



EncuentraBombas

El **EncuentraBombas** es un juego ampliamente inspirado en uno ya existente, diseñado para **un jugador** y que consiste en **encontrar una serie de bombas** en un **tablero o matriz**, evitando que nos **exploten en la cara**.

Nosotros vamos a dedicar esta práctica a realizar una versión de este famoso juego y para ello vamos a seguir una serie de pasos. Dichos pasos están pensados en ese orden, para que os resulte más fácil la compleción de la práctica, por lo que os recomiendo que los sigáis.

En la práctica vamos a hacer uso de **2 matrices**, una primera en la que estará toda la información de la partida y que **NUNCA** mostraremos al usuario (la llamaremos **plantilla**) y otra que se le **mostrará al usuario** y que será sobre la que **haga las jugadas** (la llamaremos **tablero**).

Primero vamos a construir la plantilla pero, para ello, antes tenemos que establecer los **grados de dificultad**, que nos darán el **tamaño de la matriz** y el **número de bombas** que se van a colocar. Tenemos los siguientes modos:

- **Principiante:** 8 x 8 casillas y 10 bombas.
- **Intermedio:** 16 x 16 casillas y 40 bombas.
- **Experto:** 16 x 30 casillas y 99 bombas.
- **Personalizado:** en este caso, es el usuario el que indica el tamaño y la cantidad de minas.

Esto será lo primero que le preguntaremos al usuario nada más iniciemos el programa, ya que lo utilizaremos para construir el resto del programa.

Una vez sepamos el modo de dificultad, construiremos una **matriz vacía de enteros** del tamaño indicado e iremos colocando bombas de forma aleatoria por la matriz. Como la matriz está vacía, es decir, **rellenada con todo a 0**,





utilizaremos el valor **-1** para indicar que hay una bomba en esa casilla. La metodología irá de la siguiente forma:

- 1) Seleccionamos una fila y columna al azar, **controlando los valores** para no salirnos de la matriz.
- 2) Comprobamos que en esa posición **no haya una bomba ya puesta**. En caso de que lo haya, **repetimos el paso 1**.
- 3) Modificamos el valor de la casilla de 0 a -1.
- 4) **Incrementamos en 1** el valor de las **8 casillas contiguas** a la bomba.
- 5) Repetimos el proceso para el total de bombas que se vayan a colocar.

Al final la matriz debería de quedar así:

1	1	1	0
1	-1	1	0
1	1	2	1
0	0	1	-1

Si os fijáis, las celdas que **pegan a varias bombas** tendrán el **valor incrementado** con respecto al resto, esto debería de ser el resultado natural de emplear **el proceso anterior**. Una vez tengamos la matriz completa, crearemos otra matriz del mismo tamaño pero que permanecerá vacía. Esta última será la que **verá el jugador, el tablero**.

Ahora pasaremos a lo que es el bucle principal del juego. Principalmente, se le pedirá al usuario que introduzca casillas para ir detectando las bombas. Esto se repetirá hasta que el jugador haya encontrado todas las bombas, o le haya explotado una.

El proceso irá de la siguiente forma:





- 1) Se le pregunta al usuario por la casilla a mostrar. Habrá que comprobar que los valores dados sean válidos.
- 2) Se le pregunta al usuario si quiere mostrar la casilla o marcarla. (Ahora solo nos vamos a centrar en mostrar).
- 3) Obtenemos el valor de la casilla en la matriz original y la actualizamos en el tablero, mostrando el resultado al usuario.
- 4) Comprobamos si la casilla seleccionada en la plantilla es una bomba, en dicho caso, se muestra un mensaje de fin de juego y se acaba la partida.

Para el caso de que la opción seleccionada sea la de marcar la casilla se hará lo siguiente:

- 1) Se marca en **el tablero la casilla seleccionada con un -4** (parece un banderín).
- 2) Comprobamos que dicha posición en **la plantilla** sea una bomba.
- 3) En caso de que sea una bomba, se guardan sus posiciones en una matriz de **2 filas x tantas columnas como bombas haya**. Hay que comprobar previamente que dichos valores no se hayan introducido ya.
- 4) Incrementamos en 1 el total de bombas marcadas.
- 5) En caso de que en el paso 2 **no** se haya detectado una bomba, se **ignoran los pasos 3 y 4**.
- 6) Comprobamos si el **total de bombas marcadas equivale al total de bombas de la partida**, en ese caso, se **termina la partida** y se muestra un mensaje de victoria.

Este bucle de juego se repetirá mientras no se cumplan alguna de las 2 condiciones que se han mencionado anteriormente. De igual forma, al terminar una partida, se le preguntará al usuario si quiere continuar jugando, en caso afirmativo, se repetirá la ejecución del programa **comenzando por la selección de dificultad**.





Para hacer la entrega debéis crear un archivo zip con el proyecto (el código fuente) cuyo nombre tenga el siguiente formato:

1TDAW_PR_NOMBRE_ALUMNO_AVENTURA_V2.zip

NOTA: se calificarán como NO PRESENTADO cualquier práctica que no cumpla con los requisitos de entrega

Apartado Extra

Tal cual tenemos nuestra versión del encuentra bombas, tenemos que ir revelando las casillas una por una. Este acercamiento puede ser muy tedioso para el usuario, así que vamos a implementar un algoritmo para revelar todas las **casillas vacías que estén contiguas**. Para ello, lo primero que tenemos que hacer es designar 2 tipos de casillas distintas en el tablero; la **casilla sin revelar** y la **casilla vacía**. El algoritmo a implementar va de la siguiente manera:

Cuando el jugador seleccione una casilla para mostrar:

- 1) Comprobamos si esa casilla es 0.
- 2) En caso afirmativo, revelamos todas las casillas contiguas a la casilla seleccionada.
- 3) A continuación recorreremos toda la matriz comprobando si para **cada casilla 0 en la plantilla** existe alguna **casilla contigua en el tablero** que ya **haya sido revelada**.
- 4) Si se cumple el caso, revelamos la casilla seleccionada junto con todas las casillas de alrededor de la misma. En caso negativo, se pasa a la siguiente sin hacer nada.
- 5) Se **repetirá** el proceso en **dos “pasadas”** a la matriz, una de arriba a abajo y otra de abajo a arriba.

Intenta hacer el algoritmo para que requiera la menor cantidad de iteraciones posibles.

