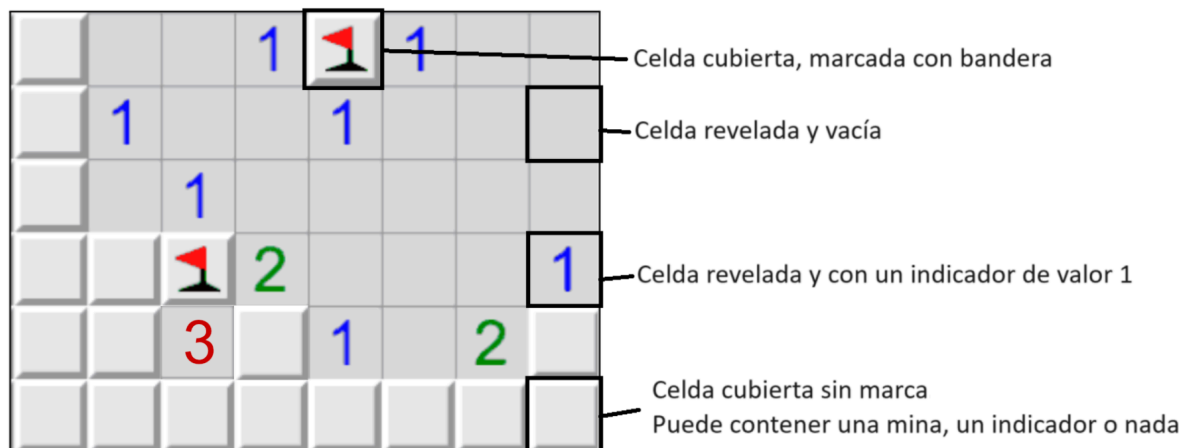


Práctica Integradora Buscaminas

"Buscaminas" es un videojuego en solitario en donde el jugador tiene por objetivo limpiar un campo minado, sin detonar ninguna mina.

Vamos a implementar este juego en Gobstones. Se utilizará el tablero para representar la grilla, donde cada celda representará una casilla en donde puede haber una **mina**, un **indicador** numérico (que es un número que indica la cantidad de minas en las casillas vecinas¹) o nada. Para que el juego tenga sentido, todas las celdas comienzan en estado **cubierta**. Esto significa que, sin importar su contenido, todas las celdas se visualizan de la misma forma.

Para cambiar el estado de una casilla se puede **revelar** su contenido. Si se revela el contenido de una casilla y esta contenía una mina, la mina explota, el juego finaliza y el jugador pierde la partida. Las casillas cubiertas también se pueden **marcar** con una bandera para indicar la (presunta) presencia de una mina. Cuando una casilla se marca con una bandera sigue estando cubierta (ya que aún no se conoce su contenido) pero en su celda ahora se muestra una bandera. En la siguiente imagen se muestra un ejemplo de juego en progreso:



Gran parte de la implementación ya está hecha y se nos solicitó completarla con ciertas funcionalidades específicas, para lo cual contamos con las siguientes primitivas de las cuales podemos hacer uso:

estáCubierta()
PROPÓSITO: Indica si la casilla actual está cubierta.
TIPO: Booleano.
PRECONDICIÓN: Ninguna.

hayVecinaCubiertaAl_(dirección)
PROPÓSITO: Indica si hay una casilla vecina en la dirección dada y además esa casilla está cubierta.
TIPO: Booleano.
PRECONDICIÓN: Ninguna.

¹ Nota: a diferencia del juego original, en esta versión del juego consideramos "vecinas" únicamente a las casillas lindantes en direcciones ortogonales.

hayIndicadorAcá()
PROPÓSITO: Indica si hay un indicador en la casilla actual.
TIPO: Booleano.
PRECONDICIÓN: La casilla actual está revelada.

indicadorAcá()
PROPÓSITO: Describe el valor del indicador numérico en la casilla actual.
TIPO: Número.
PRECONDICIÓN: La casilla actual está revelada y contiene un indicador.

tieneMarca()
PROPÓSITO: Indica si hay una marca en la casilla actual.
TIPO: Booleano.
PRECONDICIÓN: La casilla actual está cubierta.

tieneMina()
PROPÓSITO: Indica si hay una mina en la casilla actual.
TIPO: Booleano.
PRECONDICIÓN: La casilla actual está revelada.

Revelar()
PROPÓSITO: Revela el contenido de la casilla actual.
PRECONDICIÓN: La casilla actual está cubierta.

Marcar()
PROPÓSITO: Marca la casilla actual con una bandera.
PRECONDICIÓN: La casilla actual está cubierta y no está marcada.

Basados en este modelo y asumiendo que el tablero es uno válido para el juego, se pide:

Ejercicio 1)

Implementar la función **esCasillaSegura** que determina si la casilla actual es segura para revelar. Una casilla se dice segura para revelar si está cubierta y no tiene ninguna vecina cubierta ni con indicadores.

Ejercicio 2)

Implementar el procedimiento **Arriesgar** que revele la casilla cubierta más riesgosa de la grilla, suponiendo que existe una casilla más riesgosa que el resto y que esa casilla está cubierta. La casilla más riesgosa es aquella cuyo riesgo sea máximo. El riesgo de una casilla se calcula como la cantidad de casillas vecinas que están reveladas y tienen indicadores.

Ejercicio 3)

Considere la siguiente implementación de la función **hayVecinaCubiertaAl_**, cuyo contrato es correcto:

```
function hayVecinaCubiertaAl_(direcciónAAnalizar) {  
    /*  
    PROPÓSITO: Indica si hay una casilla vecina en la dirección dada y  
    además esa casilla está cubierta.  
    PARÁMETROS:  
    * direcciónAAnalizar: Dirección - La dirección a la cual analizar si hay  
    casilla cubierta.  
    TIPO: Booleano.  
    PRECONDICIÓN: Ninguna.  
    */  
    if (puedeMover(direcciónAAnalizar))  
    then {  
        Mover(direcciónAAnalizar)  
    }  
    return ( estáCubierta() )  
}
```

Se pide que determine si la solución propuesta es correcta (soluciona el problema planteado en el propósito o no), y de no serlo, proponga las correcciones adecuadas utilizando los buenos criterios vistos en la materia.

JUSTIFIQUE sus respuestas (la justificación no puede tener más de 5 renglones).