

TP08: Bataille Navale

Sommaire:

I.	Introduction.....	1
II.	Etape 1: Initialisation des variables	2
III.	Etape 2: Placement des bateaux du joueur.....	2
IV.	Etape 3: Placement des bateaux de l'ordi.....	3
V.	Etape 4:La boucle du jeu.....	3
VI.	Etape 5: Gestion du tour de l'ordinateur.....	4
VII.	Conclusion.....	4

I. Introduction:

L'objectif de ce TP est de développer une version console du jeu de la Bataille Navale. Pour ce projet, j'ai réalisé une version simplifiée sur une grille de 5 cases par 5, où le premier joueur qui coule les 5 bateaux de l'adversaire gagne la partie.

II. Etape 1 : Initialisation des variables

J'ai commencé par créer une grille. Pour celà j'ai utilisé un tableau à double entrée qui vont servir de terrain de jeu:

```
// --- 1. INITIALISATION ---
System.out.println("--- BATAILLE NAVALE 5x5 ---");

// Création des grilles 5x5 (remplies de 0 par défaut)
int[][] tabJoueur = new int[5][5];
int[][] tabOrdi   = new int[5][5];
```

III. Etape 2: Placement des bateaux du joueur

J'ai créé une boucle qui demande de choisir une ligne puis une colonne puis grâce à un scanner je récupère l'emplacement et le programme vérifie si la case n'est pas déjà utilisée:

```
// --- 2. LE JOUEUR PLACE SES BATEAUX ---
System.out.println("Placez vos 5 bateaux !");
int bateauxPlaces = 0;

while (bateauxPlaces < 5) {
    afficherGrilleJoueur(tabJoueur); // On montre ta grille

    System.out.print("Bateau " + (bateauxPlaces + 1) + " -> Ligne (1-5) : ");
    int lig = sc.nextInt() - 1;
    System.out.print("Bateau " + (bateauxPlaces + 1) + " -> Colonne (1-5) : ");
    int col = sc.nextInt() - 1;

    // Vérification : est-ce dans la grille ? Est-ce libre (0) ?
    if (lig >= 0 && lig < 5 && col >= 0 && col < 5 && tabJoueur[lig][col] == 0) {
        tabJoueur[lig][col] = 1; // 1 = Bateau
        bateauxPlaces++;
    } else {
        System.out.println("Erreur : Case invalide ou déjà prise.");
    }
}
```

IV. Etape 3: Placement des bateaux de l'ordi

L'ordinateur place aléatoirement ses bateaux sans intervention du joueur:

```
// --- 3. L'ORDINATEUR PLACE SES BATEAUX ---
System.out.println("L'ordinateur place ses bateaux...");  
int bateauxOrdi = 0;  
while (bateauxOrdi < 5) {  
    int lig = (int)(Math.random() * 5);  
    int col = (int)(Math.random() * 5);  
  
    if (tabOrdi[lig][col] == 0) {  
        tabOrdi[lig][col] = 1;  
        bateauxOrdi++;  
    }  
}
```

V. Etape 4: La boucle du jeu

Cette partie est le coeur du programme, elle permet de gérer les tirs jusqu'à la victoire d'un des deux côtés:

```
// --- 4. LA BATAILLE COMMENCE ---
int scoreJoueur = 0;
int scoreOrdi = 0;  
  
// La partie continue tant que personne n'a 5 points
while (scoreJoueur < 5 && scoreOrdi < 5) {  
  
    // --- TOUR DU JOUEUR ---
    System.out.println("\n--- A VOUS ! ---");
    afficherGrilleCachee(tabOrdi); // On montre la grille ennemie masquée  
  

    System.out.print("Tir Ligne : ");
    int lig = sc.nextInt() - 1;
    System.out.print("Tir Colonne : ");
    int col = sc.nextInt() - 1;  
  

    if (lig >= 0 && lig < 5 && col >= 0 && col < 5) {
        if (tabOrdi[lig][col] == 1) {
            System.out.println(">>> TOUCHÉ ! <<<");
            tabOrdi[lig][col] = 2; // 2 = Touché
            scoreJoueur++;
        } else if (tabOrdi[lig][col] == 0) {
            System.out.println(">>> RATÉ ! <<<");
            tabOrdi[lig][col] = 3; // 3 = Plouf
        } else {
            System.out.println("Déjà visé...");
        }
    } else {
        System.out.println("Tir hors zone !");
    }  
  

    // Vérification victoire immédiate
    if (scoreJoueur == 5) break;
}
```

VI. Etape 5: Gestion du tour de l'ordinateur

L'ordinateur doit tirer à son tour mais cherche une case qu'il n'a pas déjà choisi au paravent:

```
// --- TOUR DE L'ORDINATEUR ---
System.out.println("\n--- TOUR ORDI ---");
int ligOrdi, colOrdi;
// L'ordi cherche une case valide (pas déjà touchée ou ratée)
do {
    ligOrdi = (int)(Math.random() * 5);
    colOrdi = (int)(Math.random() * 5);
} while (tabJoueur[ligOrdi][colOrdi] == 2 || tabJoueur[ligOrdi][colOrdi] == 3);

if (tabJoueur[ligOrdi][colOrdi] == 1) {
    System.out.println("L'ordi vous a TOUCHÉ en " + (ligOrdi+1) + "," + (colOrdi+1));
    tabJoueur[ligOrdi][colOrdi] = 2;
    scoreOrdi++;
} else {
    System.out.println("L'ordi a RATÉ en " + (ligOrdi+1) + "," + (colOrdi+1));
    tabJoueur[ligOrdi][colOrdi] = 3;
}

// On affiche l'état du joueur
System.out.println("Votre État :");
afficherGrilleJoueur(tabJoueur);
}
```

VII. Conclusion

Ce projet m'a permis de comprendre la manipulation des tableaux à double entrées et la logique algorithmique d'un jeu tour par tour. Il n'a pas été très compliqué mais la difficulté était dans l'application des tableaux