

Fadi
Alouani
BTS SIO 1
groupe 1
B1
03/11/25

B1 TP : Bataille navale

Sommaire

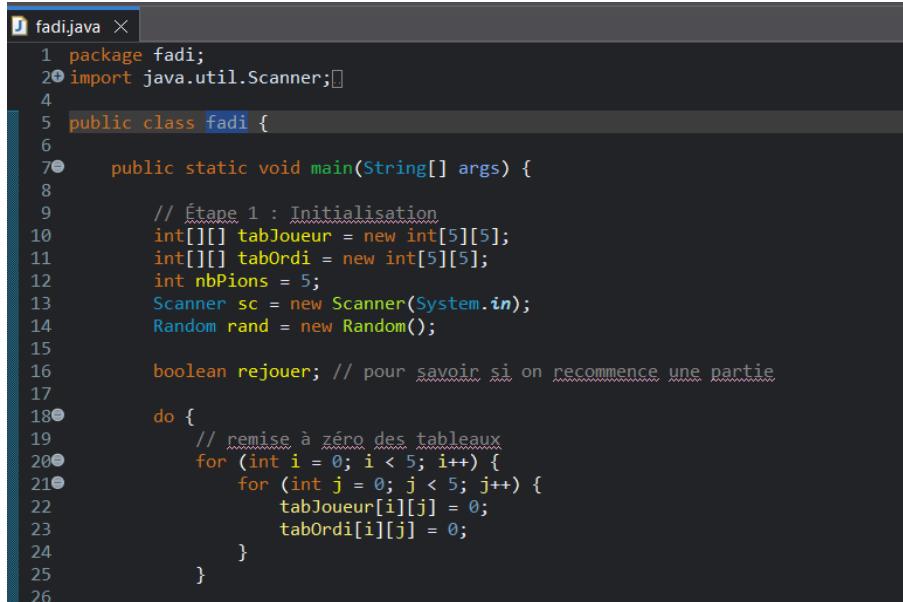
Introduction.....	1
Étape 1.....	2
Étape 2.....	2
Étape 3.....	2
Étape 4.....	3
Étape 5.....	3
Étape 6.....	3
Étape 7.....	4
Étape 8.....	4
Conclusion.....	4

Introduction

Dans ce travail pratique, j'ai développé un jeu de Bataille Navale en utilisant le langage de programmation Java sur l'environnement de développement Eclipse. Ce projet m'a permis de mettre en pratique mes connaissances en programmation orientée objet, tout en améliorant mes compétences en conception et en logique algorithmique. La réalisation de ce jeu m'a également aidé à mieux comprendre la structure d'un programme Java, la gestion des interactions entre les classes et la manipulation des tableaux pour représenter la grille de jeu.

Étape 1

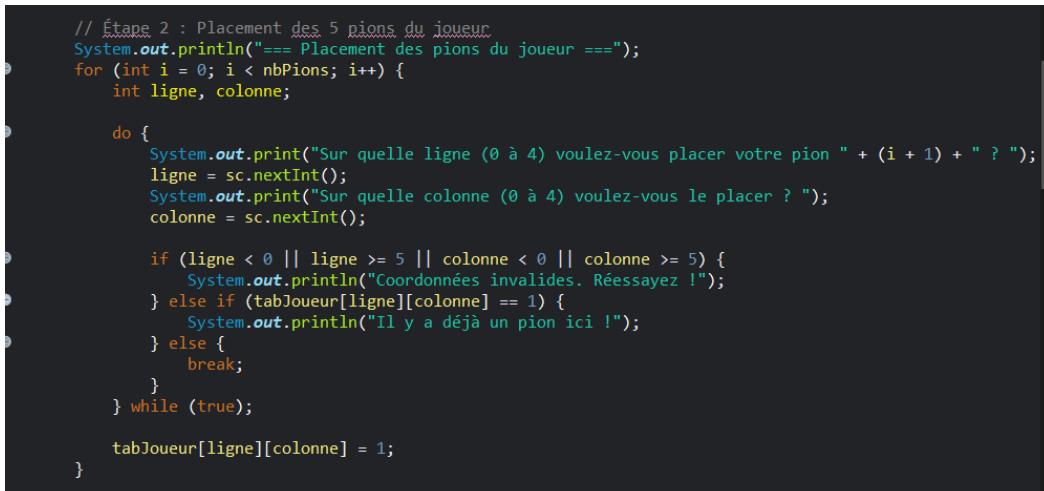
Initialisation des tableaux tabjoueur et tabordi



```
fadi.java ×
1 package fadi;
2 import java.util.Scanner;
3
4 public class fadi {
5
6     public static void main(String[] args) {
7
8         // Étape 1 : Initialisation
9         int[][] tabJoueur = new int[5][5];
10        int[][] tabOrdi = new int[5][5];
11        int nbPions = 5;
12        Scanner sc = new Scanner(System.in);
13        Random rand = new Random();
14
15        boolean rejouer; // pour savoir si on recommence une partie
16
17        do {
18            // remise à zéro des tableaux
19            for (int i = 0; i < 5; i++) {
20                for (int j = 0; j < 5; j++) {
21                    tabJoueur[i][j] = 0;
22                    tabOrdi[i][j] = 0;
23                }
24            }
25        }
26    }
```

Étape 2

Placement des 5 pions tabjoueur



```
// Étape 2 : Placement des 5 pions du joueur
System.out.println("== Placement des pions du joueur ==");
for (int i = 0; i < nbPions; i++) {
    int ligne, colonne;

    do {
        System.out.print("Sur quelle ligne (0 à 4) voulez-vous placer votre pion " + (i + 1) + " ? ");
        ligne = sc.nextInt();
        System.out.print("Sur quelle colonne (0 à 4) voulez-vous le placer ? ");
        colonne = sc.nextInt();

        if (ligne < 0 || ligne >= 5 || colonne < 0 || colonne >= 5) {
            System.out.println("Coordonnées invalides. Réessayez !");
        } else if (tabJoueur[ligne][colonne] == 1) {
            System.out.println("Il y a déjà un pion ici !");
        } else {
            break;
        }
    } while (true);

    tabJoueur[ligne][colonne] = 1;
}
```

Le mot-clé `break` en Java sert à sortir immédiatement d'une boucle (for, while, do...while).

`while(true)` signifie littéralement :

“Répète pour toujours, sans jamais t’arrêter sauf si je dis d’arrêter manuellement.”

C'est ce qu'on appelle une boucle infinie.

Voici ce que j'obtiens sur la console après l'étape deux :

```
==> Placement des pions du joueur ==>
Sur quelle ligne (0 à 4) voulez-vous placer votre pion 1 ? 1
Sur quelle colonne (0 à 4) voulez-vous le placer ? 3
Sur quelle ligne (0 à 4) voulez-vous placer votre pion 2 ? 2
Sur quelle colonne (0 à 4) voulez-vous le placer ? 1
```

Étape 3

Placement des pions 5 tabordi

```
// Étape 3 : Placement aléatoire des 5 pions de l'ordinateur
System.out.println("\nPlacement des pions de l'ordinateur...");
for (int i = 0; i < nbPions; i++) {
    int ligne, colonne;
    do {
        ligne = rand.nextInt(5);
        colonne = rand.nextInt(5);
    } while (tabOrdi[ligne][colonne] == 1);
    tabOrdi[ligne][colonne] = 1;
}
```

La variable rand (Random) sert à donner une valeur aléatoire aux choix de l'ordinateur.

Étape 4

Création de la procédure affichage

```
// Étape 4 : Affichage du plateau du joueur
System.out.println("\nPlateau du joueur :");
afficherGrille(tabJoueur);

int pionsTrouvesJoueur = 0;
int pionsTrouvesOrdi = 0;
```

Étape 5

Création de la procédure affichagetabcache

```
// Étape 5 : créer la procédure affichagetabcache
// Méthode pour afficher une grille
public static void afficherGrille(int[][] grille) {
    for (int i = 0; i < grille.length; i++) {
        for (int j = 0; j < grille[i].length; j++) {
            if (grille[i][j] == 1)
                System.out.print("O ");
            else if (grille[i][j] == -1)
                System.out.print("?");
            else
                System.out.print("x ");
        }
        System.out.println();
    }
}
```

Voici comment la grille s'affiche

Plateau du joueur :

```
x x x x x
x x 0 0 x
x 0 x x x
x x 0 x x
x x x x 0
```

Étape 6

Le joueur cherche un pion

```
// Étape 8 : fin de partie → répéter étapes 6 et 7
while (pionsTrouvesJoueur < nbPions && pionsTrouvesOrdi < nbPions) {

    // Étape 6 : le joueur cherche un pion
    System.out.println("\nA vous de jouer !");
    int tirLigne, tirCol;
    do {
        System.out.print("Entrez une ligne (0-4) : ");
        tirLigne = sc.nextInt();
        System.out.print("Entrez une colonne (0-4) : ");
        tirCol = sc.nextInt();
    } while (tirLigne < 0 || tirLigne > 4 || tirCol < 0 || tirCol > 4);

    if (tabOrdi[tirLigne][tirCol] == 1) {
        System.out.println("Touché !");
        tabOrdi[tirLigne][tirCol] = -1;
        pionsTrouvesJoueur++;
    } else if (tabOrdi[tirLigne][tirCol] == -1) {
        System.out.println("Déjà tiré ici !");
    } else {
        System.out.println("Manqué !");
        tabOrdi[tirLigne][tirCol] = -1;
    }

    if (pionsTrouvesJoueur == nbPions) break; // victoire joueur
}
```

On peut remarquer l'ajout d'instruction en cas de répétition ou d'erreur dans le choix des case on voit également la boucle de l'étape 8 pour la fin de partie.

Étape 7

L'ordinateur cherche un pion

```
// Étape 7 : L'ordinateur cherche un pion
int tirOrdiLigne, tirOrdiCol;
do {
    tirOrdiLigne = rand.nextInt(5);
    tirOrdiCol = rand.nextInt(5);
    while (tabJoueur[tirOrdiLigne][tirOrdiCol] == -1);

    System.out.println("\nL'ordinateur tire sur (" + tirOrdiLigne + ", " + tirOrdiCol + ")");
    if (tabJoueur[tirOrdiLigne][tirOrdiCol] == 1) {
        System.out.println("L'ordinateur a touché un de vos pions !");
        tabJoueur[tirOrdiLigne][tirOrdiCol] = -1;
        pionsTrouvesOrdi++;
    } else {
        System.out.println("L'ordinateur a manqué !");
        tabJoueur[tirOrdiLigne][tirOrdiCol] = -1;
    }

    // Réaffiche ton plateau pour suivre la partie
    System.out.println("\nVotre plateau :");
    afficherGrille(tabJoueur);
}

// Étape 8 : Fin de partie
if (pionsTrouvesJoueur == nbPions) {
    System.out.println("\n Vous avez gagné !");
} else {
    System.out.println("\n L'ordinateur a gagné !");
}
```

Étape 8

Fin de partie

```
// Étape 8 : Fin de partie
if (pionsTrouvesJoueur == nbPions) {
    System.out.println("\n Vous avez gagné !");
} else {
    System.out.println("\n L'ordinateur a gagné !");
}

// Demander si on recommence
System.out.print("\nVoulez-vous refaire une partie ? (o/n) : ");
char rep = sc.nextLine().toLowerCase().charAt(0);
rejouer = (rep == 'o');

} while (rejouer);

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

Voici le résultat des ligne de code sur le déroulement de la partie ,plusieurs cas :

```
L'ordinateur tire sur (2, 1)
L'ordinateur a touché un de vos pions !

Votre plateau :
x x x x x
x x 0 0 x
? ? x x x
x x 0 x x
x x x x 0

A vous de jouer !
Entrez une ligne (0-4) :
```

```
A vous de jouer !
Entrez une ligne (0-4) : 2
Entrez une colonne (0-4) : 2
Manqué !

L'ordinateur tire sur (3, 4)
L'ordinateur a manqué !

Votre plateau :
x x x x x
? x 0 0 x
? ? x x x
? ? 0 ? ?
? ? ? ? ?
```

Pour l'etape 8

```
Votre plateau :
? x ? ? x
? x ? ? ?
? ? ? x ?
? ? ? ? ?
? ? ? ? ?

L'ordinateur a gagné !

Voulez-vous refaire une partie ? (o/n) :
```

Conclusion

La réalisation de ce projet de Bataille Navale en Java m'a permis de consolider mes compétences en programmation, en particulier dans la manipulation des structures de données et l'application des principes de la programmation orientée objet. Ce travail m'a également appris à mieux organiser mon code, à et à utiliser l'environnement Eclipse de manière plus fluide. En développant ce jeu, j'ai non seulement renforcé ma logique de programmation, mais aussi acquis une expérience concrète dans la conception d'un programme interactif. Ce TP a donc été une étape formatrice et motivante dans mon apprentissage du développement en Java.