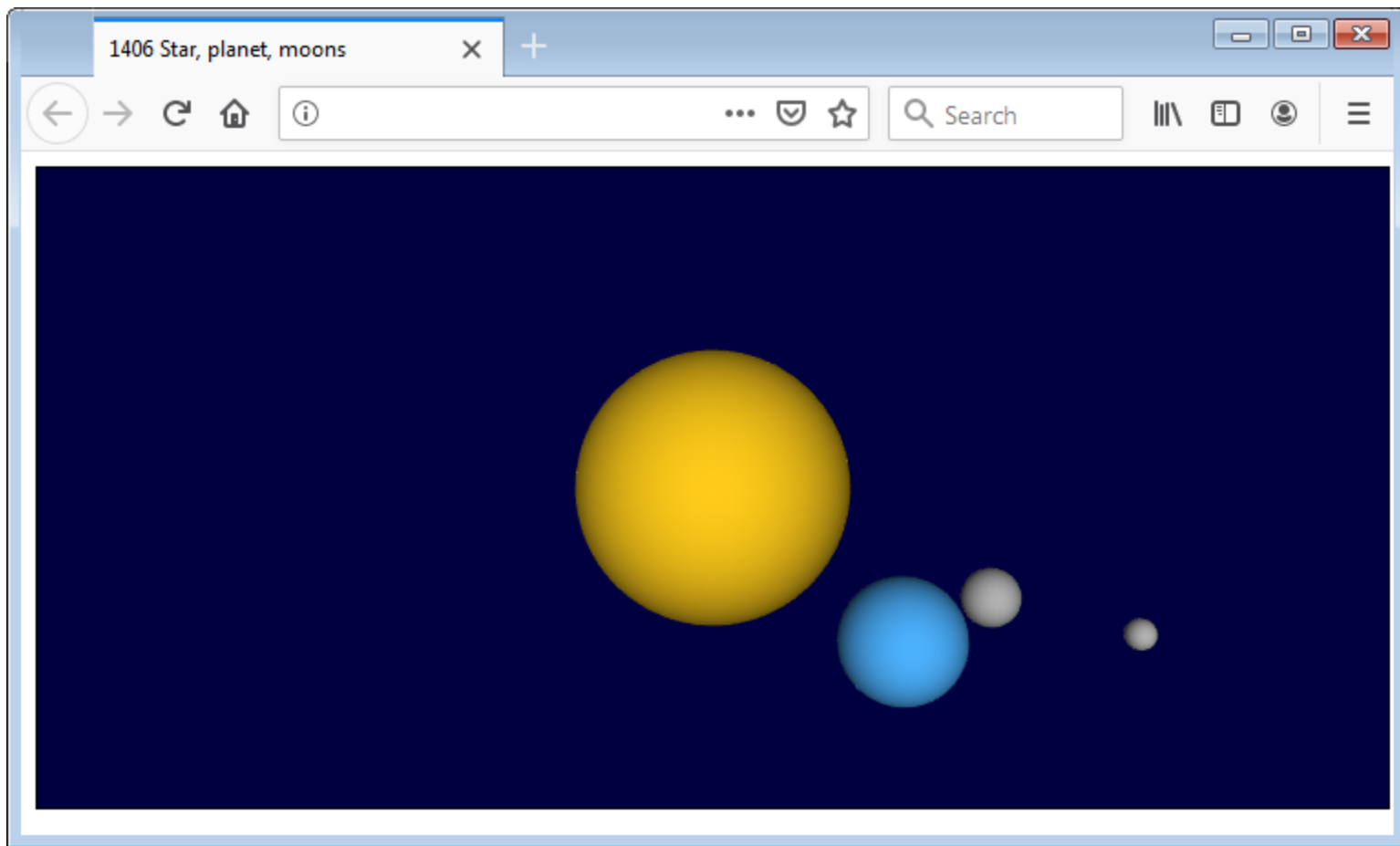
Four bodies

- A star, a planet and two moons
- Moons use the planet as a center of their circular motions

```
planet.center[0] = 30*Math.cos(t);  
planet.center[1] = 40*Math.sin(t);
```

```
moon1.center[0] = planet.center[0]+10*Math.cos(4*t);  
moon1.center[1] = planet.center[1]+10*Math.sin(4*t);
```

```
moon2.center[0] = planet.center[0]+15*Math.cos(3*t);  
moon2.center[1] = planet.center[1]+15*Math.sin(3*t);
```



TRY IT

Trajectory



Goal

- Showing a trajectory
- Nested circular motion
- Different ratios of angular velocities

Note

- Generating part of the trajectory – when the ratio is rational, the trajectory is a loop (no need to generate all of it)

Implementation

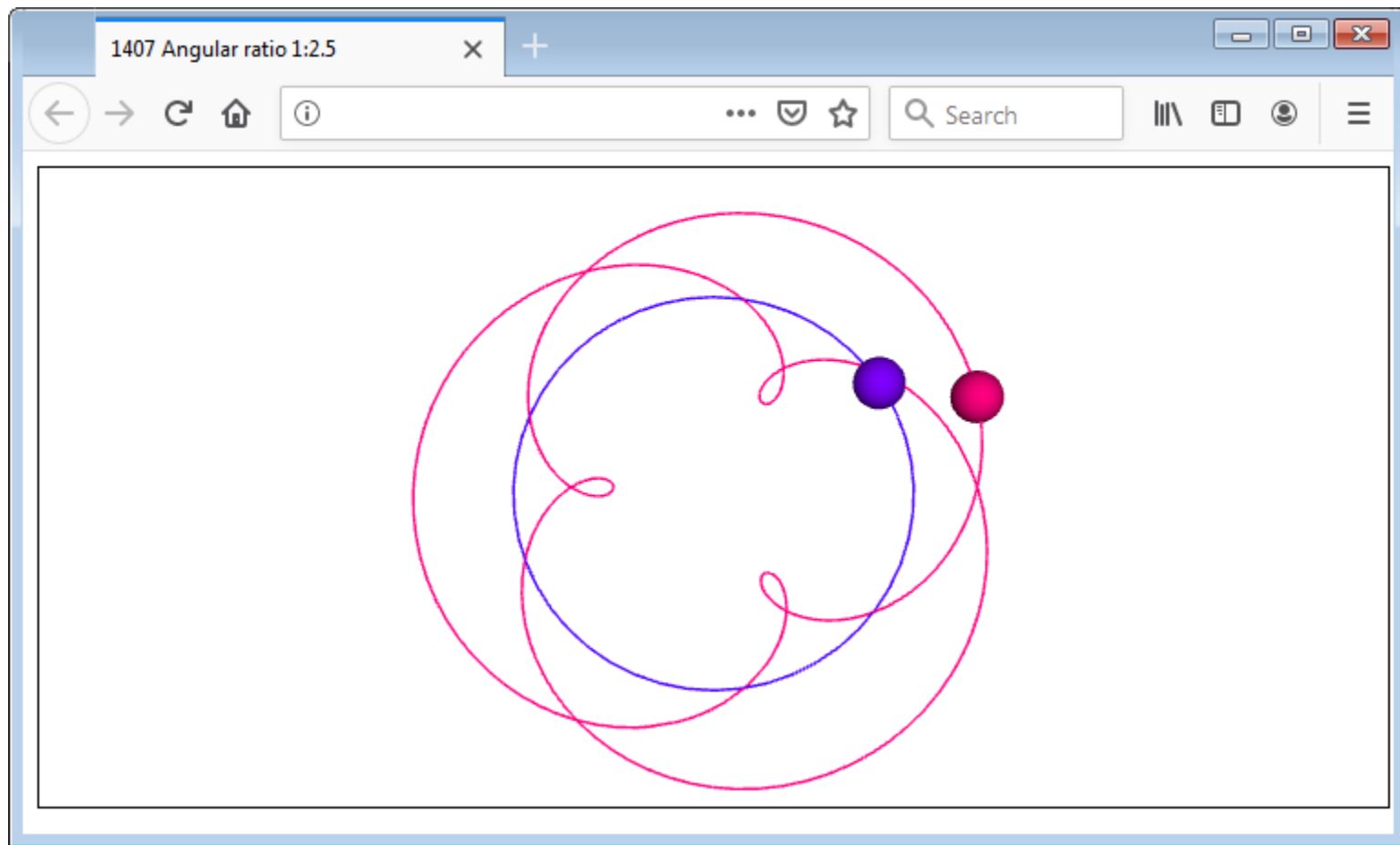
- Nested circular motion
- Ratio of angular velocities 1:2.5
- The same directions of motions

```
function loop()
{
    t = 3*Suica.time;
    a.center[0] = 30*Math.cos(t);
    a.center[1] = 30*Math.sin(t);
    b.center[0] = a.center[0] + 15*Math.cos(2.5*t);
    b.center[1] = a.center[1] + 15*Math.sin(2.5*t);
}
```

Generating a trajectory

- Adding a segment for each step
- Skipping the first step (then **from** still has no value and there are no two points needed to generate a segment)
- Generating lasts for about 4π seconds

```
function loop()
{
    ...
    if (from && 0.9<t && t<1+4*Math.PI)
        segment(from,to).custom({color:[1,0,0.5]});
    from = to;
    to = [b.center[0],b.center[1],0];
}
```

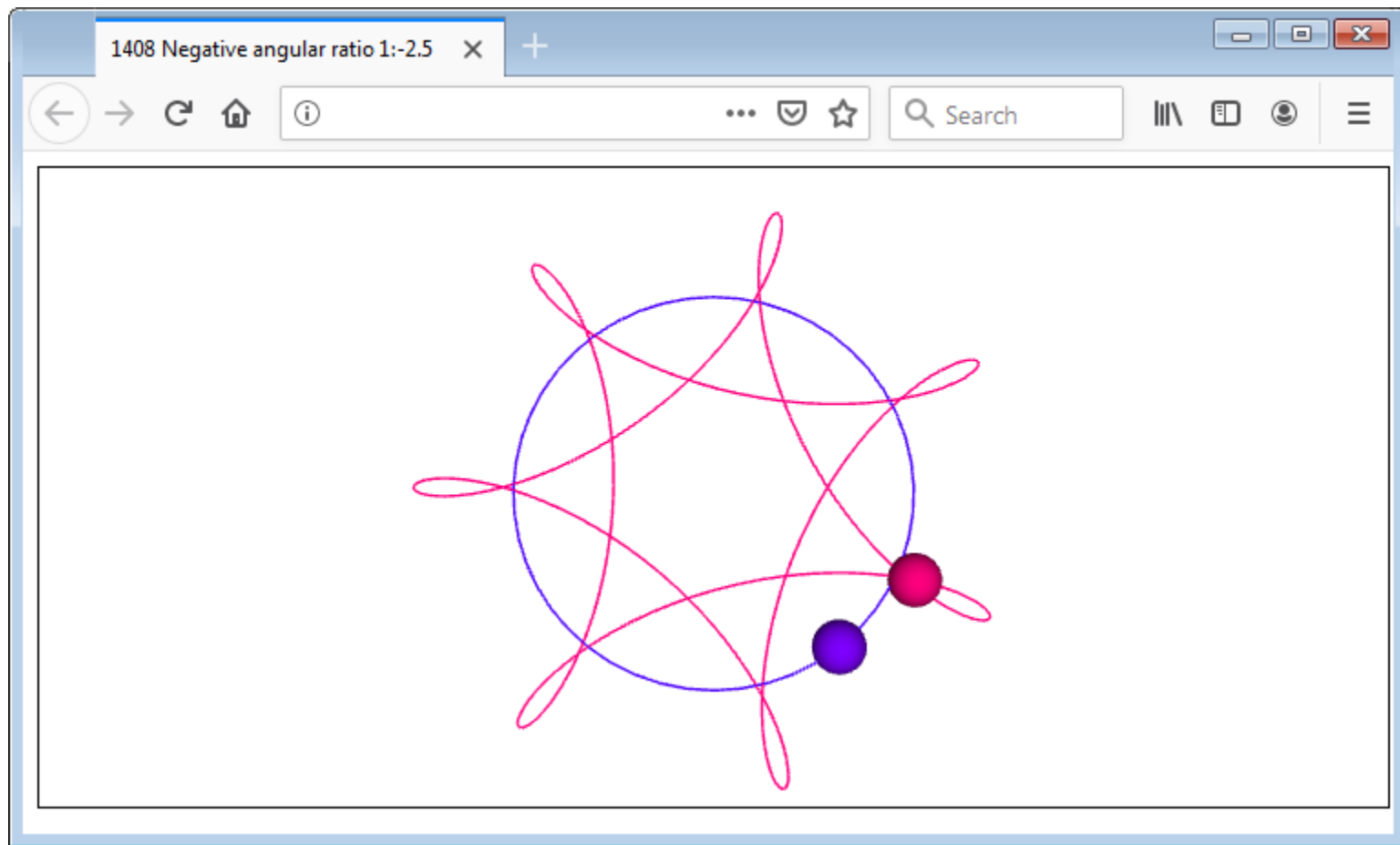



TRY IT

Negative angular ratio

- Motion is nested circular
- Ratio of angular velocities is 1:2.5
- Directions of motions are opposite

```
function loop()
{
    t = 3*Suica.time;
    a.center[0] = 30*Math.cos(t);
    a.center[1] = 30*Math.sin(t);
    b.center[0] = a.center[0] + 15*Math.cos(-2.5*t);
    b.center[1] = a.center[1] + 15*Math.sin(-2.5*t);
}
```



TRY IT

Motion on an arc

Motion on an arc



Motion on an arc

- Same as motion on a circle
- The arc is defined by two angles

When related to linear motion

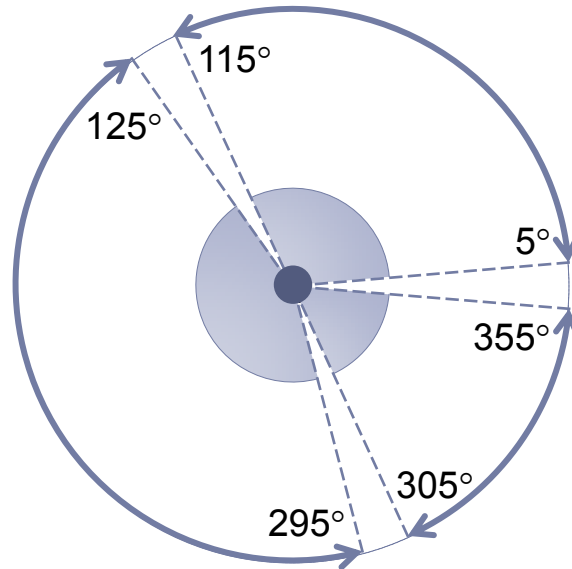
- Using “angular” vector
- Using linear combination of angles

Example



Motions on arc

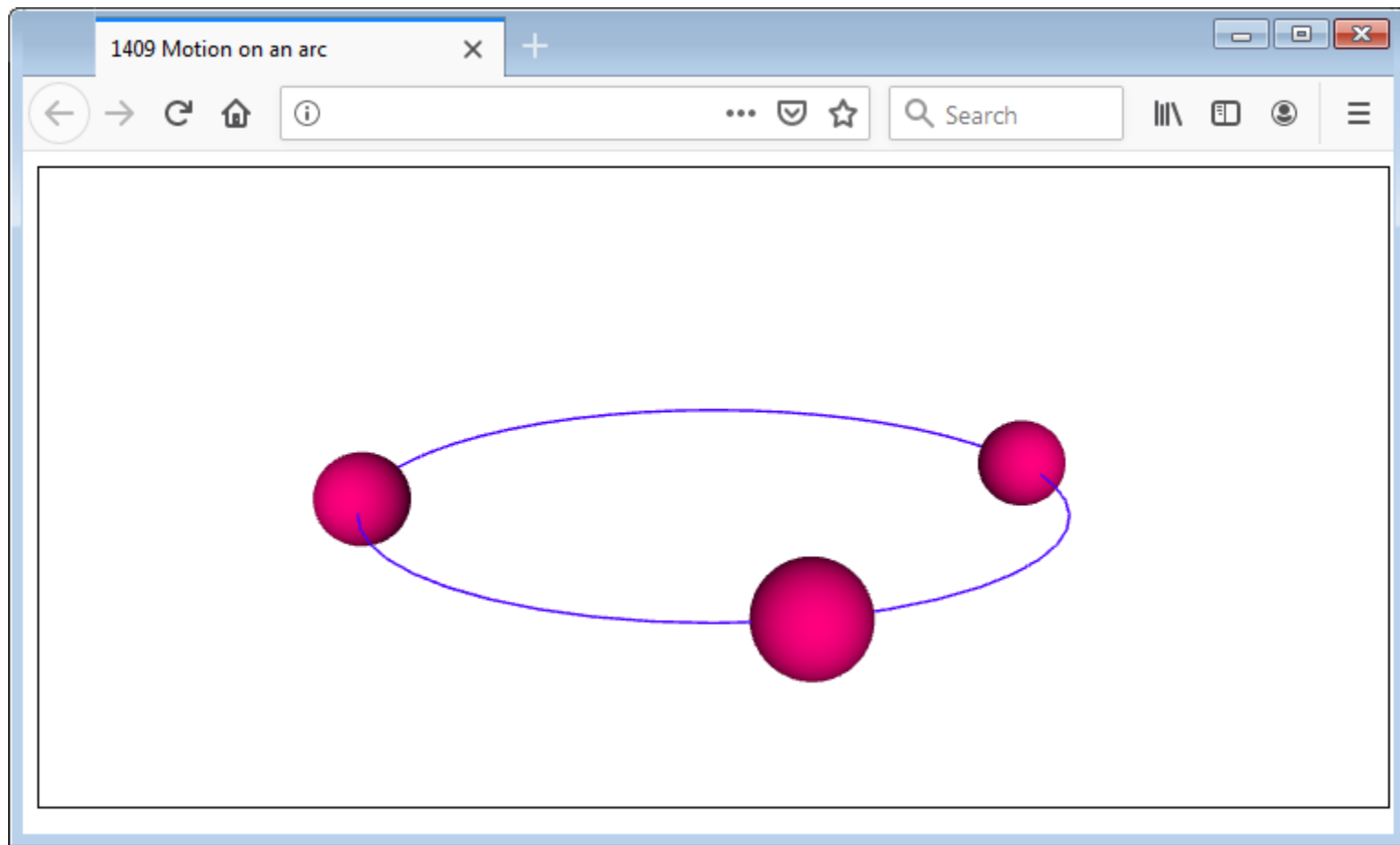
- Three spheres move on three arcs of the same circle



Implementation

- Each motion is on a circle
- The angle is changed sinusoidally
- Coefficients 1.5 and 1.25 change the angular velocity

```
ang = radians(60+55*Math.sin(t));  
a.center[0] = 30*Math.cos(ang);  
a.center[1] = 30*Math.sin(ang);  
ang = radians(210+85*Math.sin(1.5*t));  
b.center[0] = 30*Math.cos(ang);  
b.center[1] = 30*Math.sin(ang);  
ang = radians(330+25*Math.sin(1.25*t));  
c.center[0] = 30*Math.cos(ang);  
c.center[1] = 30*Math.sin(ang);
```



TRY IT

Pendulum



Chained pendulums

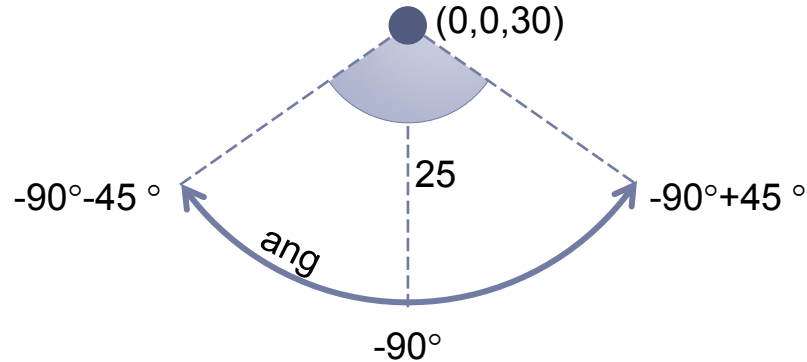
- A big swinging pendulum
- Another pendulum is attached to it
- A third pendulum is attached to the second one
- Looking for visual model, not physically correct

Idea

- Combining two motions:
 - Motion on an arc
 - Nested circular motion

Implementation of swinging

- The first pendulum: $\pm 45^\circ$ around the vertical line at -90°
- Pivot point is set to $z=30$, the length is 25
- The current swinging angle is stored in **ang**

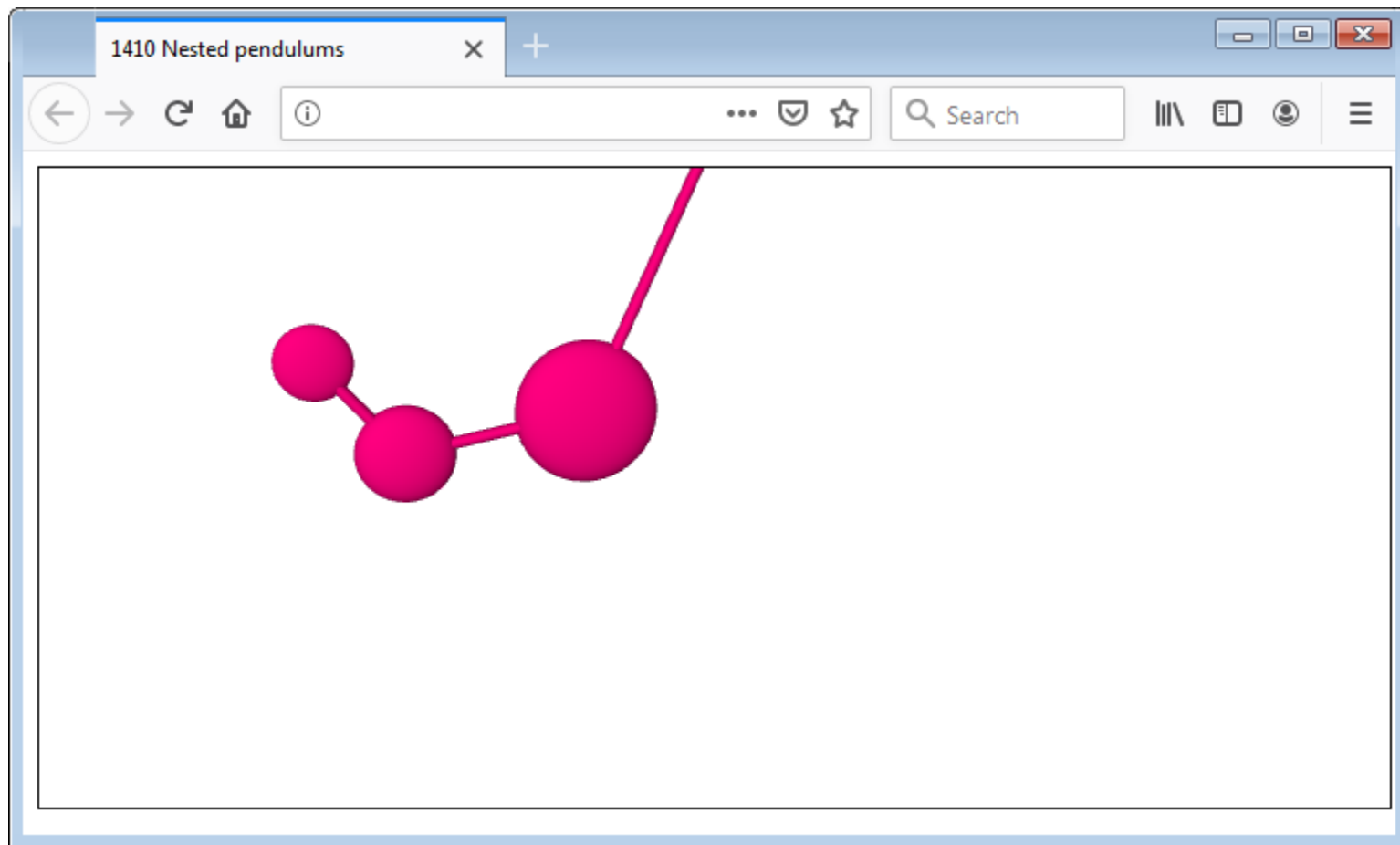


```
ang = radians(-90+45*Math.sin(t));  
a1.focus  = [0,Math.cos(ang),Math.sin(ang)];  
a2.center = [0,25*a1.focus[1],25*a1.focus[2]+30];
```

Second pendulum

- Swinging: $\pm 60^\circ$ in addition to the angle of the first pendulum
- Delayed 0.5 seconds
- Object **b1** is attached to the center of **a2**, and the center of **b2** is calculated after the circular motion in respect to **b1**
- The third pendulum is processed in the same way

```
ang = ang + radians(60*Math.sin(t-0.5));  
b1.center = a2.center;  
b1.focus = [0,Math.cos(ang),Math.sin(ang)];  
b2.center = [0,b1.center[1]+15*b1.focus[1],  
             b1.center[2]+15*b1.focus[2]];
```



TRY IT

Motion on object

Motion on cylinder



Compound motion

- Combination of two motions
- Circular motion (e.g. in XY)
- Linear motion (e.g. along Z)

$$x(u, v) = R \cos(u)$$

$$y(u, v) = R \sin(u)$$

$$z(u, v) = f(v)$$

Dominant velocity



Directional velocity

- Local velocity of a motion parameter
- Two direction when moving on a surface:

Along parameter u

Along parameter v

Dominant velocity

- The velocity in one direction is significantly larger than the velocity in another direction
- Making motion visually different

Cylinder



Motion on a cylinder

- Considering only the curved surface
- Described by two parameters – u и v
- Allows demonstration of dominant speed

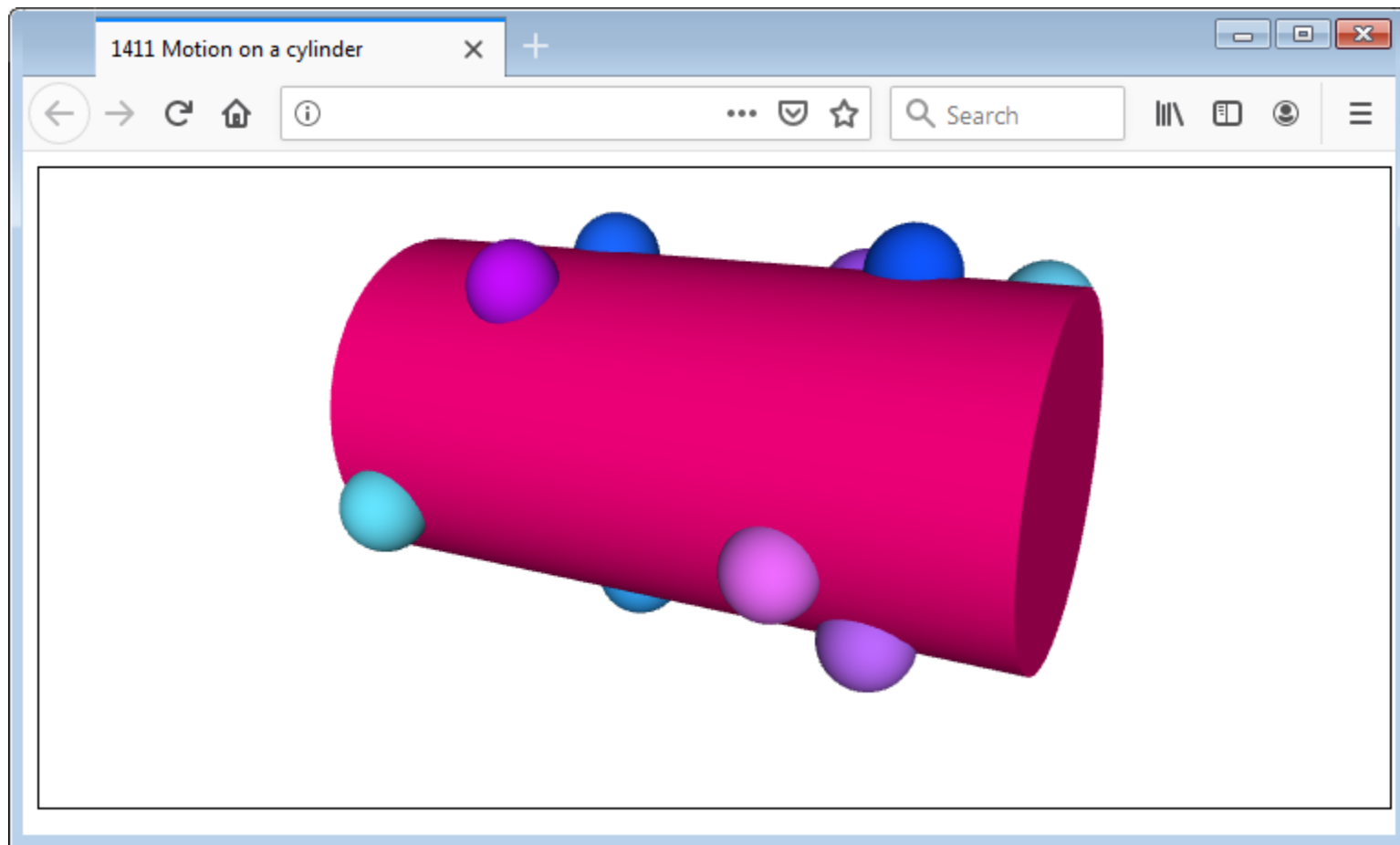
Two cases

- Dominant circular motion $u \gg v$
- Dominant linear motion $u \ll v$
(when $u \approx v$ there is no dominant motion)

Cylinder long Y axis

- Circular motion in X and Z, linear along Y
- Objects have **speed** $\in [2, 5)$
- Linear motion is shifted by **i**, to ungroup objects

```
for (var i=0; i<n; i++)  
{  
    u = a[i].speed*t;  
    v = t+i;  
    a[i].center[0] = 15*Math.cos(u);  
    a[i].center[1] = 26*Math.sin(v);  
    a[i].center[2] = 15*Math.sin(u);  
}
```

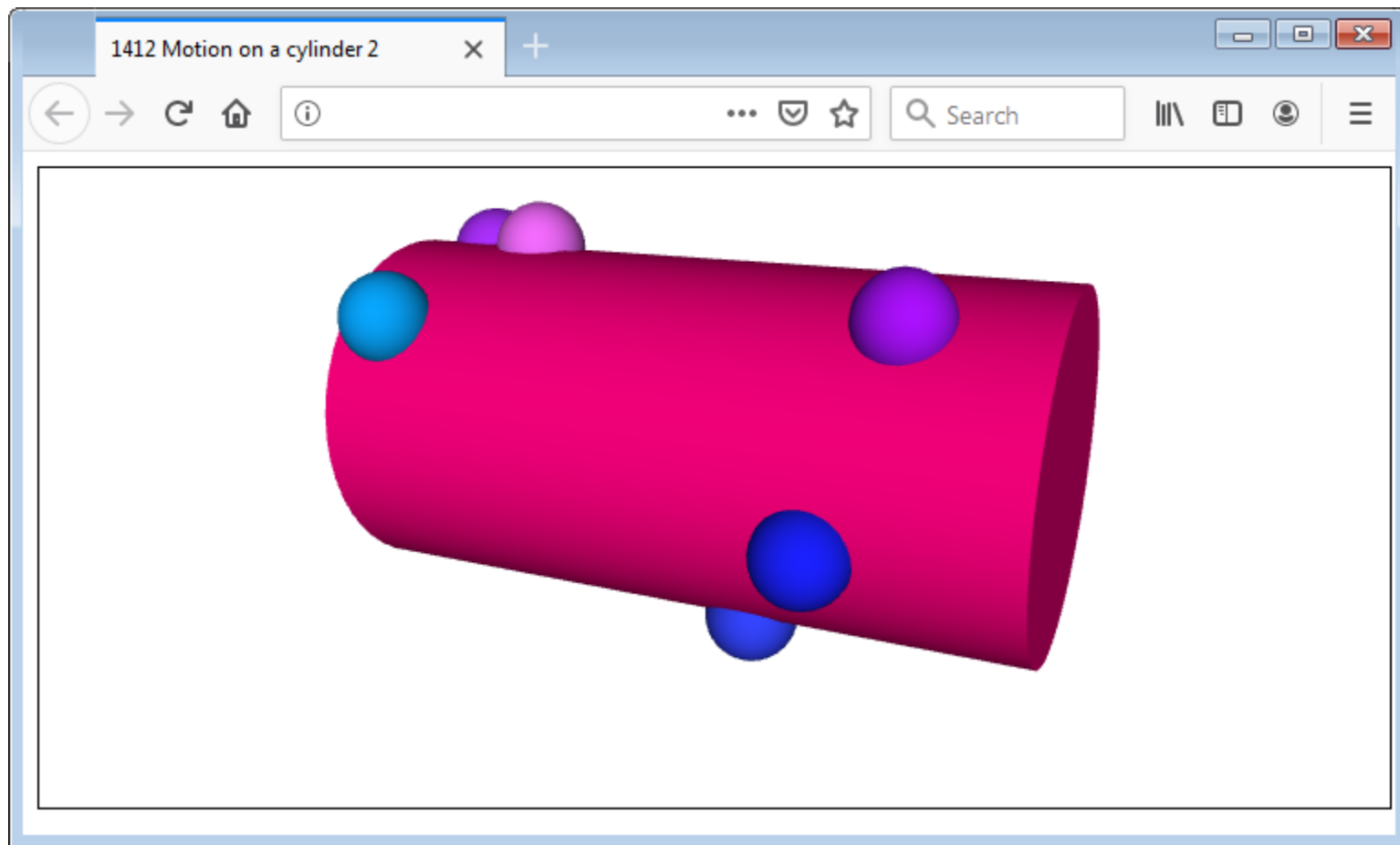


TRY IT

Change of dominant velocity

- In previous example motion for u was 2 to 5 times faster than the motion for v
- Noa making v faster than u with $\text{speed} \in [2, 5)$

```
for (var i=0; i<n; i++)  
{  
    u = t+i;  
    v = a[i].speed*t;  
    a[i].center[0] = 15*Math.cos(u);  
    a[i].center[1] = 26*Math.sin(v);  
    a[i].center[2] = 15*Math.sin(u);  
}
```



TRY IT

Sphere



Motion on a sphere

- Defined by two parameters – u and v
- Allows demonstration of dominant velocity

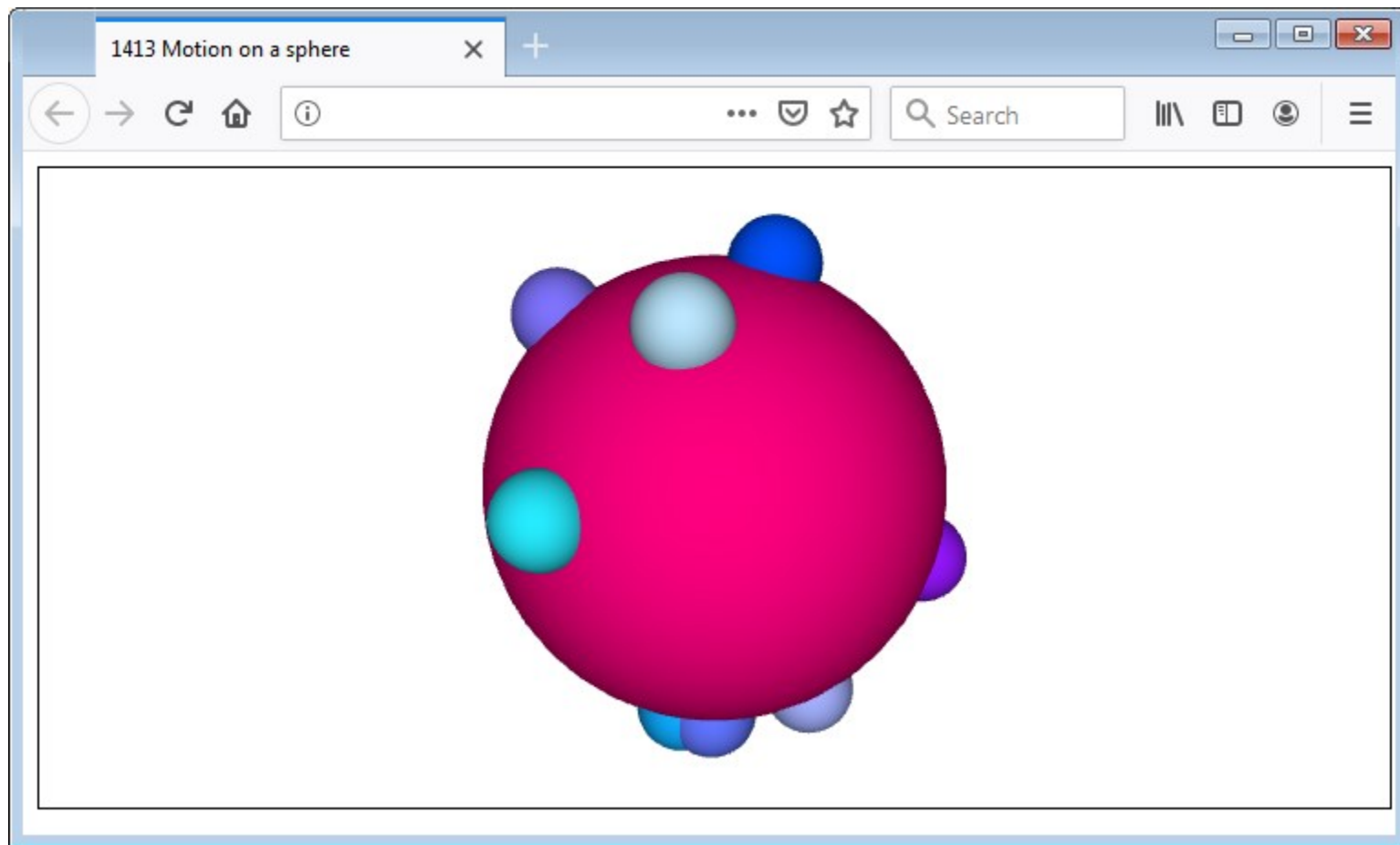
Implementation

- Spherical coordinates, u and v are angles
- Motion has two special points – the two poles

Implementation

- Transforming spherical coordinates to Cartesian coordinates
- Horizontal angle u , vertical angle v
- Two separate speeds for u and v – **uSpeed** and **vSpeed**

```
for (var i=0; i<n; i++)  
{  
    u = a[i].uSpeed*t+i;  
    v = a[i].vSpeed*t+i;  
    a[i].center[0] = 20*Math.cos(u)*Math.cos(v);  
    a[i].center[1] = 20*Math.sin(u)*Math.cos(v);  
    a[i].center[2] = 20*Math.sin(v);  
}
```



TRY IT

Motion on a cube

- Implemented as motion on a sphere
- Calculating point on a sphere p
- Largest coordinate is d
- A point on the cube has center p , but scaled with d , so that the largest coordinate becomes ± 20

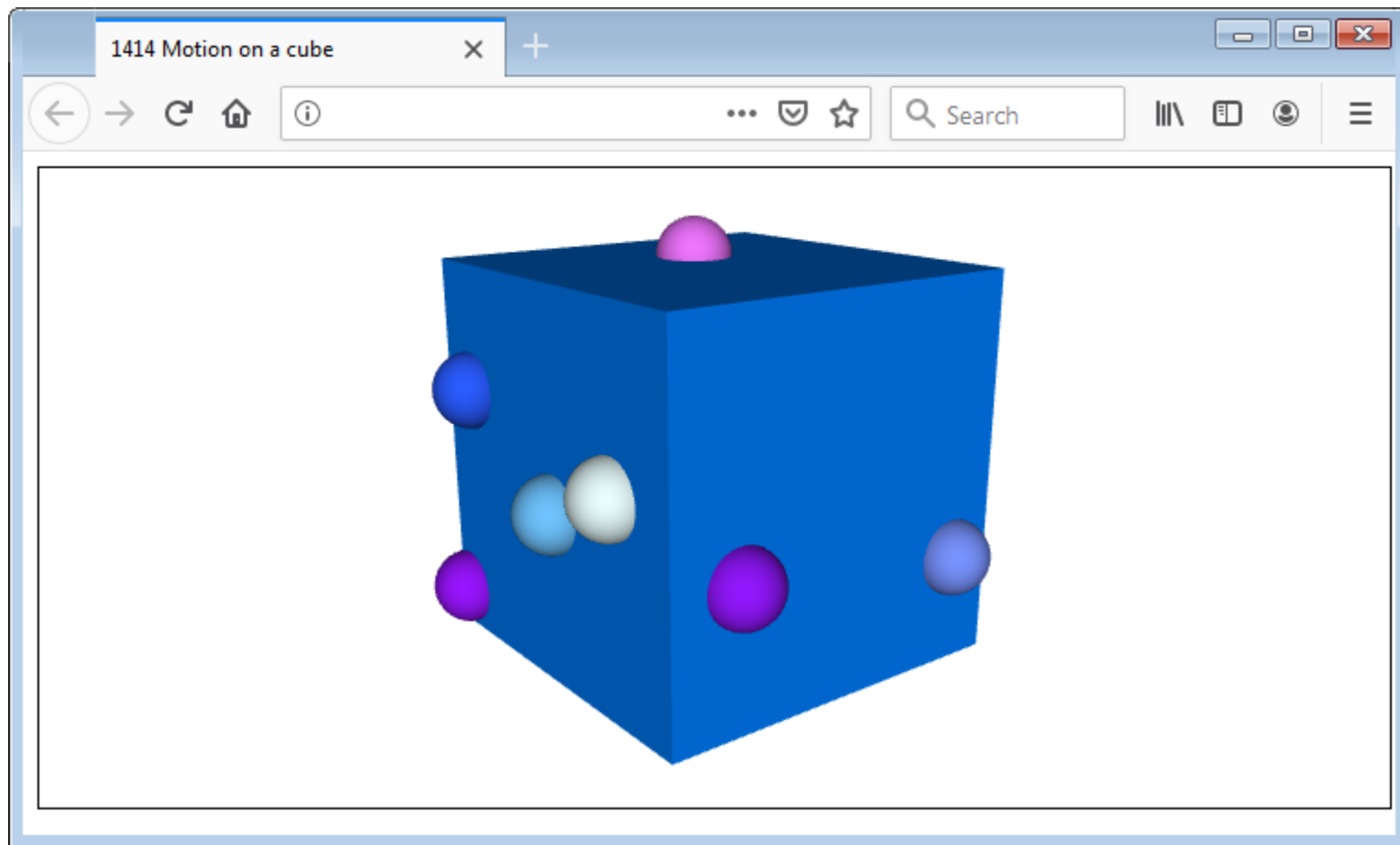
```
 $p = [\cos(u)*\cos(v), \sin(u)*\cos(v), \sin(v)];$ 
```

```
 $d = \max(\text{abs}(p[0]), \text{abs}(p[1]), \text{abs}(p[2]));$ 
```

```
 $a[i].center[0] = 20*p[0]/d;$ 
```

```
 $a[i].center[1] = 20*p[1]/d;$ 
```

```
 $a[i].center[2] = 20*p[2]/d;$ 
```

TRY IT



Summary

Circular motion



Motion on a circle

- Implemented with polar coordinates
- Linear velocity (distance per time)
- Angular velocity (angle per time)

Relative motion

- Object moves around another moving object
- Calculating via accumulation of several transformations from polar to Cartesian coordinates

Motion on an arc

- Implemented as motion on a circle
- Limitation on angle (in polar coordinates)

Motion on a surface

- There are two parameters
- Dominant velocity if one parametric velocity is significantly different from the other parametric velocity



ICT in SES

The end

Comments, questions