

PROYECTO ZOMBIES PRTY



M2 UF2: Desenvolupament
web en entorn client

Oscar Jaimez Clavero
Antonio Gálvez Madrid
Profesor: Jorge Rubio

índice

FUNCIONES IMPORTANTES.....	3
DIFICULTADES ENCONTRADAS.....	5
COMENTARIOS POSITIVOS DEL CÓDIGO.....	6

FUNCIONES IMPORTANTES

Las funciones a destacar por su importancia en la aplicación son estas:

```
//crea la tabla
pintar_tauler: function(mida_tauler){
  let taula = "<div id='taulaGeneral'>";

  for (let i = 0; i < mida_tauler ; i++) {
    taula += "<div id='fila'>";
    for (let j = 0; j < mida_tauler ; j++) {
      taula += "<div id='" + (i+1) + "," + (j+1) + "' class='casilla' ><img src='img/cespedoculto.jpg'></div>";
    }
    taula += "</div>";
  }
  taula += "</div>";

  // dentro de la variable taula esta creado con divs la tabla por lo tanto con innerHTML meteremos eso dentro de un div con Id especifico
  document.getElementById('tabla').innerHTML = taula;
},
```

Donde creamos el tablero y ponemos a cada casilla un id único.

Otra función importante es donde comprobamos que hace cuando se selecciona una de las casillas:

```
mirarLetra: function(tipo, valorX, valorY){

  console.log(this.tauler);
  partida.casillaEscogidas++;

  switch (tipo.toUpperCase()) {
    case "D":
      valorX--;
      valorY--;
      for (let i = 0; i < partida.doblePunts.length; i++) {
        //Si encuentra en esa posicion que tiene las mismas coordenadas que yo suma las estadísticas
        if (valorX == partida.doblePunts[i].x && valorY == partida.doblePunts[i].y) {
          partida.pdobles_encontrados++;
          partida.doblePunts[i].seleccionado = true;
        }
      }
      //Dobla la puntuación
      this.puntos_totales = this.puntos_totales * 2;
      this.Estadisticas();

    case "M":
      valorX--;
      valorY--;
      for (i = 0; i < partida.meitatZombis.length; i++) {
        var objeto = partida.meitatZombis[i];
        if (valorX == objeto.x && valorY == objeto.y || //centro
            valorX - 1 == objeto.x && valorY == objeto.y || //izquierda
            valorX + 1 == objeto.x && valorY == objeto.y || //derecha
            valorX == objeto.x && valorY - 1 == objeto.y || //abajo
            valorX == objeto.x && valorY + 1 == objeto.y) { //arriba
          objeto.casillas++;
          objeto.seleccionado = true;
          if (objeto.casillas == 2) {
            partida.mitadZombis_encontrados++;
            partida.eliminarMitadZombis();
          }
        }
      }
      this.Estadisticas();
  }
}
```

Que pasa cuando se selecciona.

Y por ultimo queremos también mostrar las funciones donde creamos las recompensas, y cada una de ellas:

```
crear_recompensas: function() {
    //25% del total del tauler tendran que ser recompensas
    var total = ((this.mida_casselles_tauler* 25) / 100);
    this.crear_VidaExtra();
    this.crear_MitadZombie();
    this.crear_DoblePuntos();

    while (this.recompensas_creadas < total) {
        this.crear_VidaExtra();
        this.crear_MitadZombie();
        this.crear_DoblePuntos();
    }
},

crear_DoblePuntos: function() {

    try {
        var doblesPuntos = new DoblePuntuacion (1, 0);
        var x = 0,
            y = 0;
        do {
            x = Math.floor(Math.random() * this.mida_tauler);
            y = Math.floor(Math.random() * this.mida_tauler);
        } while (this.tauler[x][y] != "g");
        doblesPuntos.x = x;
        doblesPuntos.y = y;
        this.tauler[x][y] = "d";
        this.recompensas_creadas += 1;
        this.doblePunts.push(doblesPuntos);
    } catch (e) {}
},
```

DIFICULTADES ENCONTRADAS

Al ser nuestro primer proyecto planteándolo orientado a objetos ha sido todo un reto, al ser una aplicación un poco grande nos ha costado pensar como poder hacerla orientada a objetos, así que nos volvimos a ver las clases online grabadas y nos fue de gran ayuda.

Uno de los mayores problemas que hemos tenido ha sido el no leer palabra por palabra la documentación del proyecto, ya que no habíamos leído bien que no se podía utilizar el createElement, y nosotros al principio lo utilizamos, cuando ya avanzamos el proyecto lo releímos y vimos como no se podía utilizar, eso nos frenó bastante, ya que tuvimos que cambiar bastantes cosas, por ejemplo teníamos identificadas las casillas con un solo id, y nos amoldamos a la idea, pero al ver que no podíamos utilizar el createElement tuvimos que cambiarlo a identificar las posiciones con X, Y, te adjuntamos como lo hicimos con createElement por si te da curiosidad jaja...

```
var tabla = document.getElementById("tabla");
var tbody = document.createElement("tbody");
var valor = 1;
for (var i = 0; i < casillas; i++) {
    var fila = document.createElement("tr");
    for (var j = 0; j < casillas; j++) {
        var celda = document.createElement("td");
        var imagenCelda = document.createElement("img");
        //var textoCelda = document.createTextNode(valor);
        celda.setAttribute("id", valor);
        valor++;
        imagenCelda.setAttribute("src", "/img/cespedoculto.jpg");
        celda.appendChild(imagenCelda);
        //celda.appendChild(textoCelda);
        fila.appendChild(celda);
    }
    tbody.appendChild(fila);
}
tabla.appendChild(tbody);
tabla.setAttribute("border", "2");
```

COMENTARIOS POSITIVOS DEL CÓDIGO

Creemos que nuestro código destaca en lo limpio, claro y conciso que está escrito, este proyecto aparte de tabular todo correctamente, que tengan las variables y funciones nombres lógicos hemos implementado una mejora necesaria que es la de los comentarios, hemos comentado cada función para saber de qué trata, ya que al tener ya un cierto nivel de complejidad si no tuviéramos estas buenas costumbres sería imposible poder entender el código.

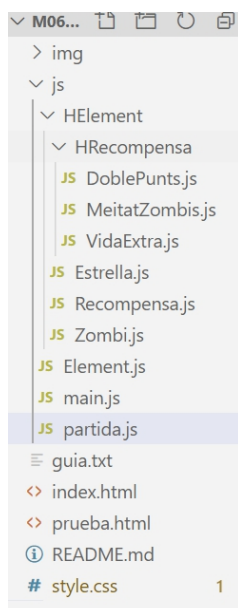
```
// Nos llenara la tabla con cesp d y segun la medida que nos diga el usuario
```

```
inicialitzar_tauler: function() {  
  for (let i = 0; i < this.mida_tauler; i++) {  
    this.tauler[i] = [];  
    for (let j = 0; j < this.mida_tauler; j++) {  
      /* iniciar tauler */  
      this.tauler[i][j]='g';  
    }  
  }  
  console.log(this.tauler);  
}
```

```
// Hace aparecer los 3 div ocultos
```

```
document.getElementById("gamediv").style.display = "block";  
document.getElementById("statsdiv").style.display = "block";  
document.getElementById("coordenadas").style.display = "block";  
// Hace desaparecer el div del valor de tablero  
document.getElementById("textdivinput").style.display = "none";
```

Tambi n nos gustar a destacar el orden de los ficheros de js.:



Donde se ve como hemos ordenado los ficheros

Con los nombres correctos y en carpetas/sub carpetas.

También hemos organizado el fichero de tal manera que estén todas las declaraciones: variables, objetos, etc. Al principio, para que abajo se vean las funciones sin nada que pueda despistar.⁹

```
let partida = {  
  mida_tauler: 0,  
  mida_caselles_tauler: 0,  
  tauler: [],  
  elements: [],  
  zombies: [], /*25% de mida_caselles_tauler*/  
  estrelles: [], /* mida_tauler */  
  recompenses: [],  
  doblePunts: [],  
  meitatZombis: [],  
  vidaExtra: [],  
  casillaEscogidas: 0,  
  ganadas: 0,  
  perdidas: 0,  
  vidas: 3,  
  puntos_totales: 0,
```