

Sección Análisis del código

1. Como se genera el tablero del juego y como se definen los tipos de celdas (vacía, numerada, bomba)?

R// El tablero se comienza a generar a partir de la función “startNewGame”. En esta función de acuerdo al tipo de juego(beginner, intermediate, expert y Custom) se asignan los valores a las variables globales.

Primera variable global importante es “gameSettings” la cual contiene el número de filas, columnas y minas en base al nivel de dificultad que se elija, a excepción del juego personalizado que, en este el número de cada parámetro necesario para crear la tabla es ingresado por el usuario.

Luego tenemos la variable global “currentType” la cual guarda el tipo de juego o mejor dicho el nivel de dificultad elegido.

Aquí también se ejecuta la función llamada “getCurrentSettings” que esta al ser ejecutada se encarga de asignarle los valores del tipo de juego elegido a la variable global “currentSettings”.

Luego teniendo todo preparado entonces se ejecuta la función “buildGameBoard” que es la encargada de construir la tabla de juego, a partir de los valores contenidos en “currentSettings”, es decir el número de filas y columnas de acuerdo al tipo de juego elegido, a cada celda se le asigna un index colocándole el atributo “index”. Y finalmente se inicia el contador al ejecutarse la función “startTimer”.

Para generar las minas se ejecuta la función “createMines”, aquí se generan números entre 1 y el número total de celdas de manera aleatoria y son guardados en la variable global mines. Luego con la función “updateCells” se recorren todas las celdas para obtener todas aquellas que están cercanas a la celda que contiene la una mina y luego esas celdas obtenidas son recorridas por la función “updateCellMineProximity” para asignarles a través de la propiedad “cm” el número que va a contener dicha celda.

2. Qué proceso se realiza para cada tipo de celda (vacía, numerada, bomba) al hacer clic izquierdo sobre ellas?

R// Al hacer click izquierdo en una celda, se ejecuta la función “onLeftClick” la cual primero verifica si la variable gameState es igual a “waiting” o el atributo status de la celda es igual a “fl” entonces no va a seguir ejecutando lo demás y en caso de no caer en estas condiciones entonces ejecutará la función “validateCellToExpand” la cual contiene una serie de condiciones:

- La primera condición verifica si en el array global de minas existe el numero de index de la celda entonces se termina el juego y se llama la función “gameOver”.
- La segunda condición verifica que si la propiedad “cm” es mayor de cero entonces se ejecuta la función “setStateFaceUp” la cual cambia el valor del atributo status a “fu” y se le asigna la imagen correspondiente de acuerdo al valor de la propiedad “cm”.
- Y la tercera condición verifica si el atributo status es igual a “fd” entonces ejecuta la función “expandEmptyCell” la cual se encargará de expandir todas aquellas celdas cercanas a la celda que se le hizo click.

3. Como se maneja la habilidad de marcar con banderas una celda?, ¿Existen limitaciones en la marcación con banderas?

R// Bien, la habilidad de marcar una determinada celda con una bandera se maneja de a través de la función “onRightClick” la cual fue asignada a cada celda con un eventListener al momento de la creación de la tabla. Esta función tiene una serie de condiciones:

- La primera es verificar que el estado del juego está en “Playing” de lo contrario no sigue ejecutando nada.
- La siguiente condición verifica que si el atributo “status” es igual a “fl” entonces significa que esa celda ya fue marcada como bandera; por lo tanto se necesita sumarle 1 al valor del input “txt_mineCount”.
- La siguiente condición verifica que todavía el valor de “txt_mineCount” sea mayor de cero para poder hacerle un decremento.
- Y la ultima condición verifica si el valor de “txt_mineCount” entonces ejecuta la función “validateWinCondition” para saber si el jugador ha ganado el juego.

La única limitación con la marcación de banderas es que la cantidad disponible es la misma que el de las minas; por lo tanto, no se puede utilizar un número mayor a ese.

4. Como se maneja la victoria (Win) y (Game Over)?

Manejo Victoria

La victoria se maneja a través de la función “validateWinCondition”, en esta se verifica que si la cantidad de celdas que fueron volteadas es igual al número de celdas en el tablero menos la cantidad de minas, si es así entonces el jugador ganó la partida

(numeroCeldasVolteadas === cantidadCeldasTablero - cantidadMinas)

La segunda condición de victoria es en base a si el jugador ha podido marcar todas las minas con banderas, esto se hace recorriendo todas las celdas marcadas con banderas y verificando si el index de la celda está incluido en el arreglo global de “mines”.

Y cuando se ha verificado que se ha ganado la partida entonces se procede a detener el contador, se pone el estado del juego en “waiting” y se muestra el alert para que el usuario pueda ingresar el nombre.

Manejo Derrota

La derrota se maneja a través de la función “gameOver” y esta se ejecuta en el momento de que el jugador hace click en una celda que contiene una mina; este proceso se hace verificando que si el index de esa celda está incluida en el arreglo de global de “mines” entonces el jugador perdió la partida.