

# DOCUMENTACIÓN

## NOMBRES

Berrío Galindo, Álvaro

García Vázquez, Sara

## DISEÑO MODULAR

### inicio

crea las matrices de ceros **matriz**, **matrizcopia**

define **contador**, **nivel**

llama al método **generapuntos**

crea (3\*nivel) puntos aleatorios

llama al método **incrementar**

genera puntos en matriz en los puntos obtenidos  
en **generapuntos**

**fin incrementar**

**fin generapuntos**

copia matriz en **matrizcopia**(llama al método copiar matriz)

recorre todos los puntos de matriz y los copia a matrizcopia

**fin copiarmatriz**

llama al método **impmatriz**

imprime el menú

imprime en pantalla matriz en forma de tablero

imprime el nivel en el que se encuentra el juego

**fin impmatriz**

llama al método **golpes**

define **matrizJuego** y la copia a matriz (con el método  
**copiarmatriz**)

define **matrizCopia** y la copia a **matrizcopia** (**copiarmatriz**)

define **indicador** y le da el valor 0

define **nivelito** como copia de nivel

**do{**

imprime el número de golpes hasta el momento

pide otro golpe

define **fila** y **col** como variables para leer de teclado

**mientras**(fila>6 ó col>6 ó fila<0 ó col<1)

imprime "Error.Elija un número entre 1 y 6".

vuelve a pedir fila y col

**fin si**

**si**(fila=0)

**switch**(en función de col)

**caso 1:** pone contador a 0  
copia matrizCopia a matrizJuego  
(**copiarmatriz**)  
imprime matrizJuego (**impmatriz**)

**caso 2:** llama al método **cerosmatriz**  
recorre matrizJuego punto a punto  
y los hace todos 0

**fin** cerosmatriz  
pone el contador a 0  
genera puntos en matrizJuego  
(**generapuntos**)  
copia matrizJuego a matrizCopia(  
**copiarmatriz**)  
imprime matrizJuego (**impmatriz**)

**caso 3:** imprime las calificaciones que se llevan en  
cada nivel

**caso 4:** pone el contador a 0  
hace de matrizJuego una matriz de ceros  
(**cerosmatriz**)  
asigna un nuevo valor a nivel (llama al  
método cambiarnivel)  
imprime Elige nivel  
define **nivel** como variable para leer  
de teclado  
devuelve nivel mientras sea menor  
que 10 y mayor que 0

**fin** cambiarnivel  
genera puntos en matrizJuego  
(**generapuntos**)  
copia matrizJuego a matrizCopia  
(**copiarmatriz**)  
imprime matrizJuego (**impmatriz**)

**caso -2:** hace de matrizJuego una matriz de ceros  
(**cerosmatriz**)  
asigna a indicador el valor 1

**fin switch**

**si no**

llama al método **decrementar**  
recorre matrizJuego punto a punto y  
disminuye en una unidad el punto (fila,col)  
y sus vecinos

**fin** decrementar  
imprime matrizJuego (**impmatriz**)  
aumenta contador en una unidad

```

}mientras(método comprobar devuelva false)
si (indicador=1)
    sale del juego
si no
    llama al método puntuaciones
        define golpes como nivel*3
        si (contador<golpes)
            imprime "Extraordinariamente bien"
        fin si
        si (contador=golpes)
            imprime "Perfecto"
        fin si
        si (contador>golpes)
            imprime el número de golpes
        fin si
    fin puntuaciones
    pone contador a 0
    hace de matrizJuego una matriz de ceros (cerosmatriz)
    llama a jugarotravez
        imprime "Quieres volver a jugar? 1 para sí, -1 para
        no"
        define jugardeNuevo como variable para leer de
        teclado
        mientras(jugardeNuevo!=1 y jugardeNuevo!=-1)
            imprime "Error" y vuelve a pedir un
            número
        fin mientras
        si (jugardeNuevo=1)
            genera puntos en matrizJuego
            (generapuntos)
            copia matrizJuego a matrizCopia
            (copiarmatriz)
            imprime matrizJuego (impmatriz)
            llama a golpes
        si no
            si (jugardeNuevo=-1)
                sale del juego
            fin si
        fin si
    fin jugarotravez
fin si
fin golpes
fin

```

A continuación voy a explicar el método comprobar que es el único que no he podido hacer debido a las circunstancias. Esta función recorre la matriz punto por punto y comprueba que todos sean 0. En caso de que sea así devuelve true, de lo contrario devolverá false.

## **DESCRIPCIÓN DE LAS ESTRUCTURAS DE DATOS UTILIZADAS**

Hemos utilizado:

- Un array unidimensional para los ficheros
- Dos datos bidimensionales para hacer la matriz de juego (int[][] matriz) y la copia de la matriz de juego (int[][] matrizcopia).

## **DESCRIPCIÓN DE LAS PRUEBAS REALIZADAS**

CASO	ENTRADA	SALIDA ESPERADA	SALIDA OBTENIDA
1	0 1	La matriz de juego (tablero fijo).	
2	0 2	La matriz de juego (tablero fijo).	
3	0 3	Se muestran las puntuaciones.	
4	0 4	Opción de elegir nivel e imprime la matriz de juego.	
5	0 -2	El juego termina	
6	1 3	Decrementa las casillas correspondientes y "Golpe (fila,columna)".	
7	8 4	"Error. Introduce otro golpe."	
8	4 9	"Error. Introduce otro golpe."	
9	0 4, 15	"Error, escribir un número del 1 al 9."	

Otras pruebas que hemos realizado han sido dar una serie de golpes al tablero e introducir la combinación 0 1, seguir jugando y probar un nuevo tablero en el mismo nivel (0 2). Continuamos jugando y cambiamos al nivel 1. Damos los 3 golpes correspondientes e introducimos 1 para seguir jugando. Este mismo proceso lo hemos realizado de nuevo pero introduciendo -1 para que termine y tecleando otro número diferente a los dichos anteriormente, caso en el que el juego imprime un mensaje de error y pide otro número. Para comprobar el mensaje que debe imprimirse en pantalla sobre las puntuaciones hemos probado a hacer el nivel correspondiente en los golpes necesarios y en más golpes.