

Reporte del juego

Gato y Raton

Jose Manuel Bautista Morales.

10 de noviembre del 2023.



Índice

[Índice](#)

[Introducción](#)

[Desarrollo](#)

[1. Código](#)

● [HTML](#)

[1. <Canvas>](#)

[2. Modal](#)

● [CSS](#)

● [JS](#)

[1. Funcion app\(\)](#)

[2. Ejecución al Cargar la Página \(window.onload\):](#)

[2. Funcionamiento](#)

[Conclusión](#)



Introducción

El presente proyecto se centra en la creación de un juego interactivo de "Gato" utilizando tecnologías web como HTML, CSS y JavaScript. El juego está diseñado para dos jugadores que compiten por alinear tres de sus imágenes consecutivas en un tablero de 3x3.

La interfaz gráfica se ha implementado mediante el uso de la etiqueta `<canvas>` de HTML5, proporcionando un lienzo de 500x500 pixeles en el cual se representan las acciones del juego. La dinámica del juego se maneja a través de eventos del mouse, permitiendo a los jugadores realizar sus movimientos mediante clics en las celdas del tablero.

El código JavaScript controla el estado del juego, gestionando los turnos de los jugadores, validando movimientos válidos y determinando el ganador o empate. Además, se ha integrado una opción para mostrar u ocultar una rejilla que facilita la ubicación de las celdas en el tablero.

Para mejorar la experiencia del usuario, se implementó una función de reinicio que restablece el tablero y los contadores de victorias y empates. Estos últimos se presentan visualmente en la interfaz, proporcionando a los jugadores una visión clara del desempeño durante varias partidas.

El juego también incluye una función de detección de victoria que resalta visualmente la línea ganadora y presenta un mensaje emergente con el nombre del jugador ganador. Además, se ha incorporado un mensaje en caso de empate.

Desarrollo

1. Código

● HTML

Iniciando por mencionar que la mayoría de este código HTML está conformado por etiquetas que son familiares y básicas para la creación de páginas web tales como (`<!DOCTYPE html>` `<head>` `<body>` `<div>``</div>`, etc), que no son relevantes considerando la etapa actual de la carrera. Sin embargo, se destacará y explicaré las etiquetas más relevantes e importantes para el proyecto actual del juego del Gato y Ratón.

1. <Canvas>

```
<canvas id="Ejercicio01">  
  
</canvas>
```

La etiqueta `<canvas>` en el código HTML del proyecto representa el área gráfica donde se visualizará el juego del Gato y Ratón. Este lienzo actúa como una pizarra en blanco que se puede manipular dinámicamente a través de JavaScript para renderizar el tablero de juego, piezas y otros elementos visuales.

En este caso específico:

- **Atributo `id="Ejercicio01"`:** Proporciona una identificación única para el lienzo, lo que facilita referenciar y manipular este elemento en el código JavaScript.
- **Uso en el proyecto:** Se utilizó, a través del código JavaScript (archivo "index.js"), para que realice la manipulación de este lienzo para dibujar el tablero del juego, las fichas del Gato y Ratón, y cualquier otro elemento visual necesario para la experiencia de juego.

2. Modal

```
<!-- Modal -->
<div class="modal fade" id="modal_ganador" data-bs-backdrop="static" data-bs-keyboard="false" tabindex="-1">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="modal-title">¡Jugador ha ganado!</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"
          id="Continuar"></button>
      </div>
      <div class="modal-body">
        <!-- Contenido del mensaje -->
      </div>
      <div class="modal-footer">
        <button id="terminar" class="btn btn-danger" data-bs-dismiss="modal"> Terminar Juego</button>
        <br>
        <button id="Continuar" class="btn btn-success" data-bs-dismiss="modal"> Continuar</button>
      </div>
    </div>
  </div>
</div>
```

Un modal diseñado con Bootstrap que despliega un mensaje indicando al ganador del juego.

- CSS

En el desarrollo de este juego, opté por emplear Bootstrap como una herramienta integral para estilizar y diseñar la interfaz. Bootstrap, un popular marco de diseño de código abierto, proporciona una extensa biblioteca de estilos predefinidos y componentes reutilizables que facilitan la creación de interfaces web atractivas y responsivas.



La elección de Bootstrap me agilizó significativamente el proceso de diseño, permitiéndome implementar rápidamente elementos como botones, tarjetas y modales con una apariencia coherente y moderna. Además, la capacidad de respuesta incorporada de Bootstrap aseguró que la interfaz del juego se adaptara de manera efectiva a distintos tamaños de pantalla, mejorando así la experiencia del usuario en dispositivos móviles y de escritorio.

- JS

1. Función app()

```
index.js > ...  
> function app() { ...  
  }  
  
```

La función **app** es la función principal que inicializa y gestiona el juego del gato y ratón. Aquí hay una breve explicación de sus componentes:

1. **Contexto del Canvas:**

```
var canvas = document.getElementById("Ejercicio01");  
canvas.width = '500';  
canvas.height = '500';  
var ctx = canvas.getContext("2d");  
canvas.style.backgroundColor = '#b2e2f2';
```

Se obtiene el contexto del elemento canvas, que se utilizará para dibujar en la pantalla. También se establece el tamaño del canvas y su color de fondo.

2. **Botón de Rejilla:**

```
const botonDeRegilla = document.getElementById('botonDeRegilla');  
let regillaMostrada = false;  
// Función para mostrar u ocultar la rejilla  
function toggleRegilla() {  
  regillaMostrada = !regillaMostrada;  
  ctx.clearRect(0, 0, canvas.width, canvas.height);  
  reiniciarJuego();  
  botonDeRegilla.innerText = regillaMostrada ? 'Ocultar Regilla' : 'Mostrar Regilla';  
}  
botonDeRegilla.addEventListener("click", toggleRegilla);
```

Se configura un botón que permite mostrar u ocultar la rejilla en el canvas cuando se hace clic. La rejilla es una cuadrícula que ayuda a los jugadores a visualizar las posiciones en el tablero.

3. Imágenes Precargadas:

```
const images = {
  gato: new Image(),
  rata: new Image(),
};
images.gato.src = 'img/gato.jpg';
images.rata.src = 'img/rata.jpg';
```

Esta parte fue diseñada para cargar las imágenes del gato y la rata antes de que comience el juego. Esto se hace utilizando objetos de la clase `Image` de JavaScript, que permiten cargar imágenes de manera asíncrona.

- **Objeto de Imágenes:** Se crea un objeto llamado `images` que contendrá las imágenes del gato y la rata.
- **Instancias de la Clase `Image`:** Se crean instancias de la clase `Image()` para las imágenes del gato y la rata.
- **Rutas de las Imágenes:** Se establecen las rutas de las imágenes utilizando la propiedad `src`. Estas rutas apuntan a los archivos de imágenes correspondientes en el directorio 'img'.

4. Jugadores y Contadores:

```
const players = {
  jugador1: null,
  jugador2: null
};
players.jugador1 = document.getElementById('nombreJugador1').value;
players.jugador2 = document.getElementById('nombreJugador2').value;

const victoriasJugador1 = document.getElementById('vicjuagdor1');
const victoriasJugador2 = document.getElementById('vicjuagdor2');
const empatesElement = document.getElementById('empates');

let victoriasPlayer1 = 0;
let victoriasPlayer2 = 0;
let empates = 0;

function updateVictoryCounts() {
  victoriasJugador1.textContent = victoriasPlayer1;
  const empatesElement: HTMLElement | null = empatesElement;
  empatesElement.textContent = empates;
}
```

- **Objeto de Jugadores y Contadores:** Se crea un objeto llamado `players` que almacenará los nombres de los jugadores. Además, se establecen variables para referenciar elementos HTML que mostrarán los contadores de victorias.
- **Asignación de Nombres de Jugadores:** Los nombres de los jugadores se obtienen de los elementos de entrada HTML con los ID 'nombreJugador1' y 'nombreJugador2'.
- **Elementos para Contadores:** Se obtienen referencias a elementos HTML con ID 'vicjuagdor1', 'vicjuagdor2', y 'empates'. Estos elementos se utilizarán para mostrar el número de victorias de cada jugador y los empates.
- **Variables de Contadores:** Se crean variables `victoriasPlayer1`, `victoriasPlayer2`, y `empates` para realizar un seguimiento de las victorias de cada jugador y los empates.
- **Función de Actualización de Contadores:** La función `updateVictoryCounts` actualiza el contenido de los elementos HTML con los contadores actuales. Esto se llama después de cada juego para reflejar los cambios en la interfaz de usuario.

En conjunto, esta sección se encarga de gestionar los nombres de los jugadores y llevar un registro de sus victorias, todo mientras actualiza dinámicamente la interfaz de usuario para reflejar estos cambios.

5. Objeto del juego (gato):

```
const gato = { ...  
};
```


- **Objeto del Juego (gato):** Se crea un objeto llamado `gato` para representar el juego. Este objeto contiene propiedades y métodos relacionados con el juego del gato y el ratón .
- **Matriz de Estados:** La propiedad `estados` es una matriz 3x3 que representa el estado actual del tablero del juego, donde 0 indica un espacio vacío, 1 indica un movimiento del jugador 1 (gato), y 2 indica un movimiento del jugador 2 (ratón).
- **Dimensiones del Tablero:** Las propiedades `ancho` y `alto` definen las dimensiones del tablero de juego y se establecen en 500x500 píxeles.
- **Turno del Jugador:** La propiedad `turnoJugador` indica cuál de los jugadores tiene el turno actualmente, siendo 1 para el jugador 1 (gato) y 2 para el jugador 2 (ratón).
- **Cuadro Resaltado:** La propiedad `cuadroResaltado` almacena la posición del cuadro que está resaltado durante el juego.
- **Métodos del Juego:** Se definen varios métodos que gestionan aspectos clave del juego, como activar un estado en respuesta a un clic, seleccionar un cuadro y analizar el estado actual para determinar si hay un ganador o empate.
- **Función de Juego Principal (play):** Esta función se encarga de inicializar y dibujar el escenario del juego, incluyendo la regilla si está activada.

En conjunto, el objeto `gato` encapsula la lógica y el estado del juego, proporcionando métodos para interactuar con él y mantener un seguimiento del progreso del juego.

6. Manejo de Eventos:

```

> canvas.addEventListener('mousedown', function (event) { ...
> });
> canvas.addEventListener('mousemove', function (event) { ...
> });
> function nombres() { ...
> };
> nombres();
> function reiniciarJuego() { ...
> }
> const botonReinicio = document.getElementById('Reinicio');
> botonReinicio.addEventListener('click', reiniciarJuego);
> const continuar = document.getElementById("Continuar");
> continuar.addEventListener('click', reiniciarJuego);
>
> const botonTerminar = document.getElementById('terminar');
> botonTerminar.addEventListener('click', function () { ...
> });
>
> gato.play();

```

- **Event Listeners:** Se registran dos eventos en el elemento `canvas` para escuchar clics y movimientos del mouse. Estos eventos están asociados a los métodos `activarEstado` y `seleccionar` del objeto `gato`.
- **Función `nombres`:** Esta función obtiene los nombres de los jugadores ingresados en los campos de entrada (`nombreJugador1` y `nombreJugador2`) y los muestra en los elementos HTML correspondientes (`name1` y `name2`).
- **Función `reiniciarJuego`:** Esta función se encarga de reiniciar el juego, limpiando el lienzo, reiniciando el estado del juego y dibujando el escenario nuevamente.
- **Event Listener para Reiniciar el Juego:** Se asigna un event listener al botón con el id `Reinicio` para que llame a la función `reiniciarJuego` cuando se haga clic en él.
- **Event Listener para Continuar el Juego:** Se asigna un event listener al botón con el id `Continuar` para que también llame a la función `reiniciarJuego` cuando se haga clic en él.
- **Event Listener para Terminar el Juego:** Se asigna un event listener al botón con el id `terminar`. Este botón cierra el modal y recarga la página, finalizando el juego.

- **Inicio del Juego (`gato.play`):** Se llama al método `play` del objeto `gato` para iniciar el juego, dibujando el escenario inicial y la regilla si es necesario.

7. **Funciones Auxiliares:** Se definen funciones adicionales, como `nombres()` para mostrar los nombres de los jugadores y `reiniciarJuego()` para reiniciar el juego.

8. **Inicio del Juego:** La función `app` se llama al cargar la página, iniciando así el juego.

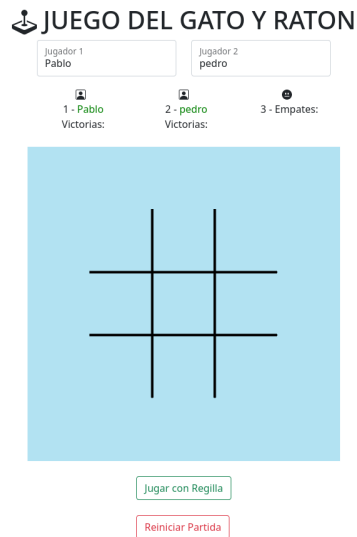
Esta función actúa como un punto de entrada para la lógica del juego, conectando los elementos del HTML con la funcionalidad del juego y coordinando las interacciones del usuario.

2. Ejecución al Cargar la Página (`window.onload`):

```
50 window.onload = function () {  
51     app();  
52 };  
53
```

1. **Evento `window.onload`:** Este código se ejecuta cuando la página web se carga por completo. El evento `window.onload` garantiza que la función `app` se ejecute después de que todos los recursos de la página, como imágenes y estilos, se hayan cargado.
2. **Llamada a la Función `app`:** La función `app` se invoca cuando la página se ha cargado por completo. En este caso, `app` contiene toda la lógica del juego del gato y ratón, y su ejecución se inicia cuando la página está lista para su interacción.
3. **Inicio del Juego:** La función `app` configura y comienza la ejecución del juego una vez que todos los elementos de la página están disponibles.

2. Funcionamiento



1. Interfaz de Usuario:

- Los jugadores ingresan sus nombres en los campos designados.
- Se visualizan los contadores de victorias y empates.
- El tablero de juego se presenta en un canvas con líneas y límites.

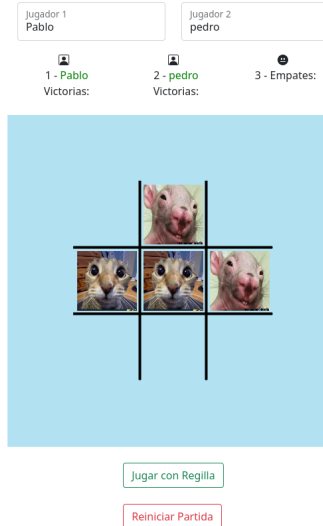
2. Inicio y Reinicio del Juego:

- Al iniciar o reiniciar el juego, se limpia el tablero y se restablece el estado inicial.
- Los nombres de los jugadores se actualizan.

3. Turnos y Jugadas:

- Los jugadores hacen clic en un espacio del tablero para realizar su jugada.
- El juego alterna entre el gato y la rata para cada turno.
- Se visualiza la imagen correspondiente al jugador en el espacio seleccionado.

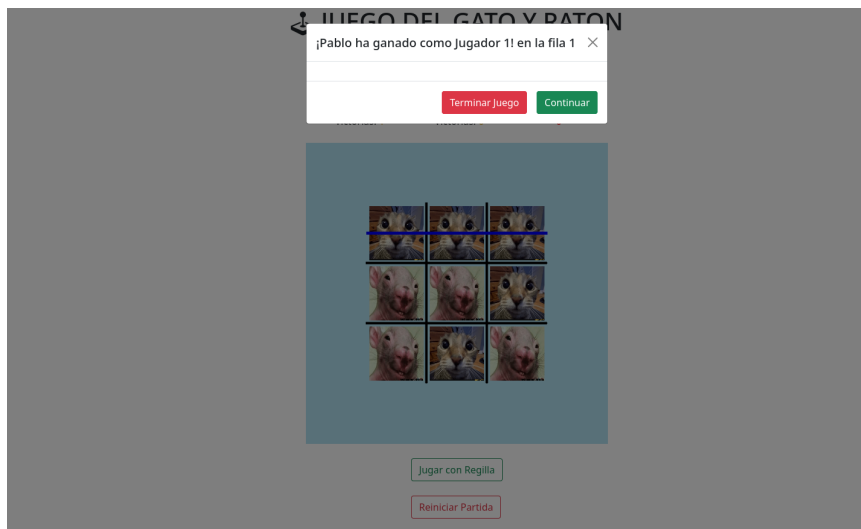
JUEGO DEL GATO Y RATON



4. Análisis de la Partida:

- Después de cada jugada, se verifica si hay un ganador o un empate.
- Se comprueba en filas, columnas y diagonales si hay tres elementos del mismo jugador.
- Si se cumple alguna condición de victoria o empate, se muestra un modal con el resultado.

5. Modal de Resultado:



- Si hay un ganador, se muestra el nombre del jugador y los detalles de la victoria.
- Si es un empate, se indica que la partida terminó sin un ganador.

- Se actualizan los contadores de victorias y empates.

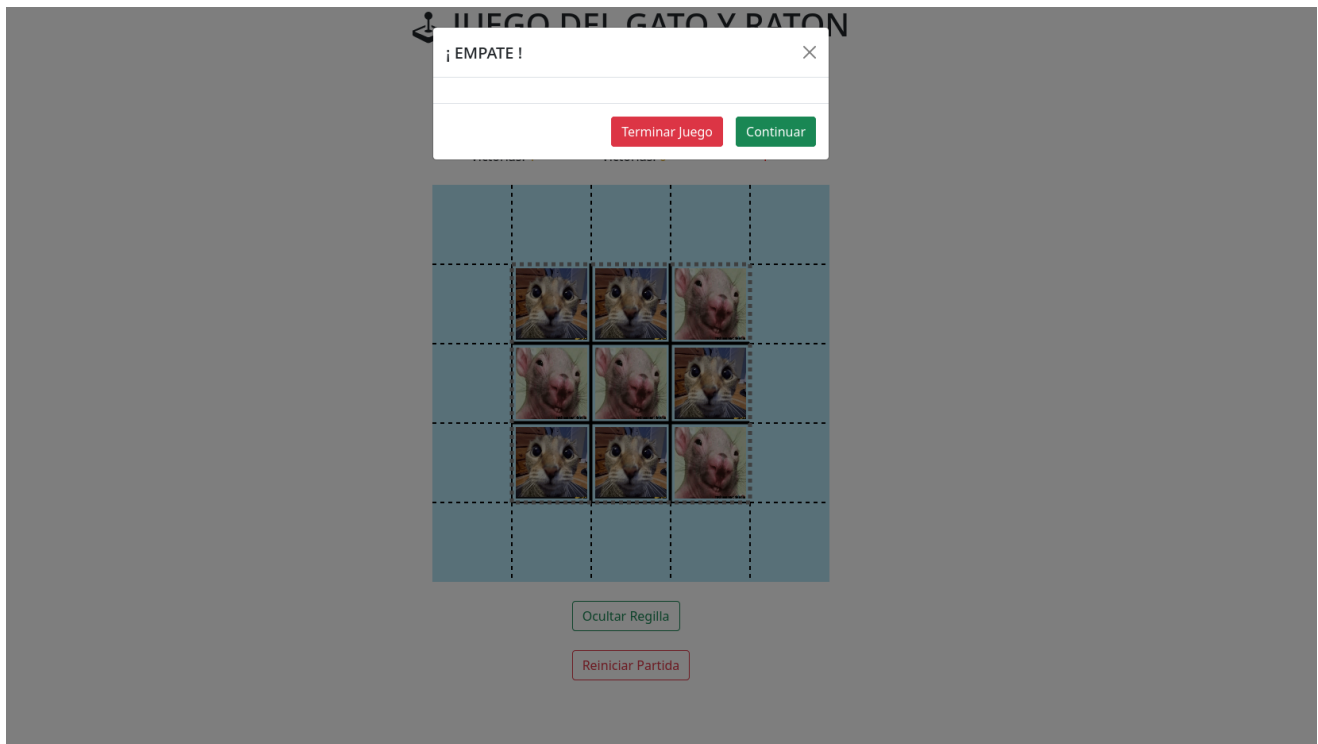


6. Continuación y Reinicio:

- El modal permite al jugador continuar o reiniciar la partida.
- Los contadores se actualizan según la decisión del jugador.

7. Terminación del Juego:

- Existe la opción de terminar el juego completamente, recargando la página y restableciendo todas las estadísticas.





Conclusión

En conclusión, el desarrollo del juego del Gato ha sido un proyecto interactivo y educativo que me ha permitido explorar y aplicar conceptos fundamentales de programación web. A través de la combinación de HTML, CSS y JavaScript, se ha logrado crear una experiencia de juego que es hasta un punto jugable.

El uso de la etiqueta `<canvas>` ha demostrado ser una herramienta poderosa para la representación gráfica en tiempo real, proporcionando un lienzo dinámico en el cual se llevan a cabo las acciones del juego. La detección de eventos del mouse ha sido esencial para permitir la interactividad del usuario, facilitando la selección de celdas y la realización de movimientos.

En resumen, este proyecto ha sido una oportunidad valiosa para aplicar y consolidar conocimientos en desarrollo web. La experiencia adquirida en este proyecto sienta las bases para futuras exploraciones y desarrollos en el ámbito de la programación web y el diseño de juegos interactivos.