

Juego de la Oca

EJERCICIO PRÁCTICO

EMDEVAPPS

1. Introducción

- Breve descripción del juego de la Oca implementado.
- Objetivo del documento.

2. Estructura del código

- Descripción del código HTML, CSS y JavaScript JQuery.
- Explicación de la estructura del juego en el DOM.

3. Funciones principales

- Asignación de turnos (``asignaturnos``)
- Construcción de objetos (``construyeobjetos``)
- Generación de selecciones de ficha (``generarSeleccionesDeFicha``)
- Mostrar fichas seleccionadas (``mostrarFichasSeleccionadas``)
- Mostrar dados seleccionados (``mostrarDadosSeleccionados``)
- Lanzamiento de dados para cada ficha (``dadosroja``, ``dadosazul``, ``dadosverde``, ``dadosamarilla``)
- Verificación del fin de la partida (``verificarFinPartida``)
- Movimiento de ficha en el tablero (``moverFicha``)
- Actualización de resultados en tiempo real (``actualizarResultados``)

El trabajo realizado aborda la implementación de un juego de la Oca mediante tecnologías web como HTML, CSS y JavaScript. El código se estructura de manera modular, con funciones específicas para gestionar diferentes aspectos del juego. A continuación, se detallan las principales funciones implementadas:

Asignación de turnos (asignaturnos):

Esta función se encarga de asignar los turnos a las fichas en juego. Utiliza un conjunto de variables para controlar el estado del juego, como fichas, podium, fichasoperativas e indiceFichaActual. Además, oculta y muestra los botones de lanzamiento de dados según el turno de cada ficha. La función también actualiza un mensaje en el DOM para informar al jugador cuyo turno es.

```
const asignaturnos = () => {
  fichasoperativas = fichas.filter((ficha) => !podium.includes(ficha));
  var fichaActual = fichasoperativas[indiceFichaActual];

  // Verificar si fichaActual es undefined y ajustar el índice en consecuencia
  if (fichaActual === undefined) {
    // Restablecer el índice a 0 cuando se alcanza el final del array
    indiceFichaActual = 0;
    fichaActual = fichasoperativas[indiceFichaActual];
  } else {
    // Incrementar el índice normalmente
    indiceFichaActual = (indiceFichaActual + 1) % fichasoperativas.length;
  }

  // Ocultar todos los divs de turno
  document.getElementById("botonrojoturno").style.display = "none";
  document.getElementById("botonazulturno").style.display = "none";
  document.getElementById("botonverdeturno").style.display = "none";
  document.getElementById("botonamarillaturno").style.display = "none";

  // Mostrar el div de turno correspondiente a la ficha actual
  if (fichaActual === "rojo") {
    document.getElementById("botonrojoturno").style.display = "block";
  } else if (fichaActual === "azul") {
    document.getElementById("botonazulturno").style.display = "block";
  } else if (fichaActual === "verde") {
    document.getElementById("botonverdeturno").style.display = "block";
  } else if (fichaActual === "amarilla") {
    document.getElementById("botonamarillaturno").style.display = "block";
  }

  // Verificar si la ficha actual es undefined
  if (fichasoperativas.length === 0) {
    $("#turnos").text("Ha terminado la partida");
  } else {
    // Si no es undefined, mostrar el turno de la ficha actual
    $("#turnos").text("Es el turno de " + fichaActual);
  }

  // Llamar a las funciones para mostrar las fichas y los dados después de actualizar la ficha actual
  mostrarFichasSeleccionadas();
  mostrarDadosSeleccionados();

  return fichaActual;
};
```

Construcción de objetos (construyeobjetos):

Se encarga de construir objetos JavaScript que representan a cada jugador en el juego. Utiliza los datos almacenados en los arrays fichas, jugadores y los resultados de cada ficha (resultadoRojo, resultadoAzul, etc.). Estos objetos se utilizan para actualizar los resultados en tiempo real y para gestionar el movimiento de las fichas en el tablero.

```

const construyeobjetos = () => {
  // este pequeño objeto tiene la posición en la cual se encuentra cada ficha.

  var fichaposicion = {
    verde: resultadoVerde,
    azul: resultadoAzul,
    amarilla: resultadoAmarillo,
    rojo: resultadoRojo,
  };

  for (var i = 0; i < fichas.length; i++) {
    var jugador = {
      jugador: jugadores[i],
      colorFicha: fichas[i],
      posicion: fichaposicion[fichas[i]],
    };
    // Asigna un nombre único al objeto jugador
    var nombreJugador = "jugador" + (i + 1);
    // Asigna el objeto jugador al nombre correspondiente
    window[nombreJugador] = jugador;
  }

  // Comprueba si cada jugador está definido y luego imprime en la consola
  for (var i = 1; i <= fichas.length; i++) {
    var nombreJugador = "jugador" + i;
    if (window[nombreJugador]) {
      console.log(window[nombreJugador]);
    }
  }
};

```

Generación de selecciones de ficha (generarSeleccionesDeFicha):

Esta función genera dinámicamente los selectores HTML para que cada jugador elija el color de su ficha al inicio del juego. Utiliza el número de jugadores seleccionado para crear los campos necesarios en el DOM.

```

function generarSeleccionesDeFicha() {
  $("#seleccionFichas").empty();
  for (var i = 0; i < numJugadores; i++) {
    var selectFicha = $("").addClass("fichaSeleccionada");
    selectFicha.append($("<option>").text("Seleccione ficha").val(""));
    selectFicha.append($("<option>").text("Rojo").val("rojo"));
    selectFicha.append($("<option>").text("Azul").val("azul"));
    selectFicha.append($("<option>").text("Verde").val("verde"));
    selectFicha.append($("<option>").text("Amarilla").val("amarilla"));

    var jugadorText = $("

").text("Jugador " + (i + 1));
    $("#seleccionFichas").append(jugadorText);
    $("#seleccionFichas").append(selectFicha);
  }
}


```

Mostrar fichas seleccionadas (mostrarFichasSeleccionadas):

Es responsable de mostrar las fichas seleccionadas por los jugadores en el tablero. Recorre el array fichas y muestra las fichas correspondientes en el DOM.

```
// en el evento onchange, hacemos la asignación de ficha seleccionada para ir rellenando el array fichas.

$(document).on("change", ".fichaSeleccionada", function () {
    var index = $(".fichaSeleccionada").index(this);
    var fichaSeleccionada = $(this).val();
    fichas[index] = fichaSeleccionada;
});

// Cuando se hace clic en el boton con identificador "iniciarpartida" iniciar partida se llaman a estas funciones

$("#iniciarpartida").click(function () {
    mostrarFichasSeleccionadas();
    mostrarDatosSeleccionados();
});

// Esta función me va a mostrar en mi documento html únicamente las fichas presentes en el array fichas

function mostrarFichasSeleccionadas() {
    $(".ficharoja, .fichaazul, .fichaverde, .fichaamarilla").hide();
    for (var i = 0; i < fichas.length; i++) {
        if (fichas[i] === "rojo") {
            $(".ficharoja").show();
        } else if (fichas[i] === "azul") {
            $(".fichaazul").show();
        } else if (fichas[i] === "verde") {
            $(".fichaverde").show();
        } else if (fichas[i] === "amarilla") {
            $(".fichaamarilla").show();
        }
    }
}
```

Mostrar datos seleccionados (mostrarDatosSeleccionados):

Similar a la función anterior, muestra los botones de lanzamiento de dados para cada ficha en el tablero. Utiliza el array fichas para determinar qué botones deben mostrarse.

```
// esta función me crea tantos jugadores como fichas existan
// Un ejemplo, si tengo fichas [azul, verde, rojo] me creará un array jugadores así--> [jugador1, jugador2, jugador3]

function mostrarDatosSeleccionados() {

    for (var i = 0; i < fichas.length; i++) {
        var jugador = "jugador" + (i + 1);
        jugadores.push(jugador);
    }

    $(".botonrojo, .botonazul, .botonverde, .botonamarilla").hide();

    for (var i = 0; i < fichas.length; i++) {
        var ficha = fichas[i];
        $(".boton" + ficha).show();
    }
}

});
```

Lanzamiento de dados para cada ficha (dadosroja, dadosazul, dadosverde, dadosamarilla):

Estas funciones gestionan el lanzamiento de los dados para cada ficha. Calculan un número aleatorio entre 1 y 6, actualizan el resultado de la ficha correspondiente y gestionan el movimiento en el tablero. Si una ficha alcanza la meta, se agrega al podium y se elimina del juego.

```
// Esta función es la encargada de lanzar los dados para la ficha roja
// Si el resultadoRojo es mayor que 20, nos cambia el resultado a -->"meta"
// además esta función llama a muchas otras funciones, como la de asignarturno(), construyeobjetos()....
// Estas funciones son elementales para ir actualizando en tiempo real los resultados.

// Despues de dadosroja(), tenemos dadosazul(), etc. Todas las funciones hacen lo mismo
// ¿Podríamos haber hecho una sola función y pasarle por parámetros la ficha? si.

function dadosroja() {
  if (resultadoRojo !== "meta") {
    var numeroAleatorio = Math.floor(Math.random() * 6) + 1;
    resultadoRojo += numeroAleatorio;
    if (resultadoRojo > 20) {
      resultadoRojo = "meta";
      podium.push("rojo");
      fichasoperativas = fichasoperativas.filter(
        (ficha) => ficha !== "rojo"
      );
      console.log(fichasoperativas);
    }
  }
  construyeobjetos();
  moverFicha(".ficharoja", numeroAleatorio);
  actualizarResultados();
  verificarFinPartida();
  asignaturnos();
}
```

Verificación del fin de la partida (verificarFinPartida):

Esta función comprueba si todas las fichas han alcanzado la meta y, en ese caso, muestra el podio con el orden de llegada de las fichas. También actualiza un mensaje en el DOM para indicar que la partida ha finalizado.

```
//Esta función nos verifica si la partida ya terminó o no, para poder mostrar el podium

function verificarFinPartida() {
  var totalFichas = fichas.length;
  var fichasMeta = 0;

  if (resultadoRojo === "meta") {
    fichasMeta++;
  }
  if (resultadoAzul === "meta") {
    fichasMeta++;
  }
  if (resultadoVerde === "meta") {
    fichasMeta++;
  }
  if (resultadoAmarillo === "meta") {
    fichasMeta++;
  }

  if (fichasMeta === totalFichas) {
    document.getElementById("partidafinalizada").style.display = "flex";
    mostrarPodium();
  }
}
```

Movimiento de ficha en el tablero (moverFicha):

Animación que simula el movimiento de una ficha en el tablero. Calcula la nueva posición de la ficha en función del resultado de los dados y actualiza su posición en el DOM con una animación suave.

```
// esta función es la que se encarga de mover las fichas, hasta ahora es la única función que logra ANIMAR, nuestro documento.
// Nos mueve hacia la derecha unos pixeles directamente relacionados con el numero aleatoreo generado cuando hemos lanzado dados.

function moverFicha(selector, numero) {
    var distancia = numero * 55;
    var posicionActual = parseFloat($(selector).css("left"));
    var nuevaPosicion = posicionActual + distancia;
    nuevaPosicion = Math.min(nuevaPosicion, 1150);
    $(selector).animate(
        {
            left: nuevaPosicion + "px",
        },
        1000,
        function () {
            if (nuevaPosicion >= 1150) {
                var color = $(selector).attr("class").split(" ")[0];
                alert("¡Ha llegado a la meta la " + color + "!");

                var partidafinalizada = 1;

                if (partidafinalizada === 1) {
                    $("#resultadoRojo").text(resultadoRojo);
                    $("#resultadoAzul").text(resultadoAzul);
                    $("#resultadoVerde").text(resultadoVerde);
                    $("#resultadoAmarilla").text(resultadoAmarilla);
                }
            }
        }
    );
}
```

Actualización de resultados en tiempo real (actualizarResultados):

Esta función actualiza la tabla de resultados en el DOM con la información más reciente sobre la posición de cada ficha. Utiliza los objetos jugador construidos previamente para obtener esta información.

En resumen, el código implementa una serie de funciones que permiten gestionar diferentes aspectos del juego de la Oca, desde la asignación de turnos hasta la actualización de resultados en tiempo real, ofreciendo una experiencia de juego dinámica y completa para los usuarios.

```
//Esta función nos actualiza en tiempo real lo que hemos pintado en el dom, y cada vez que se llama, se pinta de nuevo con los nuevos
//valores, por ejemplo si ha cambiado (jugador.posicion), se pinta con ese cambio en el dom.

function actualizarResultados() {
    var resultadosHTML = "<table>";
    for (var i = 1; i <= fichas.length; i++) {
        var nombreJugador = "jugador" + i;
        if (window[nombreJugador]) {
            var jugador = window[nombreJugador];
            resultadosHTML += "<tr>";
            resultadosHTML += "<td>Jugador " + i + "</td>";
            resultadosHTML += "<td>Color: " + jugador.colorFicha + "</td>";
            resultadosHTML += "<td>Posición: " + jugador.posicion + "</td>";
            resultadosHTML += "</tr>";
        }
    }
    resultadosHTML += "</table>";
    $("#resultadotiemporeal").html(resultadosHTML);
}
```