ИКТ в НОС

Потребителски интерфейс

Тема №19

Въвеждане на данни

Въвеждане на данни

Явно въвеждане

- Чрез изписване в текстови полета
- Чрез избор на конкретни стойности от списъци

Неявно въвеждане

- Чрез движение на мишката
- Чрез промяна на интерактивни контроли

Въвеждане чрез изписване

Основна идея

- Нужни са ни числови данни
- Въвеждат се през текстови полета

Демонстрационен пример

- Създаване на окръжности една по една
- Въвеждат се координати на център и радиус
- Натиска се бутон за създаване

Стилове

- Елементите за въвеждане са отделени в блок
- Блокът е с клас panel и е позициониран в горния-ляв ъгъл на графичното поле център на обекта

```
.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;
}
```

Елементи

- За всяка окръжност се въвеждат две координати и радиус
- Въвеждането става в три текстови полета с различни id
- Текстовите полета се ползват често, затова са създадени и променливите elemX, elemY и elemR.

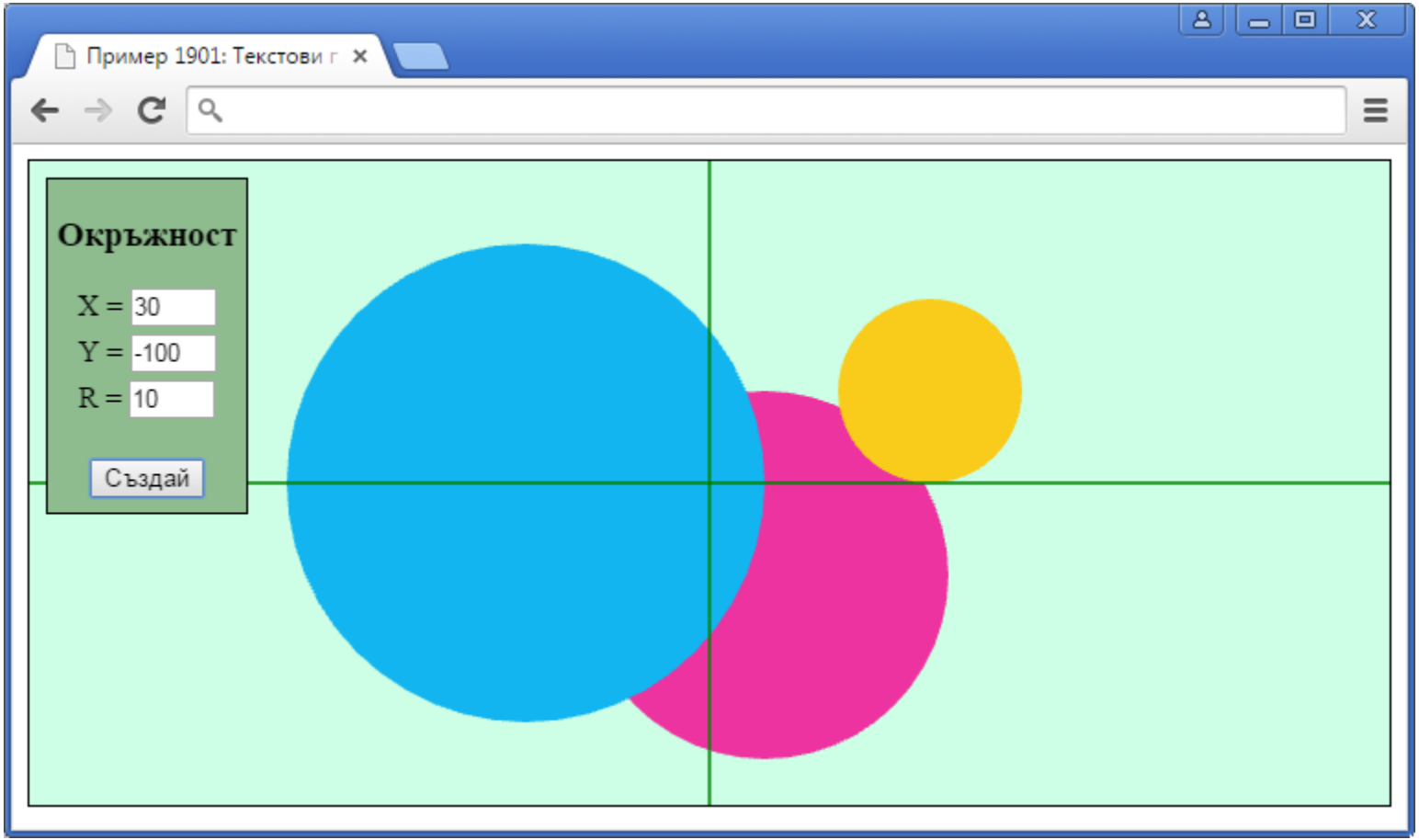
```
<div class="panel">
   <div>X = <input type="text" id="x" ...></div>
   <div>Y = <input type="text" id="y" ...></div>
   <div>R = <input type="text" id="r" ...></div>
</div>
```

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

Създаване на окръжности

- Въведените стойности се извличат през свойството value
- След създаването, текстовите полета се изчистват

```
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 = '';
}
```

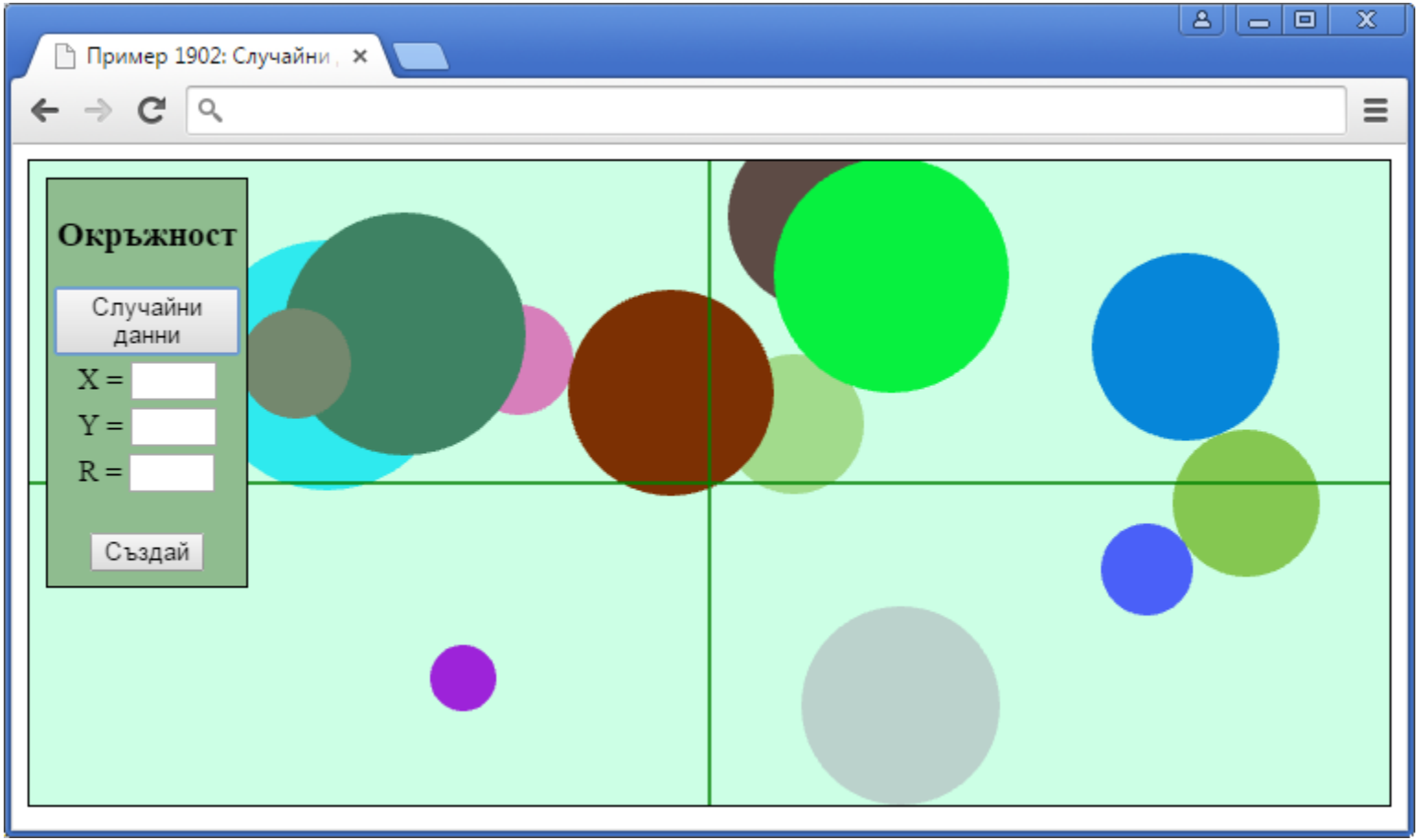


ПРОБА

Нов бутон

- Попълва полетата със случайни стойности
- Имаме възможност да фиксираме диапазона на случайността
- Закръглянето се прави от естетически причини

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



ПРОБА

Нова функционалност

Избиране на окръжност

- При кликване върху нарисувана окръжност около нея се рисува мигащ ореол
- Данните ѝ се показват в текстовите полета
- Докато има избрана окръжност, бутоните за случайни данни и за създаване на окръжност са недостъпни

Забравяне на избора

- При кликване извън окръжност се забравя, че има избрана
- Двата бутона стават достъпни

Мигащ контур

- Реализиран като черна окръжност
- Ако има избран обект, контурът мига през 0.25 секунди

```
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;
   }
}
```

Натискане на бутон върху окръжност

- Показваме контура с центъра и радиуса от окръжността
- Показваме нейните данни в текстовите полета
- Правим двата бутона неактивни

```
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;
```

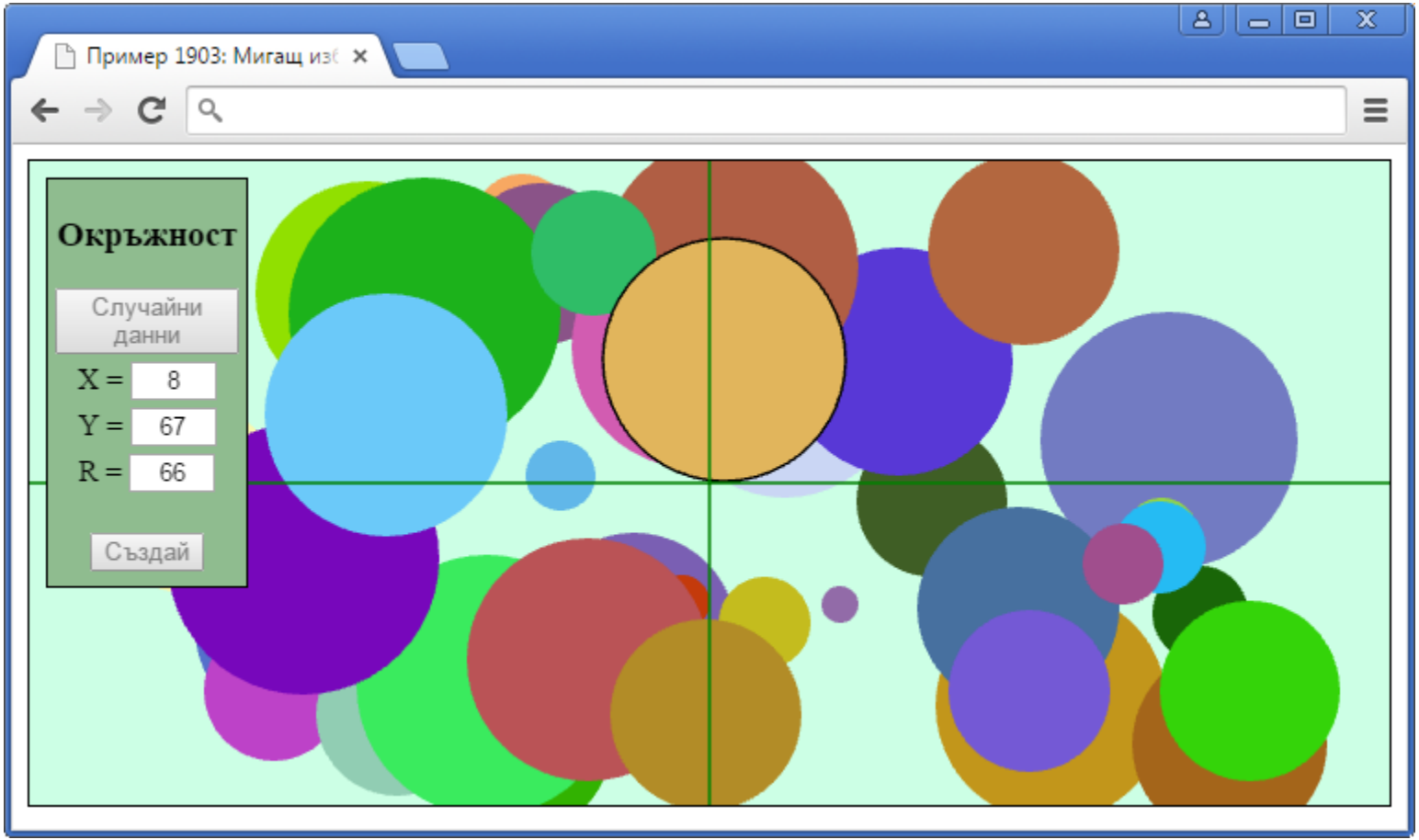
Натискане на бутон извън окръжност

- Скриваме контура
- Изчистваме текстовите полета
- Правим двата бутона отново активни

```
blinker.visible = false;

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

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


Пример 1903: Мигащ из
Окръжност
Случайни данни
X = 8
Y = 67
R = 66
Създай

ПРОБА

Последна нова функционалност

Когато е избрана окръжност

- Бутонът [Създай] става бутон [Промени]
- Ако се променят данните в текстовите полета и се натисне [Промени], избраната окръжност се актуализира

Идея

- Вместо два бутона и показване само на единия, ще ползваме един бутон със сменяемо име

Смяна на името

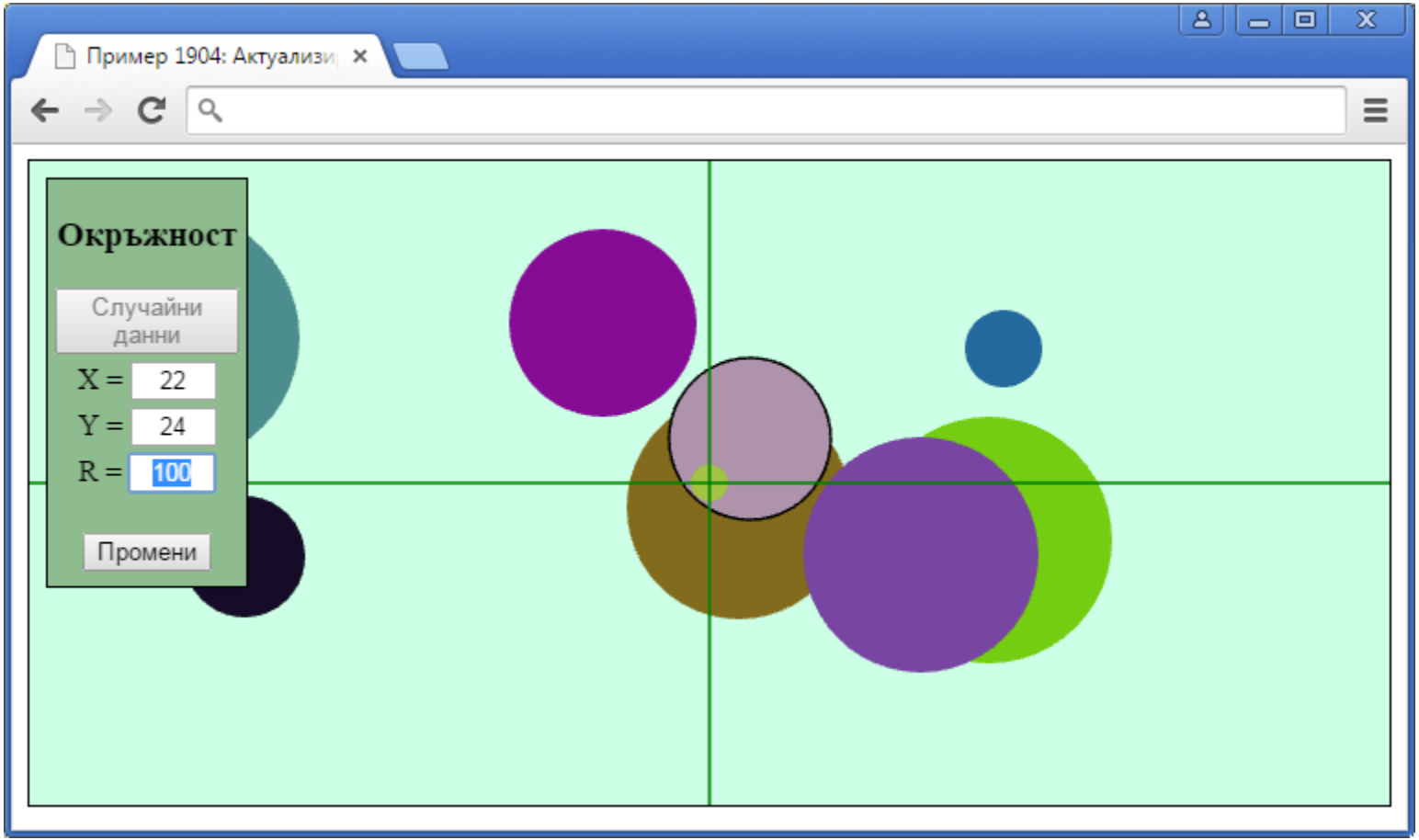
- Според това дали при кликване избираме обект или не

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

Действие на бутона

- При избрана окръжност актуализираме центъра и радиуса ѝ
- Актуализираме и мигащия контур

```
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)...}
}
```


Пример 1904: Актуализи
Окръжност
Случайни данни
X = 22
Y = 24
R = 100
Промени

ПРОБА

Интерактивни контроли

Интерактивни контроли

Роля

- Позволяват алтернативен начин за дефиниране на стойности
- Може да не са достъпни за всеки браузър

Някои видове контроли

- Чекбоксове и радио бутони
- Текстови полета за пароли
- Избор на цвят
- Избор на дата
- Избор от диапазон

Чекбоксове и радио бутони

Прилики и разлики

- Чекбоксове – позволяват избор на няколко елемента
- Радио бутони – позволяват избор на един елемент

Изобразяване

- Зависи от браузъра

Пример

- Сцена с три движения: хоризонтално, вертикално, радиално
- Всяко движение се пуска и спира с чекбокс
- Първоначално само хоризонталното движение е пуснато

Дефиниране на контролите

- Самите чекбоксове са с таг input и атрибут type със стойност checkbox
- Чекбоксовете заедно с етикета към тях са обединени с таг label, за да може да се кликва и върху етикета

```
<label><input type="checkbox" id="hor">Хоризонтално</label>
<label><input type="checkbox" id="ver">Вертикално</label>
<label><input type="checkbox" id="rad">Радиално</label>
```

Инициализация

- За удобство запомняме DOM елементите на трите чекбокса
- Променяме състоянието на чекбокса за хоризонталното движение чрез свойството checked

Алтернатива

- Можеше в HTML кода да добавим атрибут checked, за да стане чекбокса автоматично маркиран

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

hor.checked = true;
```

Реализация на движението

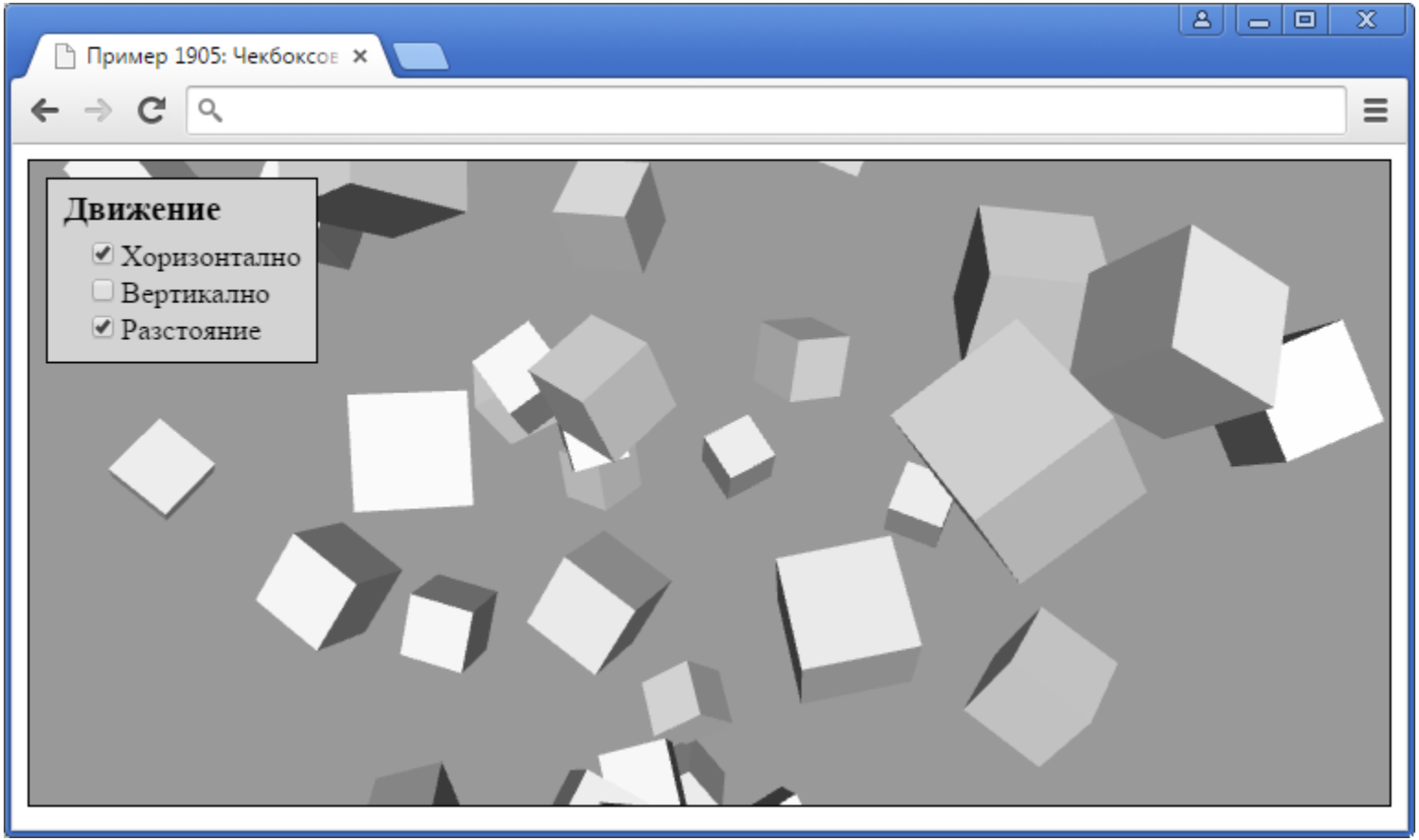
- В три променливи aHor, aVer и aRad се помнят три ъгъла
- Изминалото време от последния кадър dT се прибавя към тези ъгли, чиите съответни чекбоксове са маркирани

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

Изчисляване на гледната точка

- Позицията ѝ е по сфера, като хоризонтално се движи в една посока, а вертикално и радиално се движи синусоидно
- Хоризонталният ъгъл h съответства на aHor
- Вертикалният ъгъл v се мени от -1 до 1 радиана според aVer
- Радиалното отместване r се мени от 100 до 300 според 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] );
```


Пример 1905: Чекбоксов
Движение
Хоризонтално
Вертикално
Разстояние

ПРОБА

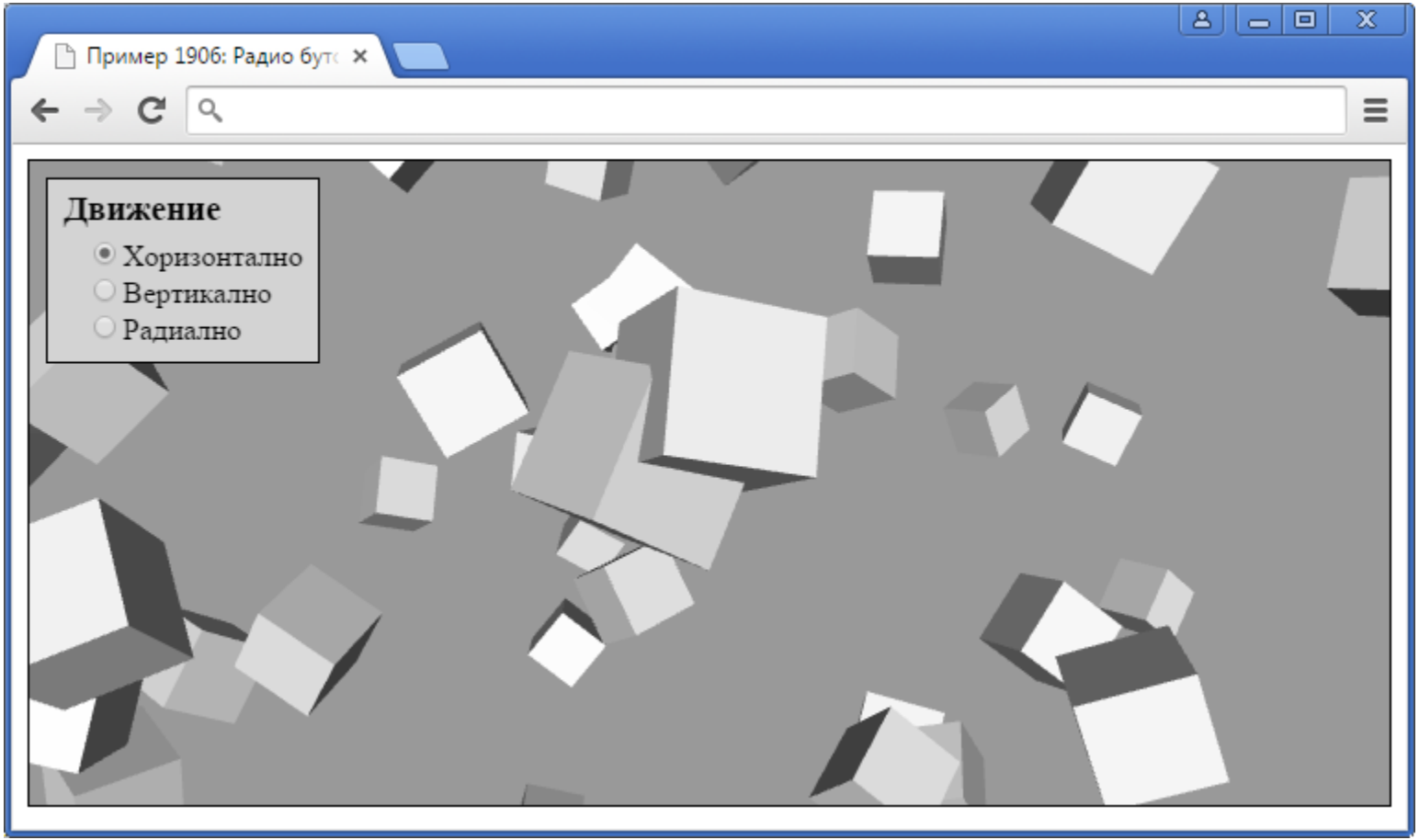
Същият пример, но с радио бутони

- Всяко движение се пуска и спира с радио бутон
- Само едно от движенията може да е активно

Дефиниране на контролите

- Отново с таг input, но с атрибут type със стойност radio
- Отново вложени в таг label
- Допълнителен атрибут name, който за трите радио бутона трябва да има една и съща стойност – така те се третират като една група от взаимно изключващи се радио бутони

```
<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>
```


Пример 1906: Радио бут
Движение
Хоризонтално
Вертикално
Радиално

ПРОБА

Пароли

Текстови полета за пароли

- Броят показвани обекти се въвежда като парола – вместо текст, се показват звездички
- Ползва се елемент input с type със стойност password
- Самото показване се прави във функцията doIt

```
<h3>Движение</h3>
<p>Въведете тайно броя кубове, число от 0 до 99,
   които да се показват</p>
Брой: <input type="password" id="num" size="2">
<button onclick="doIt()">Покажи</button>
```

Реализация на doIt

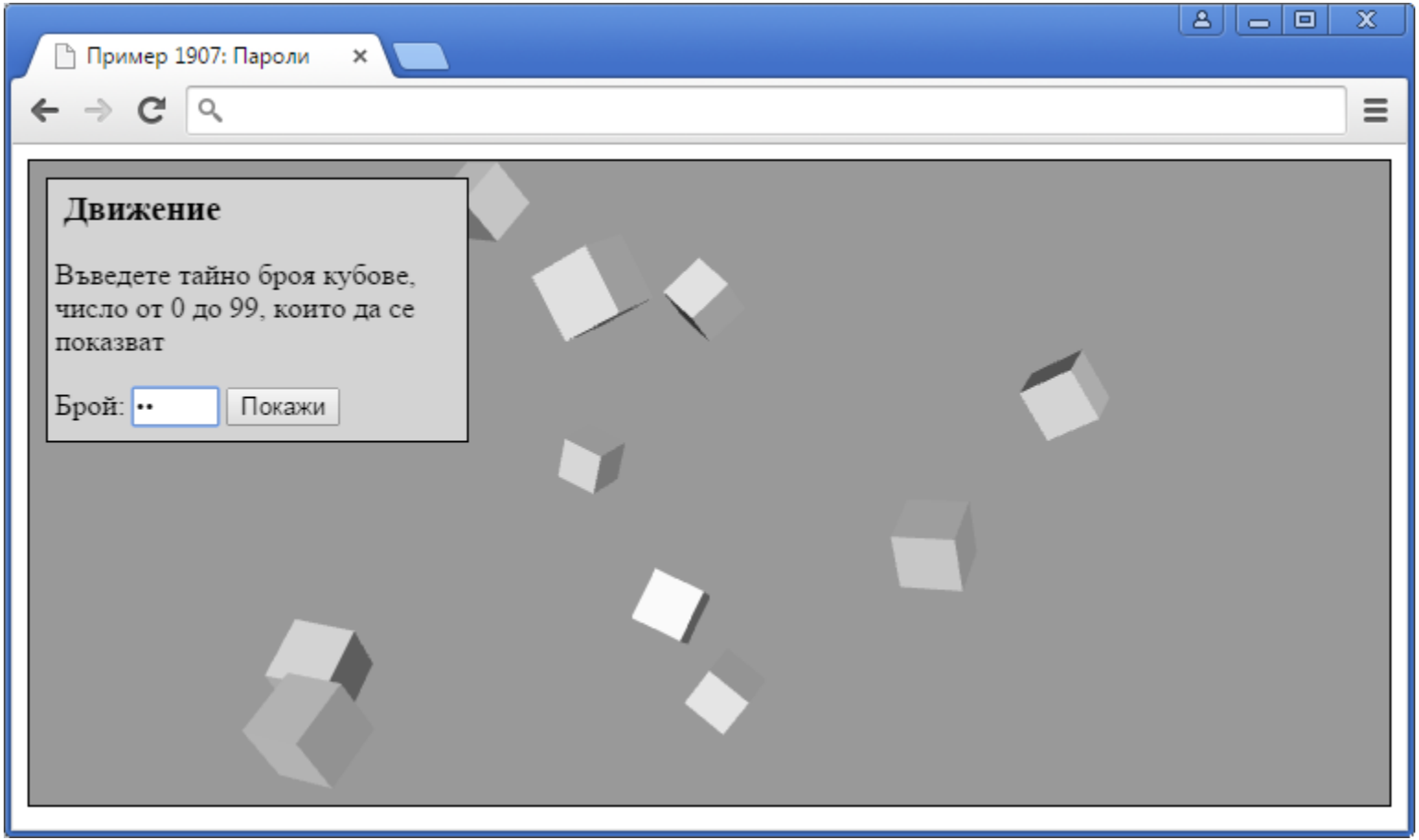
- Реалната стойност на полето-парола е във value
- Алтернативен начин да се ограничи число в интервал, без да се ползва if, е с израз с min и max

```
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 = '';
```


Пример 1907: Пароли
Движение
Въведете тайно броя кубове, число от 0 до 99, които да се показват
Брой:
Покажи

ПРОБА

Избор на цвят

Цел

- Сцена с графични обекти
- Избира се интерактивно цвят
- Всички обекти се оцветяват в този цвят

Идеи за реализация

- Библиотеки и компоненти, които дават тази функционалност
- Стандартен HTML начин за Firefox, Chrome и Opera, но не и за Internet Explorer и Safari
- В Internet Explorer елементът се показва като текстово поле

HTML начин

- Ползва се елемент input с type със стойност color
- Стойността value на елемента е цвят в 16-на бройна система във формат #rrggbb
- Всеки цветови компонент е цяло число от 0 до 255
- Първоначалната стойност е #ffffff, което е бял цвят

```
<h3>Изберете цвят</h3>
<input type="color"
       id="col"
       value="#ffffff"
       onchange="doIt()">
```

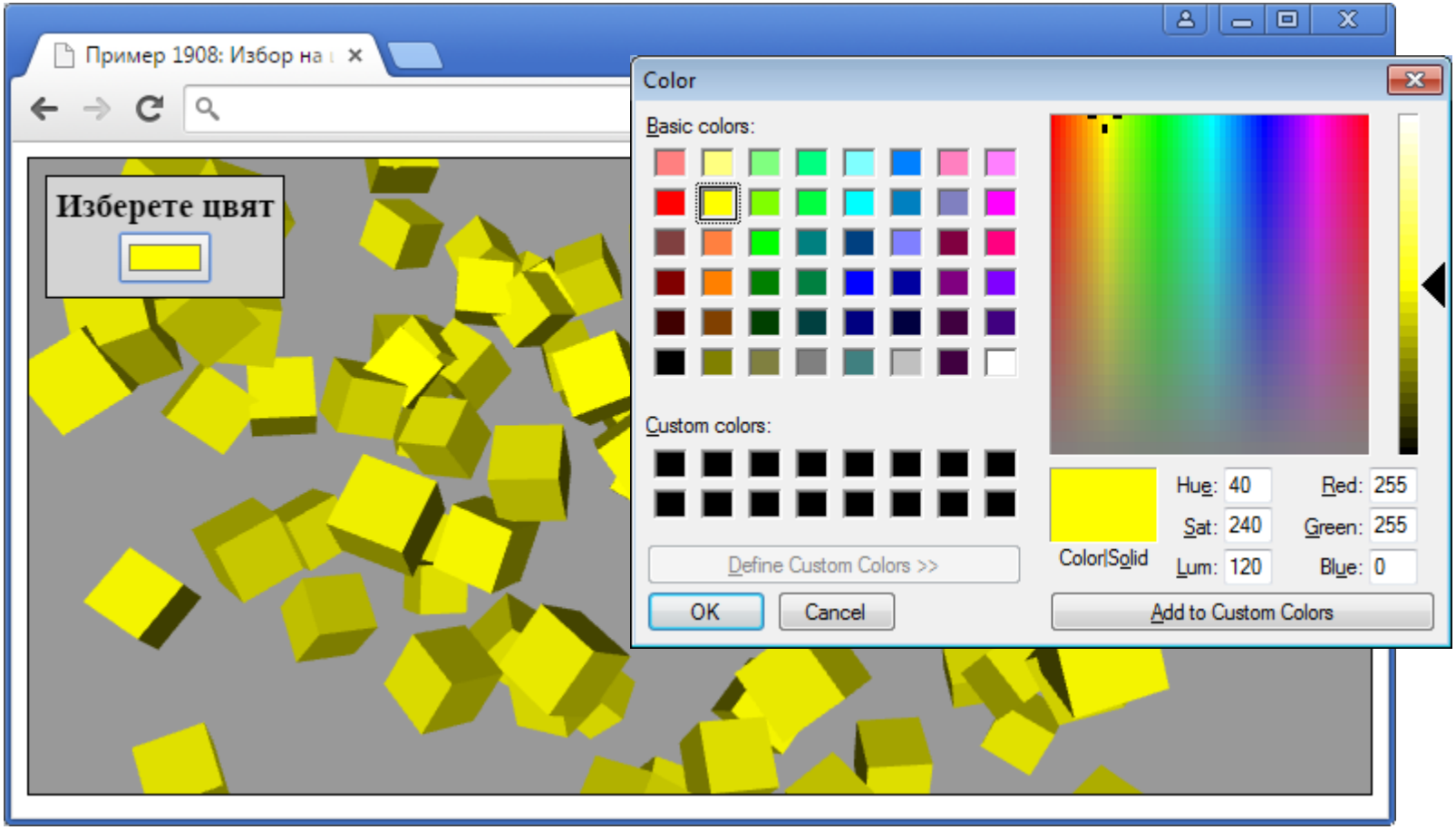
Обработване на събитието change

- Очаква се форматът на стойността #rrggbb да е спазен
- С метода substring се извличат двойка 16-ни цифри
- С parseInt(…,16) се преобразува до цяло число
- С деление на 255 се получава дробна стойност между 0 и 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];
```


Пример 1908: Избор на
Изберете цвят
Color
Basic colors:
Custom colors:
Define Custom Colors >>
OK
Cancel
Color|Solid
Hue: 40
Sat: 240
Lum: 120
Red: 255
Green: 255
Blue: 0
Add to Custom Colors

ПРОБА

Избор на дата

Цел

- Интерактивно избиране на дата от календара

Идеи за реализация

- Библиотеки и компоненти, които дават тази функционалност
- Стандартен HTML начин за Chrome, Opera и Safari, но не и за Firefox и Internet Explorer (при тях се показва обикновено текстово поле)

Реализация

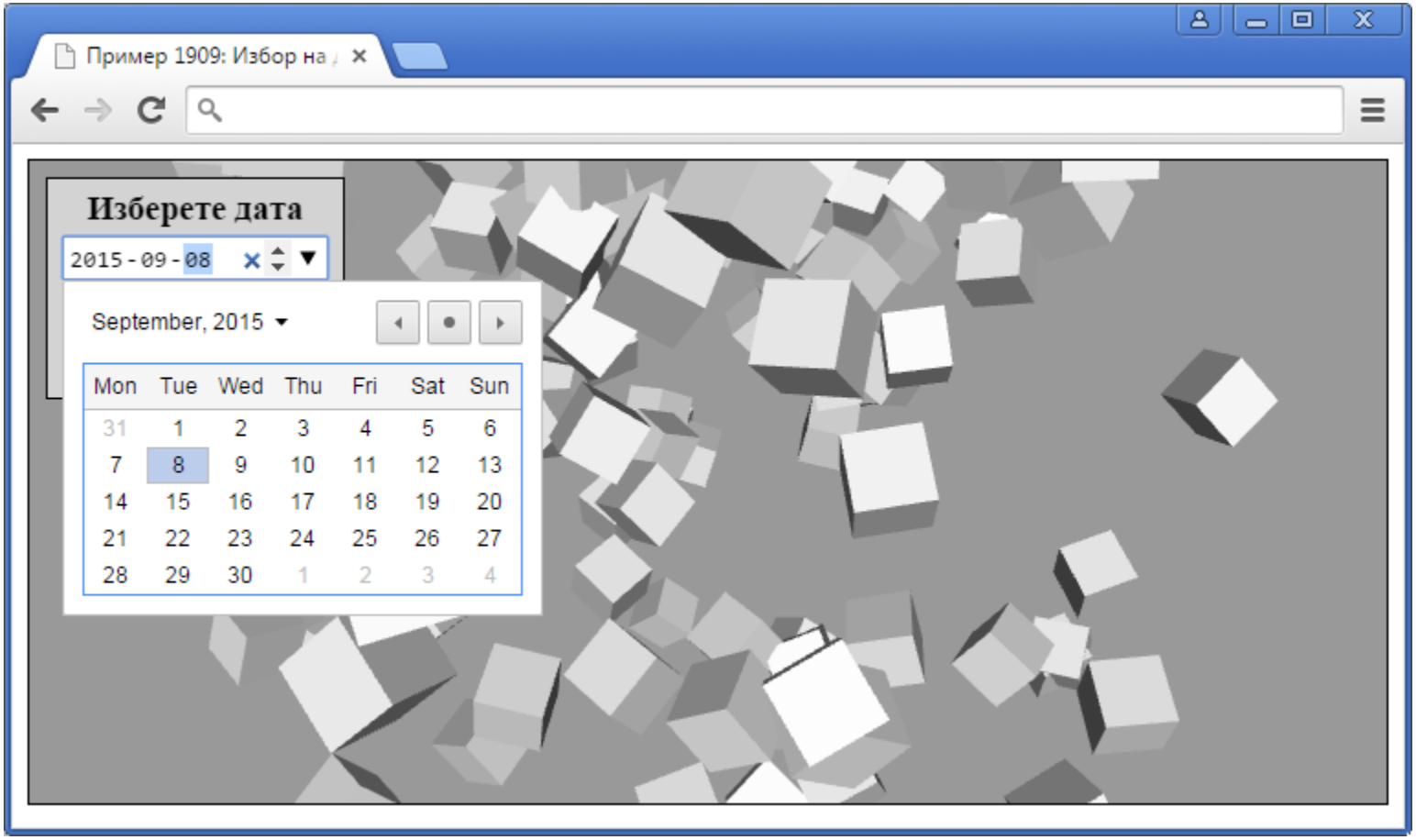
- Ползва се елемент input с type със стойност date
- Вторият input елемент е само за показване на избраната дата, затова е дефиниран с disabled като неактивен

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

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

- След избор на дата, тя се копира от единия елемент в другия

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


Пример 1909: Избор на
Изберете дата
2015-09-08
September, 2015
Mon Tue Wed Thu Fri Sat Sun
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

ПРОБА

Избор от диапазон

Цел

- Интерактивен избор на число в определен диапазон
- Числото не се въвежда цифрово

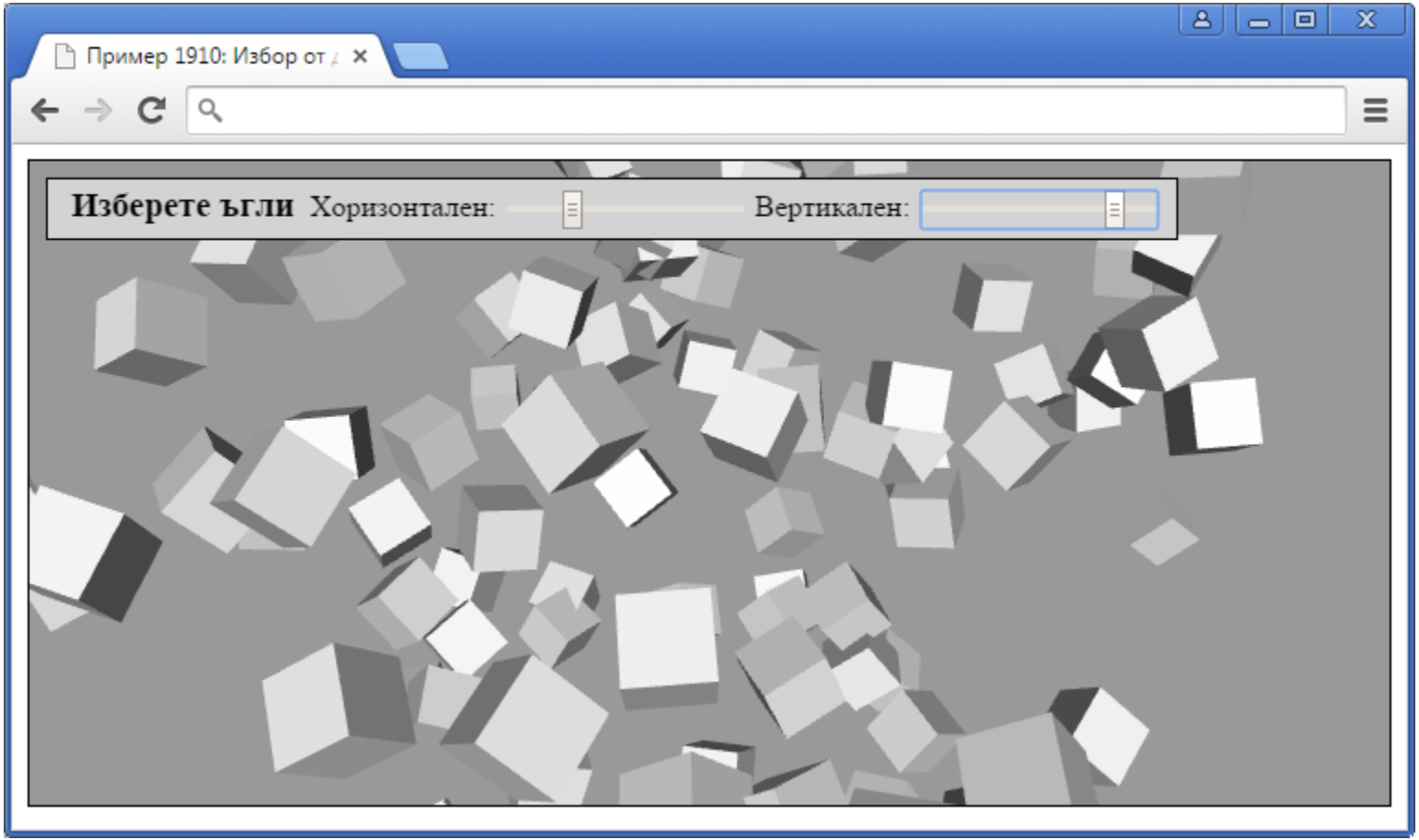
Идеи за реализация

- Стандартен HTML начин
- Различен външен вид в различните браузъри

Реализация

- Ползва се елемент input с type със стойност range
- В атрибута min се описва минималната допустима стойност
- В атрибута max се описва максималната допустима стойност
- В атрибута value се описва началната, а после и текущата стойност на елемента

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

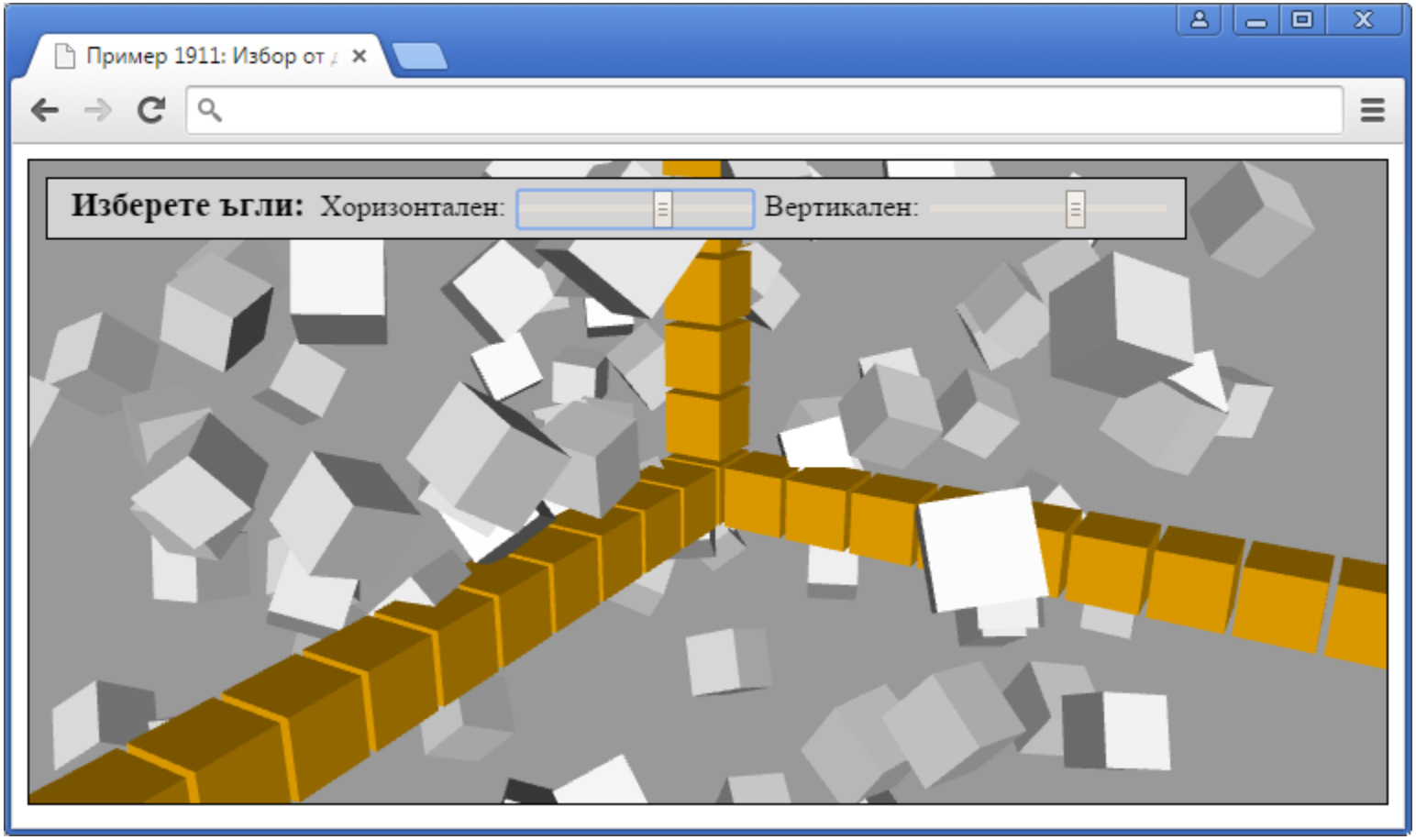

Пример 1910: Избор от
Изберете ъгли
Хоризонтален:
Вертикален:

ПРОБА

Промяна

- Възможните стойности да са през някаква стъпка
  - По хоризонтала със стъпка 30: -120, -90, -60, -30, 0, 30, 60, 90, 120
  - По вертикала със стъпка 20: -80, -60, -40, -20, 0, 20, 40, 60, 80
- Ползва се атрибут step

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


Пример 1911: Избор от
Изберете ъгли:
Хоризонтален:
Вертикален:

ПРОБА

Обобщение

Потребителски интерфейс

Въвеждане на данни

- Директно се изписват
- Използват се интерактивни елементи

Общи свойства

- Началната стойност е във value
- Текущата стойност също е във value
- Деактивирането на входното поле е със свойството disable

Чекбоксове

- Позволяват независим булев избор (т.е. само да или не)
- Входно поле <input type="checkbox"…>
- Свойството checked съдържа дали елементът е маркиран
- С елемента <label> може да се обедини чекбокс с текст

Радио бутони

- Позволяват единствен избор от няколко възможности
- Входно поле <input type="radio" name="..." …>
- Атрибутът name трябва да е еднакъв за група радио бутони
- Свойството checked съдържа дали елементът е маркиран
- С елемента <label> може да се обедини радио бутон с текст

Пароли

- Позволяват скрито въвеждане на текст в текстово поле
- Входно поле <input type="password"…>

Цветове

- Позволяват интерактивен избор на цвят
- Входно поле <input type="color"…>
- Стойността е цвят в 16-на система във формата #rrggbb
- Отделните елементи могат да се извлекат със substring и да се конвертират до десетично число с parseInt(...,16)
- Браузърите, които не поддържат избор на цвят, показват текстово поле

Дати

- Позволяват интерактивен избор на дата от календар
- Входно поле <input type="date"…>
- Стойността е дата в текстов формат
- Браузърите, които не поддържат избор на дата, показват текстово поле

Интервали

- Позволяват интерактивен избор на число от интервал
- Входно поле <input type="range"…>
- Минималната стойност е в атрибут min, максималната в max, стъпката в step, а текущата стойност е във value

ИКТ в НОС

Край

Коментари, въпроси