

MEMORIA DOCUMENTAL PROGRAMA “HUNDIR LA FLOTA” CON JAVA

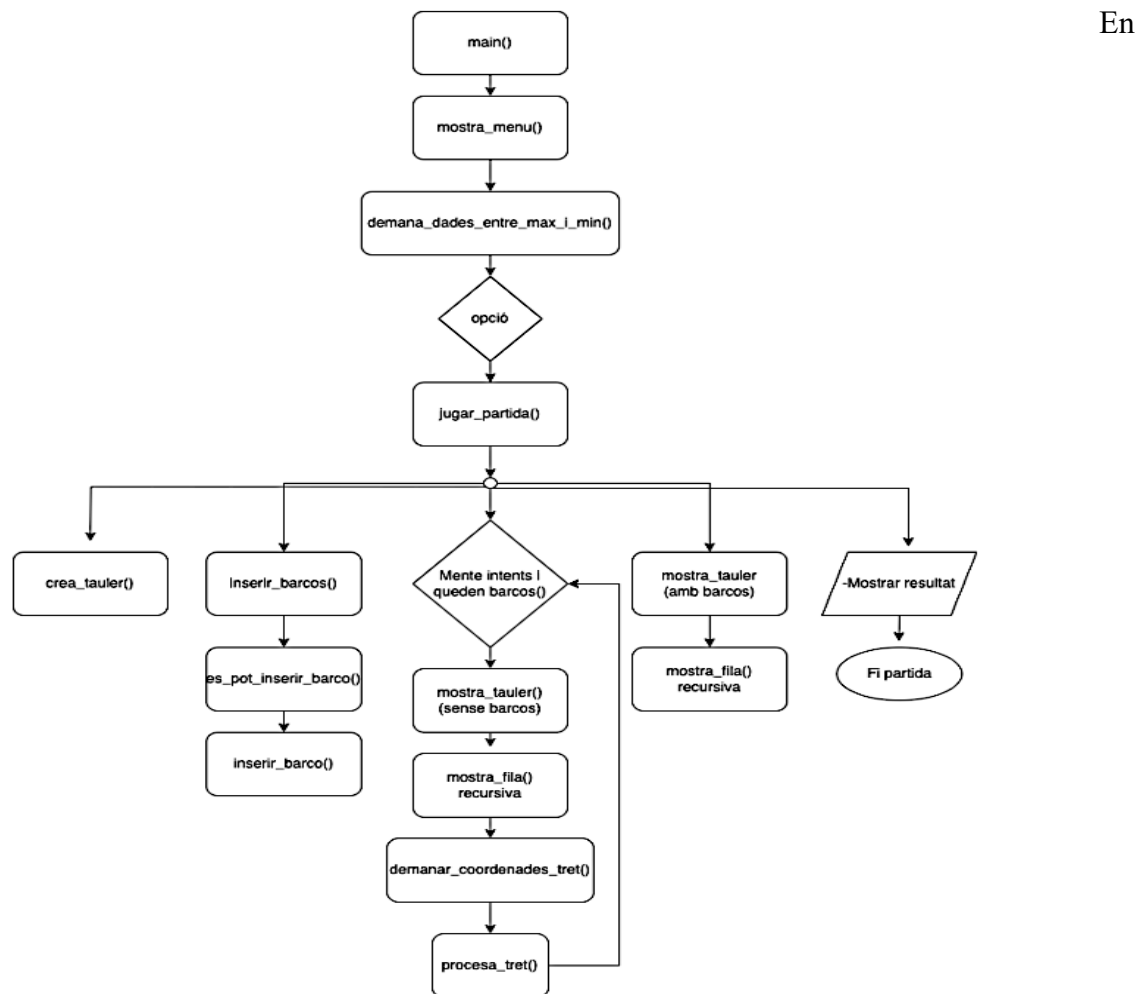
PEIRO SORIANO, L. PAOLA
1° DAM - CEEDCV

Índice

<i>Estructura del programa “Hundir la flota”</i>	2
<i>Menú inicial y método main</i>	3
<i>Jugar Partida y las opciones de dificultad</i>	3
<i>Desarrollo de funciones y método (errores)</i>	4
Función crear Tablero	4
Función insertar barcos aleatorios	4
Conclusión	5
Compilación en VSC:	5
Compilación y ejecución en Android mediante JStudio:	6

Con esta memoria documental, no obligatoria, pretendía tener, no solamente un mapa mental del procedimiento sino también un documento que me sirva para repasar los errores cometidos que menciono a lo largo del código y tener guardado mi modus operandi. Lo adjunto con fines educativos y para poder tener costumbre de documentar los trabajos realizados de cara a buenas prácticas y entorno laboral.

Estructura del programa “Hundir la flota”



primer lugar, se nos facilita un PDF que contiene unas instrucciones para el desarrollo del juego. En las mismas se nos indican los elementos que contendrá el juego y la estructura de este.

[Instrucciones juego](#)

Primero, se nos presenta un diagrama para poder facilitar la comprensión de la estructura y de las explicaciones que se dan en el PDF adjunto.

Empezamos en primer lugar con las importaciones y las declaraciones de las clases que en este caso importaremos la clase “SCANNER” para leer la entrada del usuario. Se suele

usar también el método “console read line” pero en este caso, usaré la clase Scanner y declararé la public class ‘Hundir la flota’ que contendrá el código de mi juego.

Menú inicial y método main

Mi punto de partida desde ahora será el método “main” donde comenzará la ejecución de mi código. Se mostrará el mensaje de bienvenida y las opciones de dificultad al usuario desde el nivel fácil hasta el nivel difícil.

Usaremos para ello la solicitud al usuario de un dato máximo y mínimo (invocaremos y tendremos la función guardada) que guardaremos en una variable y nos permitirá que, mediante la elección, se pueda abrir un nivel u otro. Según la opción que el usuario haya escogido, se establecen las variables que representan la cantidad de barcos. Seguidamente se invoca a la función de “jugarPartida” con los parámetros necesarios para que esta comience.

Para poder ofrecer las opciones de dificultad implementé primero un menú sencillo con la opción de mostrar en pantalla las opciones de dificultad, pero tuve un problema cuando solicité al jugador que ingresara su elección ya que en primera instancia me olvidé de incluir la validación para que se validase que hubiera un número dentro del rango esperado. Por tanto, si metía una letra de abecedario u otra cosa que no fuera del 1 al 3, causaba errores o me cerraba el juego.

Cuando el jugador seleccionaba el nivel de dificultad, se asignaban valores específicos a las variables y se representaba la cantidad de barcos y los intentos disponibles para esa dificultad en concreto. Cuando hice la prueba inicial, con pocos intentos unos 2 o 3 a lo sumo, me daba cuenta de que el juego no finalizaba cuando se agotaban los intentos. Era relativamente sencillo, se debía a un error en la lógica de control de bucles y en concreto, en la actualización de los intentos restantes a falta de → `intentos--;`

También, y por último me he encontrado con un error mientras intentaba cerrar el objeto ‘Scanner’ que he usado para que se lea la entrada del usuario, el problema es que surge un error, o más bien, advertencia y no logro resolverlo. Lo he puesto en el código para recordarme que, una vez corregido saber cuál es el error en concreto.

Jugar Partida y las opciones de dificultad

Desarrollo la función de jugarPartida y establezco el desarrollo y flujo del juego una vez que el jugador escoge la dificultad. Entonces, se crean dos tableros para cada uno (función crear tablero) y se insertan de manera aleatoria (función insertar barcos con método random) los barcos en ambos tableros y así, comienza el bucle del juego hasta agotar o los intentos o los barcos.

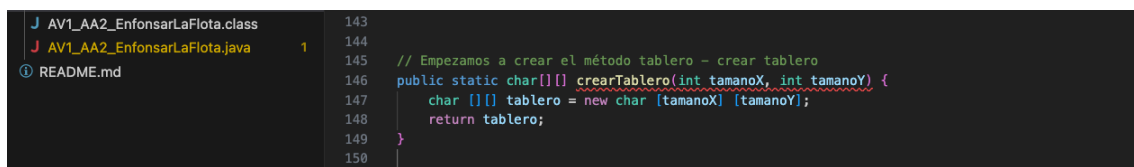
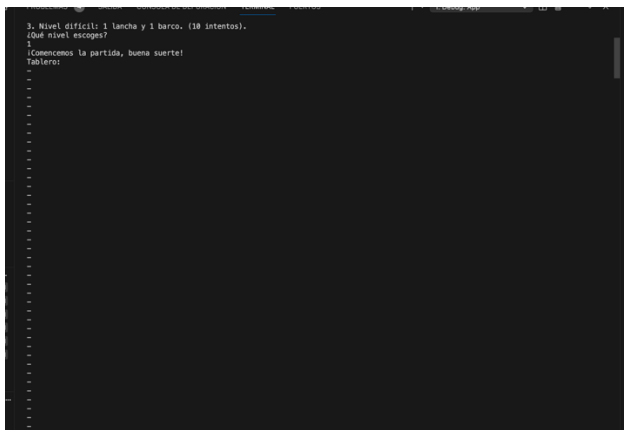
Otro error que he tenido aquí ha sido cuando intenté ajustar la dificultad del juego, luego lo resolví mediante un switch para poder poner las opciones de dificultad colocando dentro de cada “case” las cantidades de barcos e intentos. A partir de ahí intenté estar más atenta para las iteraciones futuras.

Desarrollo de funciones y método (errores)

Durante el desarrollo del juego en cuestión, he tenido varios quebraderos de cabeza. Primero en cuanto a errores se refiere. Cuando llegué a la función de “crear Tablero” tuve diferentes dificultades.

Función crear Tablero

Durante la fase inicial cometí un error al olvidar iniciar bien el tablero con el carácter ‘-’ al inicio y claro, esto hacía que hubiese una visualización no esperada del tablero y que los caracteres no fuesen los deseados.



Pude arreglarlo ajustando la función y iniciando una celda de tablero con el carácter ‘-’



Función insertar barcos aleatorios

En esta función tuve un problema relacionado con la inserción de los barcos, no hice verificación de si cabían en el tamaño del tablero antes de insertarlos y claro se superponían o excedían del tablero.

```

191
192 v public static void insertarBarcosAleatorio(char[][] tablero, int lanchas, int barcos, int acorazado, int portaaviones) {
193
194 }

```

Y para solucionarlo agregué una validación previa para verificar si los barcos cabían y así pues evitaba que no se viese bien.

```

J AV1_AA2_EnfonsarLaFlota.java 1 189
D README.md
190
191
192 public static void insertarBarcosAleatorio(char[][] tablero, int lanchas, int barcos, int acorazado, int portaaviones) {
193     if (validarCantidadBarcos(tablero, lanchas, barcos, acorazado, portaaviones)) {
194
195     } else {
196         System.out.println("Error: Los barcos no caben en el tablero");
197     }
198 }
199 private static boolean validarCantidadBarcos(char[][] tablero, int lanchas, int barcos, int acorazado, int portaaviones) {
200
201

```

Conclusión

A lo largo del desarrollo del proyecto he ido comentando cada parte y hoy en día, me seguiría costando horas de reestructurarlo de nuevo pero el detallarlo me ha ido ayudando a su resolución. La planificación, es la clave, problemas pequeños que se han ido resolviendo con bucles e iteraciones.

Finalmente, lo he compilado mediante el terminal del IDE Visual Studio Code y lo he probado en mi otro ordenador Windows 11 y en mi dispositivo Android mediante la APP JStudio para ver que compilaba y funcionaba.

Compilación en VSC:

```

> lib
> src
J AV1_AA2_EnfonsarLaFlota.class
J AV1_AA2_EnfonsarLaFlota.java 1
D README.md

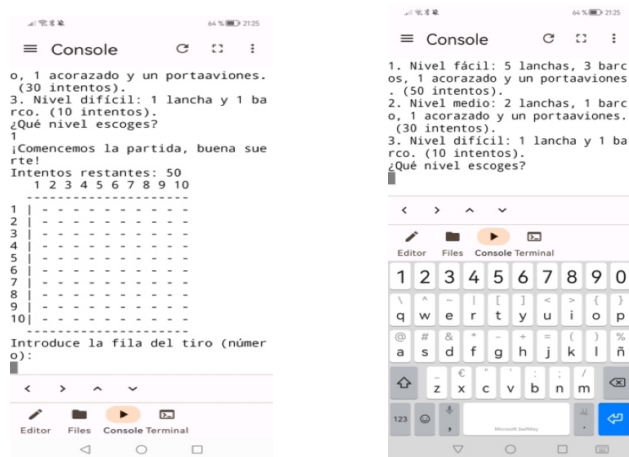
Decompile Class File

PROBLEMAS 1 SALIDA TERMINAL ...
Dendron Documents Library Music Public
PaoLa@Air-de-Liney ~ % Desktop
zsh: command not found: Desktop
PaoLa@Air-de-Liney ~ % cd Desktop
PaoLa@Air-de-Liney Desktop % ls
CAPTURAS
Captura de Pantalla 2024-01-05 a las 15.17.03.png Captura de Pantalla 2024-01-05 a las 19.25.53.png
Captura de Pantalla 2024-01-05 a las 15.17.40.png DOC GRAF PRGR
Captura de Pantalla 2024-01-05 a las 15.18.31.png ENTREGA EDICION
Captura de Pantalla 2024-01-05 a las 15.31.49.png ENTREGABLE LM
Captura de Pantalla 2024-01-05 a las 15.31.54.png Entregables_EV
Captura de Pantalla 2024-01-05 a las 19.20.19.png SIMULACROS
PaoLa@Air-de-Liney Desktop % cd Entregables_EV
PaoLa@Air-de-Liney Entregables_EV % ls
COPIA EV_BB00 EV_ED EV_FOL EV_LM EV_PRG EV_SI
PracticasBB00
PaoLa@Air-de-Liney Entregables_EV % cd EV_PRG
PaoLa@Air-de-Liney EV_PRG % ls
Flota.java
PaoLa@Air-de-Liney EV_PRG % cd flota.java
PaoLa@Air-de-Liney flota.java % ls
README.md bin lib src
PaoLa@Air-de-Liney flota.java % cd src
PaoLa@Air-de-Liney src % ls
AV1_AA2_EnfonsarLaFlota.java
PaoLa@Air-de-Liney src % javac AV1_AA2_EnfonsarLaFlota.java
PaoLa@Air-de-Liney src % java AV1_AA2_EnfonsarLaFlota

=====BIENVENIDO/A A HUNDIR LA FLOTA!=====
Niveles de dificultad:
1. Nivel fácil: 5 lanchas, 3 barcos, 1 acorazado y un portaaviones. (50 intentos).
2. Nivel medio: 2 lanchas, 1 barco, 1 acorazado y un portaaviones. (30 intentos).
3. Nivel difícil: 1 lancha y 1 barco. (10 intentos).
¿Qué nivel escoges?

```

Compilación y ejecución en Android mediante JStudio:



El proyecto me ha brindado la oportunidad de asentar conocimientos con los bucles y familiarizarme más con las funciones. Espero seguir desarrollando pequeños proyectos.

De manera opcional y por ocio, haré uso de este código y lo adaptaré a importando Swing JOptionPane para imprimirlo en mi escritorio y no imprimirlo en consola. Posteriormente, será subido a mi GitHub <https://github.com/Pao9629>