

# Bataille navale en Python

## Table of Contents

1. Présentation du problème
2. Travail demandé

## 1. Présentation du problème

On voudrait réaliser un jeu de bataille navale en Python (au moins, les éléments de base...).

Un jeu de bataille navale se compose d'une grille et d'un ensemble de bateaux, chaque bateau se compose d'un *ensemble de taille fixe* d'éléments : Un croiseur comprend 3 éléments, un escorteur 2 éléments et un sous-marin un seul élément. Chaque élément est caractérisé par sa position et par son état : sain ou touché. Les sous-marins ont la possibilité de plonger. Lorsqu'ils plongent ils ne peuvent pas être touchés.

Voici comment on instancie une flotte de bateaux (qui correspond à la figure) :

### Exemple de code utilisateur

PYTHON

```
from grille import Grille
import bateau

def affiche(code):
    if code == 0:
        print("Dans l'eau..")
    elif code == 1 or code == 2:
        print("Touché..")
    elif code == 3:
        print("Touché Coulé !")
    else:
        print("Coup non valide")

b1 = bateau.Croiseur(1, 1, True)    # True => horizontal
b2 = bateau.Escorteur(2, 5, False)  # False => vertical
b3 = bateau.SousMarin(4, 2, True)   # True => en surface

g1 = Grille(7, 9)
g1 += b1
g1 += b2
g1 += b3

print(g1)                          # Affiche "Croiseur Escorteur Sous-Marin"
affiche(g1.coup(1, 1))              # Affiche "Touché"
affiche(g1.coup(1, 2))              # Affiche "Touché"
affiche(g1.coup(1, 3))              # Affiche "Touché Coulé !"
print(g1)                          # Affiche "Escorteur Sous-Marin"
```

### Exemple de bataille navale :



En résumé, le jeu est représenté par une *grille* qui contient des *bateaux*. Un bateau est soit un *escorteur*, soit un *croiseur*, soit un *sous-marin*. Un bateau est formé d'*éléments*. Un bateau est coulé lorsque tous ses éléments ont été *touchés*.

Cependant, bien que l'utilisateur « voit » une grille, le jeu ne sera pas implémenté de cette façon...

## 2. Travail demandé

1. Créez une classe `Grille` qui implémentera le plateau de jeu comme une *liste*.

- Cette classe fournira l'opérateur `+=` pour lui ajouter un nouveau bateau.
- Elle redéfinira la méthode `__str__` afin d'afficher (sous forme de chaîne) les bateaux qui sont encore sur le jeu (voir le code au-dessus).
- Elle fournira également les propriétés `nb_lig_max` et `nb_col_max`.
- Enfin, elle fournira une méthode `coup(lig, col)` qui renvoyant `-1` si le coup est en dehors de la grille, `0` si le coup est dans l'eau, `1` si le coup tombe sur un élément déjà touché, `2` si le coup tombe pour la première fois sur un élément et `3` si le coup a coulé un bateau.

Lorsqu'un bateau est coulé, il est supprimé de la grille.



Warning

N'écrivez cette méthode qu'après avoir écrit la méthode `est_touche` de `Bateau` puisque, en gros, son travail consiste à demander à chaque bateau s'il a été touché...

2. Créez une classe `Element` qui représente un élément d'un bateau.

- Cette classe doit fournir la propriété `est_touche` qui renvoie `True` si l'élément a déjà été touché par un coup et les propriétés `lig` et `col` qui renvoient, respectivement, la ligne et la colonne de l'élément.
- Elle redéfinit la méthode `toString()` pour renvoyer un `"X"` si l'élément est touché, un `" "` sinon.
- Enfin, elle dispose d'une méthode `toucher()` qui permet de « toucher » l'élément.

3. Créez la classe `Bateau` qui sera la classe de base de tous les bateaux.

- Cette classe doit définir une méthode `est_touche(lig, col)` qui renvoie `0` si le coup de coordonnées `(lig, col)` ne touche pas le bateau, `1` si le bateau était déjà touché, `2` si c'est la première fois qu'il est touché et `3` si le bateau a été coulé.

4. Créez les classes `Escorteur`, `Croiseur` et `SousMarin`.

- Toutes ces classes redéfinissent `__str__` pour afficher leur type de bateau.
- La classe `SousMarin` doit redéfinir la méthode `est_touche(lig, col)` de `Bateau` pour prendre en compte le fait que le sous-marin doit être en surface pour être touché.

5. Ajoutez une méthode `display` à la classe `Grille` pour l'afficher comme une véritable grille de jeu (en représentant par un symbole les cases occupées et par un autre symbole les cases libres)

Last updated 2021-12-02 09:05:41 +0100