**Лабораторна робота №7**

**Події миші та клавіатури.**

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

**Хід роботи:**

***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.

Лістинг програми:

const task1=()=>{

    const task1container = document.createElement('div')

    const square = document.createElement('div')

    document.querySelector('body').appendChild(task1container).classList.add('task1container')

    task1container.appendChild(square).classList.add('square')

    square.addEventListener('mousedown', (e) => {

        if (e.button === 0) {

            task1container.addEventListener('mousemove', moveSquare);

            e.preventDefault();

        }

    });

    task1container.addEventListener('mouseup', () => {

        task1container.removeEventListener('mousemove', moveSquare);

    });

    function moveSquare(e) {

        const containerRect = task1container.getBoundingClientRect();

        const squareRect = square.getBoundingClientRect();

        const left = e.clientX - containerRect.left - squareRect.width / 2;

        const top = e.clientY - containerRect.top - squareRect.height / 2;

        if (left >= 0 && left + squareRect.width <= containerRect.width && top >= 0 && top + squareRect.height <= containerRect.height) {

            square.style.left = left + 'px';

            square.style.top = top + 'px';

          }}}

task1()

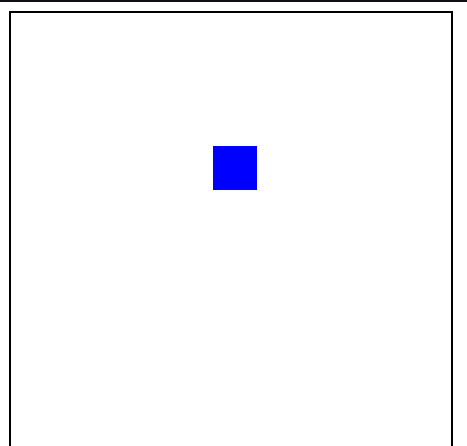
Результати виконання:

Рис. 1 – Результат 1 завдання

***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.

Лістинг програми:

const task2=()=>{

    const task2container = document.createElement('div')

    document.querySelector('body').appendChild(task2container).classList.add('task2container')

    const container = document.querySelector('.task2container')

    function createRandomCircle() {

        const circle = document.createElement('div')

        circle.classList.add('circle')

        circle.style.left = Math.floor(Math.random() \* 550)+'px'

        circle.style.top = Math.floor(Math.random() \* 550)+'px'

        circle.style.backgroundColor = getRandomColor()

        circle.style.width = Math.floor(Math.random() \* 30) + 10+'px'

        circle.style.height = circle.style.width

        return circle;

      }

      function getRandomColor() {

        const letters = '0123456789ABCDEF'

        let color = '#'

        for (let i = 0; i < 6; i++) {

          color += letters[Math.floor(Math.random() \* 16)]

        }

        return color;

      }

      function toggleActiveCircle(circles, activeIndex, newIndex) {

        circles[activeIndex].classList.remove('active')

        circles[newIndex].classList.add('active')

        circles[newIndex].focus()

      }

      function handleKeydown(event) {

        const circles = Array.from(container.children)

        const activeIndex = circles.findIndex(circle => circle.classList.contains('active'))

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

          event.preventDefault();

          const newIndex = (activeIndex + 1) % circles.length

          toggleActiveCircle(circles, activeIndex, newIndex)

        }

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

          event.preventDefault()

          const newIndex = (activeIndex - 1 + circles.length) % circles.length

          toggleActiveCircle(circles, activeIndex, newIndex)}

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

          circles[activeIndex].style.top = parseInt(circles[activeIndex].style.top) - 10+'px'}

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

          circles[activeIndex].style.top = parseInt(circles[activeIndex].style.top) + 10+'px'}

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

          circles[activeIndex].style.left = parseInt(circles[activeIndex].style.left) - 10+'px'}

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

          circles[activeIndex].style.left = parseInt(circles[activeIndex].style.left) + 10+'px'}

      }

      for (let i = 0; i < 20; i++) {

        const circle = createRandomCircle()

        container.appendChild(circle)

        if (i === 0) {

          circle.classList.add('active')

          circle.tabIndex = 0

        }

        circle.addEventListener('click', () => {

          const circles = Array.from(container.children);

          const activeIndex = circles.findIndex(circle => circle.classList.contains('active'))

          const newIndex = circles.indexOf(circle)

          toggleActiveCircle(circles, activeIndex, newIndex)

        });

      }

      document.addEventListener('keydown', handleKeydown)

}

task2()

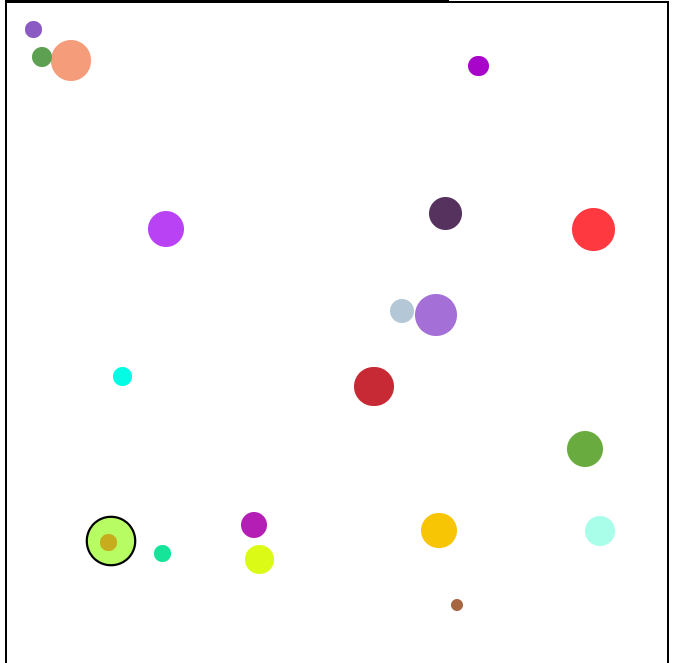
Результати виконання:

Рис. 2 – Результат виконання 2 завдання

***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

Лістинг програми:

const task3=()=>{

    const task3container = document.createElement('div')

    const runningSquare = document.createElement('div')

    document.querySelector('body').appendChild(task3container).classList.add('task3container')

    task3container.appendChild(runningSquare).classList.add('running-square')

    const container = document.querySelector('.task3container')

    const square = document.querySelector('.running-square')

    const containerWidth = container.offsetWidth

    const containerHeight = container.offsetHeight

    const squareSize = square.offsetWidth

    square.style.left = getRandomPosition(containerWidth - squareSize)+'px'

    square.style.top = getRandomPosition(containerHeight - squareSize)+'px'

    square.addEventListener('mouseover', () => {

        const maxX = containerWidth - squareSize

        const maxY = containerHeight - squareSize

        const randomX = getRandomPosition(maxX)

        const randomY = getRandomPosition(maxY)

        square.style.left = `${randomX}px`

        square.style.top = `${randomY}px`

    });

    function getRandomPosition(max) {

        return Math.floor(Math.random() \* max)

    }

}

task3()

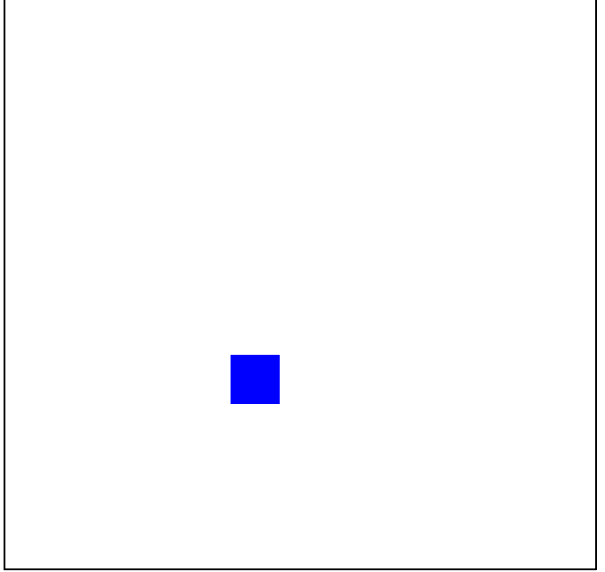
Результат виконання

Рис 3 – Результат виконання завдання

***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.

Лістинг програми:

const task4=()=>{

    const ul = document.createElement('ul')

    document.querySelector('body').appendChild(ul)

    for(let i=0;i<5;i++){

        const li = document.createElement('li')

        ul.appendChild(li)

        li.textContent=`Element ${i+1}`

    }

    const list = document.querySelector('ul')

    const listItems = list.querySelectorAll('li')

    listItems.forEach(item => {

        item.addEventListener('click', (e) => {

            if (!e.ctrlKey) {

                listItems.forEach(otherItem => otherItem.classList.remove('selected'))

            }

            item.classList.toggle('selected')

        })

    })

}

task4()

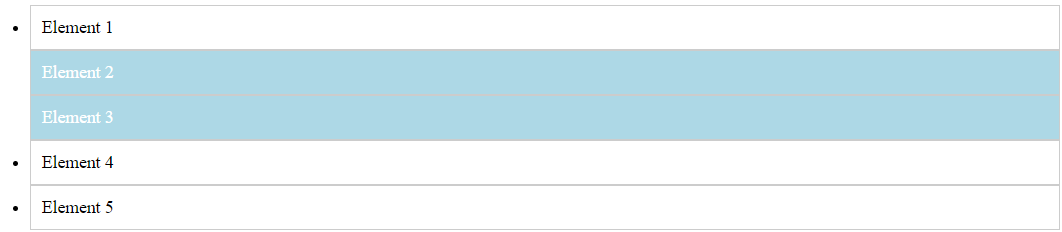
Результати виконання:

Рис. 4 – Результат виконання завдання

***Завдання 5***. Реалізуйте елемент “слайдер”.

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.Лістинг програми:

const task5=()=>{

    const task5container = document.createElement('div')

    const addTrack = document.createElement('div')

    const addThumb = document.createElement('div')

    addThumb.classList.add('thumb')

    addTrack.classList.add('track')

    document.querySelector('body').appendChild(task5container).classList.add('task5container')

    task5container.appendChild(addTrack)

    const track = document.querySelector('.track')

    track.appendChild(addThumb)

    const slider = document.querySelector('.task5container');

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

    let isDragging = false;

    thumb.addEventListener('mousedown', (e) => {

        isDragging = true;

        thumb.style.cursor = 'grabbing';

        document.addEventListener('mousemove', onMouseMove);

        document.addEventListener('mouseup', onMouseUp);

    });

    function onMouseMove(e) {

        if (isDragging) {

            const sliderRect = slider.getBoundingClientRect();

            const offsetX = e.clientX - sliderRect.left;

            const newPosition = Math.min(sliderRect.width - thumb.offsetWidth, Math.max(0, offsetX));

            thumb.style.left = newPosition + 'px';

        }

    }

    function onMouseUp() {

        isDragging = false;

        thumb.style.cursor = 'grab';

        document.removeEventListener('mousemove', onMouseMove);

        document.removeEventListener('mouseup', onMouseUp);

    }

    thumb.addEventListener('mouseleave', () => {

        if (isDragging) {

            thumb.style.cursor = 'grab';

        }

    });

    slider.addEventListener('mouseleave', () => {

        if (isDragging) {

            isDragging = false;

            thumb.style.cursor = 'grab';

        }

    });

}

task5()

Результати виконання:

Рис. 5 – Результат виконання завдання

***Висновки.*** вивчив особливості обробки подій миші та клавіатури