ICT in SES

User interface

Lesson №19

Entering data

Entering data

Explicit

- With entering in text boxes
- With selecting values from lists

Implicit

- With mouse motion
- With interactive controls

Entering with typing

Idea

- Needing numerical data
- Typing them in text boxes

Example

- Creating circles one by one
- Entering coordinates of center and radius
- Clicking on a button to create a circle

Styles

- Input elements are in a block
- Its class is “panel” and is at the top left corner

```css
.panel {
   position: absolute;
   left: 10px;
   top: 10px;
   width: 100px;
   border: solid 1px black;
   padding: 0 0.25em 0.5em 0.25em;
   text-align: center;
   background-color: DarkSeaGreen;
}
```

Elements

- Two coordinates and a radius for each circle
- Three text boxes with distinct id
- Variables elemX, elemY and elemR are references to the boxes

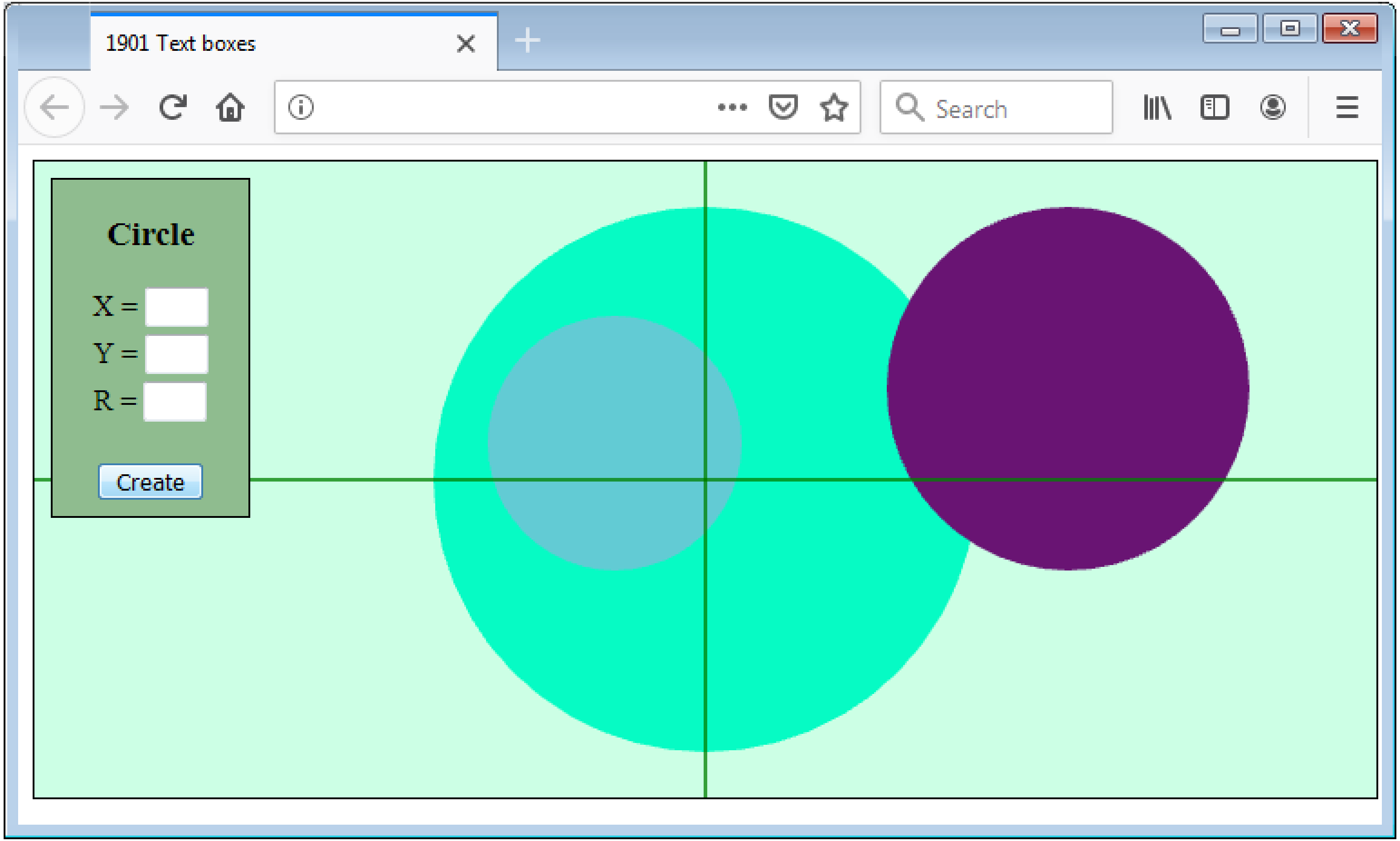
```
<div class="panel">
  <div>X = <input type="text" id="x" ...></div>
  <div>Y = <input type="text" id="y" ...></div>
  <div>R = <input type="text" id="r" ...></div>
</div>
```

```
elemX = document.getElementById('x');
elemY = document.getElementById('y');
elemR = document.getElementById('r');
```

Creating circles

- Using value to extract values
- Clearing the text boxes after circle creation

```
var z = -100;
function create()
{
   var x = Number(elemX.value);
   var y = Number(elemY.value);
   var r = Number(elemR.value);
   circle([x,y,z++],r).custom({...});
   elemX.value = '';
   elemY.value = '';
   elemR.value = '';
}
```

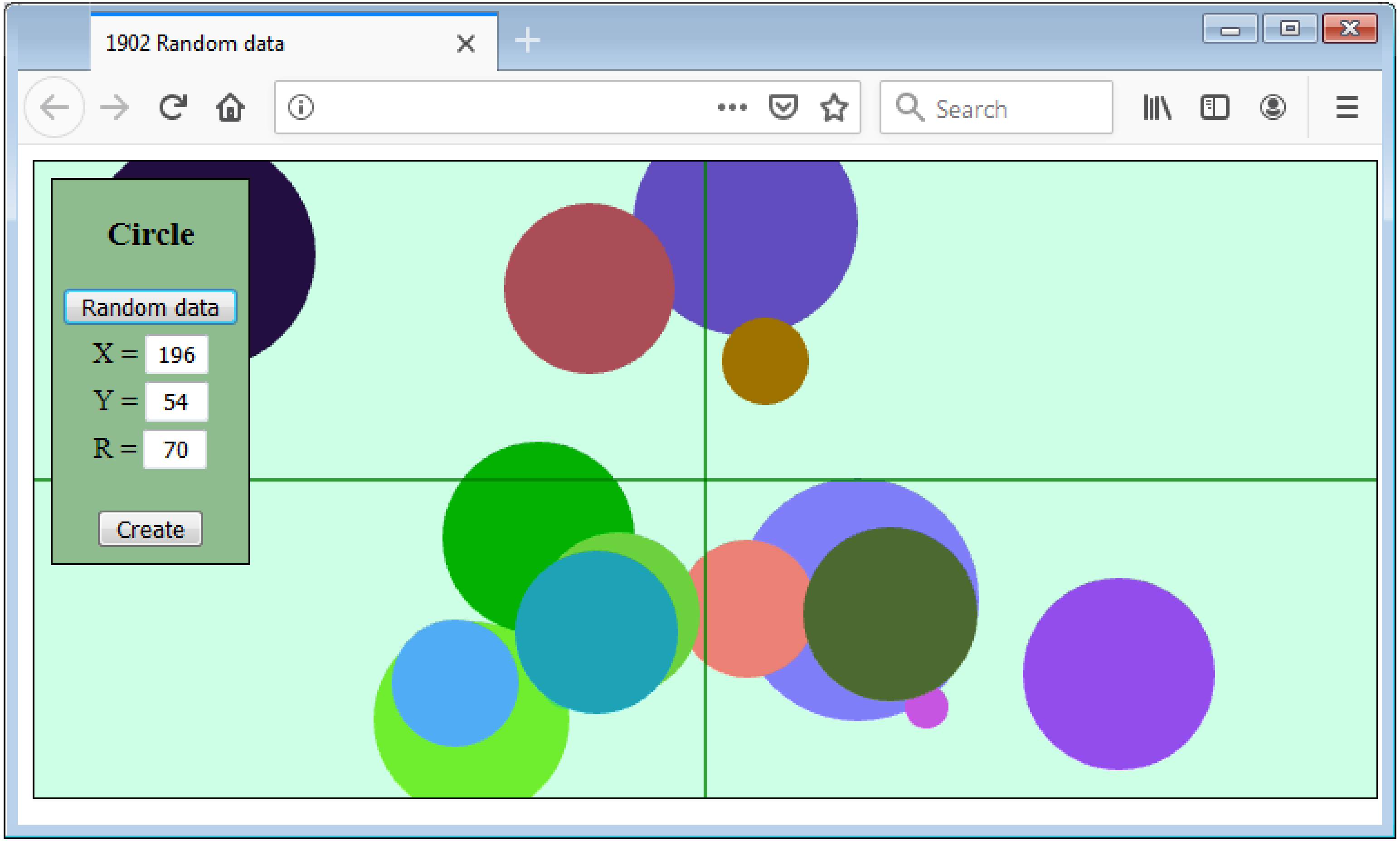


TRY IT

A new button

- Fills the text boxes with random values
- Hardcoding the range for each randomness
- Rounding is only for aesthetical reasons

```
function randomData()
{
   elemX.value = Math.round(random(-300,300));
   elemY.value = Math.round(random(-150,150));
   elemR.value = Math.round(random(10,80));
}
```



Circle

Random data

X = 196

Y = 54

R = 70

Create

TRY IT

New functionality

Selecting a circle

- Clicking on a circle adds a blinking outline
- Circle’s data are shown in the text boxes
- The buttons for random data and circle creation are disabled

Unselecting a circle

- Clicking outside a circle removes its outline
- Both buttons become enabled

Blinking outline

- Implemented as a black circle
- If a circle is selected, the outline blinks each 0.25 seconds

```
blinker = circle([0,0,0],0).custom({
   color:[0,0,0],
   mode:Suica.LINE});
...
function animate()
{
   if (obj)
   {
      blinker.visible = (2*Suica.time)%1>0.5;
   }
}
```

Clicking on a circle

- Showing the outline
- Showing circle’s data
- Disabling both buttons

```
blinker.visible = true;
blinker.center = obj.center;
blinker.radius = obj.radius;

elemX.value = obj.center[0];
elemY.value = obj.center[1];
elemR.value = obj.radius;

elemRandom.disabled = true;
elemCreate.disabled = true;
```

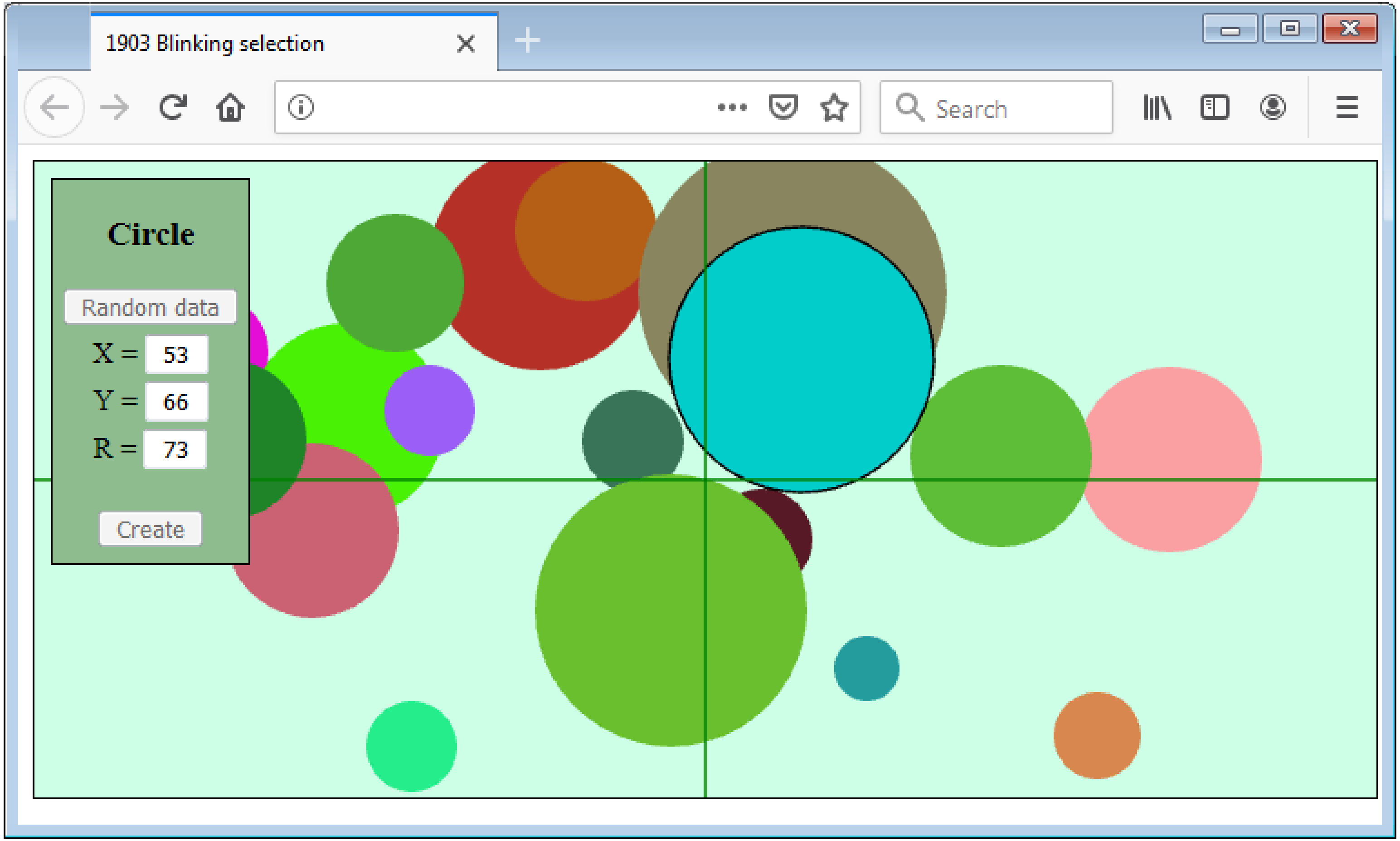
Clicking outside a circle

- Hiding the outline
- Clearing the text boxes
- Enabling both buttons

```
blinker.visible = false;

elemX.value = '';
elemY.value = '';
elemR.value = '';

elemRandom.disabled = false;
elemCreate.disabled = false;
```



1903 Blinking selection
Search
Circle
Random data
X = 53
Y = 66
R = 73
Create

TRY IT

Another addition

When a circle is selected

- Button [Create] becomes [Update]
- If text box values are modified, [Update] updates the selected circle with the new data

Idea

- Instead of two buttons there will be one with updated caption

Updating caption

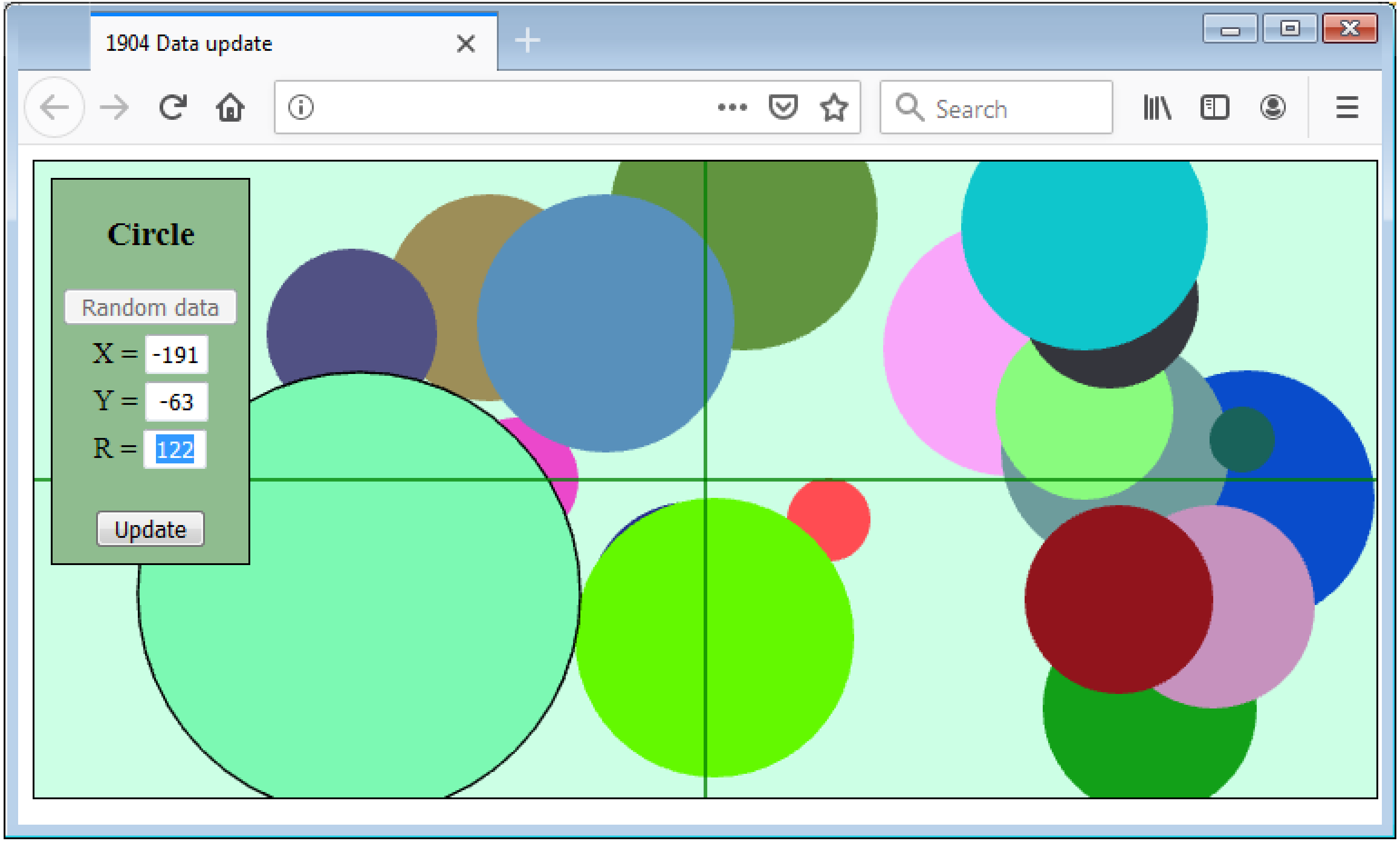
- Depends on whether clicking is on object

```
function mouseDown(event)
{
   obj = p.objectAtPoint(...);
   if (obj)
   { ...
      elemCreate.innerHTML = 'Update';
   }
   else
   { ...
      elemCreate.innerHTML = 'Create';
   }
}
```

Button

- If a circle is selected then update its center and radius
- Also update the blinking outline

```
function create()
{ ...
   if (obj)
   {
      obj.center = [x,y,0];
      obj.radius = r;
      blinker.center = [x,y,0];
      blinker.radius = r;
   }
   else { circle([x,y,z++],r)...}
}
```



X = -191

Y = -63

R = 122

TRY IT

Interactive controls

Interactive controls

Purpose

- Alternative method for entering values
- May not be available in every browser

Types

- Checkboxes and radio buttons
- Text boxes for passwords
- Colour selectors
- Date selectors
- Range selectors

Checkboxes and radio buttons

Comparison

- Checkboxes – selection of any number of elements
- Radio buttons – selecting only one element

Appearance

- Depends on the browser

Example

- Scene with three motions: horizontal, vertical, radial
- Each motion is started and stopped by a checkbox
- Initially only the horizontal motion is active

Defining controls

- Checkboxes re tags input with type set to checkbox
- Checkboxes and their labels combined in tag label, so that labels are clickable

```
<label><input type="checkbox" id="hor">Horizontal</label>
<label><input type="checkbox" id="ver">Vertical</label>
<label><input type="checkbox" id="rad">Radial</label>
```

Initialization

- Remembering DOM elements of the checkboxes
- Setting the horizontal motion checkbox status to checked

Alterntaive

- Attribute checked might have been added to the HTML declaration of the tag

```
hor = document.getElementById('hor');
ver = document.getElementById('ver');
rad = document.getElementById('rad');

hor.checked = true;
```

Motion implementation

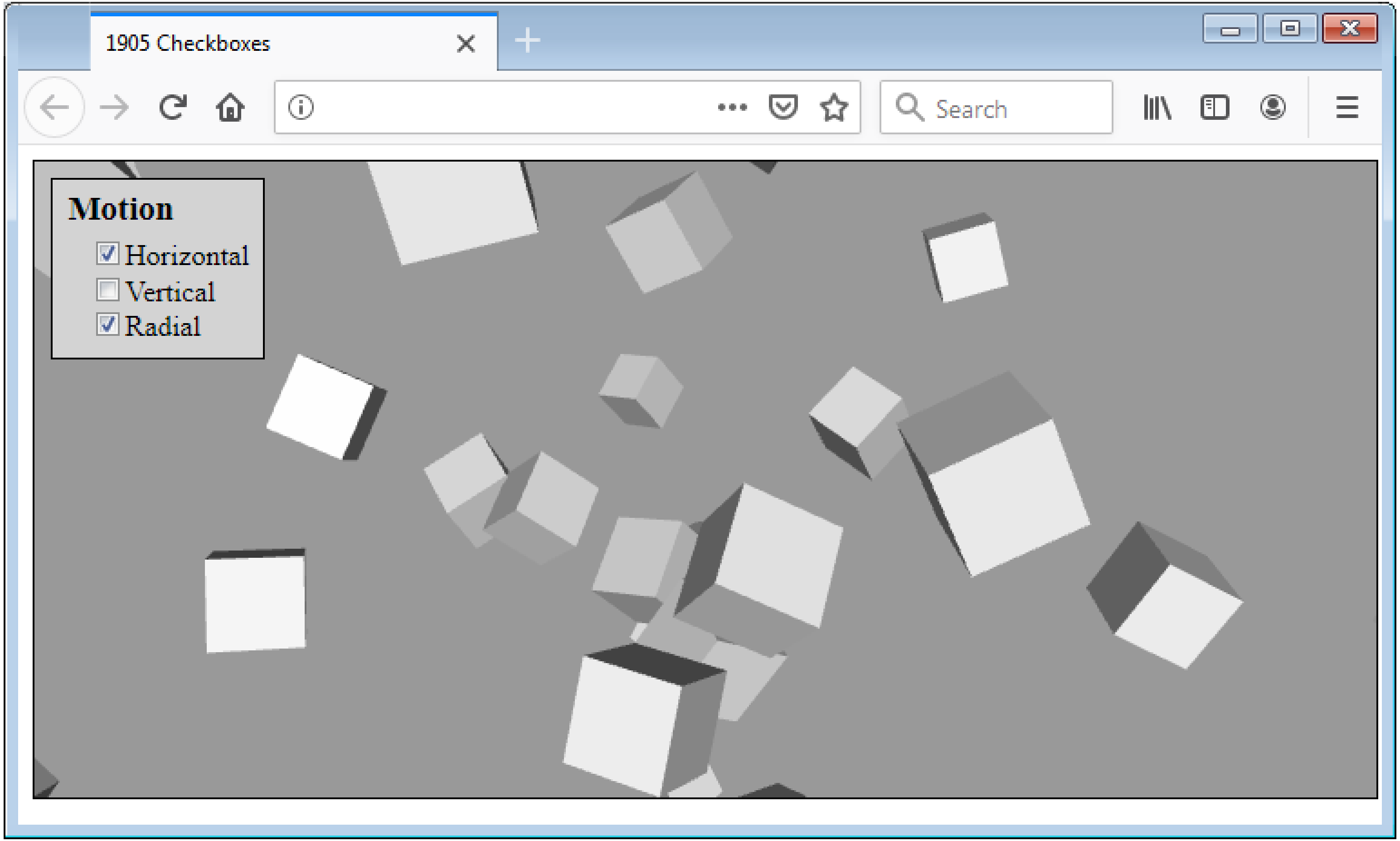
- Three variables aHor, aVer and aRad contain the angles
- Elapsed time since the previous frame dT is added to angles, which checkboxes are checked

```
function animate()
{
   ...
   if (hor.checked) aHor += dT;
   if (ver.checked) aVer += dT;
   if (rad.checked) aRad += dT;
   ...
}
```

Calculating the view point

- Position on a sphere, horizontal motion is unidirectional, vertical and radial motions are sinusoidal
- Horizontal angle h corresponds to aHor
- Vertical angle v changes from -1 to 1 radians based on aVer
- Radial distance r is from 100 to 300 depending on aRad

```
var r = 200+100*sin(aRad);
var h = aHor;
var v = sin(aVer);
lookAt( [r*cos(h)*cos(v),
         r*sin(h)*cos(v),
         r*sin(v)], [0,0,0], [0,0,1] );
```



Motion

☑ Horizontal

☐ Vertical

☑ Radial

TRY IT

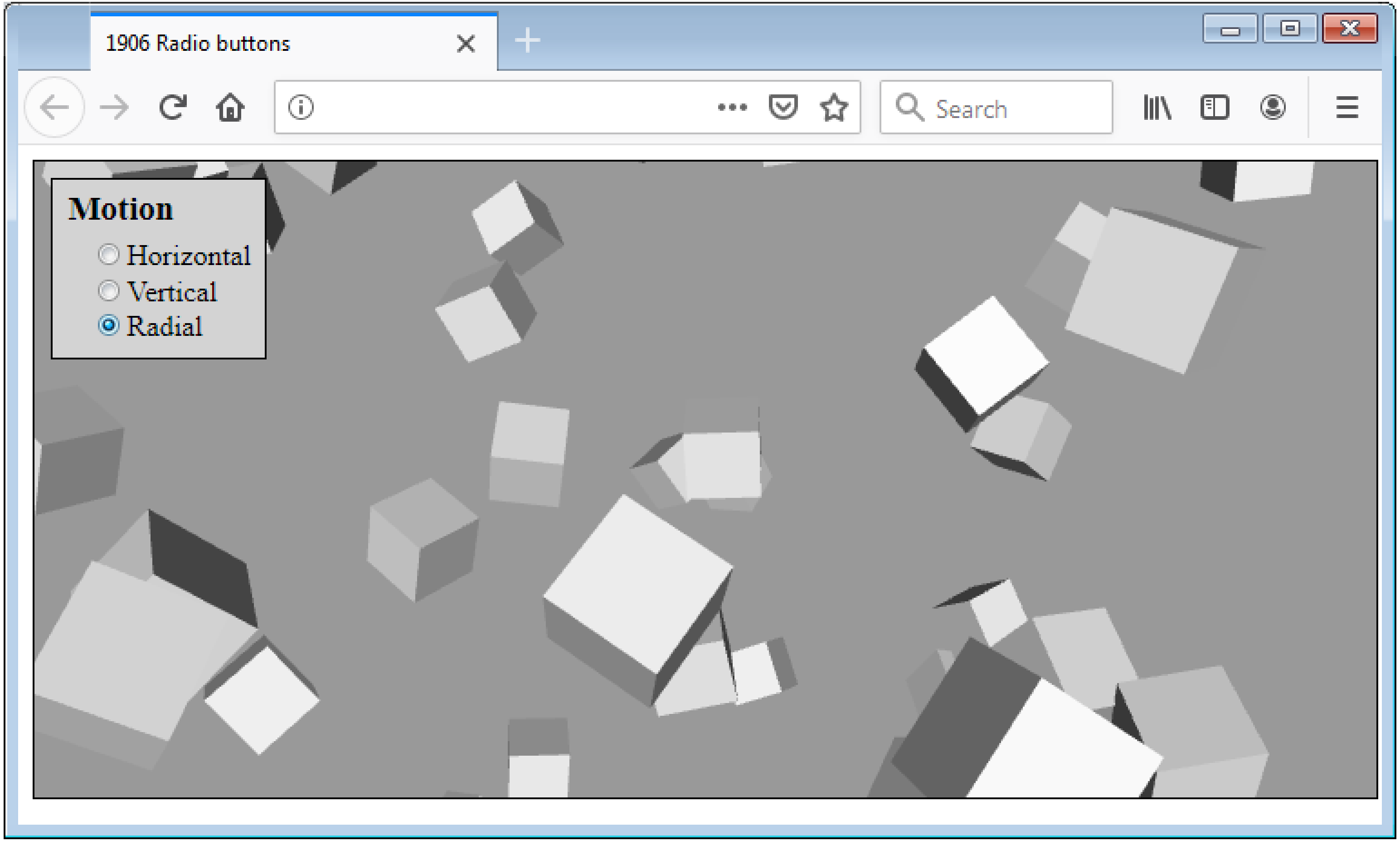
Same example, but with radio buttons

- Each motion is started by a radio button
- Only one motion could be active at any time

Defining controls

- Again with input tag, but type is radio
- Again inside label
- Additional attribute name, which has the same value for all radio buttons – thus they are treated as a group

```
<label><input type="radio" name="motion" id="hor">…</label>
<label><input type="radio" name="motion" id="ver">…</label>
<label><input type="radio" name="motion" id="rad">…</label>
```



Motion

○ Horizontal
○ Vertical
● Radial

TRY IT

Passwords

Text boxes for passwords

- The number of shown objects is types as a password – i.e. the text is hidden by *****
- Element input with type set to password
- The actual visualization is implemented in doIt

```
<h3>Motion</h3>
<p>Enter secretly the number of cubes to show,
   from 0 to 99</p>
Count: <input type="password" id="num" size="2">
<button onclick="doIt()">Show</button>
```

Implementation of doIt

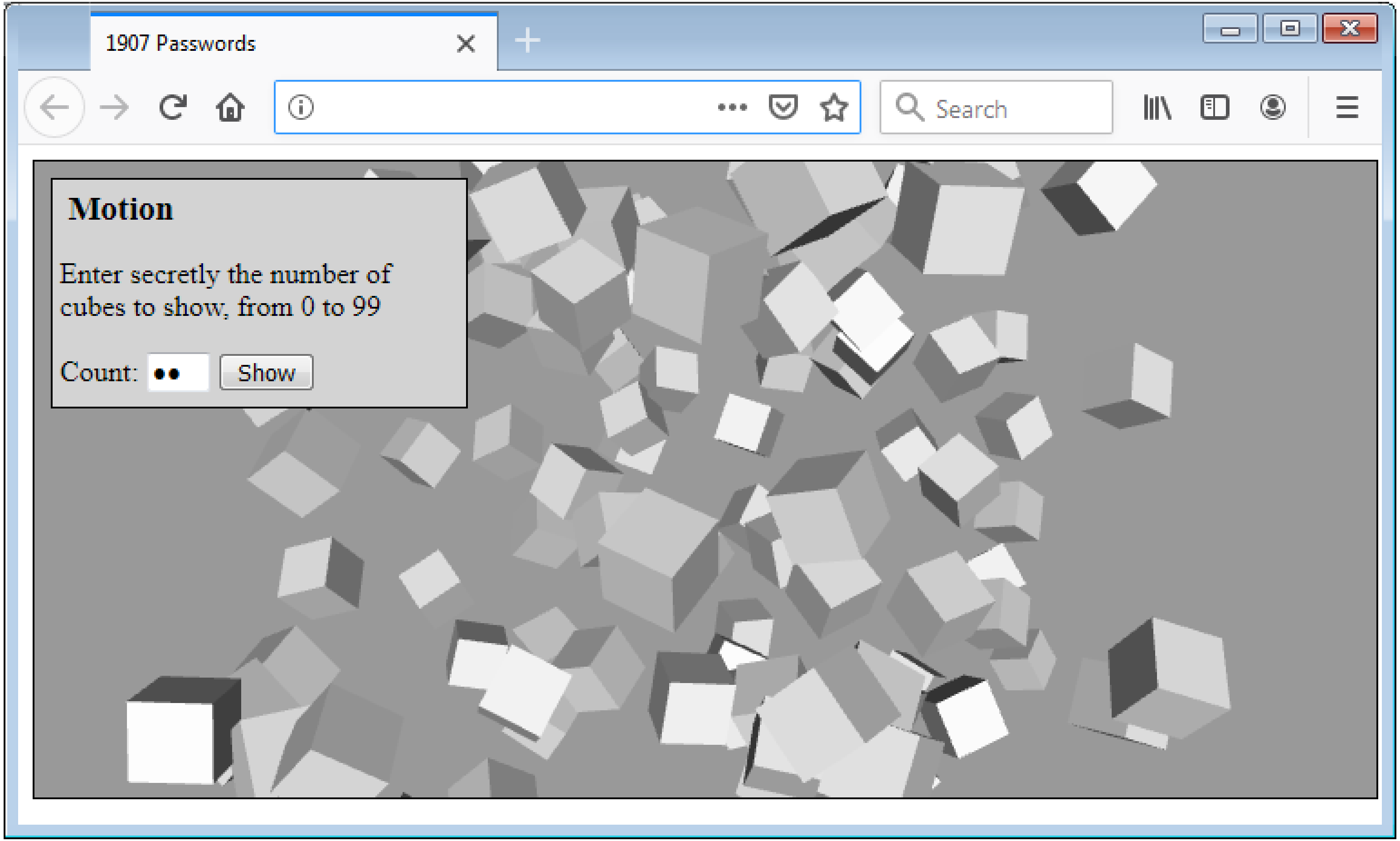
- The rel value of the password box is in value
- A way to bound a number in range, without if, is to use min and max

```
var elem = document.getElementById('num');
var num = parseInt(elem.value);

num = Math.max(Math.min(num,100),0);

for (var i=0; i<100; i++)
   c[i].visible = i<num;

elem.value = '';
```



Motion

Enter secretly the number of cubes to show, from 0 to 99

Count: ●● Show

TRY IT

Selecting a colour

Example

- Scene with graphical objects
- Interactive selection of colour
- Colouring all objects with this colour

Ideas

- Libraries and components providing this functionality
- Default HTML support in Firefox, Chrome and Opera
- In some browsers the colour element is shown as a text box

HTML approach

- Using input with type set to color
- The value is a colour in #rrggbb format
- Each colour component is from 00 (0) to FF (255)
- The initial value is #ffffff (i.e. white colour)

```
<h3>Select a colour</h3>
<input type="color"
       id="col"
       value="#ffffff"
       onchange="doIt()">
```

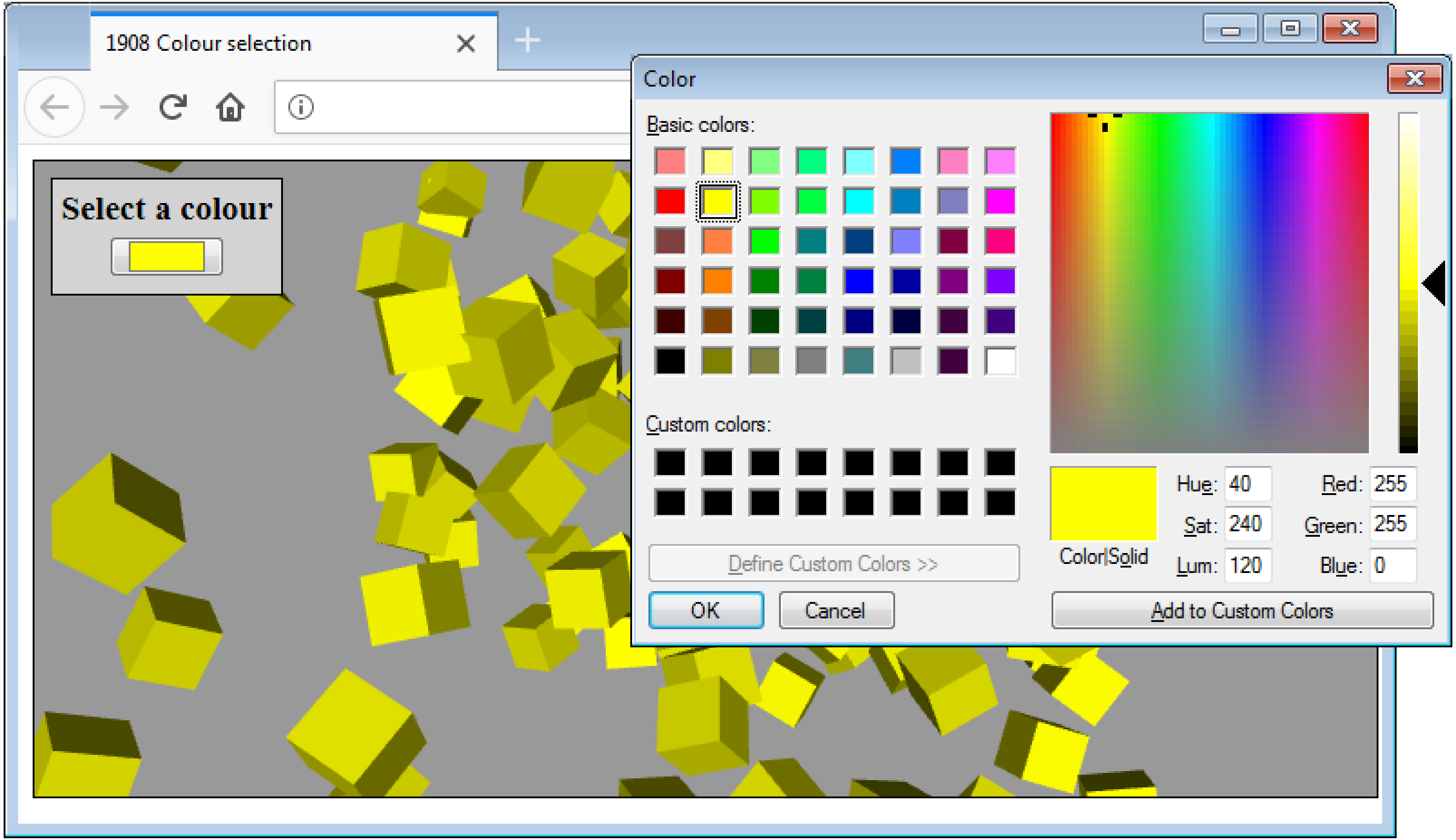
Processing change event

- Assuming the colour value is in #rrggbb format
- Pairs of hexadecimal digits are extracted with substring
- Converting to integer numbers with parseInt(…,16)
- Dividing by 255 produces values between 0 and 1

```
var elem = document.getElementById('col');

var r = parseInt(elem.value.substring(1,3),16);
var g = parseInt(elem.value.substring(3,5),16);
var b = parseInt(elem.value.substring(5,7),16);

for (var i=0; i<100; i++)
   c[i].color = [r/255,g/255,b/255];
```



Hue: 40
Sat: 240
Lum: 120
Red: 255
Green: 255
Blue: 0

TRY IT

Selecting a date

Problem

- Interactive selecting of a date (from a calendar)

Ideas

- Libraries and components providing this functionality
- Default HTML support in Firefox, Chrome and Opera
- In some browsers the date element is shown as a text box

Implementation

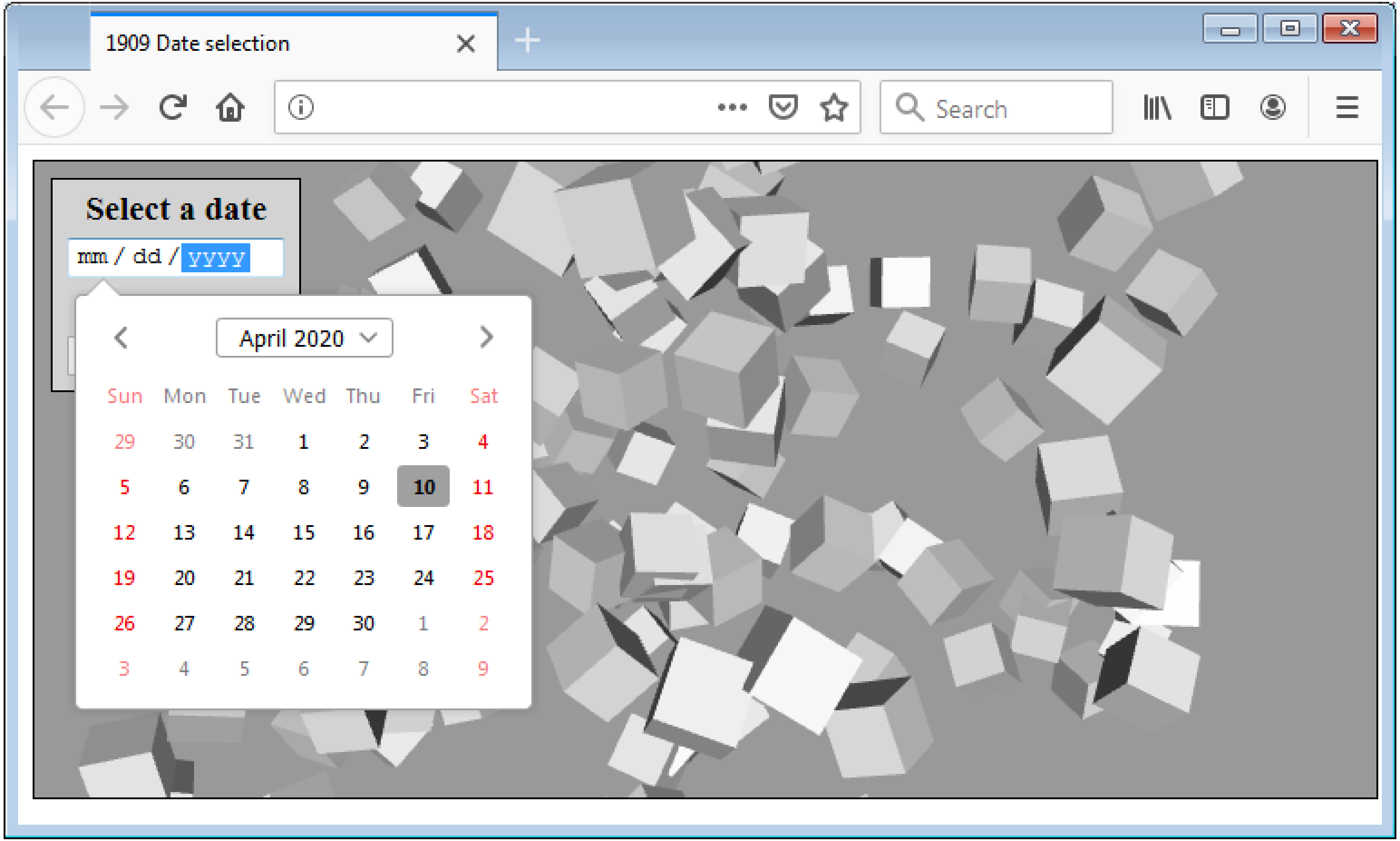
- Using element input with type set to date
- The second input element is only for showing the selected date, thus it is defined disabled

```
<h3>Select a date</h3>
<input type="date" id="dat" onchange="doIt()">

<h3>Selected date</h3>
<input type="date" id="tad" disabled>
```

- Selected date is shown in the second date box

```
var elem = document.getElementById('dat');
document.getElementById('tad').value = elem.value;
```


1909 Date selection
Search
Select a date
mm / dd / yyyy
April 2020
Sun Mon Tue Wed Thu Fri Sat
29 30 31 1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 1 2
3 4 5 6 7 8 9

TRY IT

Selecting range

Goal

- Interactive selection of a number in a range
- The number is not typed, but picked

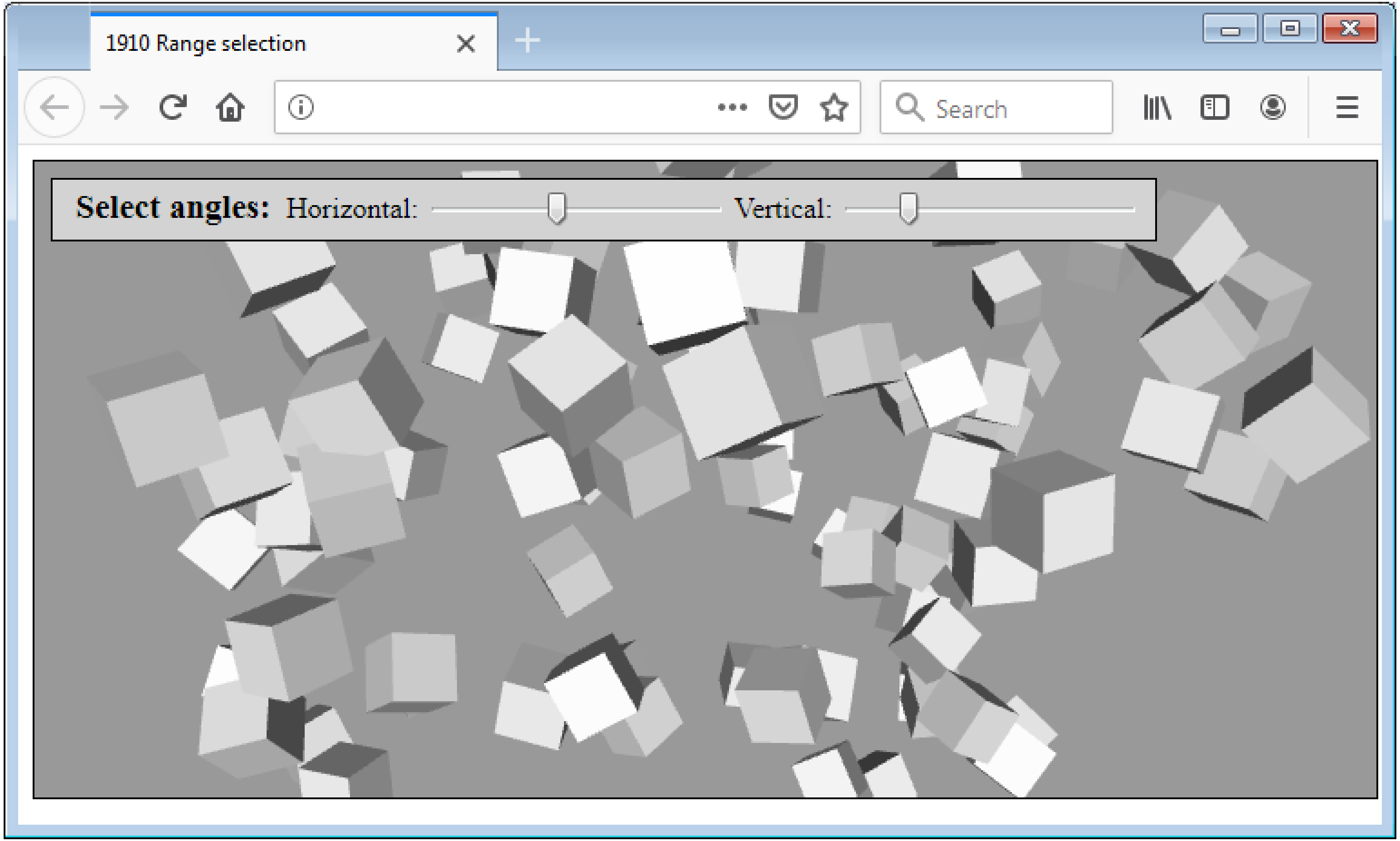
Idea

- The traditional HTML way
- Different appearance in different browsers

Implementation

- Using element input with type set to range
- Attribute min defines the minimal accepted value
- Attribute max defines the maximal accepted value
- Attribute value contains the initial (and later on – the current) value of the element

```
Horizontal:
<input type="range" id="hor" min="-120" max="120"
                                       value="0">
Vertical:
<input type="range" id="ver" min="-80" max="80"
                                       value="0">
```



Select angles: Horizontal: Vertical:

TRY IT

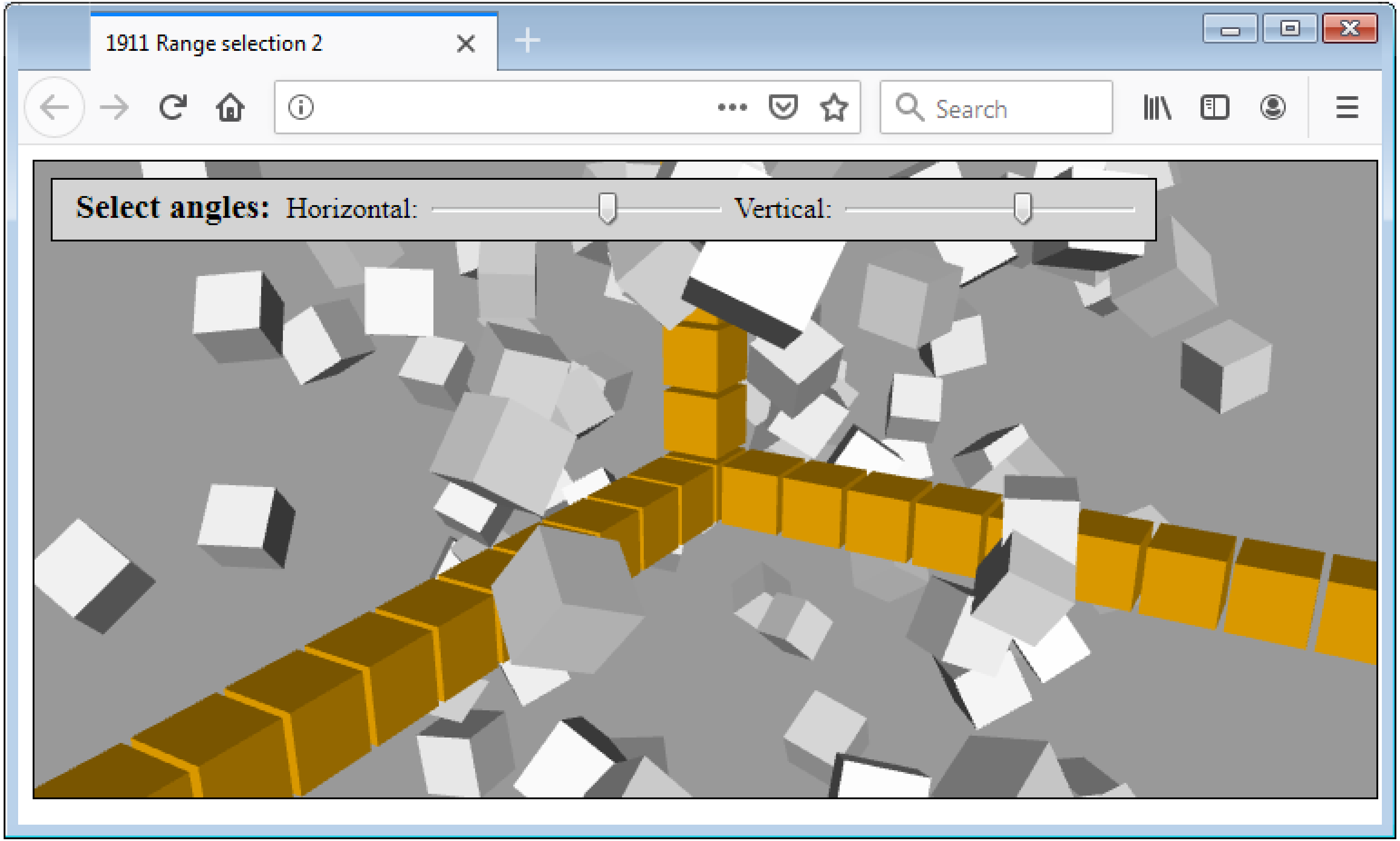
Modification

- Accepted values are at a step

  Horizontal step is 30, i.e. values are -120, -90, -60, -30, 0, 30, 60, 90, 120

  Vertical step is 20, i.e. values are -80, -60, -40, -20, 0, 20, 40, 60, 80
- Using attribute step

```
Horizontal:
<input type="range" id="hor" min="-120" max="120"
                              step="30" value="0">
Vertical:
<input type="range" id="ver" min="-80" max="80"
                              step="20" value="0">
```

1911 Range selection 2
Search
Select angles: Horizontal:
Vertical:

TRY IT

Summary

User interface

Entering data

- Explicitly typing data
- Using interactive elements

Common attributes

- Initial value is in value
- Current value is also in value
- Deactivation is done with disable

Checkboxes

- Independent boolean selection (i.e. selected or not selected)
- Defined with <input type="checkbox"... >
- Property checked determines whether the element is marked
- Tag <label> groups the element with its label

Radio buttons

- Selection of only one option from several possible
- Defined with <input type="radio" name="..." ... >
- Attribute name must be the same for all buttons in a group
- Property checked determines whether the element is marked
- Tag <label> groups the element with its label

Passwords

- Entering hidden text
- Defined with <input type="password"… >

Цветове

- Interactive selection of colour
- Defined with <input type="color"… >
- Colour is hexadecimal and formatted at #rrggbb
- Colour component could be extracted with substring and converted to a number with parseInt(…,16)
- Browser, that do not support this element show text box

Дати

- Interactive selection of a date from a calendar
- Defined with <input type="date"…>
- The value is in text format
- Browser, that do not support this element show text box

Ranges

- Interactive selection of a number in a range
- Defined with <input type="range"…>
- Minimal value in min, maximal value in max, incremental step in step, and current value in value

ICT in SES

The end

Comments, questions