Manual Técnico del Juego Tres en Raya

Este documento explica la implementación técnica del juego de Tres en Raya (también conocido como Tic Tac Toe). Se detallan tanto los archivos HTML, CSS y JavaScript que componen el juego, proporcionando una visión clara de la estructura y la lógica de la aplicación.

Índice

1. Estructura HTML
2. Estilos CSS
3. Lógica JavaScript
    1. Variables globales
    2. Eventos y Funciones Principales
    3. Funciones de la IA
    4. Funciones de finalización y reinicio del juego

Estructura HTML

El archivo `index.html` define la estructura básica del juego. Los elementos principales son:

- **Encabezados y Mensajes de Información:**
    - `<h1>`: El título principal del juego "TRES EN RAYA".
    - `<h2 id="modo-juego">`: Muestra el modo de juego seleccionado ("2 Jugadores" o "Jugador vs IA").
    - `<p id="turno">`: Informa a los jugadores de quién es su turno.
    - `<h2 id="resultado">`: Muestra el resultado del juego (victoria, derrota, empate).
- **Botones de Selección de Modo de Juego:**
    - `<button id="btn-jugador-vs-jugador">`: Inicia el juego en modo "2 Jugadores".
    - `<button id="btn-jugador-vs-ia">`: Inicia el juego en modo "Jugador vs IA".
- **Tablero de Juego:**
    - `<div id="tablero">`: Contenedor para las casillas del tablero (9 botones).
    - Cada casilla se representa como un `<button>` dentro de este contenedor.
- **Botón de Reinicio:**
    - `<button id="btn-reiniciar">`: Botón para reiniciar el juego.

La estructura HTML es simple y contiene los elementos necesarios para jugar, controlar los modos y reiniciar el juego.

Estilos CSS

El archivo `estilo.css` define el diseño visual del juego. Aquí se detallan los aspectos clave:

- **Colores y Fuentes:**

  - Fondo negro para el cuerpo del juego (`background-color: black`).
  - Títulos en color `chartreuse` para resaltar los encabezados.

- **Tablero y Botones:**
  - El tablero está oculto inicialmente con `display: none` y se muestra al iniciar el juego (`display: flex`).
  - Las casillas son botones (`<button>`) de 190px de ancho y alto, con un fondo negro y texto blanco.
  - Las casillas ganadoras se resaltan con una clase `.winning` que cambia el fondo a `chartreuse` y el texto a negro.
  - Los botones interactivos, como los de selección de modo de juego y el botón de reinicio, tienen bordes redondeados y efectos de hover (cambio de color al pasar el mouse).

- **Estado Final del Juego:**
  - El mensaje final se muestra en un tamaño más grande con diferentes colores según el resultado (victoria en `chartreuse`, derrota en `red`).

Lógica JavaScript

El archivo `script.js` contiene la lógica principal del juego. Aquí se controlan todos los eventos y la funcionalidad de la partida, desde el manejo de los turnos hasta la inteligencia artificial (IA) y la verificación de ganadores.

Variables globales

1. **Elementos del DOM:**
   - `tablero`: Contenedor del tablero de juego.
   - `resultado`: Muestra el resultado del juego (victoria, derrota, empate).
   - `turno`: Muestra de quién es el turno (Jugador 1, Jugador 2, IA).
   - `btnReiniciar`: Botón para reiniciar el juego.
   - `btnJugadorVsJugador` y `btnJugadorVsIA`: Botones para seleccionar el modo de juego.
   - `modoJuego`: Muestra el modo de juego elegido ("JUGADOR VS JUGADOR" o "JUGADOR VS IA").
   - `botonesOpciones`: Contenedor de los botones para seleccionar el modo de juego.
2. **Estado del Juego:**
   - `casillas`: Array que mantiene el estado de cada una de las 9 casillas del tablero. Inicialmente todas están vacías (`null`).
   - `esTurnoX`: Booleano que indica si es el turno de "X" (Jugador 1) o "O" (Jugador 2 o IA).
   - `juegoActivo`: Booleano que indica si el juego está en curso.
   - `tipoJuego`: Almacena el tipo de juego ("jugador" o "ia").
3. **Combinaciones Ganadoras:**
   - `combinacionesGanadoras`: Array de arrays que define las posiciones ganadoras del tablero. Incluye filas, columnas y diagonales.

Eventos y Funciones Principales

1. **Selección del Modo de Juego:**
    - **Eventos:**
        - `btnJugadorVsJugador.addEventListener("click", () => iniciarJuego("jugador"))`: Cuando se selecciona "2 Jugadores", se inicia el juego en modo jugador contra jugador.
        - `btnJugadorVsIA.addEventListener("click", () => iniciarJuego("ia"))`: Cuando se selecciona "Jugador vs IA", se inicia el juego en modo jugador contra IA.
    - **Función `iniciarJuego(modo)`**:
        - Establece el tipo de juego (jugador vs jugador o jugador vs IA).
        - Muestra el tablero de juego y el botón de reinicio.
        - Si es modo IA, determina si la IA empieza o no.
        - Cambia el estado del juego a activo y asigna el primer turno.
2. **Manejo del Turno:**
    - **Función `actualizarTurno()`**:
        - Actualiza el mensaje que muestra de quién es el turno. Si es "X" o "O", o "IA".
3. **Juego en el Tablero:**
    - **Evento `tablero.addEventListener("click", (e) => {...})`**:
        - Detecta cuando el jugador hace clic en una casilla. Si la casilla está vacía, la marca con "X" o "O", dependiendo del turno.
        - Después de cada jugada, se verifica si hay un ganador o si el juego ha terminado en empate.
        - Si hay un ganador, se llama a la función `finalizarJuego()`. Si no, cambia el turno.
4. **Verificación de Ganador:**
    - **Función `verificarGanador()`**:
        - Recorre las combinaciones ganadoras y verifica si alguna de ellas está llena con el mismo símbolo. Si encuentra una combinación ganadora, marca las casillas ganadoras y retorna `true`.
        - Si no hay ganador, retorna `false`.

Funciones de la IA

1. **Función `turnoIA()`**:
    - Es llamada cuando es el turno de la IA. La IA primero intenta buscar una jugada ganadora, luego intenta bloquear una jugada ganadora del jugador, y si no hay jugadas críticas, hace una jugada aleatoria.
    - La IA juega con el símbolo "O".
2. **Función `buscarJugadaGanadora(simbolo)`**:
    - Analiza las combinaciones ganadoras para ver si hay una oportunidad de ganar o bloquear al jugador. Si encuentra una casilla vacía en una combinación ganadora, la retorna.
3. **Función `realizarJugadaIA(indice)`**:

- Realiza la jugada de la IA en la casilla indicada. Actualiza el tablero y verifica si la IA ha ganado o si es empate.

Funciones de Finalización y Reinicio del Juego

1. **Función** `finalizarJuego(ganador)`:
   - Cuando el juego termina (ya sea por victoria, derrota o empate), esta función se encarga de mostrar el resultado final en la interfaz.
   - Si hay un ganador, se muestra el nombre del ganador. Si la IA gana, se muestra un mensaje de derrota.
   - Si no hay ganador, muestra el mensaje "¡EMPATE!".
2. **Evento** `btnReiniciar.addEventListener("click", () => {...})`:
   - Resetea el juego a su estado inicial, limpiando el tablero y restableciendo las variables de estado.

Resumen de la Lógica del Juego

El flujo principal de la lógica del juego es:

1. El jugador selecciona el modo de juego (2 Jugadores o Jugador vs IA).
2. El juego se inicia, y el turno comienza con el Jugador 1 (si es contra otro jugador) o con la IA (si es contra la máquina).
3. Los jugadores hacen clic en las casillas para marcar su símbolo ("X" o "O").
4. Se verifica si hay un ganador después de cada jugada. Si hay un ganador, el juego termina y se muestra el resultado.
5. En caso de empate (todas las casillas ocupadas sin ganador), también se termina el juego y se muestra un mensaje de empate.
6. El botón de reinicio permite resetear el juego a su estado inicial.

Cada parte del juego está controlada por una función específica que se llama en el momento adecuado, y la interacción entre estas funciones asegura que el juego funcione correctamente.