



ICT in SES

Effects and selected topics

Lesson №22

Working with multiple canvases

Multiple canvases



Functionality in Suica

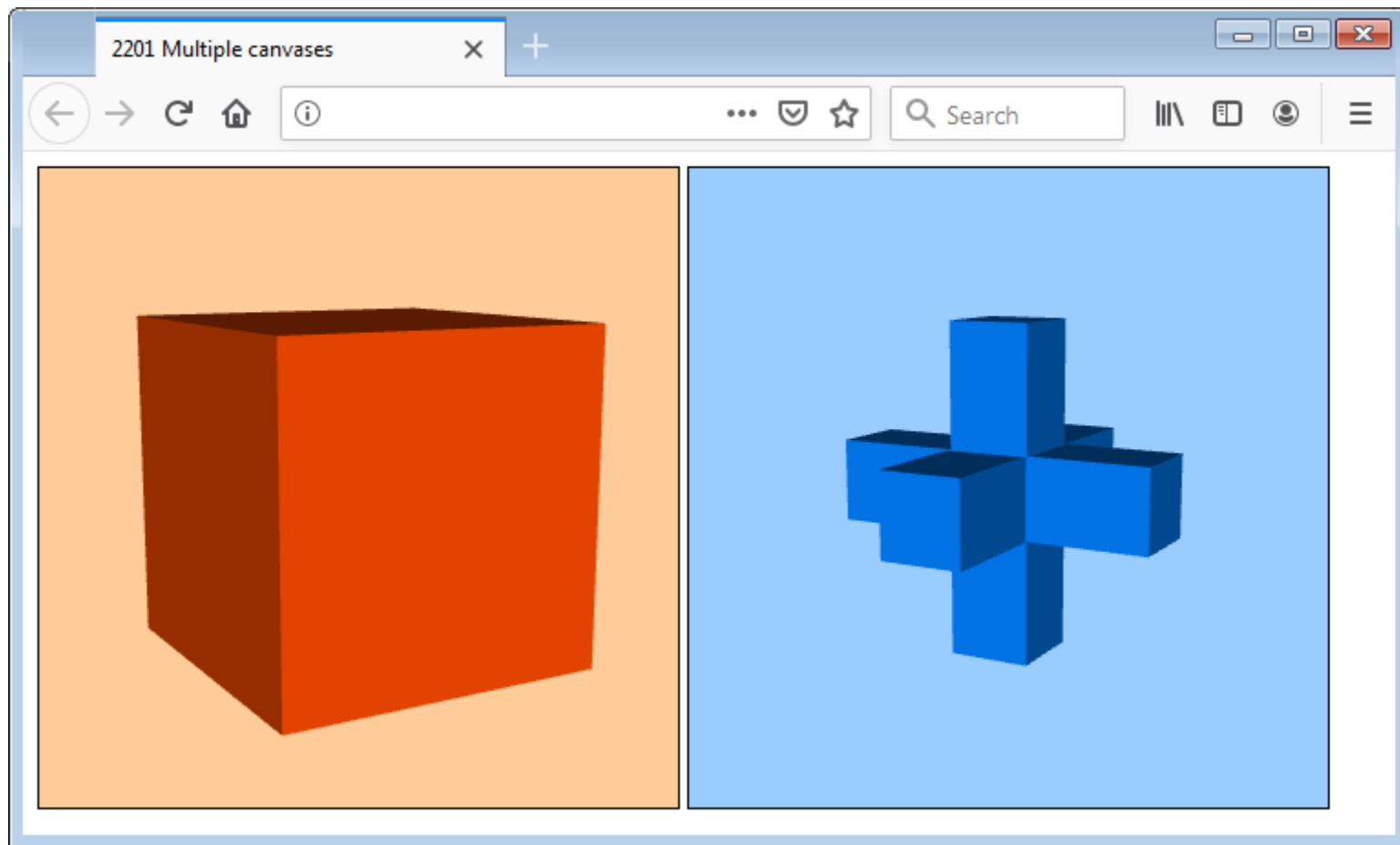
- Each created Suica instance is bound to an HTML **canvas**
- Most of Suica functions work with the last Suica instance
- It is possible to create several instances, but each must be bound to a distinct **canvas**

Implementation

- Two **canvas** elements with 3D objects
- Passing the **id** of the canvas to **new Suica**

```
new Suica('one');  
demo(70,1,0.2);  
background([1,0.8,0.6]);  
cube([0,0,0],20).custom({color:[1,0.3,0]});
```

```
new Suica('two');  
demo(70,-2.5,0.2);  
background([0.6,0.8,1]);  
cuboid([0,0,0],[5,5,20]);  
cuboid([0,0,0],[5,20,5]);  
cuboid([0,0,0],[20,5,5]);
```



TRY IT

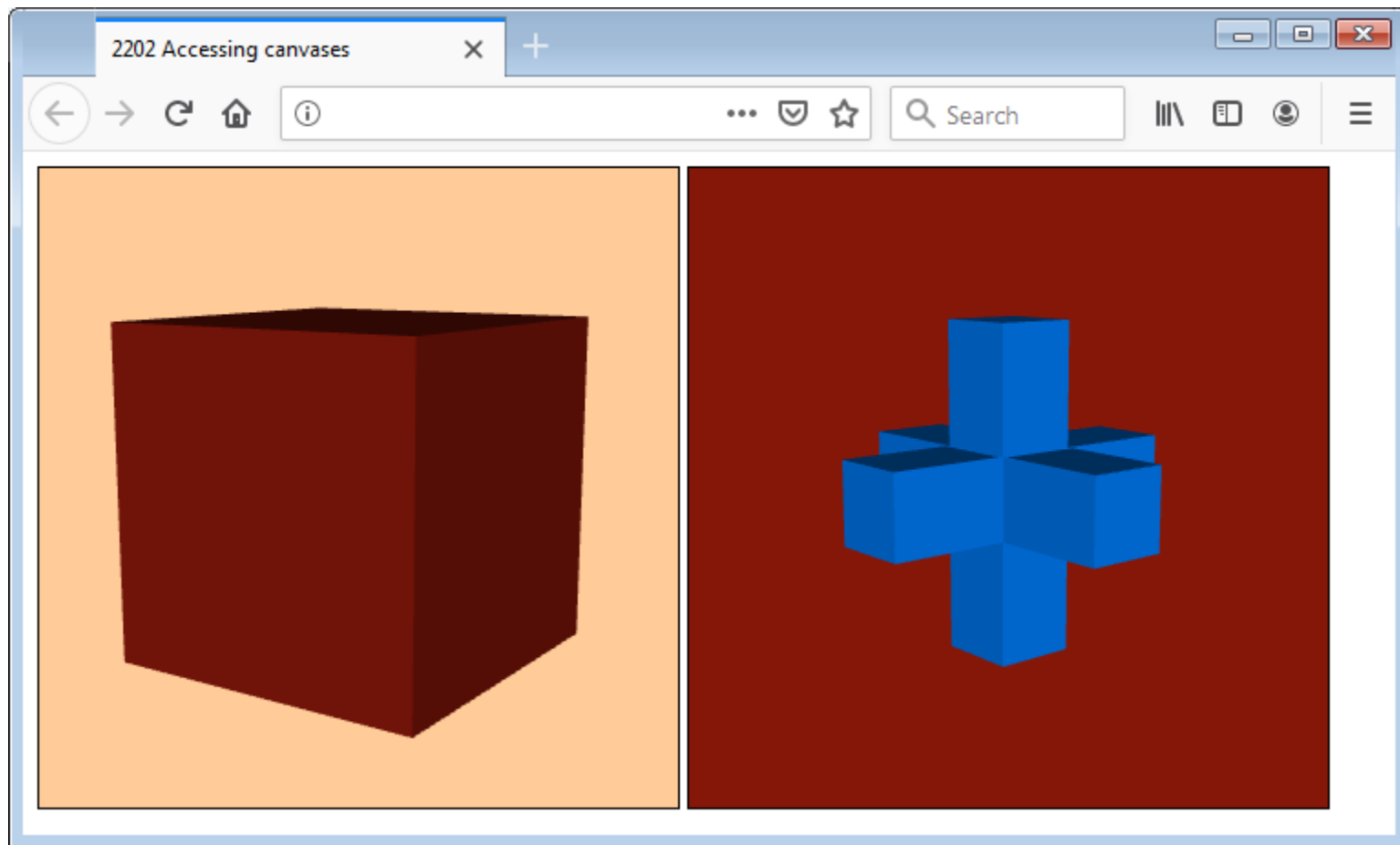
Addressing canvases

- Each **Suica** instance has its own animation cycle
- If a method of a specific instance is needed, the instance must be in a JS variable

```
a = new Suica('one');  
a.nextFrame = animateA;
```

```
b = new Suica('two');  
b.nextFrame = animateB;
```

```
function animateA() {...}  
function animateB() { b.background (...); }
```



TRY IT

HTML overlapping

HTML overlapping



So far

- Drawing the labels to graphical objects

In general

- Any HTML element could overlap over canvas
- This simulates HUD (Heads-up display) – transparent layer with additional information on top of the main window

HUD example

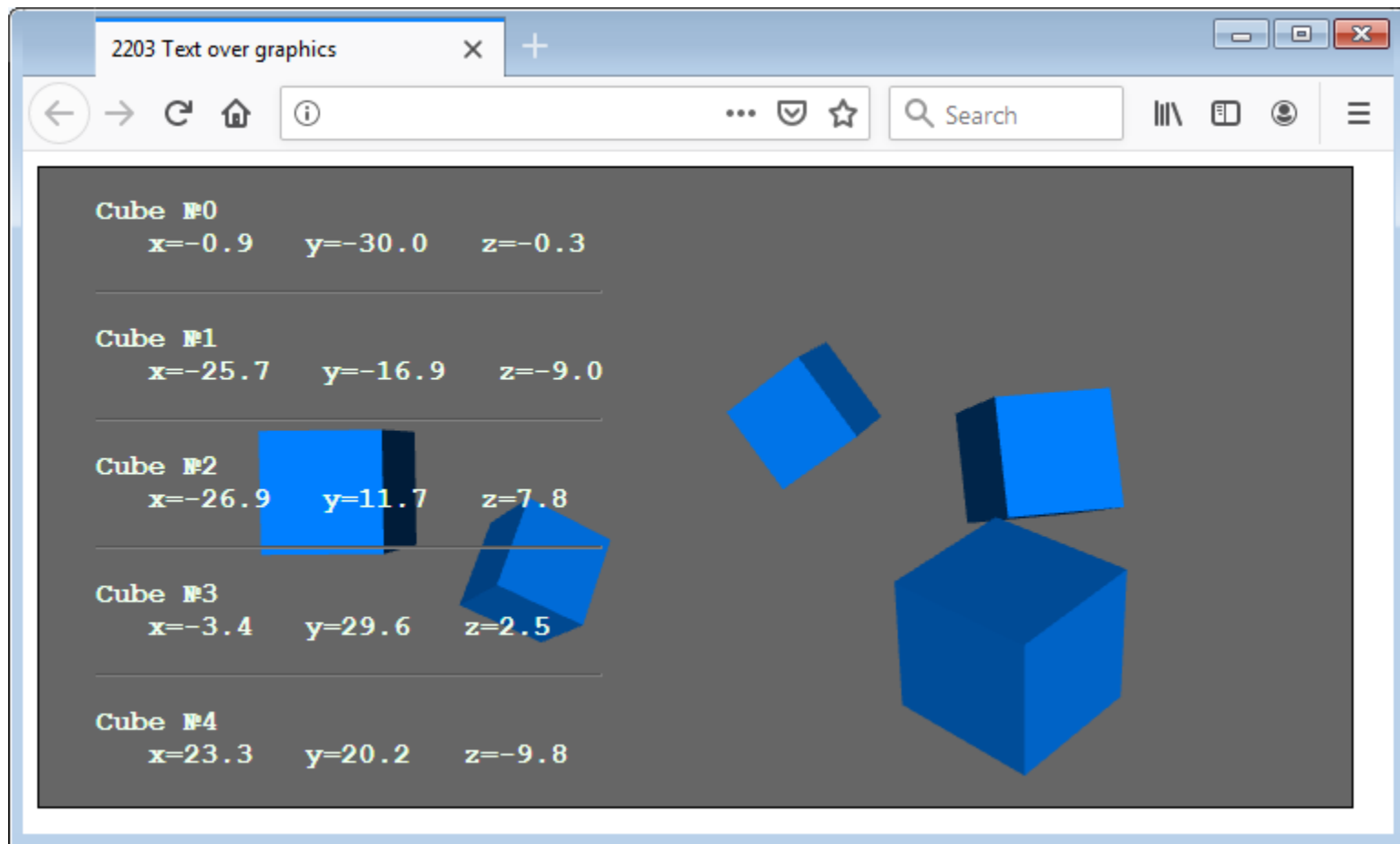
- Showing coordinates of several flying cubes
- Data is shown on top of the canvas
- Defining a style with **position**, **background-color** and **z-index**

```
.hud {  
  position: absolute;  
  background-color: transparent;  
  z-index: 10;  
  color: HoneyDew;  
  font-size: 1em;  
  font-family: 'Courier New', monospace;  
}
```

Using

- Positioning the text element over the graphics with `style.left` and `style.top`
- Changing content via property `innerHTML`

```
hud = document.getElementsByClassName('hud')[0];  
hud.style.left = s.gl.canvas.offsetLeft+32+'px';  
hud.style.top = s.gl.canvas.offsetTop+'px';  
:  
hud.innerHTML = s;
```



TRY IT



Full screen

Variable screen size



Goal

- Canvas occupying the whole browser window

Problems

- Users may change the browser window size
- On mobile devices size changes when device is rotated

Canvas

- The canvas is positioned fixed at (0,0)
- Width and height are set to 100% of the browser window

```
canvas {  
    position: fixed;  
    top: 0;  
    left: 0;  
    width: 100%;  
    height: 100%;  
};
```

Changing the size

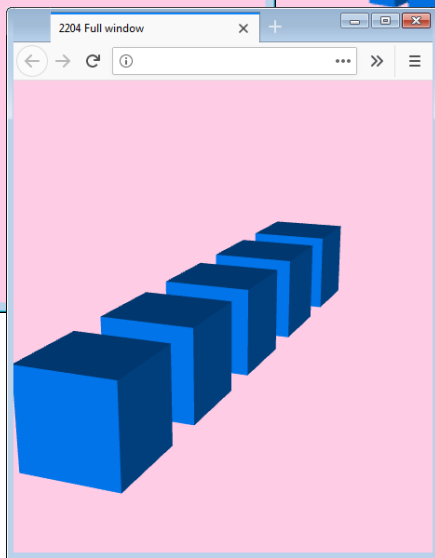
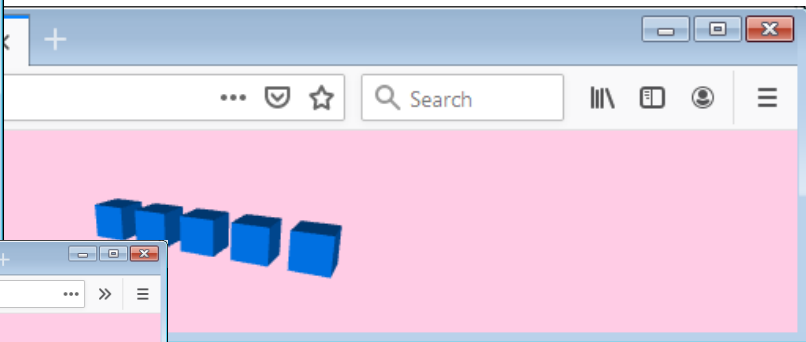
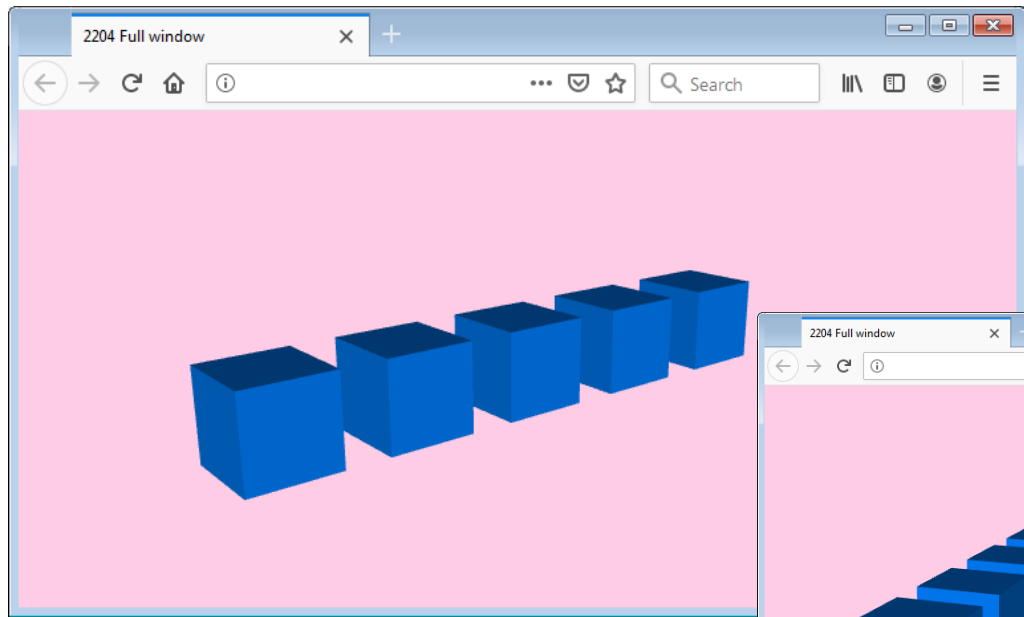
- Triggers event **resize** of the window object
- Adding event listener after Suica is activated
- Launching the event listener manually to fix the initial size of the canvas

```
s = new Suica();  
window.addEventListener('resize', onResize, false);  
onResize();  
  
function onResize(event)  
{  
    if (s) {...}  
}
```


Event listener

- Accessed canvas via `gl.canvas`
- Changes size to match the internal window size stored in `window.innerWidth` and `window.innerHeight`
- Resets the view port with `gl.viewport`
- Reapplies the perspective projection

```
s.gl.canvas.width = window.innerWidth;  
s.gl.canvas.height = window.innerHeight;  
  
s.gl.viewport(0,0,window.innerWidth,  
              window.innerHeight);  
perspective(30,1,40000);
```



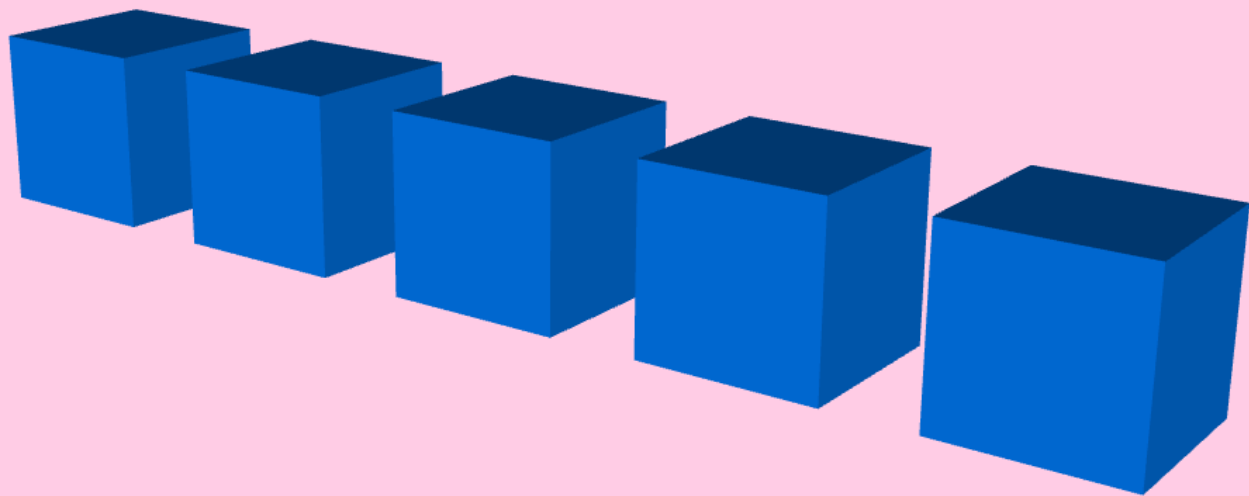
TRY IT

Full screen (not window)



Easy alternative

- Browsers support full screen switching
- Usually it is with F11 (on Windows machines)
- If the canvas is full window, then in full screen mode it will occupy the whole screen



TRY IT



Touch screens

Touching



Differences

- Mouse generates single interaction point
- Touching may generate multiple interaction points

Concept

- Events for touch screens are separate DOM events
- Sometimes it is necessary to process mouse events, keyboard events and touch events

Events



Touching events

- **touchstart** – activated when the user touches an HTML element and creates interaction point
- **touchend** – activated when the interaction point exits the HTML element or the screen edge

More events

- **touchmove** – activated when the user moves the interaction points on the touch screen
- **touchcancel** – activated when the interaction point disappears:
 - Moving outside the window
 - Showing important popup window
 - Entering the graphical area of a plug-in

Properties



Properties of touch events

- **touches** – array of all interaction points
- **changedTouches** – array of interaction points that changed since the last touch event
- The number of points in these lists are in their **length** property (e.g. `event.touches.length`)

Sound and sound effects

Sound effects



Types

- Background music – continuous music, relatively long
- Sound effects – short sounds, related to events

Example

- Bouncing balls
- Background music during bouncing
- Short sound effect when a ball hits the ground
- Sound effects overlap the music

Background music



Tag <audio>

- HTML5 for background music is **audio**
- Attribute **controls** define visibility of audio buttons
- Tag **source** point to the audio file

```
<audio id="music" controls>  
  <source src="Arktype_-_Hot_Pursuit.mp3"  
                                     type="audio/mpeg">  
  No music!  
</audio>
```

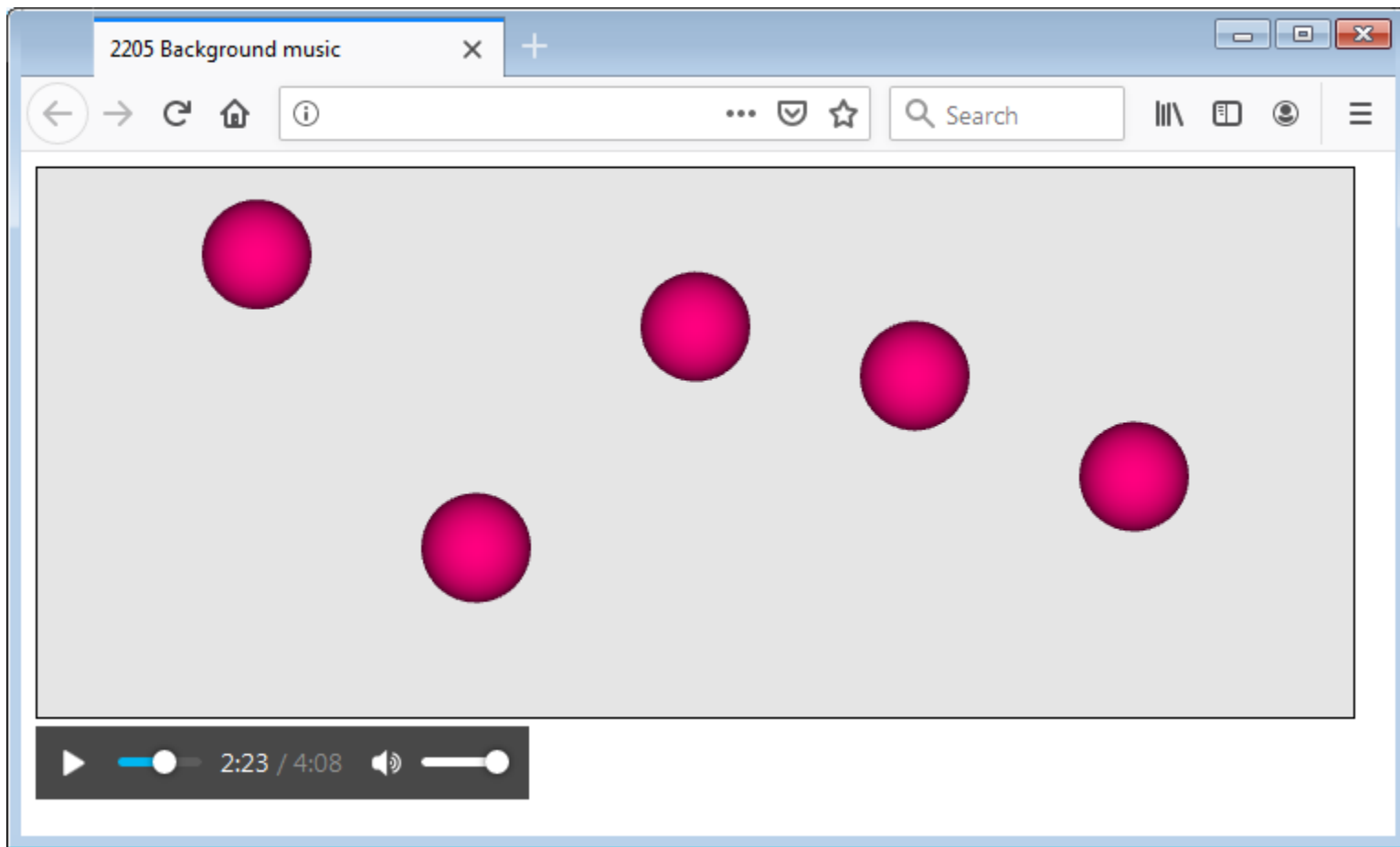
Methods

- Element **audio** support methods to manage audio
- Starting playing is done with **play**

Note

- One **audio** may have several **source**, pointing to different format of the same audio file, the browser will select the most appropriate file format

```
function main()  
{  
    :  
    document.getElementById('music').play();  
}
```



TRY IT

Music: Hot Pursuit by Arktype, license CC BY-NC-SA from 2011
<https://www.jamendo.com/en/track/856963/hot-pursuit>

Sound effects



Main problem

- Tag **audio** is suitable for long continuous music
- It cannot be played before the end of the previous playing

Solution

- Several instances of the same sound effect
- They can be played together when needed

Problem

- Add sound effect for bouncing
- Activated when any of the balls hits the ground

Overlapping sound effects

- Several instances
- Stored in array `hitEffect`

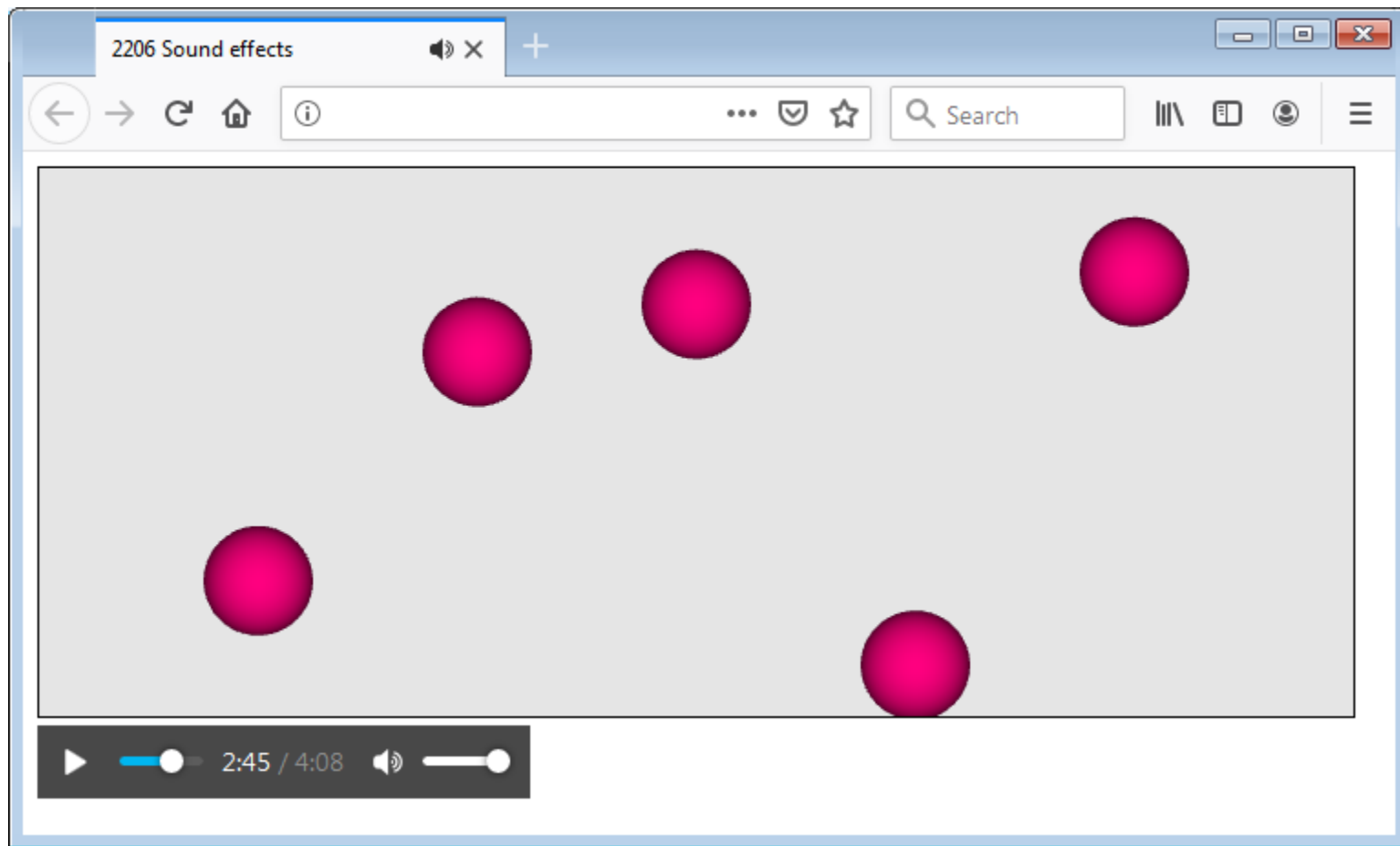
```
hits = 0;
hitEffect = [];

for (var i=0; i<n; i++)
    hitEffect[i] = new Audio('246281...sion2.wav');
```


Bounce

- Every ball moves in an absolute value of a sine curve $|\sin(q)|$
- There is a hit when $\sin(q)$ changes its sign
- The value of **hits** determines which instance of the sound effect to activate

```
a[i].center[1] = -175+a[i].radius+250*abs(sin(q));  
var sign = Math.sign(sin(q));  
if (a[i].sign != sign)  
{  
    a[i].sign = sign;  
    hitEffect[hits%n].play();  
    hits++;  
}
```



TRY IT

A Guitar Chord Hit Percussion by afleetingsspeck, license CC 0 from 2014
<https://www.freesound.org/people/afleetingsspeck/sounds/246281/>

Selected topics

Selected topics



Demonstrations of topics, selected by the students

- Additional graphical effects
- Creating other objects
- Complex objects and motions



Summary

Summary



Canvas

- Could be several in a page
- Each has its own instance of **Suica**
- Each has its own animation loop with **nextFrame**

Canvas and other HTML elements

- HTML elements could be on top of the canvas
- This includes text (with transparent or non-transparent background), buttons, text boxes, etc.

Full window and full screen

- Change of window size is captured by the **resize** event, the canvas must be adjusted to the new size

Touch screens

- There are events which are activated then the screen is touched
- The objects for these events contain array of interactive points

Background music and sound effects

- Used with tags **audio** and **sound**, activated with **init**
- Could be overlapped to create more vibrant sound



ICT in SES

The end

Comments, questions