

Bataille Navale

1ere partie : bataille navale avec un bateau d'une case

	1	2	3	4	5	6	7	8	9	10
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

Au départ on se fixe la case contenant le bateau. Cela permet de vérifier si le programme fonctionne correctement.

Exemple : colonne : 4 et ligne : 3

On crée alors deux variables ColonneBateau et LigneBateau donnant les coordonnées du bateau. On relève l'importance de donner des noms de variables explicite pour que d'autres lecteurs peuvent comprendre facilement.

1er travail : après avoir expliqué le jeu, demander aux élèves de simuler le jeu avec leur voisin sur 3 lancers en écrivant ce qu'ils disent et comment ils procèdent pour vérifier si c'est coulé ou non..

Commentaire : les élèves ont beaucoup de mal à décrire leur test.

A ce niveau on introduit les variables i et j.

En programmation, on a l'habitude de mettre i le numéro de la ligne du tableau et j celui de la colonne

2e travail : décrire en langage naturel si **un** lancer de coordonnées (i ; j) coule le bateau ou s'il est dans l'eau.

					j					
	1	2	3	4	5	6	7	8	9	10
1										
2										
3				X						
i										
4										
5										
6										
7										
8										
9										
10										

Algorithme pouvant convenir :

```
Début
  ColonneBateau prend la valeur 4
  LigneBateau prend la valeur 3
  Si (i = LigneBateau et j = ColonneBateau)
  Alors
    Afficher "couler"
  Sinon
    Afficher "Dans l'eau"
  FinSi
Fin
```

On introduit ensuite la saisie de i et de j :

```
Début
    ColonneBateau prend la valeur 4
    LigneBateau prend la valeur 3
    Saisir i
    Saisir j
    Si (i = LigneBateau et j = ColonneBateau)
    Alors
        Afficher "couler"
    Sinon
        Afficher "Dans l'eau"
    FinSi
Fin
```

3e travail : introduire la boucle "tant que", qui permet de renouveler l'opération jusqu'au moment où le bateau coule.

On relèvera l'importance de donner une valeur aux variables avant de faire des tests.

Algorithme pouvant convenir :

```
Début
    ColonneBateau prend la valeur 4
    LigneBateau prend la valeur 3
    i prend la valeur -1
    j prend la valeur -1
    Tant que (i ≠ LigneBateau ou j ≠ ColonneBateau)
        Saisir i
        Saisir j
        Si (i = LigneBateau et j = ColonneBateau)
        Alors
            Afficher "couler"
        Sinon
            Afficher "Dans l'eau"
        FinSi
    FinTantque
Fin
```

Remarque : on peut optimiser l'algorithme avec des élèves vraiment motivés.

4e travail : donner une valeur aléatoire aux deux variables ColonneBateau et LigneBateau.

```
ColonneBateau prend la valeur aléatoire compris entre 1 et 10
LigneBateau prend la valeur aléatoire compris entre 1 et 10
```

5e travail : (facultatif) ajouter un compteur de lancer à l'algorithme
Algorithme pouvant convenir :

```
Début
    ColonneBateau prend la valeur aléatoire compris entre 1 et 10
    LigneBateau prend la valeur aléatoire compris entre 1 et 10
    i prend la valeur -1
    j prend la valeur -1
    Compteur prend la valeur 0
    Tant que (i ≠ LigneBateau ou j ≠ ColonneBateau)
        Saisir i
        Saisir j
        Compteur prend la valeur Compteur +1
        Si (i = LigneBateau et j = ColonneBateau)
            Alors
                Afficher "couler"
                Afficher Compteur
            Sinon
                Afficher "Dans l'eau"
        FinSi
    FinTantque
Fin
```

6e travail : (facultatif) limiter le nombre de lancer
Algorithme pouvant convenir :

```
Début
    ColonneBateau prend la valeur aléatoire compris entre 1 et 10
    LigneBateau prend la valeur aléatoire compris entre 1 et 10
    NbLancer prend la valeur 5
    i prend la valeur -1
    j prend la valeur -1
    Compteur prend la valeur 0
    Tant que (Compteur ≤ NbLancer et (i ≠ LigneBateau ou j ≠ ColonneBateau))
        Saisir i
        Saisir j
        Compteur prend la valeur Compteur +1
        Si (i = LigneBateau et j = ColonneBateau)
            Alors
                Afficher "couler"
                Afficher Compteur
            Sinon
                Afficher "Dans l'eau"
        FinSi
    FinTantque
    Si (Compteur > NbLancer)
        Alors
            Afficher "Vous n'avez plus de munitions"
        FinSi
Fin
```

2ème partie : bataille navale avec un bateau de 3 cases horizontales.

On cherche à couler un bateau de taille 3 cases horizontales.

On crée cette fois 4 variables :

LigneBateau ; ColonneBateau1 ; ColonneBateau2 ; ColonneBateau3.

On procède de la même manière en donnant au départ la position du bateau :

```
LigneBateau prend la valeur 3
ColonneBateau1 prend la valeur 4
ColonneBateau2 prend la valeur 5
ColonneBateau3 prend la valeur 6
```

Il faut penser à neutraliser les éléments coulés, pour ne pas les comptabiliser deux fois.

Algorithme pouvant convenir :

```
Début
  LigneBateau prend la valeur 3
  ColonneBateau1 prend la valeur 4
  ColonneBateau2 prend la valeur 5
  ColonneBateau3 prend la valeur 6
  BateauTouche prend la valeur 3
  Tant que (BateauTouche ≠ 0)
    Saisir i
    Saisir j
    Si (i = LigneBateau)
      Alors
        Si (j = ColonneBateau1)
          Alors
            BateauTouche prend la valeur BateauTouche - 1
            ColonneBateau1 prend la valeur -2
          Sinon
            Si (j = ColonneBateau2)
              Alors
                BateauTouche prend la valeur BateauTouche - 1
                ColonneBateau2 prend la valeur -2
              Sinon
                Si (j = ColonneBateau3)
                  Alors
                    BateauTouche prend la valeur BateauTouche - 1
                    ColonneBateau3 prend la valeur -2
                  Sinon
                    Afficher "Dans l'eau"
                FinSi
            FinSi
          FinSi
        FinSi
      Si (BateauTouche = 0)
        Alors
          Afficher "couler"
        FinSi
      Sinon
        Afficher "Dans l'eau"
      FinSi
    FinTantque
  Fin
```

Travail suivant : donner une valeur aléatoire aux deux variables ColonneBateau1 et LigneBateau, en faisant en sorte de ne pas sortir de la grille :

LigneBateau prend la valeur aléatoire compris entre 1 et 10 ColonneBateau1 prend la valeur aléatoire compris entre 1 et 8 ColonneBateau2 prend la valeur ColonneBateau1+1 ColonneBateau3 prend la valeur ColonneBateau1+2
--

Pour généraliser la bataille navale, on peut expliquer aux élèves qu'il vaut mieux créer un tableau de variables pour placer les positions des bateaux, vu qu'il y a au total 100 cases possibles. Mais ce n'est pas au programme.