**PRUEBA DE ANÁLISIS Y PROGRAMACIÓN**

Realizado por: Edison Omar Ordoñez Almendares

Sección 1 – Análisis

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

* Tablero del juego

Este es generado por la función “startNewGame” la cual se enlaza a un elemento html por medio del id “gameBoard” donde tiene un estilo que al visualizarse muestre los bloques del buscaminas.

Luego tenemos la función “openCloseGameOptions” la cual nos traerá el nivel de dificultad que queremos el buscaminas y a raíz de este visualizar el tablero de juego.

* Tipos de celdas
  + Vacía

La función comienza recuperando un enlace a un elemento html por medio del id “gameTable". Luego recorre cada fila de la tabla y, para cada fila, recorre cada columna de esa fila. Si el atributo de índice de la celda coincide con el argumento “cellIndex” pasado a la función, la función establece el estado de esa celda para que esté boca arriba mediante la función “setStateFaceUp”.

* + Numerada
  + Bomba

La función “createMines” contiene un array vacio “mine []” y utilizada el objeto “currentSettings” que contiene el numero de minas, filas y columnas, el cual por medio de un bucle for genera números aleatorios utilizando Math.random(), El número aleatorio generado se compara con la lista “mine” utilizando el Array.includes, método para garantizar que cada mina tenga una ubicación única. Si el número no está en la lista, se agrega a la lista usando Array.push.

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

* Tipos de celda
  + Vacía

Si el estado de la celda es "fd" (boca abajo), se llama a la función expandEmptyCell con el índice de la celda como argumento. Esto es para expandir las celdas vacías (celdas con un recuento de minas de 0) y revelar las celdas cercanas.

* + Numerada
  + Bomba

Al hacer click izquierdo en una bomba ocasiona la función “gameOver” la cual detiene el tiempo y pone el estado del juego en espera y muestra las demás minas dando por perdida la partida.

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

R// Si el estado de la celda es "fl" (marcado), la función llama a la función “setStateFaceDown” para cambiar el estado de la celda hacia abajo e incrementa el recuento de minas en 1.

Si el estado de la celda no está marcado y el recuento de minas no es igual a 0, la función llama a “setStateFlagged” para cambiar el estado de la celda ha marcado y reduce el recuento de minas en 1.

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

* Win

La función se llama con el argumento "normalWin", que es un valor booleano que indica si el juego se ganó con un juego normal (revelando todas las celdas que no son mínas).

Si "normalWin" es falso, la función selecciona todas las celdas con un estado de "fl" (marcado) utilizando el método querySelectorAll y almacena el resultado en la variable "f". Si el número de celdas marcadas es igual al número de minas, la función verifica si el índice de cada celda marcada está incluido en la matriz de "minas". Si todas las celdas marcadas son realmente mínas, la variable "flagsWin" se establece en verdadero.

* Game Over

Si el índice de la celda está presente en la matriz "minas" y el modo del click es "leftClick", se llama a la función gameOver con el índice de la celda como argumento.

Sección 3 – EXTRA

1. Después de haber visto el código del juego, ¿Hay algún punto del código original que se pueda mejorar?, ¿Qué cambiarias?

R// Comentar la funcionalidad de cada función y utilizar variables mas detalladas o mas referentes a lo que se realizara.

2. Después de haber visto el código del juego, ¿Sabes si hay alguna forma de hacer trampa para ganar sin modificar físicamente los archivos del juego?, si es así como lo harías?

R// Por medio de la consola del navegador, ingresamos el código “showBombs(true)” y muestra la ubicación de todas las minas en el tablero del juego.