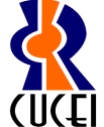
Logotipo

Descripción generada automáticamenteUniversidad de Guadalajara.

Centro Universitario de Ciencias Exactas e Ingenierías.

DIVISIÓN DE TECNOLOGÍAS PARA LA INTEGRACIÓN CIBER HUMANA.

DEPARTAMENTO DE CIENCIAS

COMPUTACIONALES.

Practica6

**NOMBRE DE LOS ESTUDIANTES:**

Padilla Perez Jorge Daray

Juan Jesús Sámano Juárez

Ernesto Macias Flores

NOMBRE DE LA MATERIA: Seminario Sistemas Operativos

SECCION: D04

CICLO ESCOLAR: 2024-A

NOMBRE DEL PROFESOR: Julio Esteban

Esta actividad esta conformada por 3 códigos a continuación se mostrarán.

**Códigos:**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Frames con Iconos</title>

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

</head>

<body>

  <div id="frame1" class="frame">

    <img src="gallina.png" id="gallina" draggable="true" class="icon">

    <img src="zorro.png" id="zorro" draggable="true" class="icon">

    <img src="maiz.png" id="semillas" draggable="true" class="icon">

  </div>

  <div id="frame2" class="frame">

    <img src="bote (1).png" id="bote" draggable="true" class="icon">

  </div>

  <div id="frame3" class="frame"></div>

  <script src="script.js"></script>

</body>

</html>

**Dice.css**

.frame {

    width: 23.23%;

    height: 70vh;

    float: left;

    box-sizing: border-box;

    text-align: center;

    font-size: 24px;

  }

  #frame1, #frame3 {

    background-image: url('pasto5.jpg'); /\* Cambia 'pasto-verde.jpg' por la ruta de tu imagen de pasto verde \*/

    background-size: cover;

  }

  #frame2 {

    background-image: url('agua2.jpg'); /\* Cambia 'rio-azul.gif' por la ruta de tu imagen de río azul \*/

    background-size: cover;

  }

  .frame {

    width: 33.33%;

    height: 100vh;

    float: left;

    box-sizing: border-box;

    text-align: center;

    font-size: 24px;

    position: relative;

  }

  .icon {

    width: 50px;

    height: 50px;

    background-color: white;

    border: 1px solid black;

    border-radius: 50%;

    display: inline-block;

    cursor: pointer;

    position: absolute;

  }

  #zorro {

    top: 20%;

    left: 20%;

  }

  #gallina {

    top: 50%;

    left: 20%;

  }

  #semillas {

    top: 80%;

    left: 20%;

  }

  #bote {

    top: 50%;

    left: 50%;

  }

 /\* Estilos generales de los frames y los iconos \*/

.frame {

    width: 33.33%;

    height: 100vh;

    float: left;

    box-sizing: border-box;

    text-align: center;

    font-size: 24px;

    position: relative;

  }

  .icon {

    width: 50px;

    height: 50px;

    background-color: white;

    border: 1px solid black;

    border-radius: 50%;

    display: inline-block;

    cursor: pointer;

    position: absolute;

    transition: all 0.3s ease; /\* Transición suave para los íconos \*/

  }

  /\* Estilos específicos para el bote \*/

  #bote {

    width: 50px;

    height: 50px;

    background-color: blue;

    border: 1px solid black;

    border-radius: 50%;

    display: inline-block;

    cursor: pointer;

    position: absolute;

    top: 50%;

    left: 10px; /\* Ajusta la posición horizontal según sea necesario \*/

    transform: translateY(-50%);

  }

**Scripy.js**

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

    const gallina = document.getElementById('gallina');

    const zorro = document.getElementById('zorro');

    const semillas = document.getElementById('semillas');

    const bote = document.getElementById('bote');

    const frame1 = document.getElementById('frame1');

    const frame2 = document.getElementById('frame2');

    const frame3 = document.getElementById('frame3');

    // Función para animar el movimiento del bote de izquierda a derecha

    function animateBoteRight() {

        const frameWidth = frame2.offsetWidth;

        const boteWidth = bote.offsetWidth;

        const boteLeft = bote.offsetLeft;

        const targetLeft = frameWidth - boteWidth - 10; // Ajuste de margen derecho

        const animation = bote.animate(

        [{ left: boteLeft + 'px' }, { left: targetLeft + 'px' }],

        { duration: 1000 }

        );

        animation.onfinish = function() {

            // Mover el ícono seleccionado al frame derecho

            const selectedIcon = frame2.querySelector('.selected');

            if (selectedIcon) {

                selectedIcon.classList.remove('selected');

                frame3.appendChild(selectedIcon);

                // Verificar si el bote debe quedarse en el lado derecho del frame2

                if (bote.offsetLeft >= targetLeft) {

                    frame2.appendChild(bote);

                }

                // Verificar la validez de la situación

                checkSituation();

            }

        };

    }

    // Función para animar el movimiento del bote de derecha a izquierda

    function animateBoteLeft() {

        const frameWidth = frame2.offsetWidth;

        const boteWidth = bote.offsetWidth;

        const boteLeft = bote.offsetLeft;

        const targetLeft = 10; // Ajuste de margen izquierdo

        const animation = bote.animate(

        [{ left: boteLeft + 'px' }, { left: targetLeft + 'px' }],

        { duration: 1000 }

        );

        animation.onfinish = function() {

            // Mover el ícono seleccionado al frame izquierdo

            const selectedIcon = frame2.querySelector('.selected');

            if (selectedIcon) {

                selectedIcon.classList.remove('selected');

                frame1.appendChild(selectedIcon);

                // Verificar si el bote debe quedarse en el lado izquierdo del frame2

                if (bote.offsetLeft < targetLeft) {

                    frame2.appendChild(bote);

                }

                // Verificar la validez de la situación

                checkSituation();

            }

        };

    }

// Función para verificar la validez de la situación

function checkSituation() {

    // Verificar si la gallina y las semillas están solas en algún lado

    const ladoIzquierdo = frame1.children.length > 1;

    const ladoDerecho = frame3.children.length > 1;

    const boteEnElMedio = frame2.children.length > 1;

    if ((gallina.parentNode && gallina.parentNode.id === 'frame1') &&

        (semillas.parentNode && semillas.parentNode.id === 'frame1') &&

        (!zorro.parentNode || zorro.parentNode.id !== 'frame1')) {

        alert("¡Has perdido! La gallina se ha comido las semillas.");

    } else if ((gallina.parentNode && gallina.parentNode.id === 'frame3') &&

        (semillas.parentNode && semillas.parentNode.id === 'frame3') &&

        (!zorro.parentNode || zorro.parentNode.id !== 'frame3')) {

        alert("¡Has perdido! La gallina se ha comido las semillas.");

    } else if (gallina.parentNode && semillas.parentNode && gallina.parentNode === semillas.parentNode) {

        alert("¡Has perdido! La gallina se ha comido las semillas.");

    } else if (gallina.parentNode && zorro.parentNode && gallina.parentNode === zorro.parentNode) {

        alert("¡Has perdido! El zorro se ha comido la gallina.");

    } else if (ladoIzquierdo && ladoDerecho && !boteEnElMedio) {

        alert("¡Has ganado! Has llevado a la gallina, al zorro y las semillas al otro lado sin que se coman entre ellos.");

    }

}

    // Agregar evento de clic para mover el bote al lado opuesto y quedarse en su nueva posición

    bote.addEventListener('click', function() {

        // Determinar la posición actual del bote

        const boteLeft = bote.offsetLeft;

        const frameWidth = frame2.offsetWidth;

        const targetLeft = frameWidth - bote.offsetWidth - 10; // Ajuste de margen derecho

        // Si el bote está en el lado izquierdo, animarlo hacia la derecha

        if (boteLeft < targetLeft) {

            animateBoteRight();

            // Quedarse en la nueva posición

            bote.style.left = targetLeft + 'px';

        } else { // Si el bote está en el lado derecho, animarlo hacia la izquierda

            animateBoteLeft();

            // Quedarse en la nueva posición

            bote.style.left = '10px'; // Ajuste de margen izquierdo

        }

    });

    // Agregar evento de arrastre para cada icono

    [gallina, zorro, semillas, bote].forEach(icon => {

        icon.addEventListener('dragstart', function(e) {

            // Configurar los datos de transferencia para el ícono arrastrado

            e.dataTransfer.setData('text/plain', icon.id);

            // Si es un ícono seleccionable, agregar la clase 'selected' para destacarlo

            if ([gallina, zorro, semillas].includes(icon)) {

                icon.classList.add('selected');

            }

        });

    });

    // Agregar eventos de soltura para el bote solo en el frame del río

    frame2.addEventListener('dragover', function(e) {

        e.preventDefault();

    });

    frame2.addEventListener('drop', function(e) {

        e.preventDefault();

        const iconId = e.dataTransfer.getData('text/plain');

        const icon = document.getElementById(iconId);

        // Verificar si el elemento arrastrado es un ícono

        if (icon && icon !== frame2) {

            // Mover el ícono al bote si no es el bote mismo

            if (icon === bote) {

                frame2.appendChild(bote);

                // Mover los otros íconos junto al bote

                [gallina, zorro, semillas].forEach(icon => frame2.appendChild(icon));

            } else {

                frame2.appendChild(icon);

            }

            // Animar el movimiento del bote hacia la derecha

            animateBoteRight();

        }

    });

    // Agregar evento de clic para seleccionar un ícono

    [gallina, zorro, semillas].forEach(icon => {

        icon.addEventListener('click', function() {

            // Remover la clase de selección de todos los íconos

            [gallina, zorro, semillas].forEach(i => i.classList.remove('selected'));

            // Agregar la clase de selección al ícono clicado

            icon.classList.add('selected');

        });

    });

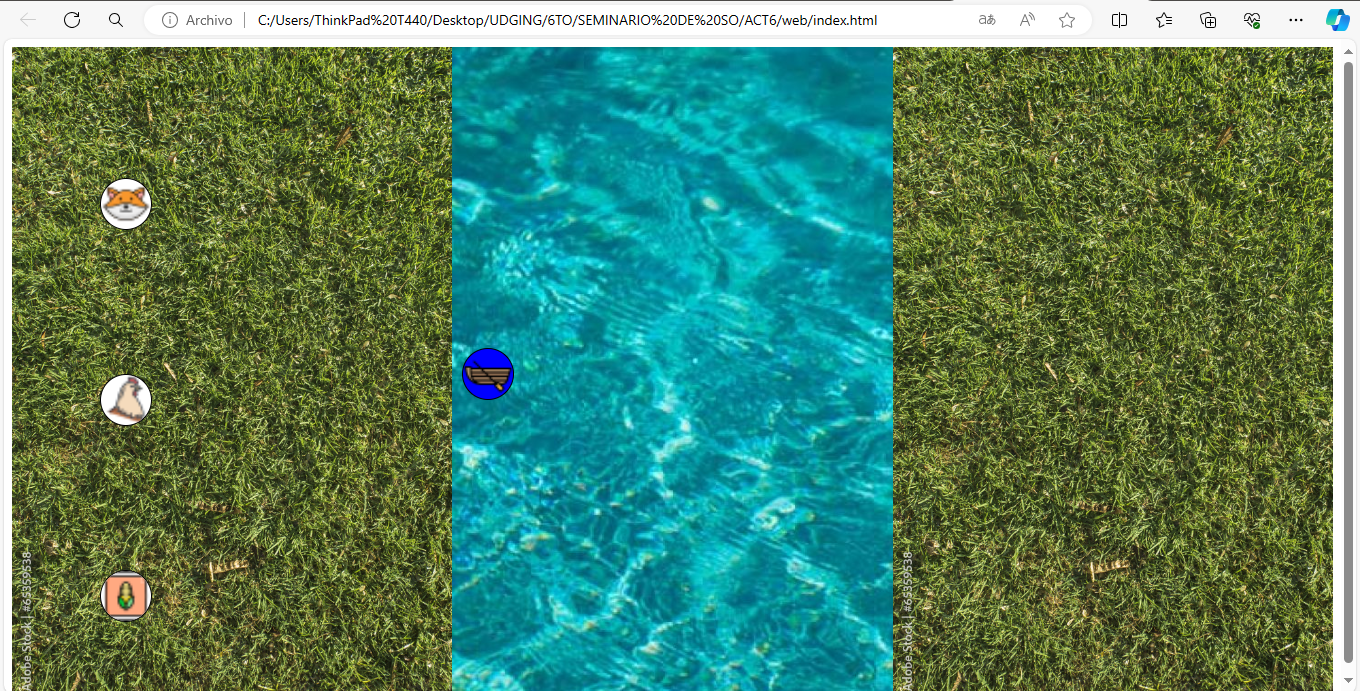
});

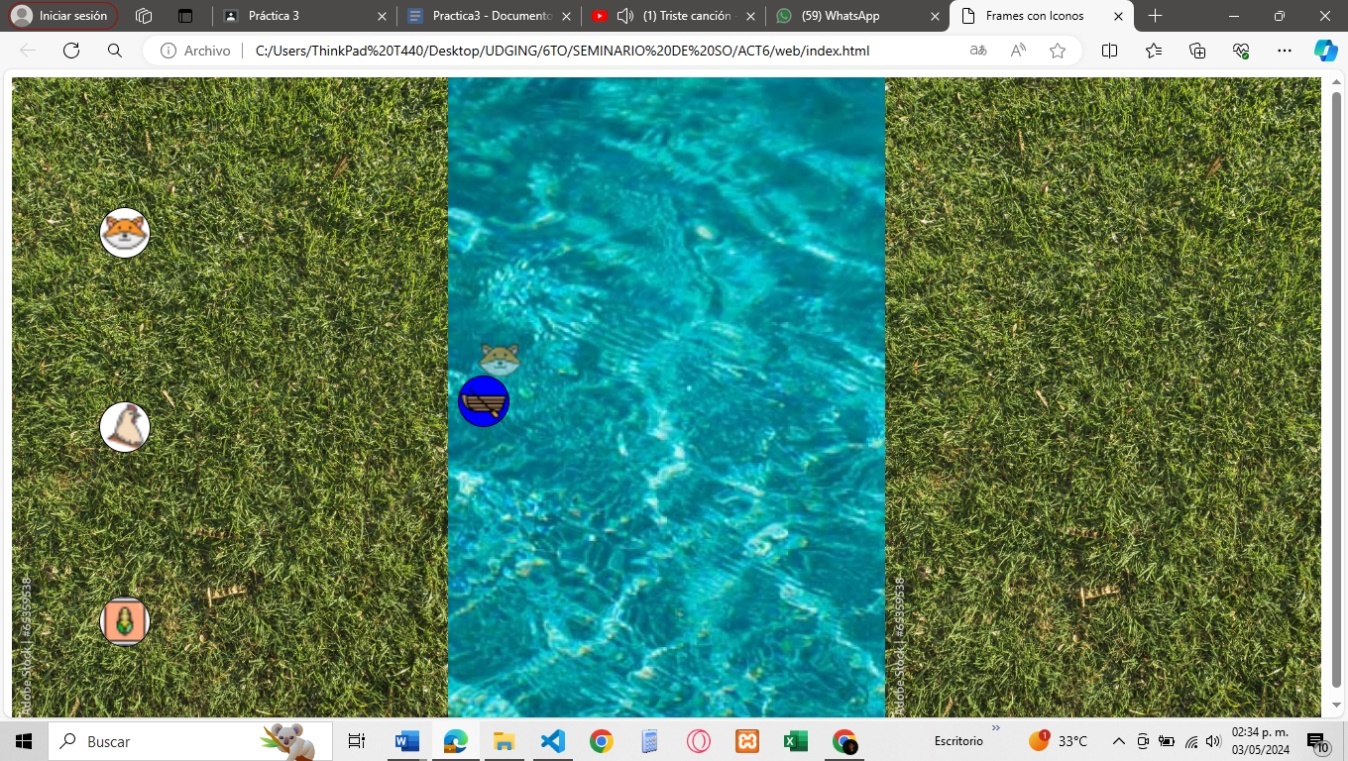
(es.javascript, 2024)

(Arciniegas, 2024)

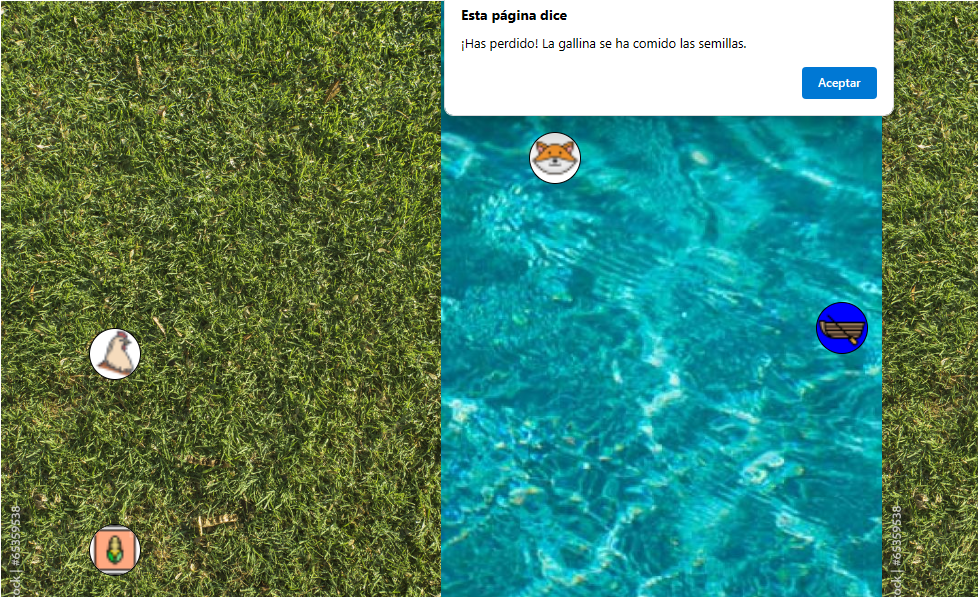
**Capturas de ejecución:**

Lo primero que s muestra al usuario es esta página, la cual mostrará todas las funciones que se requieren para esta actividad.



Los iconos se puedes arrastrar

Se cumplen las condicionales arroja alertas.



Interfaz de usuario gráfica

Descripción generada automáticamente

Referencias:

# Referencias

Arciniegas, J. E. (29 de 04 de 2024). *YouTube*. Obtenido de Frames: https://www.youtube.com/watch?v=6hiRoq1Fmeg

es.javascript. (29 de 04 de 2024). *Mover eventos*. Obtenido de es.javascript: https://es.javascript.info/mouse-drag-and-drop

**Conclusiones:**

**Juan Jesús Sámano Juárez**:

Al realizar este proyecto fue un poco complicado el manejo de frames o también conocidos como marcos, después de haber solucionado este paso fue la elaboración de icono, los cuales en esta actividad son (gallina, zorro, semillas y bote) esto fue relativamente fácil de este punto la siguiente tarea era asignarles un evento de movimiento y arrastre para que el usuario pueda manejarlos y por último agregar las condicionales del juego.

**Padilla Perez Jorge Daray:**

Para concluir con esta actividad interesante ya que es raro tener que simular que una computadora actual tarde tanto en hacer ciertos procesos y más cuando es algo tan sencillo, el tema de tener varias ventanas fue todo un reto que el compañero Ernesto soluciono, la comunicación en equipo fue importante para la realización de esta actividad, aprendí juntos a mis compañeros y espero que podamos seguir así de comunicados.

**Ernesto Macias Flores:**

Durante la ejecución de este proyecto, nos encontramos con algunos desafíos al gestionar los marcos o frames. Una vez resueltos, nos enfocamos en el diseño de los íconos necesarios para la actividad: una gallina, un zorro, semillas y un bote. Este proceso resultó relativamente sencillo. A continuación, nos centramos en dotar a cada ícono con eventos de movimiento y capacidad de arrastre, brindando así al usuario el control necesario. Finalmente, completamos el proyecto integrando las condiciones del juego para una experiencia de usuario completa.