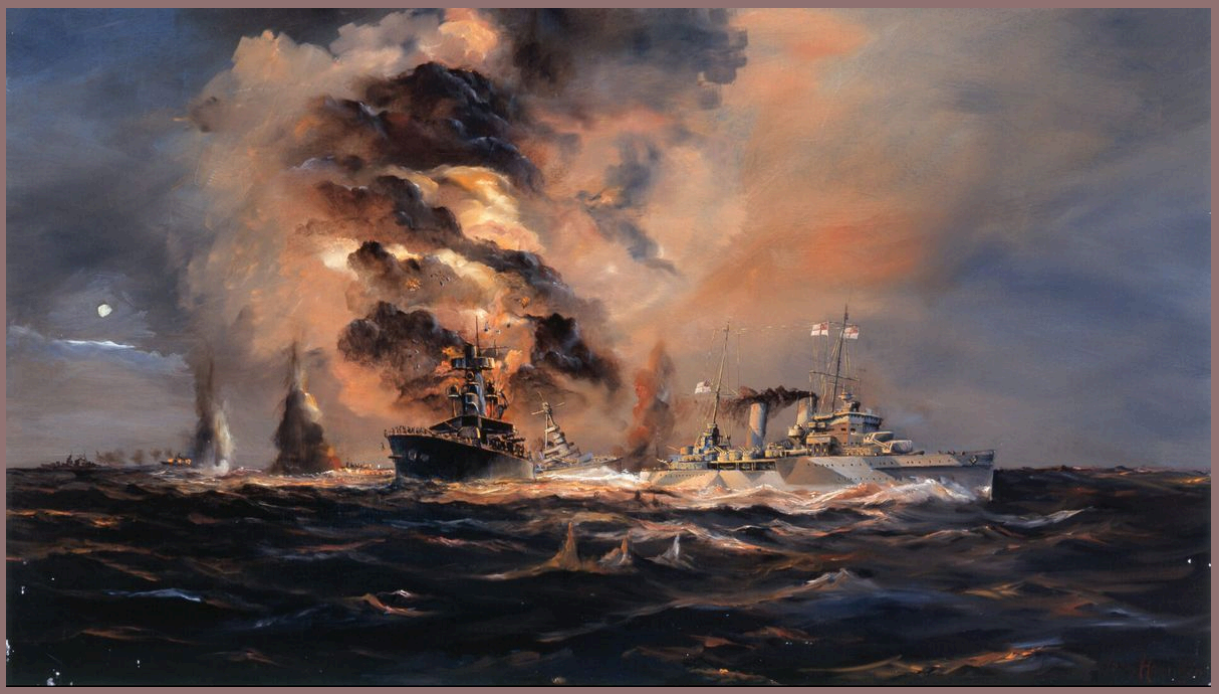
Nicolas Lilou BTS SIO 1

LA BATAILLE NAVALE



Introduction : Nous allons devoir créer le jeu de la bataille navale en respectant plusieurs conditions.

Étape 1: Initialisation des bateaux

Nous allons devoir créer deux tableaux, un pour le joueur et l'autre pour l'ordinateur, que nous devrons tous deux initialiser à 0.

```java
// ÉTAPE 1 : Initialisation des tableaux
int lignes, colonnes;
System.out.println("Veuillez entrer le nombre de lignes que vous désirez avoir dans votre tableau : ");
lignes = sc.nextInt();
System.out.println("Veuillez entrer le nombre de colonnes que vous désirez avoir dans votre tableau : ");
colonnes = sc.nextInt();

int[][] tabJoueur = new int[lignes][colonnes];
int[][] tabOrdi = new int[lignes][colonnes];

// Initialiser toutes les cases à 0
for (int i = 0; i < lignes; i++) {
    for (int j = 0; j < colonnes; j++) {
        tabJoueur[i][j] = 0;
        tabOrdi[i][j] = 0;
    }
}
```

Étape 2: Placement des pions du joueur

Il va falloir faire une boucle jusqu'à ce que 5 bateaux soient placés, établir les limites du tableau, mais aussi empêcher que le joueur place son bateau sur une case déjà occupée.

```java
System.out.println("PLACEMENT DE VOS BATEAUX");
do {
    System.out.println("Sur quelle ligne voulez vous placer votre pion ?");
    int batUserL = sc.nextInt();
    System.out.println("Sur quelle colonne voulez vous placer votre pion ?");
    int batUserC = sc.nextInt();

    if (batUserL >= 1 && batUserL <= lignes && batUserC >= 1 && batUserC <= colonnes) {
        int ligneIndex = batUserL - 1;
        int colonneIndex = batUserC - 1;

        // Vérifier si case déjà prise
        if (tabJoueur[ligneIndex][colonneIndex] == 1) {
            System.out.println("Cette case est déjà occupée par un bateau !");
        } else {
            // Placement du bateau
            tabJoueur[ligneIndex][colonneIndex] = 1;
            bateauxPlaces++;
            System.out.println("Le bateau " + bateauxPlaces + " est bien placé en (" + batUserL + "," + batUserC + ") !");

            // Affichage du plateau
            System.out.println("\nVotre plateau de jeu :");
            affichagetabjoueur(tabJoueur, lignes, colonnes);
        }
    } else {
        System.out.println("Vous êtes hors des limites! Veuillez saisir des valeurs entre 1 et " + lignes + " pour les ligne
    }

} while (bateauxPlaces < nombreBateaux);
```

```
Sur quelle ligne voulez vous placer votre pion ?
2
Sur quelle colonne voulez vous placer votre pion ?
2
Le bateau 2 est bien placé en (2,2) !
```

```
Sur quelle ligne voulez vous placer votre pion ?
5
Sur quelle colonne voulez vous placer votre pion ?
5
Cette case est déjà occupée par un bateau !
```

Étape 3: Placement des pions de l’ordinateur

Il va falloir dans cette étape placer aléatoirement les 5 bateaux pour l'ordinateur. Mais on vérifie d’abord si la case est libre.

```
System.out.println("\nL'ordinateur place ses bateaux...");
do {
    int batOrdiL = (int)(Math.random() * lignes) + 1;
    int batOrdiC = (int)(Math.random() * colonnes) + 1;

    int ligneIndOrdi = batOrdiL - 1;
    int colonneIndOrdi = batOrdiC - 1;

    // Vérifier si case libre
    if (tabOrdi[ligneIndOrdi][colonneIndOrdi] == 0) {
        tabOrdi[ligneIndOrdi][colonneIndOrdi] = 1;
        bateauxPlacesOrdi++;
    }
} while (bateauxPlacesOrdi < nbBatOrdi);

System.out.println("L'ordinateur a placé tous ses " + nbBatOrdi + " bateaux !");
```

Étape 4: Création de la procédure affichageTab

On va afficher un tableau avec tous les bateaux visibles dedans, les cases contenant la le signe “o” seront prises par un bateau et les autres afficheront le signe “~”.

```
// ÉTAPE 4 : Procédure d'affichageTab
static void affichagetabjoueur(int[][] tab, int nbLignes, int nbColonnes) {
    // Afficher les numéros de colonnes
    System.out.print("  ");
    for (int j = 1; j <= nbColonnes; j++) {
        System.out.print(j + " ");
    }
    System.out.println();

    // Afficher le contenu du tableau
    for (int i = 0; i < nbLignes; i++) {
        // Afficher le numéro de ligne
        System.out.print((i + 1) + " ");
        if (i < 9) System.out.print(" ");

        for (int j = 0; j < nbColonnes; j++) {
            if (tab[i][j] == 1) {
                System.out.print("o ");
            } else {
                System.out.print("~ ");
            }
        }
        System.out.println();
    }
}
```

```
Votre plateau de jeu :
  1 2 3 4 5
1 o ~ ~ ~ ~
2 ~ o ~ ~ ~
3 ~ ~ o ~ ~
4 ~ ~ ~ o ~
5 ~ ~ ~ ~ o
```

Étape 5: Création de la procédure affichagetabCache

Nous allons afficher un tableau avec pour signe "?" pour les cases non découvertes (0 et 1),
"o" pour les bateaux touchés (2)
et "x" pour les coups dans l'eau (3)

```
Plateau de l'ordinateur :
  1 2 3 4 5
1  x ? ? ? ?
2  ? ? ? ? ?
3  ? ? ? ? ?
4  ? ? ? ? ?
5  ? ? ? ? ?
```

```java
for (int i = 0; i < nbLignes; i++) {
    // Afficher le numéro de ligne
    System.out.print((i + 1) + " ");
    if (i < 9) System.out.print(" ");

    for (int j = 0; j < nbColonnes; j++) {
        if (tab[i][j] == 0) {
            // aucune action sur cette case → ?
            System.out.print("? ");
        } else if (tab[i][j] == 1) {
            // un pion non découvert → ?
            System.out.print("? ");
        } else if (tab[i][j] == 2) {
            // un pion découvert → o
            System.out.print("o ");
        } else if (tab[i][j] == 3) {
            // case découverte sans pion → x
            System.out.print("x ");
        } else {
            // par défaut
            System.out.print("? ");
        }
    }
    System.out.println();
```

Étape 6: Tour de jeu de l'utilisateur

On demande au joueur de saisir la ligne et la colonne qu'il veut découvrir, on laisse une pause de 2 secondes pour le suspense.
Les trois issues possibles sont raté, touché ou tir à blanc.

```java
// ÉTAPE 6 : Tour du joueur
System.out.println("TOUR DU JOUEUR");
System.out.println("A vous de jouer !");

// 6.1-6.2 Demander les coordonnées
System.out.println("Sur quelle ligne voulez-vous découvrir une case ? (1-" + lignes + ")");
int ligneDecouverte = sc.nextInt();
System.out.println("Sur quelle colonne voulez-vous découvrir une case ? (1-" + colonnes + ")");
int colonneDecouverte = sc.nextInt();

// Vérification des limites
if (ligneDecouverte >= 1 && ligneDecouverte <= lignes &&
    colonneDecouverte >= 1 && colonneDecouverte <= colonnes) {

    int ligneIndex = ligneDecouverte - 1;
    int colonneIndex = colonneDecouverte - 1;

    // 6.3 Simulation du tir
    System.out.println("Tir en cours...");
    try {
        Thread.sleep(2000);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }

    // 6.4 Résultat du tir
    if (tabOrdi[ligneIndex][colonneIndex] == 0) {
        System.out.println("Raté !");
        tabOrdi[ligneIndex][colonneIndex] = 3;
    } else if (tabOrdi[ligneIndex][colonneIndex] == 1) {
        System.out.println("Touché !");
        tabOrdi[ligneIndex][colonneIndex] = 2;
```

Étape 7: Tour de jeu de l’ordinateur

L’ordinateur doit générer aléatoirement des coordonnées non encore jouées, nous ajoutons une pause de 2 secondes pour le suspense et affichons le résultat du jeu de l’ordinateur.

```
TOUR DE L'ORDINATEUR
L'ordinateur tire en (4,2)
Tir en cours...
L'ordinateur a raté !
```

```java
do {
    caseValide = true;
    ligneOrdi = (int)(Math.random() * lignes) + 1;
    colonneOrdi = (int)(Math.random() * colonnes) + 1;

    int ligneIndexOrdi = ligneOrdi - 1;
    int colonneIndexOrdi = colonneOrdi - 1;

    // Vérifier si case déjà découverte
    if (tabJoueur[ligneIndexOrdi][colonneIndexOrdi] == 2 ||
        tabJoueur[ligneIndexOrdi][colonneIndexOrdi] == 3) {
        caseValide = false;
    }
} while (!caseValide);

System.out.println("L'ordinateur tire en (" + ligneOrdi + "," + colonneOrdi + ")");

// 7.4 Simulation du tir
System.out.println("Tir en cours...");
try {
    Thread.sleep(2000);
} catch (InterruptedException e) {
    e.printStackTrace();
}

int ligneIndexOrdi = ligneOrdi - 1;
int colonneIndexOrdi = colonneOrdi - 1;

// 7.5 Résultat du tir
if (tabJoueur[ligneIndexOrdi][colonneIndexOrdi] == 0) {
    System.out.println("L'ordinateur a raté !");
    tabJoueur[ligneIndexOrdi][colonneIndexOrdi] = 3;
} else if (tabJoueur[ligneIndexOrdi][colonneIndexOrdi] == 1) {
    System.out.println("L'ordinateur vous a touché !");
    tabJoueur[ligneIndexOrdi][colonneIndexOrdi] = 2;
```

Étape 8: Fin de la partie

Dans cette étape on va vérifier si le joueur ou l’ordinateur a trouvé tous les bateaux de son adversaire, puis affiche le résultat final et propose une autre partie.

```java
        // ÉTAPE 8.1 : Vérifier fin de partie
        if (nbPionTrouveJoueur >= nbBatOrdi || nbPionTrouveOrdi >= nombreBateaux) {
            partieEnCours = false;
        }
    }

    // ÉTAPE 8.2 & 8.3 : Fin de partie et rejouer
    System.out.println("FIN DE LA PARTIE");
    if (nbPionTrouveJoueur > nbPionTrouveOrdi) {
        System.out.println("VOUS AVEZ GAGNÉ !");
    } else if (nbPionTrouveOrdi > nbPionTrouveJoueur) {
        System.out.println("L'ORDINATEUR A GAGNÉ !");
    } else {
        System.out.println("MATCH NUL !");
    }

    System.out.println("\nVoulez-vous rejouer ? (o/n)");
    rejouer = sc.next();

} while (rejouer.equalsIgnoreCase("o") || rejouer.equalsIgnoreCase("oui"));

System.out.println("Merci d'avoir joué !");
sc.close();
```

```
Votre plateau :
  1 2 3 4 5
1  o ? o ? x
2  ? o ? x x
3  x ? ? x ?
4  ? x ? o ?
5  x ? x ? o
L'ordinateur a gagné !
FIN DE LA PARTIE
L'ORDINATEUR A GAGNÉ !

Voulez-vous rejouer ? (o/n)
```

Conclusion : Ce TP a permis la mise en place de procédure pour l'affichage des tableaux et approfondir les autres notions.