



CENTRE ESPECÍFIC
D'EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA

ACTIVITAT AVALUABLE

Programació
CFGs DAW

Autors:

Carlos Cacho y Raquel Torres

Revisat per:

Joan V. Cassany – jv.cassany@edu.gva.es

Guillermo Garrido – g.garrido@edu.gva.es

José Cantó Alonso – j.cantoalonso@edu.gva.es

Salvador Rue – s.rueorquin@edu.gva.es

2023/2024

Llicència



[CC BY-NC-SA 3.0 ES](#) Reconeixement – No Comercial – Compartir Igual (by-nc-sa) No

es permet un ús comercial de l'obra original ni de les possibles obres derivades,
la distribució de les quals s'ha de fer amb una llicència igual a la que regula l'obra original.
Aquesta és una obra derivada de l'obra original de Carlos Cacho i Raquel Torres.

ACTIVITAT AVALUABLE

‘AFONAR LA FLOTA’

EL VIDEOJOC



1. OBJECTIUS

Realitzar un programa Java que permeta jugar a una versió simplificada d'Afonar la flota.

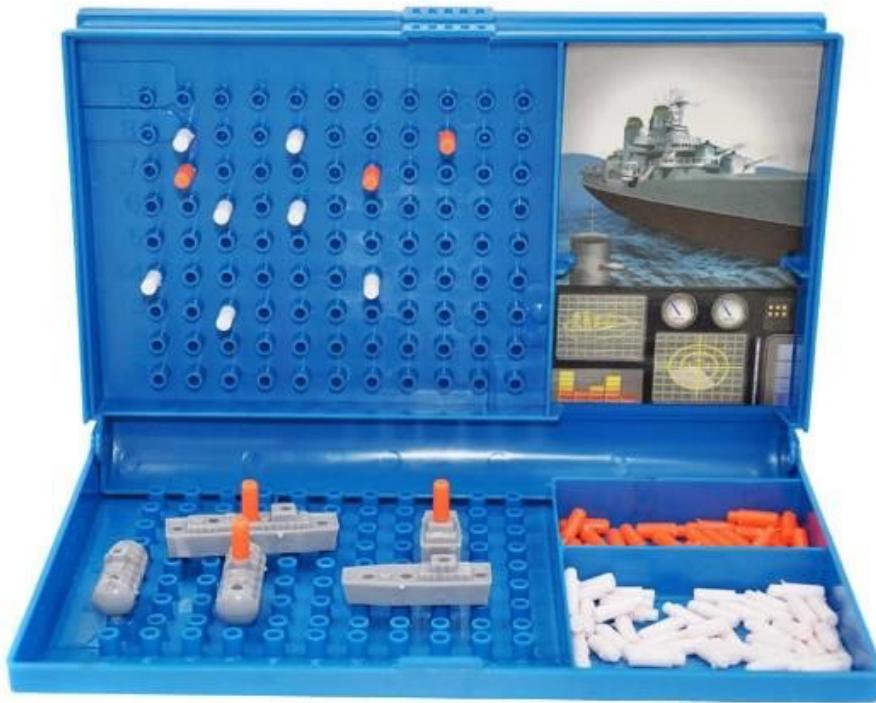


Imagen extraída de todocoleccion.net

2. DESCRIPCIÓ DEL JOC ORIGINAL

En el **joc original**, cada jugador disposa de dos taulers dividits en caselles. Cada tauler representa una zona diferent de la mar oberta: la pròpia i la contrària.

En el tauler propi el jugador col·loca els seus barcos i registra els trets de l'oponent llança contra ell, de manera que té el control de on ha fet els trets i quins barcos propis han segut abatuts.

En el altre tauler es registren els trets propis contra l'adversari i on es van identificant les posicions dels barcos del contrincant

En començar, cada jugador posiciona els seus barcos en el primer tauler, de manera secreta, invisible a l'oponent.

Una vegada tots els barcos han sigut posicionats, s'inicia una sèrie de rondes. En cada ronda, cada jugador, en el seu torn, dispara cap a la flota del seu oponent, indicant una posició (les coordenades d'una casella) i la registra en el segon tauler. Si aqueixa posició és ocupada per part d'un barco contrari, l'oponent cantarà **Tocat!** si encara queden parts del barco (caselles) sense danyar, o **Enfonsat!** si amb aqueix tret el barco ha quedat totalment destruït. Si la posició indicada no correspon a cap part d'un barco, cantarà **Aigua!**.

Guanyarà el jugador que destruïsca primer tots els barcos del seu oponent.

Text extret de Wikipedia: [https://es.wikipedia.org/wiki/Batalla_naval_\(juego\)](https://es.wikipedia.org/wiki/Batalla_naval_(juego))

3. DESCRIPCIÓ DE LA VERSIÓ SIMPLIFICADA

En la versió simplificada que heu de programar només hi haurà **un jugador** humà (usuari) l'objectiu del qual serà **afonar tots els barcos de l'ordinador** en un nombre determinat d'intents. L'ordinador tindrà un tauler amb barcos i l'usuari dispararà per a intentar afonar-los.

L'ordinador col·locarà els barcos en unes posicions aleatòries que no coneixerem fins el final del joc quan es mostre el tauler complet amb el resultat de la partida. Els barcos es situaran en el tauler sempre en posició horitzontal sempre a partir de la posició aleatòria calculada i cap a la dreta. Cal assegurar-se que els barcos en cap moment es solapen.

Nosaltres, a cada tret que llancen podrem veure el tauler dels nostres atacs que ens servirà per a decidir on fer el següent tret.

Les columnes del tauler s'identificaran de forma numèrica començant pel 0. Igualment les files s'identificaran de forma alfabètica començant per l' A.

Els tipus de barcos són:

- Llanxa (L): ocupa una casella del tauler.
- Vaixell (B): ocupa 3 caselles horizontals consecutives del tauler.
- Cuirassat (Z): ocupa 4 caselles horizontals consecutives del tauler.
- Portaavions (P): ocupa 5 caselles horizontals consecutives del tauler.

Per tant, el contingut de les caselles al llarg de la partida podrà ser:

- Llanxa: L
- Vaixell: B
- Cuirassat: Z
- Portaavions: P
- Aigua: A
- Posició oculta: -
- Posició tocada o afonada: X

Un exemple de **tauler amb els barcos** que l'ordinador ha situat en el tauler seria:

	0	1	2	3	4	5	6	7	8	9
A	-	-	B	B	B	-	-	-	-	-
B	-	-	-	-	-	-	-	-	-	-
C	-	-	-	-	-	-	-	-	-	-
D	L	-	P	P	P	P	P	-	-	-
E	-	-	-	-	-	-	-	-	-	-
F	-	-	-	-	-	-	-	-	-	-
G	-	-	-	-	-	B	B	B	-	-
H	-	C	C	C	C	-	-	-	-	-
I	-	-	B	B	B	-	-	-	-	-
J	L	L	-	L	-	-	-	L	-	-

Aquest tauler romandrà ocult i sols es mostrerà al finalitzar la partida amb tots els trets efectuats per l'usuari.

Al principi el tauler visible al jugador mostrarà totes la seua posicions a ocultes (-), així:

	0	1	2	3	4	5	6	7	8	9
A	-	-	-	-	-	-	-	-	-	-
B	-	-	-	-	-	-	-	-	-	-
C	-	-	-	-	-	-	-	-	-	-
D	-	-	-	-	-	-	-	-	-	-
E	-	-	-	-	-	-	-	-	-	-
F	-	-	-	-	-	-	-	-	-	-
G	-	-	-	-	-	-	-	-	-	-
H	-	-	-	-	-	-	-	-	-	-
I	-	-	-	-	-	-	-	-	-	-
J	-	-	-	-	-	-	-	-	-	-

Aquest serà el tauler que se'ns anirà mostrant després de cada tret.

Per a que l'usuari llance cada tret se li demanen les coordenades (fila i columna) del tret, les quals no s'acceptaran fins que siguin vàlides.

Si eixa posició està ocupada per un barco es mostrerà el missatge “**Tocat!**” . **No es contempla el missatge “Enfonsat!” per a reduir la dificultat.** En el cas que no hi haja cap barco en aqueixa posició, es mostrerà el missatge “**Aigua!**”. Al mateix temps s'actualitzarà eixa posició en el tauler marcant-la amb una “**X**” (si hi havia barco) o amb una “**A**” (si hi era aigua).

A continuació es torna a mostrar el tauler visible actualitzar i es demana un nou tret a l'usuari.

L'usuari continuarà realitzant trets fins que afone tots els barcos, i en aquest cas li apareixerà el missatge de “**Has guanyat!**”, o bé, si el jugador no ha aconseguit afonar tots els barcos en els intents que tenia, amb el que li semblarà el missatge de “**Has perdut!**”. En tots dos casos es mostrerà el tauler ocult de l'ordinador i el joc finalitzarà.

4. CONSELLS I CONSIDERACIONS

Abans de començar a programar llig bé l'enunciat, subratlla l'important, fes-te notes, escriu un esborrany de les possibles funcions, etc. Intenta entendre el problema i com realitzar-lo abans de començar a escriure codi.

No intentes fer-ho tot de cop. Una estratègia clau en programació és dividir un problema gran en diversos problemes xicotets i després resoldre'ls un a un (és a dir, programa i prova les funcions una a una).

Comença fent el funcionament bàsic i a poc a poc veu afegint els altres. És molt aconsellable **fer còpies de seguretat** del codi després de cada etapa.

És obligatori escriure a dalt de cada funció un comentari explicant què fa.

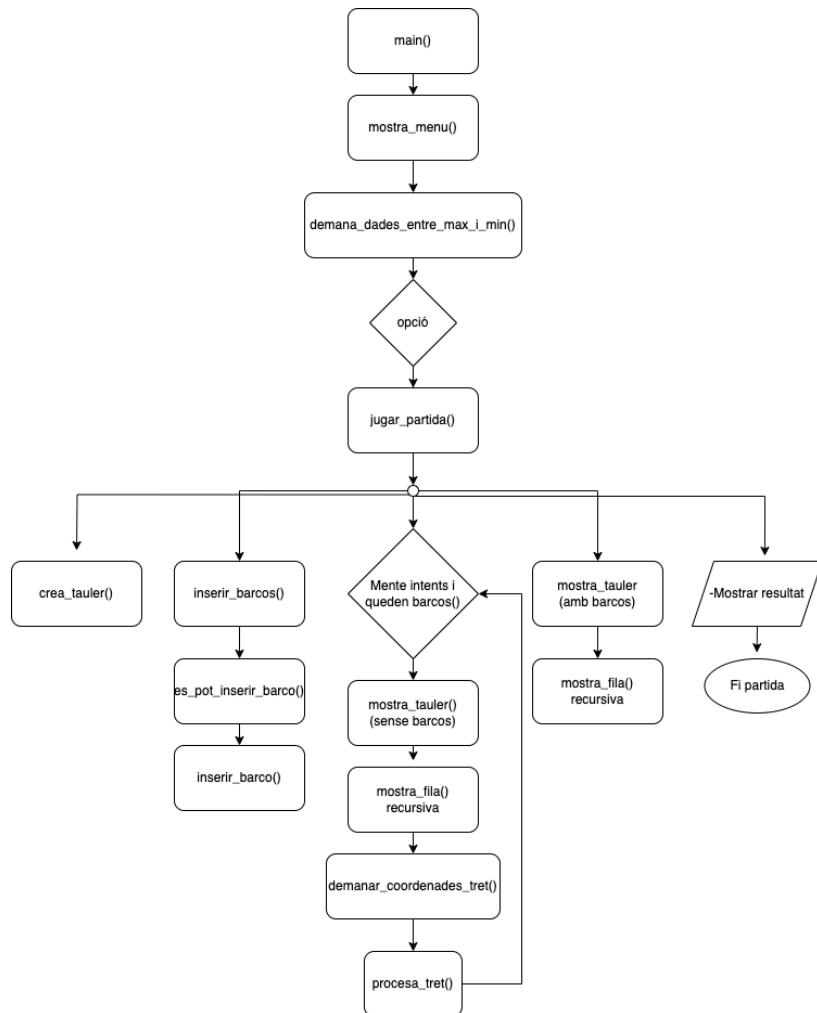
Una bona pràctica és que les funcions no tinguen més de **50-60 línies de codi**, inclosa la funció main. No és una norma d'obligatori, però sí estem fent una bona divisió del problema generalment no han d'aparèixer funcions massa llargues.

IMPORTANT!!: A continuació es descriu la funcionalitat completa del videojoc però es recomana desenvolupar-lo per etapes, començant amb el funcionament bàsic i després afegint els altres aspectes a poc a poc.

5. ESTRUCTURA DEL PROGRAMA I ÚS DE FUNCIONS

5.1. DIAGRAMA DEL JOC

A continuació es mostra un diagrama del joc per a facilitar la comprensió de la seua estructura i de les explicacions que es donen en aquest apartat.



5.2. INICI DEL JOC

Menú inicial

Partirem de la creació del menú principal on se mostrerà a l'usuari les següents opcions per a triar el nivell:

===== BENVINGUTS A AFONAR LA FLOTA =====

Nivells de dificultat:

1. Fàcil: 5 llanxes, 3 vaixells, 1 cuirassat i 1 portaavions (50 trets).
2. Mitjà: 2 llanxes, 1 vaixell, 1 cuirassat i 1 portaavions (30 trets).
3. Difícil: 1 llanxa y 1 vaixell (10 trets).

Quin nivell tries?

Açò es programarà en el mètode `main` des de on cridarem a un mètode per a validar les dades `demanas_dades_entre_max_i_min` i posteriorment s'accedeix al mètode d'inici de partida denominat `jugar_partida` que rebrà els **paràmetres** corresponents segons el nivell triat. A mes a mes, també rebrà la mida del tauler que sempre serà de 10x10.

Validar les dades del nivell de dificultat

Per a simplificar el procés de validació de les dades (totes són numèriques), pots crear una sola funció per a validar el nivell que vol amb `demanas_dades_entre_max_i_min`.

5.3. JUGAR LA PARTIDA

Crearem el mètode `jugar_partida` la qual contemplarà:

- Indicar inici de joc donant la benvinguda
- Crear tauler (`crear_tauler`) segons la mida especificada
- Situar els barcos (`inserir_barco`)
- Processar els trets mentre queden intents i queden barcos sense afonar (`queden_barco`) -
Mostrarem el tauler (`mostrar_tauler`) sense barcos per que l'usuari puga triar les coordenades del tret
- Demanar coordenades del tret (`demanas_coordenades_tret`)
- Processar tret (`processa_tret`)
- Mostrarem el resultat final de tauler (`mostrar_tauler`) amb la situació final en que han quedat els trets i els barcos.
- Donar el resultat de la partida segons haja guanyat (afonat tots els barcos) o no.

5.4. MÈTODES DEL TAUER

Creació del tauler

La creació del tauler (`crear_tauler`) es farà a partir del nombre de files i columnes triades i retornarà el tauler inicialitzat amb totes les posicions ocultes “-”.

Visualització del tauler

El tauler es mostrerà (`mostrar_tauler`) de dos formes: la que s'utilitza durant la partida amb els trets per a que el usuari desidisca el següent tret, i una altra per al final de la partida amb el contingut complet del tauler. Per fer les dues opcions amb un sol procediment simplement haurem d'indicar-li el tipus de visualització que volem.

En la primera no es visualitzarà cap barco, sols es mostrerà els trets realitzats fins el moment: “X” en posicions de tocat, “A” en posicions de aigua i “-” on no s'ha fet cap tret.

En la segona es visualitzar el tauler complet, amb els barcos i la situació final dels trets que s'han realitzat.

5.5. MÈTODE D'INSERIR ELS BARCOS

Obtindre una posició aleatòria

Crearem una funció **coordenada_aleatoria** que ens dona una posició aleatòria dins del tauler. La funció rebrà el tauler per poder fer ús de les dimensions i la mida del barco que es vol inserir, per a evitar donar una coordenada que situe al barco o part de aquest fora del tauler, es a dir, condicionarà la posició final de la fila on es pot inserir. La funció ens tornarà les coordenades (en un vector de dos posicions, en la primera la fila i en la segona la columna).

Recorda que el més senzill és utilitzar el mètode **Math.random()** que proporciona un número pseudoaleatori tipus 'double' entre 0.0 i 1.0.

Afegint algunes operacions matemàtiques podem obtindre un nombre enter aleatori en el rang que ens interessa:

Código Java	Número aleatorio generado
<code>Math.random()</code>	Tipo double entre 0.0 y 1.0
<code>(int) (Math.random() * (N+1))</code>	Tipo int entre 0 y N
<code>(int) (MIN + Math.random() * (MAX - MIN + 1))</code>	Tipo int entre MIN y MAX

Inserir tots els barcos de cadascun dels tipus

El mètode **inserir_barcos** situarà tots els barcos en el tauler que vindrà determinat pel nivell seleccionat. Per tant, necessitarà coneixer el tauler i el nombre de cadascun dels tipus barcos que ha d'inserir. Igualment, per cada barco a inserir haurà d'obtindre unes coordenades aleatòries i validar que al inserir un barco aquest lloc no es solapa amb altre ja existent en el tauler. Si després de 100 intents de inserir el barco no es capaç de aconseguir-ho, no l'insertarà i mostrara el text "ERROR: No s'ha pogut inserir el barco".

Si tot és correcte, es procedirà a inserir el barco, el procés es repartirà per a tots els barcos de cadascun dels tipus.

5.6. MÈTODES DE TRETS

Sol·licitud de coordenades del tret

Després d'haver mostrat el tauler amb els trets realitzats a l'usuari (com ja s'ha explicat previament) li demanarem que ens dona les coordenades del següent tret. Es sol·licitarà la fila i la columna per separat i les verificarem fins que siguen vàlides.

Processar el tret

Processarem el tret amb la funció **processa_tret** a partir del tauler i les coordenades donades per l'usuari. El tret podrà donar com a resultat tres casos:

- **Aigua!** Si hi havia un»-»
- **Tocat!** Si hi havia una lletra de barco
- **Error! ja hi havies fet un tret ací** Si ja hi havia un tret previ en la casella.

A continuació s'actualitzarà la posició corresponent a les coordenades amb el valor adequat.

5.7. FI DE LA PARTIDA

Es mostrerà el tauler complet (barcos i trets), es donarà un missatge indicant a l'usuari si ha guanyat o no i es mostrerà un missatge d'acomiadament.

6. LLISTA DEL MÈTODES A IMPLEMENTAR Nom de la clase

```
public class AV1_Avaluable2_EnfonsarLaFlota {
```

Mètode main

```
//PROCEDIMENT: Mètode principal que crida al mètode mostrar_menu  
public static void main (String[] args){}
```

Mètode mostrar menú

```
//PROCEDIMENT: mostra el menu d'opcions d'inici de partida, crida al mètode  
demana_dades_entre_max_i_min  
public static void mostrar_menu() {}
```

//FUNCió: demana l'opció i comprova que el valor es troba entre el nombre d'opcions disponibles.

```
public static int demanda_dades_entre_max_i_min (String text, int min, int max) {}
```

Mètode jugar partida public static void jugar_partida(int files, int columnes, int llanxes, int vaixells, int cuirassats, int portaavions, int max_intents) {}

Mètodes tauler

```
// FUNCió: Crear tauler buit amb "-" en totes les posicions public  
static char[][] crear_tauler(int files, int columnes) {}
```

// PROCEDIMENT: Mostra per pantalla el tauler, primer mostra la capçalera i en acabant recorre un for on, per cada fila crida la funció recursiva mostra_fila.

```
public static void mostra_tauler(char[][] tauler, boolean veureTot) {}
```

//PROCEDIMENT RECURSIU: Mostra per pantalla cadascuna de les files de la matriu.

```
private static void mostra_fila(char[][] tauler, int i, int j, boolean mT){}
```

Mètodes situar barcos

// PROCEDIMENT: Inserir els barcos en posicions aleatòries, per a cada barco, crida al mètode inserir_barco.

```
public static void inserir_barcos (char[][] tauler, int llanxes, int vaixells, int cuirassats, int portaavions) {}
```

//FUNCió: Inserir un barco, primer comprova que el barco cap al tauler i es pot inserir mitjançant la funció es_pot_inserir_barco, si esta es positiva insereix el barco i tornarà true.

```
public static boolean inserir_barco(char[][] tauler, int tamany, char lletra){}
```

//FUNCIÓ: Comprova si cap i é spot inserir un barco al tauler, tornarà true si es possible.

```
public static boolean es_pot_inserir_barco(char[][] tauler, int tamany, int f, int c){}
```

Mètodes tret

// FUNCIÓ: Demana coordenades de tret

```
public static int[] demana_coordenades_tret(char[][] tauler) {}
```

// PROCEDIMENT: Processem el tret amb les coordenades indicades public
static void processa_tret(char[][] tauler, int f, int c) {}

Mètodes genèriques i de validació

// FUNCIÓ: Rep el tauler i dimensions de l'objecte i ens retorna una posició aleatòria public
static int[] coordenada_aleatoria(char[][] tauler, int mida) {}

// FUNCIÓ: Comprovar si quede barcos en el tauler public
static boolean queden_barcos(char[][] tauler) {}

// FUNCIÓ: Demana el valor indicat del menú a l'usuari i valida que siga correcte. public
static int demana_dades_entre_max_i_min (String text, int min, int max) {}

7. DESENVOLUPAMENT PER ETAPES I CRITERIS DE QUALIFICACIÓ

1.- **Videojoc Bàsic (5 punts):** Es pot jugar però només hi ha 10 llanxes (no hi ha altres tipus de barcos) i sense nivells de dificultat (el joc acaba després de 50 trets).

2.- **Afegir tots els barcos (2 punts):** Afegir tots els tipus de barcos (Llanxa, Vaixell, Cuirassat i Portaavions). Segueixen sense haver-hi nivells de dificultat, es juga en nivell fàcil.

3.- **Afegir 3 nivells de dificultat (3 punts):** Afegir l'opció de triar el nivell de dificultat (només fàcil, mitjà i difícil).

En cada etapa desenvolupada es valorarà:

- Correcte funcionament del videojoc.
- Correcte ús de funcions i evitar repetir codi (en la mesura que siga possible).
- Codi ordenat i ben estructurat.
- Noms de variables apropiats i autoexplicatius.
- Comentaris útils i breus que ajuden a entendre el codi.