

Memòria



Index

Introducció.....	3
HTML.....	4
CSS.....	5
JavaScript.....	8
Conclusions i Complicacions.....	16

Introducció

Faré una breu introducció on explicaré com he dividit el joc i com ho he organitzat.

Principalment ho he dividit en variables, funcions i finalment poguer jugar al joc, no he fet més implementacions al joc

HTML

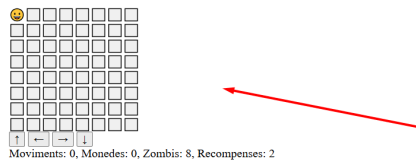
```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Joc de zombis y recompensas - Nico Mesa</title>
  <link rel="stylesheet" href="css/style.css">
</head>
<body>
  <h1>Joc de zombis y recompensas</h1>
  <div id="tabla"></div> <!-- Aquí fare que surti el tauler -->
  <div id="direccio"> <!-- Controls direccionals -->
    <button onclick="moureJugador('nord') ">↑</button> <!-- Nord -->
    <button onclick="moureJugador('oest') ">←</button> <!-- Oest -->
    <button onclick="moureJugador('est') ">→</button> <!-- Est -->
    <button onclick="moureJugador('sud') ">↓</button> <!-- Sud -->
  </div>
  <div id="estat"></div> <!-- Aquí poso tots els esdeveniments que pasin al
joc -->
  <script src="js/script.js"></script>
</body>
</html>
```

L'HTML ho he fet de la següent manera:

- He posat com a ID la taula que he de generar dins del joc (ho faig en JavaScript)
- Faig un div on implemento els botons de que representaran les direccions segons l'usuari vol moure's
- Un ID d'estat per posar l'estat actual generat pel joc on recollim les dades del moviment, monedes, zombies i recompenses

El joc sense CSS sortiria així:

Joc de zombis y recompensas

CSS

```
body {
  font-family: 'Arial'; /* Aplica la font Arial al text dins el body */
  display: flex; /* Utilitza flexbox per al disseny del body */
  flex-direction: column; /* Els elements es col·locaran en una columna */
  align-items: center; /* Centra els elements */
  background-color: #2c3e50; /* Estableix el color de fons a un to fosc */
  color: white; /* Estableix el color del text a blanc */
  margin: 0; /* Elimina el marge predeterminat del body */
  padding: 20px; /* Afegeix 20px al seu voltant */
}

h1 {
  text-align: center; /* Centra el text del títol */
  margin: 30px 0; /* Afegeix un marge superior e inferior de 30px */
}

#direccio, #estat {
  display: flex; /* Utilitza flexbox per als controls direccionals i
l'estat */
  justify-content: center; /* Centra els elements */
  align-items: center; /* Centra els elements */
  margin-top: 10px; /* Afegeix una separacio a dalt de 10px */
}

#direccio button {
  background-color: #e74c3c; /* Estableix el color de fons dels botons */
  color: white; /* Estableix el color del text dels botons */
  border: none; /* Elimina les vores dels botons */
  padding: 10px; /* Afegeix 10 px al seu voltant de separacio */
  margin: 5px; /* Afegeix una separacio de 5px al voltant dels botons */
  border-radius: 5px; /* Arrodoneix les cantonades dels botons */
}
```

```
        cursor: pointer; /* Canvia el cursor al passar-lo pel botó */
    }

#direccio button:hover {
    background-color: #c0392b; /* Canvia el color de fons quan es passa el
cursor sobre el botó */
}

#tabla {
    display: table; /* Defineix la taula com un element de taula */
    border-collapse: collapse; /* Col·lapsa les vores de la taula */
    margin: 20px; /* Afegeix 20 px de separacio al seu voltant */
}

.fila {
    display: table-row; /* Defineix les files de la taula */
}

span {
    display: table-cell; /* Defineix les cel·les de la taula */
    width: 30px; /* Amplada de les cel·les a 30px */
    height: 30px; /* Altura de les cel·les a 30px */
    text-align: center; /* Centra el text dins les cel·les */
    border: 1px solid #ddd; /* Afegeix una vora de 1px i color gris clar a
les cel·les */
    background-color: #95a5a6; /* Estableix el color de fons de les cel·les
*/
}

span.jugador {
    background-color: #2980b9; /* Estableix el color de fons per la cel·la
del jugador */
}

span.visitada {
    background-color: #7f8c8d; /* Estableix el color de fons per les cel·les
visitades */
}

#estat {
    background-color: #34495e; /* Estableix el color de fons per l'estat */
    padding: 10px; /* Afegeix una separacio de 10px dins l'estat */
    border-radius: 5px; /* Arrodoneix les cantonades del contenidor de
l'estat */
}
```

El CSS principalment com volia fer-ho és que tot estigui en el centre, amb els colors que he escollit combinant el blau amb el vermell, a més de combinar el color blanc a les lletres faig que es vegi molt millor (dins del CSS he afegit els comentaris i el que fa cada etiqueta i funció)

Així es veurà el tauler amb el CSS:



JavaScript

```
// Variables
let medidaTauler = parseInt(prompt("Introdueix la mida del tauler:")); //
// Demanar la medida del tauler
let tauler = []; // Crear el tauler buit
let posicioJugador = [0, 0]; // Indicar on esta el jugador en el tauler
let contadorMonedes = 0; // Comptador de monedes
let contadorMoviments = 0; // Comptador de moviments
let numZombies = medidaTauler; // Numero de zombies en el tauler
let numRecompenses = Math.floor(medidaTauler / 4); // Numero de recompenses
// en el tauler

//Joc
inicialitzarTauler(medidaTauler);
taulerHTML()

// Funcions
// Inicialitzar el tauler per poder jugar amb ell
function inicialitzarTauler(medidaTauler) {
    tauler = [];
    for (let i = 0; i < medidaTauler; i++) {
        let fila = [];
        for (let j = 0; j < medidaTauler; j++) {
            fila.push(0); // Afegim a la fila que contengui 0
        }
        tauler.push(fila); //Fiquem que el tauler contingui les files
    }

    tauler[0][0] = 1; // Posició inicial del jugador
```



```
posicioJugador = [0, 0]; // Emmagatzemar la posició del jugador
colocarZombie(); //Col.loquem zombis al tauler
colocarRecompensa(); //Col.loquem recompenses al tauler
taulerHTML(); //Incialitzem el tauler en HTML
dibuixarTauler(); // Inicialitzem el en consola
mostrarEstat(); // Mostrar i actualitzar l'estat
}

// Actualitzar el tauler HTML per poguer dibuixar-ho al nostre HTML
function taulerHTML() {
  const tabla = document.getElementById('tabla');
  tabla.innerHTML = ''; // Limpiar tabla abans de dibuixar
  for (let i = 0; i < tauler.length; i++) {
    let fila = document.createElement('div');
    fila.className = 'fila';
    for (let j = 0; j < tauler[i].length; j++) {
      let celda = document.createElement('span');
      if (tauler[i][j] === 1) {
        celda.textContent = '😄'; // Emoji del jugador
      } else if (tauler[i][j] === 4) {
        celda.textContent = '✔'; // Marqum com casella visitada
      } else {
        celda.textContent = '■'; // Emoji de espai buit
      }
      fila.appendChild(celda); // Afegir la cel.la a la fila
    }
    tabla.appendChild(fila); // Afegir fila al tauler
  }
}

// Dibuixar el tablero
function dibuixarTauler() {
  console.clear();
  for (let i = 0; i < tauler.length; i++) {
    let fila = [];
    for (let j = 0; j < tauler[i].length; j++) {
      if (tauler[i][j] === 1) {
        fila.push(1); // Mostrar al jugador que representa un 1
      } else if (tauler[i][j] === 4) {
        fila.push("✔"); // Casella visitada
      } else {
        fila.push(0); // Mostrar un 0 para ocultar zombis y recompensas
      }
    }
  }
}
```

```
        console.log(fila.join(" ")); // Fem separacions en consola
    }
    taulerHTML(); // Mostrar en l'HTML
}

// Mostrar el estat del jugador (amb recompenses, moviments, zombis, etc)
function mostrarEstat() {
    const estat = document.getElementById('estat');
    estat.textContent = `Moviments: ${contadorMoviments}, Monedes:
${contadorMonedes}, Zombis: ${numZombies}, Recompenses: ${numRecompenses}`;
}

// Moure al jugador
function moureJugador(direccio) {
    let [x, y] = posicioJugador;
    let nouX = x;
    let nouY = y;

    // Moure al jugador depenen de la direcció
    switch (direccio.toLowerCase()) {
        case 'nord':
            nouX--;
            break;
        case 'est':
            nouY++;
            break;
        case 'oest':
            nouY--;
            break;
        case 'sud':
            nouX++;
            break;
        default:
            mostrarEstat('Posa una direcció vàlida');
            return;
    }

    // Validar si el moviment es vàlid (no pot moure's fora del mapa)
    if (!movimentValid(nouX, nouY)) {
        alert('No pots sortir del tauler!');
        return; // Fa que el jugador no es mogui
    }

    // Validar si ja se ha visitat la casella (quan es dibuixi el moviment i
    // si vol tornar sortira que no pot anar aquesta casella)
```

```
    if (tauler[nouX][nouY] === 4) {
        alert('Ja has visitat aquesta casella!');
        return; //Fa que el jugador no es mogui
    }

    contadorMoviments++; // Incrementar el numero de moviments si el moviment
// es vàlid
    validarRecompensaZombi(nouX, nouY); //Validar recompensa en el lloc

    // Marcar la posició anterior com visitada
    tauler[posicioJugador[0]][posicioJugador[1]] = 4;

    // Actualitzar la posició del jugador
    posicioJugador = [nouX, nouY];
    tauler[nouX][nouY] = 1;
    dibuixarTauler(); // Actualitza tauler consola
    taulerHTML(); // Actualitzar tauler HTML
    mostrarEstat();

    // Comprovar si se ha visitat tot el tauler
    verificarFinalJoc();
}

// Verificar final del joc
function verificarFinalJoc() {
    for (let i = 0; i < tauler.length; i++) {
        for (let j = 0; j < tauler[i].length; j++) {
            if (tauler[i][j] === 0) return; // Si queda alguna casella sense
// visitar, no finalitza
        }
    }

    alert(';Has visitat totes les caselles, joc acabat!');//Retorna si ja
// s'ha finalitzat el joc
}

// Col.locar zombis
function colocarZombie() {
    for (let i = 0; i < numZombies; i++) {
        let x = Math.floor(Math.random() * medidaTauler); //Ho col.loquem en
// les files en random
        let y = Math.floor(Math.random() * medidaTauler); //Ho col.loquem en
// les cel.lles en random
        if (tauler[x][y] === 0 && (x !== 0 || y !== 0)) { //Si compleix que
// les posicions son correctes
            tauler[x][y] = 2; // Col.loca un zombi
        }
    }
}
```

```
        } else {
            i--; // Repetir si intenta col.locar un zombi en la posició
            inicial o a sobre altre element
        }
    }
}

// Col.locar recompenses
function colocarRecompensa() {
    for (let i = 0; i < numRecompenses; i++) {
        let x = Math.floor(Math.random() * medidaTauler); //Ho col.loquem en
        les files en random
        let y = Math.floor(Math.random() * medidaTauler); //Ho col.loquem en
        les cel.les random
        if (tauler[x][y] === 0 && (x !== 0 || y !== 0)) {
            tauler[x][y] = 3; // Col.locar una recompensa
        } else {
            i--; // Repetir si intenta col.locar una recompensa en la posició
            inicial o a sobre altre element
        }
    }
}

// Validar si s'ha trobat un zombi o recompensa
function validarRecompensaZombi(x, y) {
    if (tauler[x][y] === 2) { //Si ho troba llançem la funcio troba un zombie
        trobaUnZombie();
    } else if (tauler[x][y] === 3) { //Si ho troba llançem la funcio troba una
        recompensa
        trobaUnaRecompensa();
    } else {
        contadorMonedes += contadorMoviments; //I si no incrementa les
        monedes ja que continua el joc
    }
}

// Trobar un zombi
function trobaUnZombie() {
    alert('Has trobat un zombi!'); //Posem un alert de que hem trobat un
    zombi
    numZombies--; //Restem el numero de zombis
    contadorMonedes = Math.floor(contadorMonedes / 2); // Reduir monedes a la
    meitat
    alert('Monedes a la meitat');
    contadorMoviments = 0; // Reinitziar comptador de moviments
}
```

```
}

// Trobar una recompensa
function trobaUnaRecompensa() {
    alert('Has trobat una recompensa!'); //Posem un alert
    numRecompenses--; //Restem el numero de recompenses
    contadorMonedas = (contadorMoviments + contadorMonedas) * 5; //
Incrementar monedes x5
    alert('Monedes augmentades per 5');
}

//Verifiquem que no es surti del tauler
function movimentValid(x, y) {
    // Verificar que el moviment sigui dins dels limits del tauler
    return x >= 0 && x < medidaTauler && y >= 0 && y < medidaTauler;
}
```

1. Variables

- medidaTauler on demanem a l'usuari la mesura del nostre tauler (obliguem a que sigui int amb parseINT).
- tauler inicialitzat buit per omplir-ho més endavant.
- posicioJugador indiquem que es col.locara a la posició 0,0 dins el tauler.
- contadorMonedas a 0 per omplir-ho i mostrar-ho dins el joc.
- contadorMoviments a 0 per omplir-ho i mostrar-ho dins el joc.
- numZombis iguaem a la mesura del tauler (o sigui si l'usuari posa 8) es posaran 8 zombis al mapa.
- numRecompenses posem les recompenses entre la mesura del tauler entre 4.

2. Joc

- Inicialitzo el joc trucant a les funcions inicialitzarTauler(medidaTauler) i taulerHTML()
- (Les direccions les representa els botons, o sigui la funció moureJugador() està dins l'HTML)
- Perquè truco aquestes funcions, dins la lògica que he fet inicialitzo el tauler i després d'inicialitzar-ho truco al tauler HTML per poguer jugar, on fa la transició de demanar i generar el tauler i només quedaria jugar.

3. Funcions

- inicialitzarTauler(medidaTauler)

1. Aquesta funció (inicialitzar el tauler) agafant amb un doble bucle for i fent push on el que fa es inicialitzar la `medidaTauler*medidaTauler`.
Posa al jugador inicialitzat dins d'aquest en la posició 0,0 i posa un 1 en el nostre cas adalt a l'esquerra un 1.
Col.loquem Zombis, col.loquem Recompenses, inicialitzem el nostre `taulerHTML` per registrar els zombis i recompenses, dibuixem `Tauler`(o sigui mostrar-ho per consola) e inicialitzem l'estat de les nostres recompenses, monedes, moviments, etc...
- `taulerHTML()`
1. Aquesta funció generem en base al ID #tabla el tauler en HTML utilitzant les funcions `.createElement` les files ('span') en un doble bucle for.
La lògica que he implementat es generar el tauler i després indicar que es cadascuna de les coses, en aquest cas 1 representa el jugador, 4 representa una casella ja visitada i el reste que s'ompli amb quadrats blancs.
Ho pugem amb `appendChild` i ja ho tindrem tot ficat a la nostra taula.
- `dibuixarTauler()`
1. Faig la misma lògica que en el tauler pero aquest no es interactiu, únicament mostra el tauler dibuixat amb 0.
- `moureJugador(direccio)`
1. Em vaig basar en els moviments en el tauler, o sigui nord i sud pertany a la posició X i oest a est la posició Y, el faig moure segons la direcció posada (switch-case), llavors reps el valor de x,y fent un validador de moviments i actualitzar el tauler mentre es mogui i sigui un moviment vàlid, sumes el moviment, valides si n'hi ha zombis o recompenses i actualitzes la posició del jugador, marques la casella com ja visitada i el tauler actualitzat.
- `verificarFinalJoc()`
1. Faig un doble for per recorre el tauler i poso un condicional que mentre hi hagin tauler en 0 que no finalitzi el joc, si no retorna un alert on hi hagi un missatge de final de joc
- `colocarZombie()`
1. Faig un únic for ja que ha de recorre el numero de zombis ja predeterminats. Els col.loco en les posicions X,Y amb `Math.random * mesura del tauler`, marquem en el mapa que es un 2 el zombi mentre

compleix el condicional (o sigui que es col.loquin en posició 0 i diferents a 0 per les recompenses) faig un else que si no compleix el if que repeteixi si si intenta col.locar un zombi en la posició inicial o on n'hi hagi recompenses.

- colocarRecompensa()

1. Faig la mateixa lògica que en col.locar zombis.

- validarRecompensaZombi(x,y)

1. Faig un if-if else-else on validi si troba un zombi que s'executi la funció has trobat un zombie, el mateix amb recompensa i si no que sumi monedes al contador.

- trobaUnZombi()

1. Poso un alert que has trobat un zombi, resto el número de zombis, passo a la meitat les monedes i faig que el moviment passi a 0

- trobaUnaRecompensa()

1. Poso un alert que hem trobat una recompensa, restem el número de recompenses i multipliquem les monedes * 5

- movimentValid(x,y)

1. Verifico que el moviment sigui valid dins del paràmetre del joc

Conclusions i Complicacions

Ha sigut un projecte que en un principi em va costar molt arrancar-ho ja que tenia la idea ficada de com fer-ho i com podia executar-ho però gràcies a algunes aclaracions de Jorge he pogut resoldre aquest projecte amb algunes complicacions:

- Al principi d'aquest projecte el fet de fer 2 tauler no sabia com plasmar-ho ja que un servia per jugar pero un altre era per veure i actualitzar-ho
- Els moviments al principi no sabia com plasmar-ho pero al fer-ho únicament amb X,Y vaig poguer fer com actualitzar i moure el jugador dins el tauler
- El col.locar Zombis sense repetir i que no es col.loqui a sobre que altre recompensa o del jugador al principi no sabia com fer-ho pero simplement era aplicar uns condicionals
- Fer el tauler en HTML utilitzant JS, ja que abans no ho sabia com fer-ho pero Jorge ho va explicar i vaig poguer fer-ho sense gaire complicació