

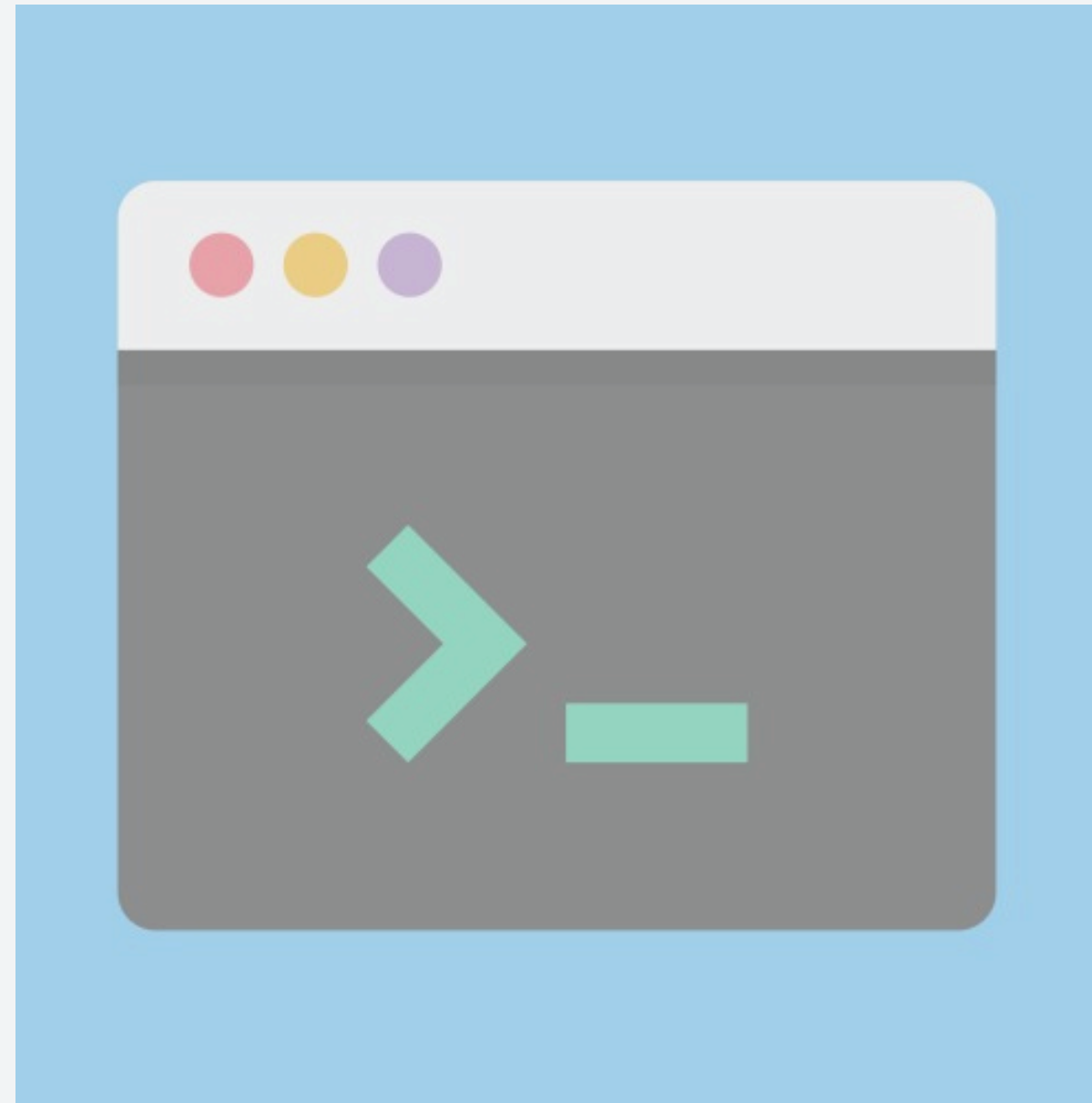


# **BATAILLE NAVALE**

Résultats console

**ZEKI INCEOGLU & DENIS ZAPOI**

**VOUS TROUVEREZ DANS CE PDF PLUSIEURS  
FONCTIONS/PROCÉDURES DONT ONT VA ANALYSER LE  
CONTENU AINSI QUE LES RÉSULTATS DANS LA CONSOLE**



# **La méthode “ajouterSurPlateau”**

**Cette méthode est le point central de notre logiciel.**

**En effet, elle nous permet de placer les bateaux de façon graphique sur le plateau de jeu. Les bateaux en eux même sont saisis grâce à la fonction saisirBat, les bateaux sont stockés sous forme de liste, de la façon suivante :**

- Si le bateau est de longueur 1 , la liste sera de longueur 2, avec la position Y et X du bateau**
- Si le bateau est de longueur 2 ou 3 , la liste sera de longueur 3, avec la position Y et X du bateau mais aussi sa direction (1 pour Nord, 2 pour Sud etc...)**

**Cette méthode utilise encore une autre fonction (booléenne) qui vérifie si le bateau peut être placé ou si sa position est illégale (en dehors du plateau ou chevauche un autre**

# TESTS DE CETTE PROCÉDURE

COMMENCONS PAR UN CAS SIMPLE, PRENONS UN BATEAU DE LONGUEUR 1 EN (2,4), UN BATEAU DE LONGUEUR 2 EN (3,5 ORIENTATION SUD)

ET ENFIN UN BATEAU DE LONGUEUR 3 EN (6,9, DIRECTION SUD)

**NB: ON  
METS LES  
Y AVANT  
LES X**

-	1	2	3	4	5	6	7	8	9	10
A	-	-	-	-	-	-	-	-	-	-
B	-	-	-	-	-	-	-	-	-	-
C	-	-	-	-	-	-	-	-	-	-
D	-	-	-	-	-	-	-	-	-	-
E	-	-	-	-	-	-	-	-	-	-
F	-	-	-	-	-	-	-	-	-	-
G	-	-	-	-	-	-	-	-	-	-
H	-	-	-	-	-	-	-	-	-	-
I	-	-	-	-	-	-	-	-	-	-
J	-	-	-	-	-	-	-	-	-	-

VOICI NOTRE TABLEAU DE JEU  
VIDE, AVANT LES REQUETES

```
Le bateau à mettre est de taille : 1  
donner y et après x
```

```
2
```

```
4
```

```
Le bateau à mettre est de taille : 2  
donner y et après x
```

```
3
```

```
5
```

```
saisir la direction de la façon suivante : 1 pour nord 2 pour sud 3 pour est et 4 pour ouest
```

```
2
```

```
Le bateau à mettre est de taille : 3  
donner y et après x
```

```
6
```

```
9
```

```
saisir la direction de la façon suivante : 1 pour nord 2 pour sud 3 pour est et 4 pour ouest
```

```
2
```

**VOICI LES 3  
INPUTS QUE  
LE JOUEUR  
FAIT AFIN DE  
DONNER LES 3  
BATEAUX**

# RESULTAT GRAPHIQUE SUR LE PLATEAU DE JEU

-	1	2	3	4	5	6	7	8	9	10
A	-	-	-	-	-	-	-	-	-	-
B	-	-	-	X	-	-	-	-	-	-
C	-	-	-	-	X	-	-	-	-	-
D	-	-	-	-	X	-	-	-	-	-
E	-	-	-	-	-	-	-	-	-	-
F	-	-	-	-	-	-	-	-	X	-
G	-	-	-	-	-	-	-	-	X	-
H	-	-	-	-	-	-	-	-	X	-
I	-	-	-	-	-	-	-	-	-	-
J	-	-	-	-	-	-	-	-	-	-

COMME VOUS POUVEZ VOIR ,TOUT EST BIEN PLACÉ ,  
COMME ATTENDU

Le batteau à mettre est de taille : 1

donner y et après x

11

11

donner y et après x

11

11

donner y et après x

1

1

T'as mal saisi le bateau1

Le batteau à mettre est de taille : 1

donner y et après x

1

1

donner y et après x

1

1

Le batteau à mettre est de taille : 2

donner y et après x

2

4

**SI LE BATEAU EST HORS LIMITES  
IL NE SERA PAS PLACÉ ET LE  
PROGRAMME REDEMANDE DES  
INPUTS ET PASSE AU PROCHAIN  
BATEAU SI LE BATEAU  
PRÉCÉDENT EST VALIDE**

Le batteau à mettre est de taille : 3

donner y et après x

5

6

saisir la direction de la façon suivante : 1 pour nord 2 pour sud 3 pour est et 4 pour ouest

2



**LES BATEAUX  
SONT BIEN  
PLACÉS MEMEME  
APRES DE  
MAUVAIS INPUTS**

[illegible]



**Voyons maintenant la fonction  
booléenne peutEtrePlace**

**CETTE FONCTION RETOURNE DONC UN  
RÉSULTAT VRAI OU FAUX QUI PERMET DE  
SAVOIR SI UN BATEAU EST BIEN PLACÉ  
OU NON, D'AUTRE SOUS-PROGRAMMES  
DÉPENDENT DE CETTE FONCTION**

# FONCTIONNEMENT:

**LA FONCTION PREND EN PARAMETRE LE TABLEAU DE JEU, LE BATEAU STOCKÉ SOUS FORME DE LISTE (Y, X, DIRECTION) ET LA LARGEUR DU BATEAU**

**ENSUITE, EN FONCTION DE LA DIRECTION, ELLE REGARDE SI TOUTES LES CASES CONCERNÉES SONT LIBRES OU NON, SI OUI ELLE RETOURNE VRAI.**

**UNE CASE EST CONSIDÉRÉE LIBRE SI ELLE EST ÉGALE À “-”**

**EXEMPLE PRATIQUE, SI LA LISTE RECUE EN PARAMETRE EST (2, 4, 2) ET LONGUEUR 3, ALORS ELLE VA VÉRIFIER À PARTIR DE LA CASE 2, 4 JUSQU'À LA LONGUEUR DONNÉE, DANS LA DIRECTION DONNÉE (ICI 2, DONC SUD)**

```
ArrayList<Integer> tabtestFaux = new ArrayList<>();  
tabtestFaux.add(1);  
tabtestFaux.add(10);  
tabtestFaux.add(3);  
System.out.println(peutEtrePlacé(tabtestFaux, plateau, l: 3));
```

```
ArrayList<Integer> tabtestVrai = new ArrayList<>();  
tabtestVrai.add(5);  
tabtestVrai.add(5);  
tabtestVrai.add(1);  
System.out.println(peutEtrePlacé(tabtestVrai, plateau, l: 1));
```

# VOICI DEUX TESTS, UN VRAI ET UN FAUX

```
false  
true
```

## ÇA MARCHE BEL ET BIEN DONC