

**ИКТ В НОС**

# **Ефекти и избрани теми**

Тема №22

# Работа с няколко графики

# Няколко графични полета

---



## Функционалност в СУИКА

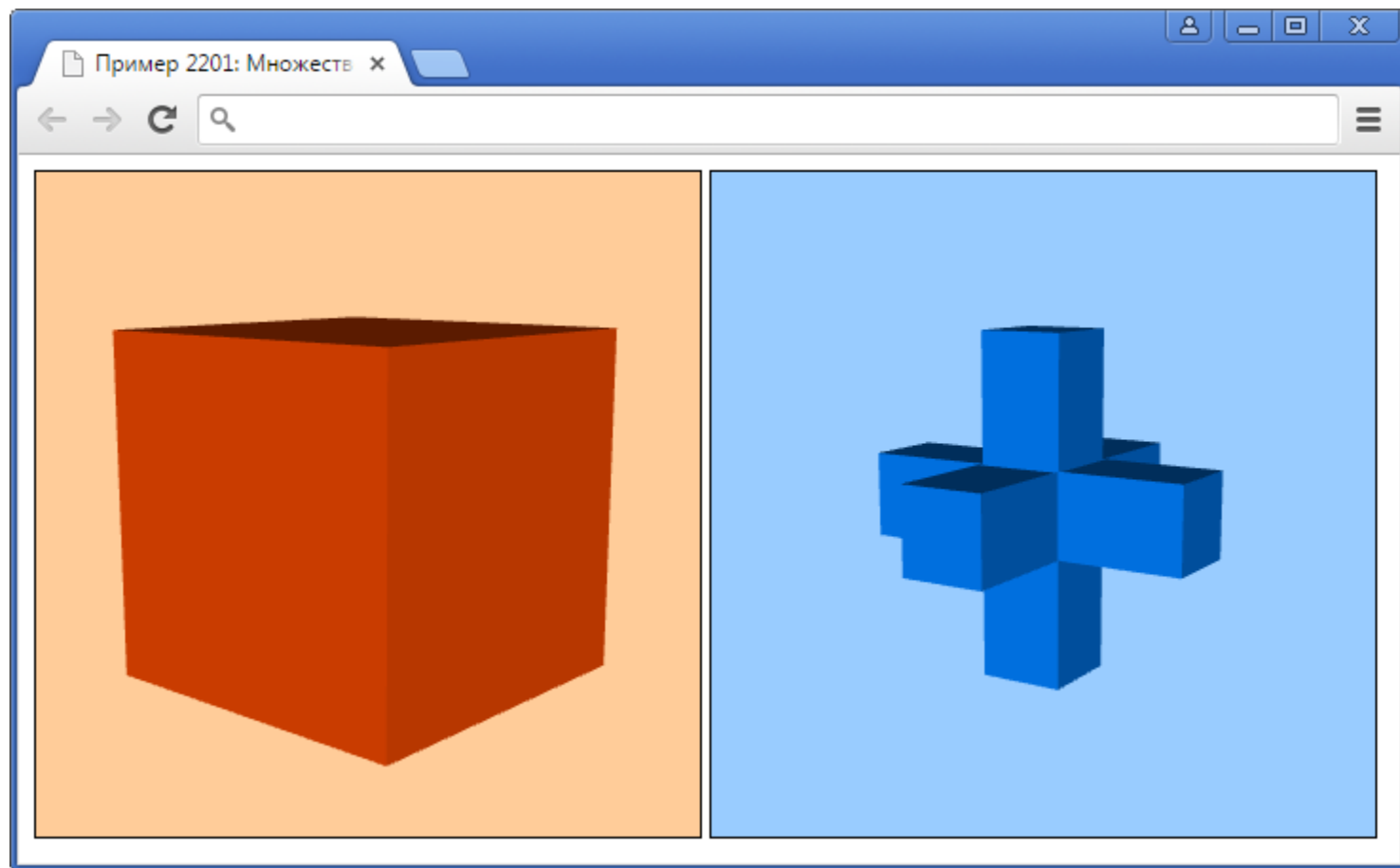
- При създаването на инстанция на Suica тя се свързва с конкретен HTML елемент от тип **canvas**
- Голяма част от функциите работят с последно създадената инстанция на Suica
- Има възможност да се създадат няколко инстанции, но всяка трябва да е свързана към различен **canvas**

## Примерна реализация

- Два **canvas** елемента с някакви 3D обекти в тях
- На **new Suica** подаваме **id** на елемента canvas

```
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]);
```



ПРОБА

# Адресиране на графично поле

- Всяка инстанция на **Suica** има собствен цикъл за анимация
- Ако трябва да се използва метод на конкретна инстанция, то тя трябва да е записана в променлива

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

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

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

# HTML наслагване

# HTML наслагване

---



## Използване до момента

- Рисуване на имена на графични обекти

## По принцип

- Могат да се използват всички HTML елементи като са насложени върху графично поле
- Така се симулира HUD (Heads-up display) – прозрачен слой с допълнителна информация над основния екран



## Пример с HUD

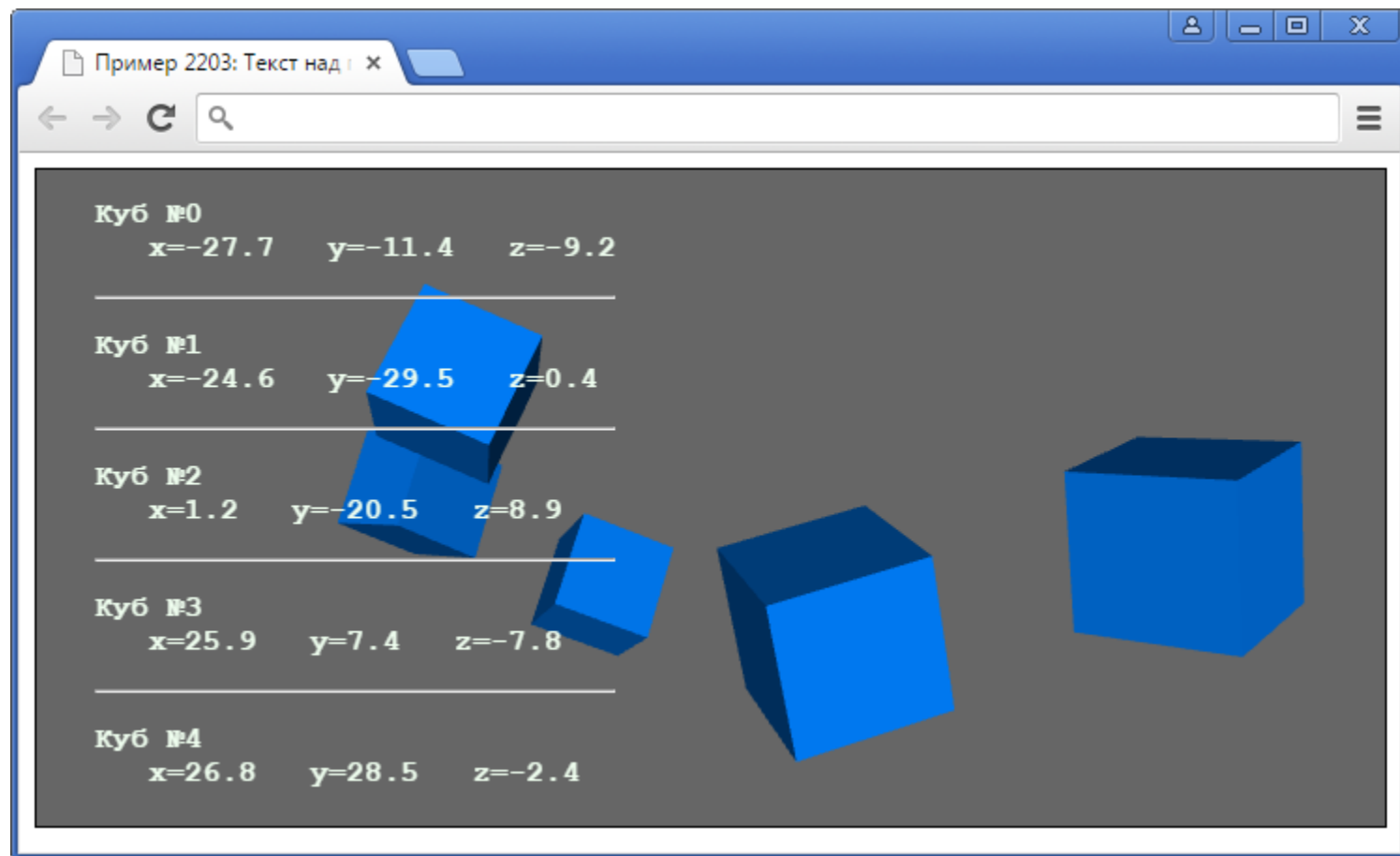
- Няколко куба летят и им се изписват координатите
- Те се изписват над графичното поле
- Аналогично на етикетите, дефинираме си стил, като най-важните свойства са **position**, **background-color** и **z-index**

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

## Използване

- Позиционираме текстовият елемент да е над желаното място над графиката със **style.left** и **style.top**
- Променяме съдържанието чрез **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;
```



ПРОБА

**Работа на цял прозорец**

# Променлив размер

---



## Желана ситуация

- Графичното поле заема целия прозорец на браузъра

## Проблеми

- Потребителят може да промени размера на прозореца
- Ако се ползва мобилно устройство потребителят може да го завърти и размерите на прозореца да се сменят

## Стил на canvas

- Графичното поле трябва да е позиционирано фиксирано на координати (0,0)
- Ширината и височината трябва да заемат на 100% прозореца на браузъра

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

## Промяна на размера

- Това е събитие **resize** към обекта window
- След създаването на Suica добавяме слушател на събитието
- Понеже canvas вече е бил променен преди създаването на слушателя, изпълняваме слушателя „ръчно“

```
s = new Suica();
window.addEventListener('resize', onResize, false);
onResize();

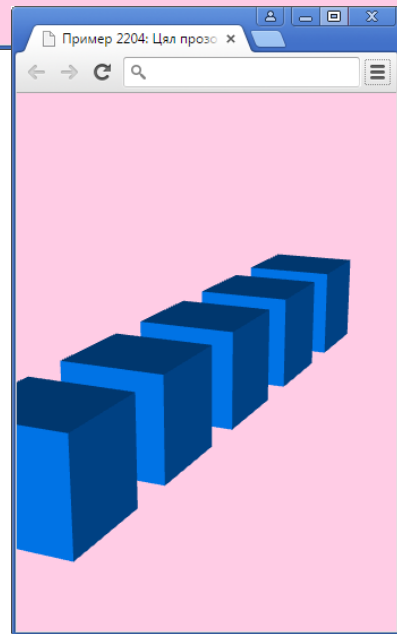
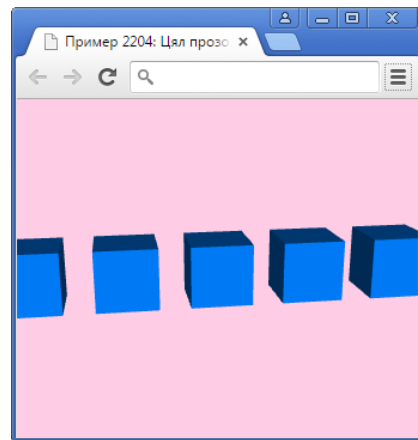
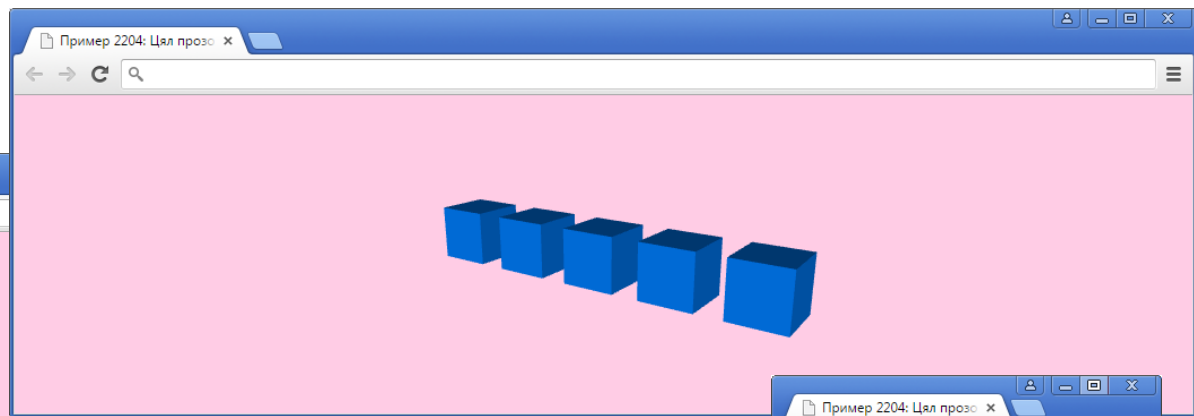
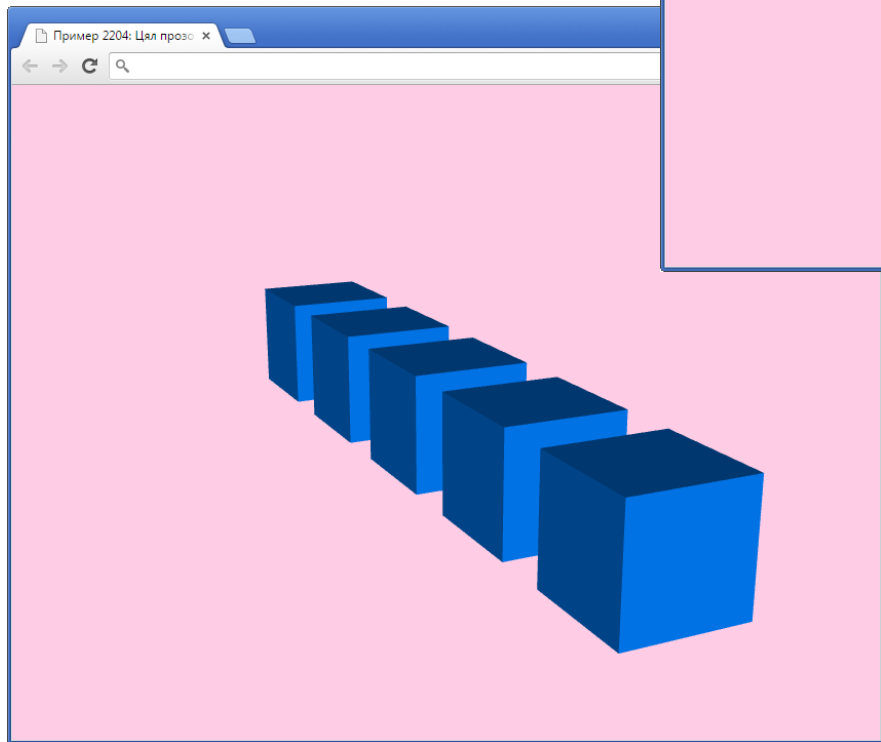
function onResize(event)
{
    if (s) {...}
}
```

## Самият слушател

- Достига до canvas елемента през свойството `gl.canvas`
- Променя размера да съответства на вътрешния размер на прозореца – `window.innerWidth` и `window.innerHeight`
- Дефинира видимата зона да е цялата с `gl.viewport`
- Наново указва перспективата

```
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);
```





ПРОБА

# Работа на цял екран

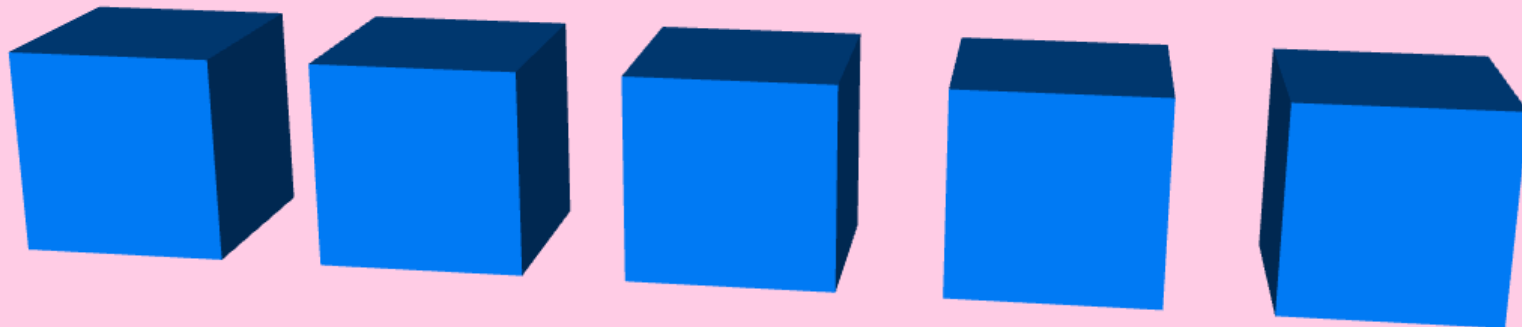
---



## Лесна алтернатива

- Браузърите имат опция за превключване на цял екран
- Например, в Chrome се прави с F11
- Ако графичното поле заема целия прозорец, след влизане в режим на цял екран то ще заема целия екран

You have gone full screen. [Exit full screen \(F11\)](#)



ПРОБА

# Работа с докосване

# Работа с докосване

---



## Разлика

- При работа с мишка точката на докосване е една
- При докосване с пръсти има няколко точки на докосване

## Концепция

- Събитията за работа със сензорни екрани са отделни събития в DOM
- Понякога е нужно да се обработват събитията както на мишката, така и на клавиатурата и на сензорния екран



## Събития при работа с докосване

- **touchstart** – активира се, когато потребителят докосне съответният HTML елемент и с това създава точка на докосване в него
- **touchend** – активира се, когато точката на докосване излезе извън съответният HTML елемент или когато потребителят плъзне пръст извън ръба на екрана

# Събития при работа с докосване

- **touchmove** – активира се, когато потребителят премества точка на докосване по повърхността на сензорният екран
- **touchcancel** – активира се, когато точката на докосване престане да съществува; това включва:

Преместването ѝ извън прозореца на браузъра

Появяване на изскачащ прозорец с важно съобщение

Навлизване в графичното поле на приставка (plugin)



## Свойства на touch-събитията

- **touches** – масив на всички текущи точки на докосване
- **changedTouches** – масив от индивидуалните точки на докосвания, които са се променили от последното touch-събитие
- Броят на точките на докосване в тези списъци се получава от свойството им **length** (напр. `event.touches.length`)



# **Звук и звукові ефекти**

# Звукови ефекти

---



## Видове

- Фонова музика – постоянна музика, относително дълга
- Звукови ефекти – кратки звукове, свързани със събития

## Примерен модел

- Топащи се топчета
- Фонова музика по време на топането
- При удар на топче в земята се чува кратък звуков ефект
- Звуковият ефект ще се наслагва над музиката

# Фонова музика



## Таг <audio>

- За фонова музика ще ползваме HTML5 таг **audio**
- Атрибутът **controls** на **audio** указва да се виждат бутоните
- Тагът **source** описва къде е файла и какъв тип е

```
<audio id="music" controls>  
  <source src="Arktype_-_Hot_Pursuit.mp3"  
                                         type="audio/mpeg">
```

Искаме музика, но няма!  
</audio>

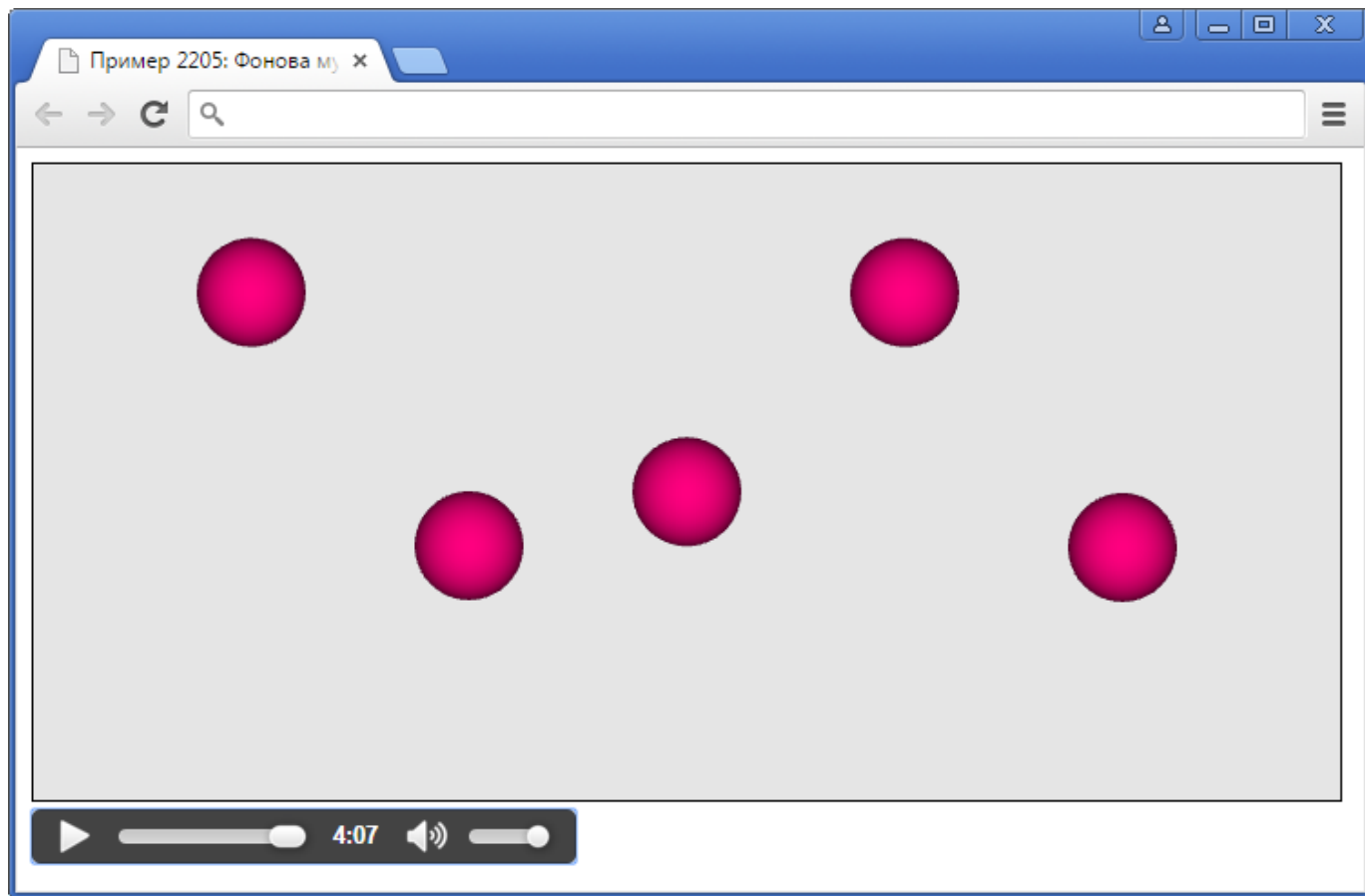
# Методи

- Елементът **audio** има методи за работа с него
- За пускане на музиката извикваме метода **play**

## Забележка

- В един **audio** може да има няколко тага **source**, сочещи различни формати на един и същ музикален файл, браузърът ще подбере първия формат, който се поддържа

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



ПРОБА

# Звукови ефекти

---



## Основен проблем

- Тагът **audio** е удобен за дълга и непрекъснатата музика
- Не може да се просвирва преди да е свършило предходното просвирване

## Идея за звуковите ефекти

- Няколко инстанции на един и същ звуков ефект
- При пускането те се редуват

## Задача

- Да добавим звуков ефект на удар
- Активира се когато някое от топчетата отскочи

## Застъпващи се звукови ефекти

- Ще имаме няколко инстанции на звуковия ефект
- Зареждаме ги в масива `hitEffect`

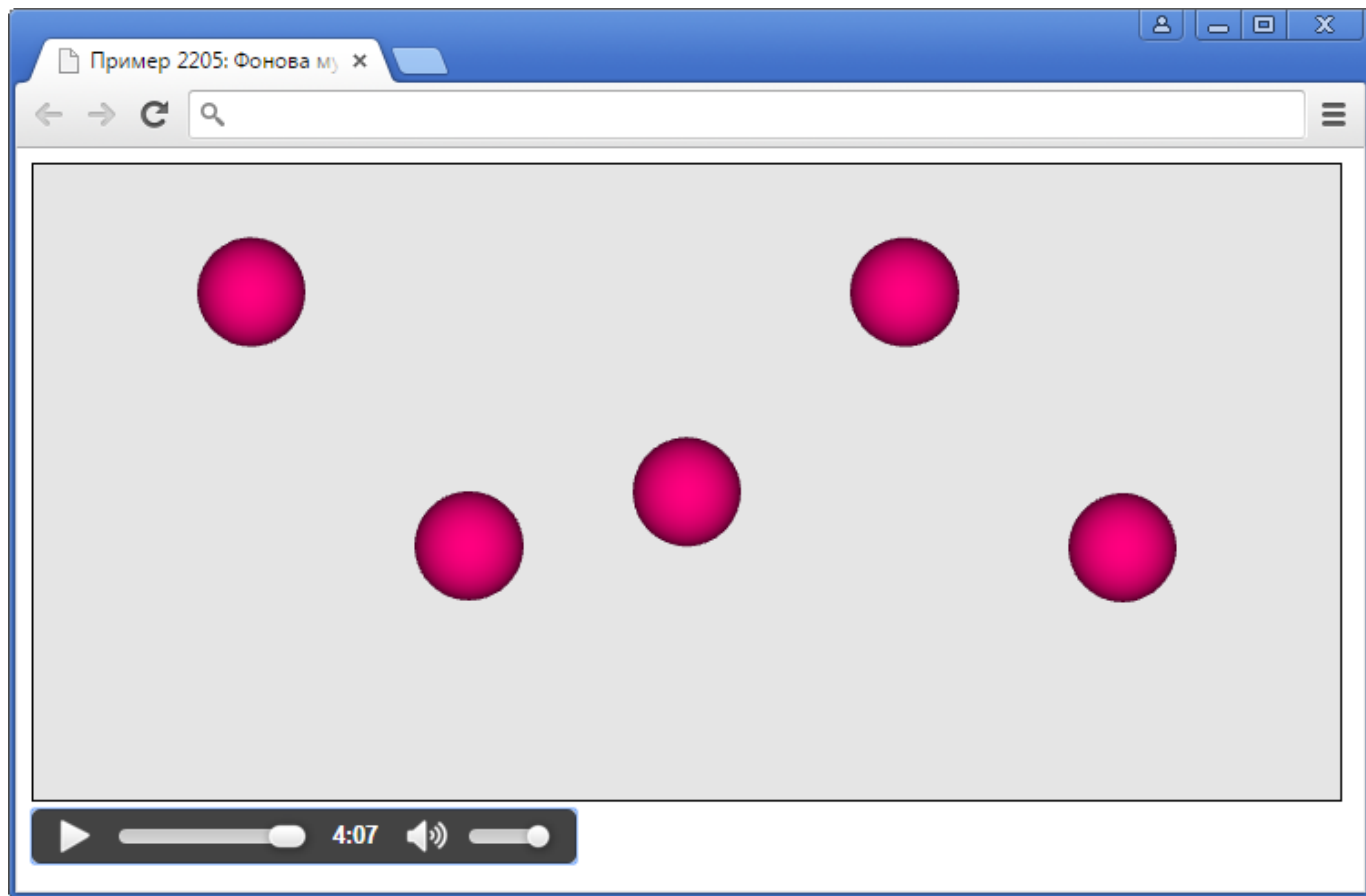
```
hits = 0;  
hitEffect = [];  
  
for (var i=0; i<n; i++)  
    hitEffect[i] = new Audio('246281...sion2.wav');
```

# Удар

- Всяко топче се движи по синусоида, но в модул
- Удар има когато синусоидата, без модул, си смени знака
- При засичане на удари редуваме кой от ефектите да пуснем
- Променливата **hits** е брояч и реализира редуването

```
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++;  
}
```





ПРОБА

# **Избрани теми**

# Избрани теми

---



## Демонстрации по теми, избрани от студентите

- Допълнителни графични ефекти
- Създаване на други обекти
- Сложни обекти и/или движения

# Обобщение



## Графичните полета

- Могат да са няколко на страница
- Всяко си е със собствена инстанция на **Suica**
- Всяко си има собствен анимационен цикъл **nextFrame**

## Връзка с HTML

- HTML елементи могат да се поставят над графично поле
- Това включва текст (с прозрачен или непрозрачен фон), бутони, полета за въвеждане на данни и т.н.

## Цял прозорец

- При работа с цял прозорец се улавя събитието **resize** и се настройва размера на графичното поле според новия размер

## Работа със сензорни екрани

- Има събития, които се активират при докосване или движение върху сензорен екран
- Обектите на тези събития поддържат списък на активните точки на докосване

## Фонова музика и звукови ефекти

- Включват се с тагове **audio** и **sound**, активират се с метод **init**
- За наслагване се използват по няколко инстанции



# ИКТ В НОС

**Край**

Коментари, въпроси