

B1 : Bataille Navale

B1 : Bataille Navale.....	1
Etape 1:.....	2
Etape 2:.....	2
Etape 3:.....	3
Etape 4:.....	4
Etape 5:.....	5
Etape 6:.....	5
Etape 7:.....	7
Etape 8:.....	8

Etape 1:

```
/* Définition plateau de jeu */  
int tabjoueur[][] = new int [5][5];  
int tabordi[][] = new int [5][5];
```

Etape 2:

2.1

Création des scanner pour demander où placer le pion.

```
Scanner scanner1= new Scanner(System.in);  
System.out.println(x:"Choisir la ligne: 1 à 5");  
int li = scanner1.nextInt()-1;
```

```
scanner1= new Scanner(System.in);  
System.out.println(x:"Choisir la colonne: 1 à 5");  
int col = scanner1.nextInt()-1;
```

2.3

Boucle de vérification pour ne pas placer 2 fois le même pion

```
while (tabjoueur[li][col] == 1){  
    System.out.println(x:"Un pion est déjà ici recommencez");  
    Scanner scanner2= new Scanner(System.in);  
    System.out.println(x:"Choisir la ligne: 1 à 5");  
    li = scanner2.nextInt()-1;  
  
    scanner2= new Scanner(System.in);  
    System.out.println(x:"Choisir la colonne: 1 à 5");  
    col = scanner2.nextInt()-1;  
}  
tabjoueur[li][col] = 1;  
affichage(tabjoueur);
```

2.4

Boucle permettant de placer 5 pion

```
for (int i = 1 ; i <= 5 ; i++){  
    /* Demande de placement des pion et vérification */  
}
```

Etape 3:

3.1½

Demande du placement des pions a l'ordinateur

```
int lior = (int)(Math.random()*(4))+0;  
int color = (int)(Math.random()*(4))+0;
```

3.3

Vérification pour ne pas placer 2 fois le même pion.

```
while (tabordi[lior][color] == 1){  
    lior = (int)(Math.random()*(4))+0;  
    color = (int)(Math.random()*(4))+0;  
}
```

3.4

Boucle permettant de placer 5 pion

```
for (int i = 1 ; i <= 5 ; i++){  
    /* Demande de placement des pion et vérification */  
}
```

Etape 4:

Orange: Définition de la procédure

Rouge: Numérotation ligne

Vert: numérotation des ligne et mise des icônes pour les pion

Bleu: Boucle pour afficher le plateau de jeu entier

```

/* affichage du plateau avec les pions placés visibles */
public static void affichage(int tab[][], int nbcase){
    int n = 1;
    System.out.print(s:"");
    for (int l = 1 ; l <= nbcase ; l++){
        System.out.print(l+"");
    }
    System.out.println(x:"");
    System.out.println(x:"");
    for (int[] i: tab) {
        System.out.print( n + "\t");
        for (int j: i) {
            if (j == 1){
                System.out.print( "o" + "\t");
            }
            else if (j == 0){
                System.out.print( "~" + "\t");
            }
        }
        System.out.println(x:"\n");
        n ++;
    }
}

```

Etape 4.1

Etape 4.2

Etape 4.3

Etape 4.4

Etape 4.5 :

Permet d'afficher le plateau de jeu du joueur

```
affichage(tabjoueur, nbcase:5);
```

Etape 5:

Vert : Les if permettent de créer les états demandés

```
/* affichage du plateau avec les pions cachés et avec les information de tire */
public static void affichageTabCache(int tab[][], int nbcase){
    int n = 1;
    System.out.print(s:"");
    for (int l = 1 ; l <= 5 ; l++){
        System.out.print(l+"");
    }
    System.out.println(x:"");
    System.out.println(x:"");
    for (int[] i: tab) {
        System.out.print( n + "\t");
        for (int j: i) {
            if (j == 0){
                System.out.print( "?" + "\t");
            }
            else if (j == 1){
                System.out.print( "?" + "\t");
            }
            else if (j == 2){
                System.out.print( "o" + "\t");
            }
            else if (j == 3){
                System.out.print( "x" + "\t");
            }
        }
        System.out.println(x:"\n");
        n ++;
    }
}
```

Etape 5

Etape 6:

6.0

Affiche "A vous de jouer !"

```
System.out.println(x:"A vous de jouer !");
```

6.1½

Vert: permet de demander la ligne ou tirer avec les vérification nécessaire

Rouge: permet de demander la colonnes ou tirer avec les vérification nécessaire

```
/* Choix ligne pour le tire */
System.out.println(x:"A vous de jouer !"); Etape 6.1
Scanner scanner1= new Scanner(System.in);
System.out.print(s:"Choisir la ligne où tirer: 1 à 5 : ");
int li = scanner1.nextInt()-1;
li = verificationscan(li);

/* Choix colonne pour le tire */ Etape 6.2
scanner1= new Scanner(System.in);
System.out.print(s:"Choisir la colonne où tirer: 1 à 5 : ");
int col = scanner1.nextInt()-1;
col = verificationscan(col);
```

6.3

```
/* temps de trajet du boulet de canon */
System.out.println(x:"Tir en cours");
Thread.sleep(2*1000);
```

6.4

Vérification pour ne pas tirer 2 fois au même endroit

```
/* Vérification pour ne pas pouvoir tiré 2 fois au même endroit */
while (tabordi[li][col]==2 || tabordi[li][col]==3){
    scanner1= new Scanner(System.in);
    System.out.print(s:"Choisir la ligne où tirer: 1 à 5 : ");
    li = scanner1.nextInt()-1;
    scanner1= new Scanner(System.in);
    System.out.print(s:"Choisir la colonne où tirer: 1 à 5 : ");
    col = scanner1.nextInt()-1;
}
```

Permet d'annoncer au joueur le résultat de son tir.

```
/* resultat du tire */ Etape 6.4.3
if (tabordi[li][col]==1){
    tabordi[li][col] = 2;
    System.out.println(x:"Touché, coulé !");
    scoreJ ++;
}
else if (tabordi[li][col]==0){
    tabordi[li][col] = 3;
    System.out.println(x:"Raté !");
}

/* affichage grille adversaire avec info sur les tire */
affichageTabCache(tabordi, nbcase:5); Etape 6.5
```

Etape 7:

7.0

```
System.out.println(x:"A l'ordinateur de jouer !");
```

7.1/2

```
/* Choix Ordinateur ligne pour le tire */  
int lior = (int)(Math.random()*(4))+0;
```

Etape 7.1

Etape 7.2

```
/* Choix Ordinateur colonne pour le tire */  
int color = (int)(Math.random()*(4))+0;
```

7.3

```
/* vérification pour ne pas tirer au même endroit */  
while (tabjoueur[lior][color]==2 || tabjoueur[lior][color]==3){  
    lior = (int)(Math.random()*(4))+0;  
    color = (int)(Math.random()*(4))+0;  
}
```

7.4

```
/* trajet boulet de canon */  
System.out.println(x:"Tir en cours");  
Thread.sleep(2*1000);
```

7.5

Permet de donner le résultat du tir de l'ordinateur avec la grille joueur

```
/* résultat du tire */  
if (tabjoueur[lior][color] == 1){  
    tabjoueur[lior][color] = 2;  
    System.out.println(x:"Touché, coulé !");  
    score0 ++;  
}  
else{  
    tabjoueur[lior][color] = 3;  
    System.out.println(x:"Raté !");  
}
```

Etape 7.5

Etape 7.6

```
/* affichage grille adverser avec info */  
affichageTabCache(tabjoueur, nbcase:5);
```

Etape 8:

8.1

permet de jouer jusqu'à que l'un des camps ai perdu tous ses bateaux

```
/* Début du jeux */
while (run == true){
    /* Tour Joueur */
    if(scoreJ != 5 && score0 != 5){
        /* code de la manche joueur */
    }
    else{
        run = false;
    }

    /* Tour ordinateur */
    if(scoreJ != 5 && score0 != 5){
        /* code de la manche ordinateur */
    }
    else{
        run = false;
    }
}
```

8.2

mise en forme et résultat de la partie.

```
/* Résultat de la partie */
if (scoreJ>score0){
    System.out.println(x:"");
    System.out.println(x:"");
    System.out.println(x:"Le joueur à gagné !!!");
    System.out.println(x:"");
    System.out.println(x:"");
}
else{
    System.out.println(x:"");
    System.out.println(x:"");
    System.out.println(x:"L'ordinateur à gagné !!!");
    System.out.println(x:"");
    System.out.println(x:"");
}
```


8.3

placement du code pour rejouer.

```
public static void main(String[] args) throws InterruptedException{
    boolean rejoue = true;
    while (rejoue==true){

        /* Code entier du jeu */

        /* ----- */

        /* Demande pour rejouer ou pas */
        Scanner Rejoueoupas = new Scanner(System.in);
        System.out.println("Voulez vous rejouer: 1=Oui 2=Non");
        int resultat = Rejoueoupas.nextInt();
        if (resultat == 1) {
            rejoue = true;
        }
        else if (resultat == 2) {
            rejoue = false;
        }

        /* vérification de la réponse */
        while (resultat != 1 && resultat != 0) {
            Rejoueoupas = new Scanner(System.in);
            System.out.println("| Vous vous êtes tromper il
                               | faut répondre par: 1=Oui 2=Non");
            resultat = Rejoueoupas.nextInt();
            if (resultat == 1) {
                rejoue = true;
            }
            else if (resultat == 2) {
                rejoue = false;
            }
        }
    }
}
```