**Лабораторна робота №7**

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

1

*ДУ «Житомирська політехніка».23.121.03.00 – Лр7*

Розроб.

Білотіл Я. О.

Перевір.

Дмитренко І. А.

*Т. контр.*

Н. контр.

Затверд.

Звіт

з лабораторної роботи

Літ.

Аркушів

*13*

*ФІКТ Гр. IПЗк-23-1*

**Тема:** Події миші та клавіатури.

**Мета роботи:** вивчити особливості обробки подій миші та клавіатури.

**Завдання на лабораторну роботу**

**Завдання 1.**

**Надати квадратикам можливість перетягування (draggable) в межах контейнера*.*** При переміщенні курсора, наведеному на даний блок-квадратик, та утриманні кнопки миші даний елемент повинен перетягуватись. Квадратики не повинні заходити навіть частково за межі контейнера.

Код програми:

index.html

<!DOCTYPE html>

<html>

<head>

<title>Перетягування квадратиків в межах контейнера</title>

<link rel="stylesheet" href="style.css">

<script src="script.js"></script>

</head>

<body>

<div class="container">

<div class="draggable square" id="square1"></div>

<div class="draggable square" id="square2"></div>

<div class="draggable square" id="square3"></div>

<div class="draggable square" id="square4"></div>

<div class="draggable square" id="square5"></div>

</div>

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

<script src="https://code.jquery.com/ui/1.13.0/jquery-ui.min.js"></script>

</body>

</html>

style.css

.container {

width: 400px;

height: 400px;

border: 2px solid #ccc;

position: relative;

overflow: hidden;

}

.square {

width: 50px;

height: 50px;

background-color: #3498db;

position: absolute;

}

#square1 {

width: 50px;

height: 50px;

background-color: #3498db;

top: 50px;

left: 50px;

}

#square2 {

background-color: #9b59b6;

top: 200px;

left: 200px;

}

#square3 {

background-color: #e74c3c;

top: 100px;

left: 300px;

}

#square4 {

background-color: #f39c12;

top: 250px;

left: 100px;

}

#square5 {

background-color: #2ecc71;

top: 150px;

left: 200px;

}

script.js

$(function () {

function randomPositionAndSize($element) {

var maxSize = 100;

var minSize = 30;

var containerWidth = $(".container").width();

var containerHeight = $(".container").height();

var newPositionTop = Math.random() \* (containerHeight - maxSize);

var newPositionLeft = Math.random() \* (containerWidth - maxSize);

var newSize = Math.random() \* (maxSize - minSize) + minSize;

$element.animate({

width: newSize,

height: newSize,

top: newPositionTop,

left: newPositionLeft,

});

}

$(".draggable").each(function () {

randomPositionAndSize($(this));

});

$(".draggable").draggable({

containment: ".container",

scroll: false,

});

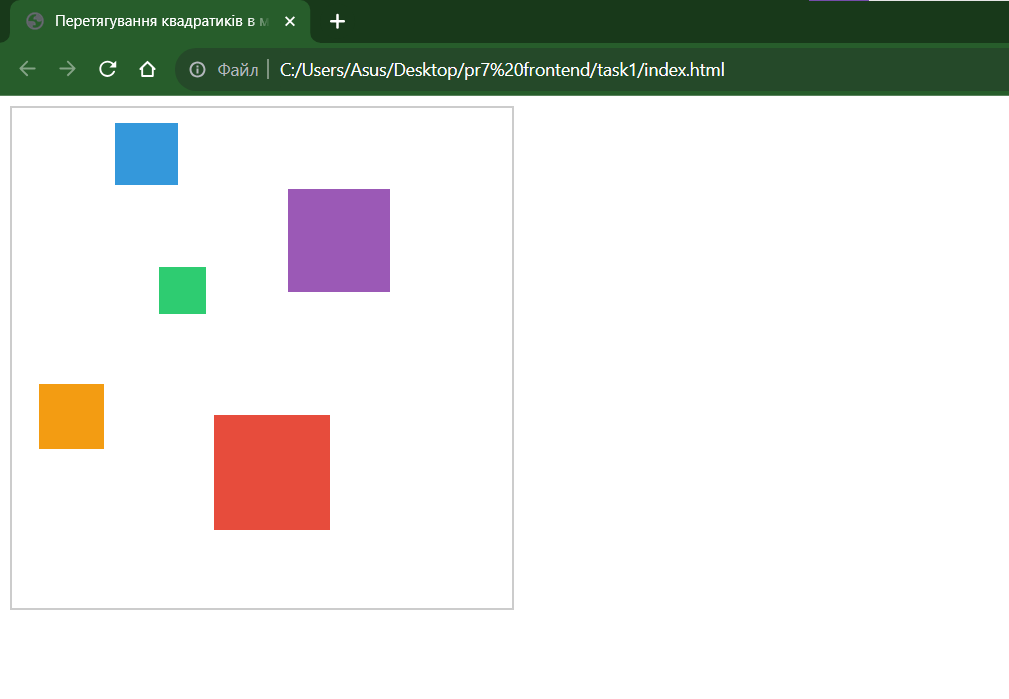
$(".draggable").on("click", function () {

randomPositionAndSize($(this));

});

});

Результат роботи програми:



Мал. 1 – Завдання 1

**Завдання 2.**

**Розмістіть на сторінці випадковим чином 20 кольорових кружечків з радіусами від 10 до 30**. Перший з них є активним (реалізуйте візуальне виділення активного кружечка). Передбачте такі можливості:

* при натисканні на клавішу «Tab» повинен ставати активним наступний кружечок.
* при натисканні комбінації клавіш “Shift-Tab”, активним повинен ставати попередній кружечок.
* клавіші стрілок повинні зміщувати активний кружечок у відповідну сторону на 10 пікселів.

Код програми:

index.html

<!DOCTYPE html>

<html>

<head>

<title>Кольорові кружечки</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="container" tabindex="0"></div>

<script src="script.js"></script>

</body>

</html>

style.css

.container {

width: 100%;

height: 100vh;

position: relative;

}

.circle {

width: 40px;

height: 40px;

border-radius: 50%;

position: absolute;

cursor: pointer;

}

.active {

border: 2px solid #000;

}

script.js

document.addEventListener("DOMContentLoaded", function () {

const container = document.querySelector('.container');

const colors = ['#3498db', '#e74c3c', '#2ecc71', '#f39c12', '#9b59b6', '#1abc9c', '#34495e', '#f1c40f', '#e67e22', '#95a5a6'];

for (let i = 0; i < 20; i++) {

const circle = document.createElement('div');

const radius = Math.floor(Math.random() \* (30 - 10 + 1)) + 10;

const color = colors[Math.floor(Math.random() \* colors.length)];

circle.classList.add('circle');

circle.style.width = `${radius}px`;

circle.style.height = `${radius}px`;

circle.style.backgroundColor = color;

const left = Math.random() \* (container.clientWidth - radius \* 2);

const top = Math.random() \* (container.clientHeight - radius \* 2);

circle.style.left = `${left}px`;

circle.style.top = `${top}px`;

container.appendChild(circle);

}

const circles = document.querySelectorAll('.circle');

let activeIndex = 0;

function setActiveCircle(index) {

circles.forEach((circle, i) => {

if (i === index) {

circle.classList.add('active');

} else {

circle.classList.remove('active');

}

});

}

container.addEventListener('keydown', function (event) {

if (event.key === 'Tab' && !event.shiftKey) {

event.preventDefault();

activeIndex = (activeIndex + 1) % circles.length;

setActiveCircle(activeIndex);

} else if (event.key === 'Tab' && event.shiftKey) {

event.preventDefault();

activeIndex = (activeIndex - 1 + circles.length) % circles.length;

setActiveCircle(activeIndex);

} else if (event.key === 'ArrowRight') {

circles[activeIndex].style.left = `${parseInt(circles[activeIndex].style.left, 10) + 10}px`;

} else if (event.key === 'ArrowLeft') {

circles[activeIndex].style.left = `${parseInt(circles[activeIndex].style.left, 10) - 10}px`;

} else if (event.key === 'ArrowUp') {

circles[activeIndex].style.top = `${parseInt(circles[activeIndex].style.top, 10) - 10}px`;

} else if (event.key === 'ArrowDown') {

circles[activeIndex].style.top = `${parseInt(circles[activeIndex].style.top, 10) + 10}px`;

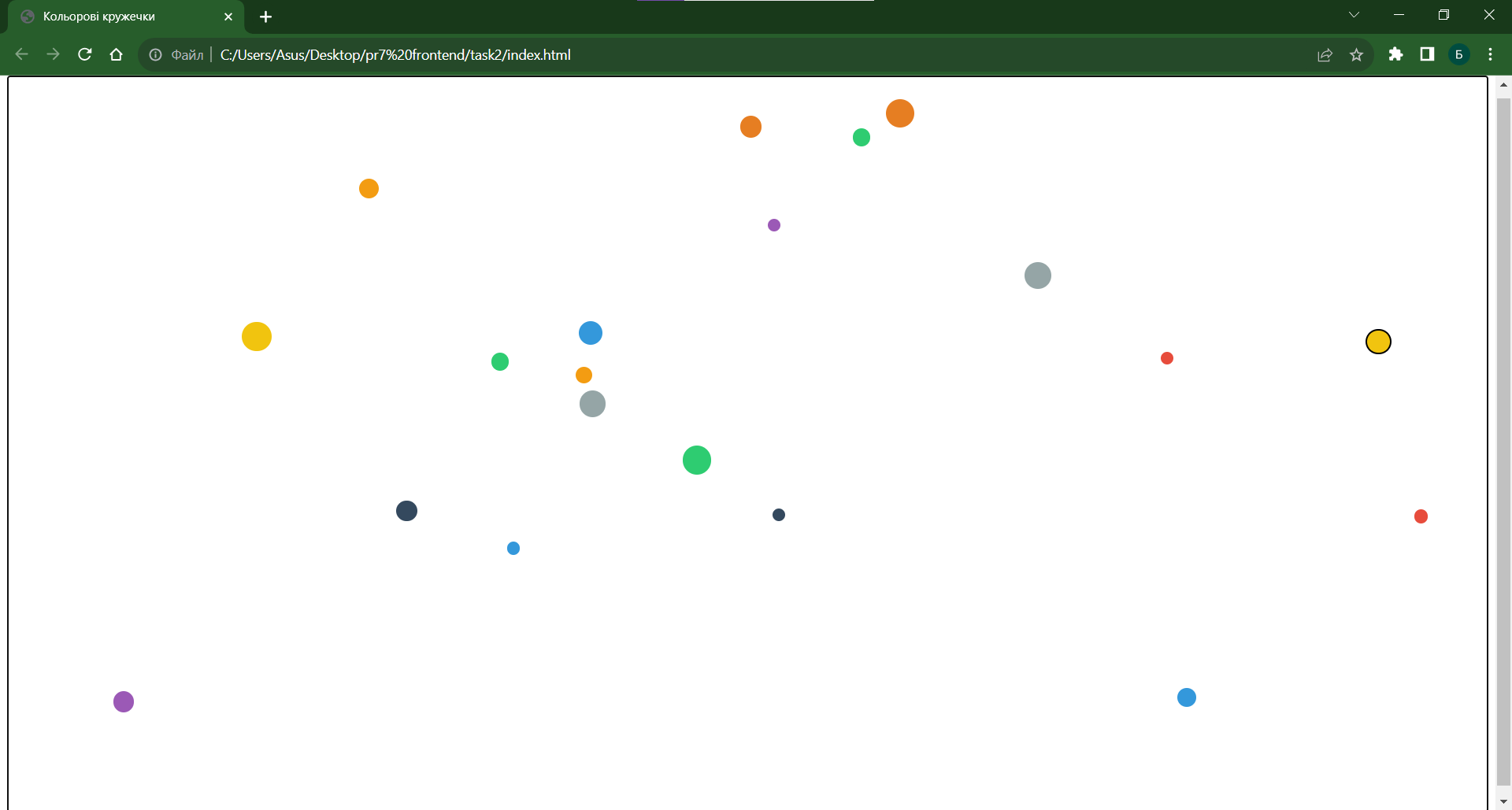
}

});

setActiveCircle(activeIndex);

});

Результат роботи програми:



Мал. 2 – Завдання 2

**Завдання 3.**

Реалізуйте «утікаючу» кнопку. В **div**-елементі знаходиться кнопка. Проте при наведенні на неї курсора миші, вона змінює своє положення так, що унеможливлює натиснення на неї. Кнопка не повинна виходити за межі  **div**-елемента.

Код програми:

index.html

<!DOCTYPE html>

<html>

<head>

<title>Утікаюча кнопка</title>

<style>

.container {

width: 300px;

height: 200px;

border: 1px solid #ccc;

position: relative;

overflow: hidden;

}

.button {

position: absolute;

padding: 10px 20px;

background-color: #3498db;

color: white;

border: none;

cursor: pointer;

}

</style>

</head>

<body>

<div class="container">

<button class="button">Натисніть мене</button>

</div>

<script src="script.js"></script>

</body>

</html>

script.js

document.addEventListener("DOMContentLoaded", function () {

const container = document.querySelector('.container');

const button = document.querySelector('.button');

let isHovered = false;

function getRandomPosition() {

const containerRect = container.getBoundingClientRect();

const buttonRect = button.getBoundingClientRect();

const maxX = containerRect.width - buttonRect.width;

const maxY = containerRect.height - buttonRect.height;

const newX = Math.floor(Math.random() \* maxX);

const newY = Math.floor(Math.random() \* maxY);

return { x: newX, y: newY };

}

function moveButton() {

if (isHovered) {

const newPosition = getRandomPosition();

button.style.left = `${newPosition.x}px`;

button.style.top = `${newPosition.y}px`;

}

setTimeout(moveButton, 1000);

}

button.addEventListener('mouseenter', function () {

isHovered = true;

moveButton();

});

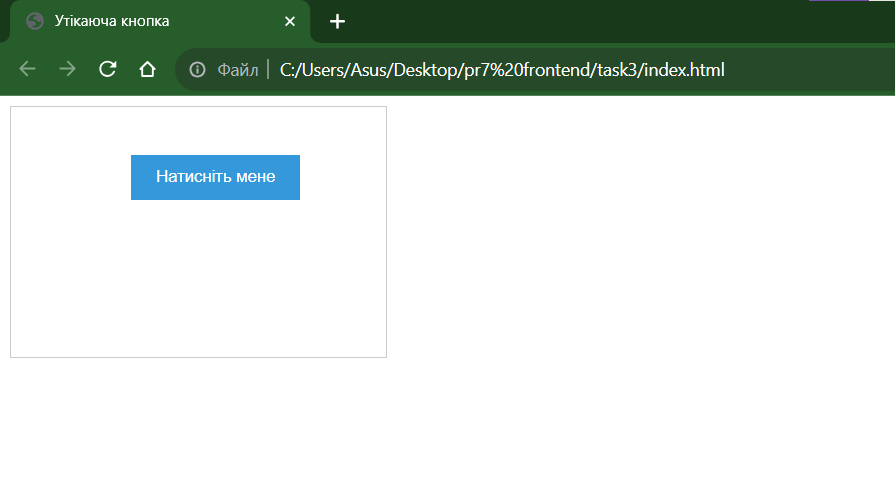
button.addEventListener('mouseleave', function () {

isHovered = false;

});

});

Результат роботи програми:



Мал. 3 – Завдання 3

**Завдання 4.**

Розмістіть список з елементів, які можна буде виділяти, як у файлових менеджерах. Передбачте такі можливості:

* клік на елементі списку виділяє лише цей елемент (додає клас **.selected**), та знімає виділення з усіх інших;
* клік за допомогою **Ctrl** (**Cmd** для **Mac**) виділяє елемент (або якщо елемент виділений, то знімає виділення з нього), виділення інших елементів при цьому не змінюються.

Код програми:

index.html

<!DOCTYPE html>

<html>

<head>

<title>Виділення елементів списку</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<ul>

<li>Елемент 1</li>

<li>Елемент 2</li>

<li>Елемент 3</li>

<li>Елемент 4</li>

<li>Елемент 5</li>

</ul>

<script src="script.js"></script>

</body>

</html>

style.css

.selected {

background-color: #3498db;

color: white;

}

ul {

list-style: none;

padding: 0;

}

li {

padding: 8px;

cursor: pointer;

}

script.js

document.addEventListener("DOMContentLoaded", function () {

const listItems = document.querySelectorAll('li');

function clearSelection() {

listItems.forEach(item => {

item.classList.remove('selected');

});

}

listItems.forEach(item => {

item.addEventListener('click', function (event) {

if (!event.ctrlKey && !event.metaKey) {

clearSelection();

}

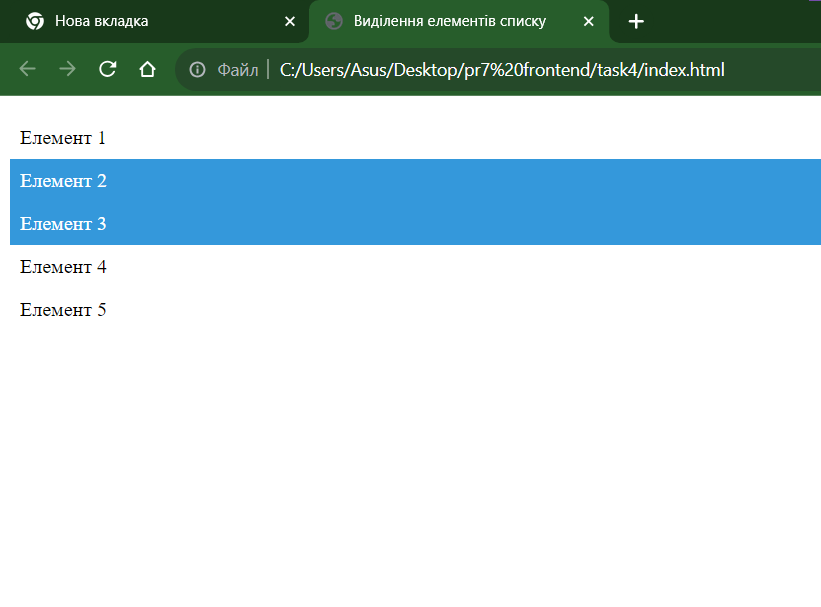
item.classList.toggle('selected');

});

});

});

Результат роботи програми:



Мал. 4 – Завдання 4

**Завдання 5.**

Реалізуйте елемент “слайдер”, який має такий вигляд:



Передбачте такі особливості його роботи:

* при наведенні курсору миші на «повзунок», користувач затискає кнопку миші і рухає «повзунок» переміщаючи курсор миші;
* при натиснутій кнопці миші, курсор може виходити за межі слайдера, але слайдер все одно має працювати (це зручно для користувача);
* слайдер повинен нормально працювати при різкому русі миші ліворуч або праворуч за межі слайдера. При цьому «повзунок» повинен зупинятися чітко біля його краю.

Код програми:

index.html

<!DOCTYPE html>

<html>

<head>

<title>Слайдер</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="slider">

<div class="slider-bar"></div>

<div class="slider-handle"></div>

</div>

<script src="script.js"></script>

</body>

</html>

style.css

.slider {

width: 300px;

height: 20px;

background-color: #ccc;

position: relative;

cursor: pointer;

}

.slider .slider-bar {

width: 100%;

height: 100%;

background-color: #ccc;

}

.slider .slider-handle {

width: 20px;

height: 20px;

background-color: #f39c12;

border-radius: 5px;

position: absolute;

top: 50%;

transform: translateY(-50%);

cursor: pointer;

}

script.js

document.addEventListener("DOMContentLoaded", function () {

const slider = document.querySelector('.slider');

const handle = document.querySelector('.slider-handle');

let isDragging = false;

handle.addEventListener('mousedown', function (event) {

isDragging = true;

handle.style.transition = 'none';

});

document.addEventListener('mouseup', function () {

isDragging = false;

handle.style.transition = '';

});

slider.addEventListener('mousemove', function (event) {

if (isDragging || event.buttons === 1) {

const sliderRect = slider.getBoundingClientRect();

let newPosition = event.clientX - sliderRect.left;

if (newPosition < 0) {

newPosition = 0;

} else if (newPosition > sliderRect.width) {

newPosition = sliderRect.width;

}

handle.style.left = `${newPosition}px`;

}

});

handle.addEventListener('mouseleave', function () {

if (isDragging) {

isDragging = false;

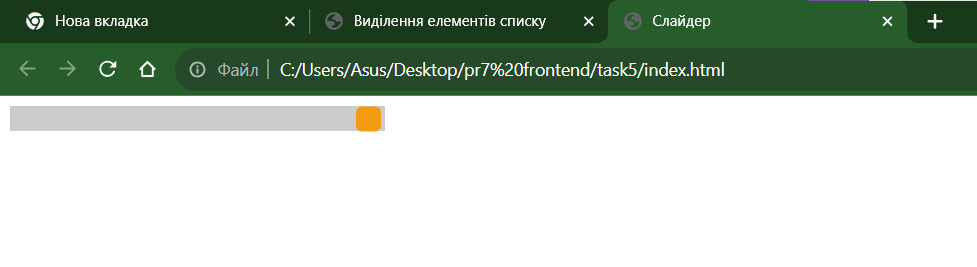
handle.style.transition = '';

}

});

});

Результат роботи програми:



Мал. 5 – Завдання 5