

Tarea 3

JavaScript en contexto Frontend Web

1 Problema

Deberán desarrollar un juego del estilo Match-3. Este tipo de juegos, se basan en un tablero rectangular con fichas, teniendo como objetivo para el jugador seleccionar 3 o más elementos que formen un grupo conectado. El grupo seleccionado desaparece cayendo las fichas de arriba e incorporándose nuevas fichas al tablero (por arriba). El puntaje se define en base a la cantidad de fichas del grupo.

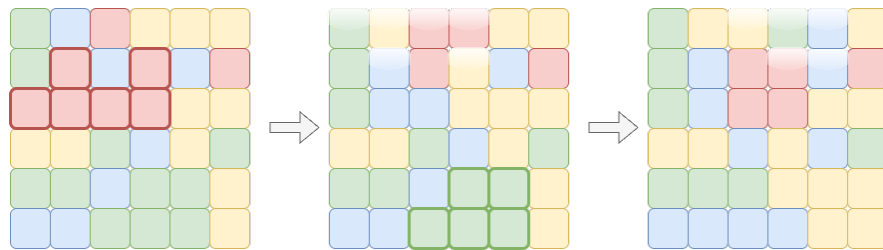


Figure 1: Ejemplo transiciones, primero selecciona el grupo Rojo, luego el grupo Verde. Las casillas sobre el grupo que desaparece bajan, y se rellenan las de arriba de forma aleatoria.

1.1 Detalles

Su juego deberá considerar lo siguiente:

- El juego tendrá niveles, en cada uno habrá un límite de movimientos en los cuáles el jugador deberá cumplir un objetivo.
- Su juego debe tener 16 niveles, con dificultad ascendente. Estos niveles se construyen variando las siguientes características:
 - Tamaño del tablero, soportando 3 opciones: 10x12, 8x10, 6x6 (mientras más pequeño, más difícil).
 - Cantidad de fichas diferentes, pudiendo ser de 3 a 6 (mientras más tipos de ficha más difícil).
 - Dos tipos de objetivo:
 - * Cantidad de puntos.
 - * Cantidad de fichas de distintos tipos (por ejemplo: 10 verdes y 6 rojas).
- Cada uno debe definir sus 16 niveles, pero deben considerar al menos una vez cada uno de los valores diferentes de las características antes definidas.
 - Es decor, debe haber al menos un nivel con cada tamaño de tablero, al menos un nivel con cada cantidad de fichas diferentes (3, 4, 5 y 6), y al menos un nivel por cada tipo de objetivo.



- El puntaje se define en base a la cantidad de fichas del grupo seleccionado. Un grupo de 3 fichas otorga 100 puntos, luego se agregan 60 puntos por cada ficha adicional.
- Las fichas deben disponerse aleatoriamente, al igual que las fichas que ingresan al tablero cuando se elimina un grupo.
- El jugador seleccionará el grupo a eliminar presionando en cualquiera de las fichas que lo componen.
- Deben utilizar colores e íconos para identificar el tipo de fichas.
- Siempre debe estar visible la cantidad de movimientos que le quedan al jugador, así como su avance hacia el objetivo.
- Si no hay más grupos que eliminar, el juego debe declarar perdedor al jugador y permitirle comenzar de nuevo.
- Al iniciar, el jugador deberá poder escoger el nivel que quiere jugar.
- Todo el desarrollo debe ser en un único archivo HTML, todo en una misma vista (la inicialización y luego el juego, y luego poder jugar un nuevo nivel).

2 Entrega

La tarea se dividirá en 2 entregas:

- Entrega Parcial 1 (EP1) [1.5 pts] → Martes 24 de junio a las 10:30, mostrar en horario de clases.
 - Selección de 3 niveles cada uno con un tamaño distinto (10x12, 8x10, 6x6), todos con 4 tipos de fichas.
 - Se debe presentar el tablero creado de forma aleatoria.
 - * No es necesario preocuparse por los aspectos visuales en esta entrega parcial, si se deben diferenciar las fichas.
 - * Que sea aleatorio significa que si se selecciona el mismo nivel varias veces, cada vez será una distribución diferente.
 - Principios de programación funcional en el código utilizado.
- Entrega Final → Martes 8 de julio, presencial, horario por definir.
 - Deberán inscribirse en un horario de corrección.
 - En el horario el ayudante/profesor revisarán su tarea y resolverán con uds. cualquier duda.



3 Consideraciones

Las siguientes son consideraciones y restricciones para el desarrollo de su tarea. El no cumplimiento de una o más de estas implicará que su tarea NO sea revisada y se evalúe con nota 1.0.

- El código JavaScript de la tarea debe estar completamente definido en base al paradigma funcional, según los principios vistos en clases.
- Todo el estilo debe ser manejado en archivos CSS, no puede haber nada de estilo directo en el HTML.
- El archivo HTML no puede contener Javascript, todo debe realizarse en archivos .js específicos.
- Todo el trabajo de su tarea deberán hacerlo utilizando un repositorio en github según las instrucciones que les entregará el profesor.