
TP BATAILLE NAVALE

Table des matières

INTRODUCTION	1
Etape 1 : Initialisation des Tableaux.	2
Etape 2 Placement des 5 Pions (Tabjoueur).....	2
Etape 3 : On place les 5 Pions de l'ordi.	3
Etape 4 Afficher les placements des pions sous forme d'un Plateau.....	4
Etape 5 création de la procédure affichage tab Cache.	4
ETAPE 6 :	5
ETAPE 7 :	6
ETAPE 8 :	6
ETAPE 9 :	7
Conclusion :	8

INTRODUCTION : Dans ce TP nous allons apprendre comment faire un jeu de bataille en posant des Pions a des cases on a déjà vu ca dans le TP SHIFUMI c'est un peu le même concept nous allons procéder par des étapes et nous allons voir ca ensemble comment on doit faire ce genre de TP.

Etape 1 : Initialisation des Tableaux.

- On doit faire 2 tableaux vu qu'on vas jouer contre l'ordinateur un tableau pour l'utilisateur et un autre pour l'ordi, et on vas attribuer les valeurs, dans ce cas on va mettre 5 lignes et 5 colonnes pour les deux joueurs.

```
public class TP_Bataille {  
  
    public static void main(String[] args) {  
        int[][] tabjoueur = new int[5][5];  
        int[][] tabordi = new int[5][5];  
    }  
}
```

- Une fois on a attribué les tableaux on va procéder aux prochaines étapes.

Etape 2 Placement des 5 Pions (Tabjoueur)

- Dans cet étape on vas demander aux joueurs ou ils souhaitent placer leurs Pions entre les 5 lignes et les 5 colonnes, pour ca on vas tout simplement faire un `system.out.print("")` et mettre ce qu'on veut demander, dans mon cas je vais demander où il souhaite placer le pion sur quelle ligne et quelle colonne.

```
if (pions < nbPions) {  
    System.out.print("Entrez la ligne (0 à 4) pour le pion " + (pions + 1) + " : ");  
    int ligne = sc.nextInt();  
  
    System.out.print("Entrez la colonne (0 à 4) pour le pion " + (pions + 1) + " : ");  
    int colonne = sc.nextInt();  
}
```

- Il ne doit pas placer un pion dans un case déjà prise ou dans un endroit hors de case par exemple sur la ligne 7 et on doit demander à l'utilisateur de saisir et placer 5 pions donc pour ça on doit faire une boucle pour demander à l'utilisateur jusqu'au `Nbpions == Pions placer`.

```
// Placement des pions du joueur
while (pions < nbPions) {
    System.out.print("Entrez la ligne (0 à 4) pour le pion " + (pions + 1) + " : ");
    int ligne = sc.nextInt();

    System.out.print("Entrez la colonne (0 à 4) pour le pion " + (pions + 1) + " : ");
    int colonne = sc.nextInt();

    if (ligne >= 0 && ligne < 5 && colonne >= 0 && colonne < 5) {
        if (tabjoueur[ligne][colonne] == 0) {
            tabjoueur[ligne][colonne] = 1;
            pions++;
            System.out.println(" Pion placé !");
        } else {
            System.out.println(" Cette case est déjà occupée !");
        }
    } else {
        System.out.println(" Position invalide !");
    }
}
```

- Après avoir demander a l'utilisateur de placer les Pions on va passer pour les choix de l'ordi pour cela on doit faire des choix aléatoires.

Etape 3 : On place les 5 Pions de l'ordi.

- Pour cette étape on va faire des choix aléatoires et pour faire des choix aléatoires on va faire un import.java. util. Random ;

```
import java.util.Scanner;
import java.util.Random;
```

- Maintenant on va attribuer des lignes et des colonnes aléatoires pour le choix de l'ordinateur entre 0 et 4 et pour cela on va écrire le code en bas.

```
// Placement des pions de l'ordinateur
int pionsOrdi = 0;

while (pionsOrdi < nbPions) {
    int ligneOrdi = rand.nextInt(5);
    int colonneOrdi = rand.nextInt(5);

    if (tabordi[ligneOrdi][colonneOrdi] == 0) {
        tabordi[ligneOrdi][colonneOrdi] = 1;
        pionsOrdi++;
    }
}

System.out.println("L'ordinateur a placé ses pions !");
sc.close();
}
```

- Int pions ordi = 0, est le nombre de pions placer au départ vu que l'ordi na pas encore placer les pions donc c'est 0.
- Pareil pour les choix ordi on doit lui faire placer les 5 pions donc on vas faire une boucle while, temps que pions ordi < a nb pions qui est 5 le boucle demanderas ou choisiras une ligne et une colonne aléatoire jusqu'au nb pions == pions ordi.

Etape 4 Afficher les placements des pions sous forme d'un Plateau.

- Une fois qu'on a enregistré les choix des placements des pions de joueur et les choix aléatoire de l'ordinateur on va les afficher sous forme d'un Plateau pour cela il faudrait faire.

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

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

Etape 5 création de la procédure affichage tab Cache.

- Pour avoir une grille il faudrait écrire ce code pour un résultat.

```

Étape 5 : créer la procédure affichageetabcache
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 "); // pion
            else if (grille[i][j] == -1)
                System.out.print("? "); // tir déjà effectué
            else
                System.out.print("x "); // vide
        }
        System.out.println();
    }
}

```

- Le résultat.

```

Plateau du joueur :
x x x x x
x x O O x
x O x x x
x x O x x
x x x x O

```

ETAPE 6 :

Dans cet étape le joueur doit choisir un placement de pion et pour cela il faut avoir ce code.

```

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
}

```

ETAPE 7 :

Dans cette étape ça sera à l'ordinateur de choisir et chercher un endroit de placement de 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 !");
}
}

```

ETAPE 8 :

Après ça on aura un résultat pour savoir qui a gagné et qui a perdu.

```
// 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.next().toLowerCase().charAt(0);
rejouer = (rep == 'o');

} while (rejouer);

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

ETAPE 9:

Différent cas et différent résultat des lignes de code.

```
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
```

Conclusion :

Ce projet de création d'un jeu de Bataille Navale en Java m'a permis de mieux comprendre comment construire une application étape par étape. J'ai pu manipuler des tableaux, gérer des interactions entre l'utilisateur et le programme, et appliquer concrètement les notions vues en cours. La mise en place de la logique du jeu, des placements aléatoires et des vérifications de cases m'a également aidé à progresser en raisonnement algorithmique.

Finalement, ce TP a été une bonne occasion de pratiquer Java de manière ludique tout en renforçant ma méthode de travail et ma maîtrise de l'environnement Eclipse.