

**B1 TP Compte Rendu
TP BATAILLE NAVALE**

Sommaire :

Analyse du problème

Conception de la solution

Réalisation

Test

Retour expérience

Quelques remarques

Analyse du problème

Nous souhaitons réaliser un programme JAVA proposant un jeu de la bataille navale à partir d'un tableau. Ce code contiendra 2 tableaux stockant les positions des pions du joueur et de l'ordinateur .

Conception de la solution

Le jeu de la Bastille navale contient 6 étapes: la première initié 2 tableaux entiers de 5 indices pour le joueur et l'ordinateur, puis étape 2 on sélectionne 5 pions pour chaque tableaux puis nous vérifions que nos pions se situent sur différentes cases. A l'étape 3 on place les pions de l'ordinateur avec un fonction aléatoire.Pour l'étape 4 et 5 on affiche des tableaux du jeu avec lignes/colonnes en affichant des symboles selon différents cas (pion touché, case vide, pion non découvert...). Pour la dernière étape on configure le tir du joueur sur l'ordinateur.

Réalisation

Nous utilisons la boucle while pour sélectionner les pions du joueur de l'utilisateur et de l'ordinateur tout en utilisant la fonction random pour le choix de l'ordinateur. Pour le parcours des lignes/colonnes nous utilisons des boucles for , puis des boucles if / elif / else dans les nos boucles pour vérifier les conditions. Nous avons utiliser également la condition case pour définir des cas pour l'affichage des états du tableau à l'étape 5.

Test

Voici notre code :

```
//EXERCICE
//ECRIRE UN PROGRAMME PERMETTANT D'AFFICHER UN JEU DE BATAILLE NAVALE
// @author : Côme
// version 1.0
import java.util.Scanner;
public class TP7 {

    static void affichagetabJoueur(int tab[][], int nbcase) {
        System.out.println(" 0 1 2 3 4"); //Affiche un espace avant la position des lignes
        int l = 0; // Déclare une variable l et la met à 0
        while (l < nbcase) { // Tant que l est plus petit que nbcase on répète l'insstruction
            System.out.print((l) + " "); //Afficher la ligne plus un espace
            int l2 = 0; // Déclare une variable l et la met à 0

            while (l2 < nbcase) { // Tant que l est plus petit que nbcase on répète l'insstruction
                if (tab[l][l2] == 1) {
                    System.out.print("o ");
                } else {
                    System.out.print("~ ");
                }
                l2=l2+1; // Incrémente l2 de 1 à chaque tour
            }
            System.out.println(); // retour à la ligne après chaque rangée
            l=l+1; // Incrémente l de 1 à chaque tour
        }
    }

    static void affichageAvecEtats(int tab[][], int nbcase) {
        System.out.println(" 0 1 2 3 4"); // Affiche un espace avant la position des lignes + les numeros de lignes
        int l = 0; // Déclare une variable l et la met à 0
        while (l < nbcase) { // Tant que l est plus petit que nbcase on répète l'insstruction
            System.out.print((l) + " "); // Affiche le numéro de la ligne
            int l2 = 0; // Déclare une variable l et la met à 0

            while (l2 < nbcase) { // Tant que l est plus petit que nbcase on répète l'insstruction
                switch (tab[l][l2]) { // compare les valeurs du tableau
                    case 0: // Cas avec aucun pion sur la case
                    case 1: // Cas avec une case non découvert
                        System.out.print("?");
                    break; // sort de la boucle
                    case 2: // Cas avec un pion découvert
                }
                l2=l2+1;
            }
            l=l+1;
        }
    }
}
```

```

        System.out.print("o ");
        break;                                // sort de la boucle
    case 3:                                // Cas avec une case découverte sans pion
        System.out.print("x ");
        break;                                // sort de la boucle
    default:
        System.out.print("~"); // Cas par défaut
    }
    i2=i2+1;      // Incrémente i de 1 à chaque tour
}
System.out.println(); // Retour à la ligne
i=i+1; // Incrémente i de 1 à chaque tour
}
}

public static void main(String[] args) {
    try (Scanner sc = new Scanner(System.in)) {
        int tabjoueur[][] = new int [5][5];
        int tabordi[][] = new int [5][5];

        int i = 1; //initialisation de i

        while(i<=5){ // boucle pour i de 1 à 5
            System.out.println("Sur quelle ligne voulez vous placez votre pion (0-4)");
            int ligne = sc.nextInt();
            System.out.println("Sur quelle colonne voulez vous placez votre pion (0-4)");
            int colonne = sc.nextInt();

            // Vérification si la case est déjà occupée
            if (tabjoueur[ligne][colonne] != 0) {
                System.out.println("Cette case est déjà occupée! Choisissez en une autre");
                i=i-1;          //redemander l'itération
            }
            tabjoueur[ligne][colonne] = 1;
            System.out.println("Vous avez placé votre pion n°" + i + " à la position " + ligne + ", " + colonne );
            ++i; // Ajouter +1 à i pour recommencer la boucle
        }

        affichagetabJoueur(tabjoueur, 5);      //affichage tableau joueur
    }
}

int i2 = 1; //initialisation de i

while(i2<=5){ // boucle pour i de 1 à 5
int ligneordi = (int)(Math.random()*5);
int colonneordi = (int)(Math.random()*5);

// Vérification si la case est déjà occupée
if (tabordi[ligneordi][colonneordi] != 0) {

```

```

i2=i2-1;           //redemander l'itération
}
tabordi[[ligneordi][colonneordi] = 1;
System.out.println("L'ordinateur a placé son pion n°" + i2 + " à la position " + ligneordi + ", " +
colonneordi ); // a enlever apes test
++i2; // Ajouter +1 à i pour recommencer la boucle
}

//Répétition Étape 6 & 7
int pionTrouve = 0; //compteur pion trouvé joueur
int pionTrouveOrdi = 0; //compteur pion trouvé ordi

while(pionTrouve < 5 || pionTrouveOrdi < 5) {

    System.out.println(" A vous de jouer !");
    System.out.println("Sur quelle ligne voulez-vous découvrir une case (0-4) ?");
    int ligneTir = sc.nextInt();
    System.out.println("Sur quelle colonne voulez-vous découvrir une case (0-4) ?");
    int colonneTir = sc.nextInt();

    System.out.println("Tir en cours");
    try {
//solutions eclipse
        Thread.sleep(2000);
    } catch (InterruptedException e) { // lancer l'attente de 2 sec
malgres les problemes
        e.printStackTrace(); //solution eclipse
    }
    // pause de 2 secondes

    // Vérification du résultat du tir
    if(tabjoueur[ligneTir][colonneTir] == 0) {
        System.out.println("Raté !");
        tabordi[ligneTir][colonneTir] = 3; // case découverte sans pion -> x
    } else if(tabordi[ligneTir][colonneTir] == 1) {
        System.out.println("Touché !" + pionTrouve);
        tabordi[ligneTir][colonneTir] = 2; // pion découvert -> o
        pionTrouve++;
        // compteur de pion trouvée = +1
    } else {
        // pion non découvert + aucune action sur la case -> ?
        System.out.println("Tir à blanc !"); // affiche tir a blanc pour les autres alternatives
    }

    // Affichage du plateau ordinateur après le tir
    System.out.println("Tableau de l'ordinateur après le tir :");
}

```

```

affichageAvecEtats(tabordi, 5);

System.out.println("Tour de l'ordinateur");
// compteur

de pions touchés pour l'ordinateur
int ligneTirOrdi = (int)(Math.random()*5); //ligne pion
int colonneTirOrdi = (int)(Math.random()*5); //colonne pion

// case découverte joueur = plus dispo dans random
// case = 2 ou 3 alors recommencer choix
System.out.println("Tir en cours");
try { //solutions
eclipse
    Thread.sleep(2000);
} catch (InterruptedException e) { // lancer l'attente de 2 sec malgré
les problèmes
    e.printStackTrace(); //solution eclipse
}
// pause de 2 secondes

// Vérification du résultat du tir Ordi
if(tabordi[ligneTirOrdi][colonneTirOrdi] == 0) {
    System.out.println("Raté !");
    tabjoueur[ligneTirOrdi][colonneTirOrdi] = 3;
// case découverte sans pion -> x
} else if(tabjoueur[ligneTirOrdi][colonneTirOrdi] == 1) {
    System.out.println("Touché !"+ pionTrouveOrdi);
    tabjoueur[ligneTirOrdi][colonneTirOrdi] = 2;
// pion découvert -> o
    pionTrouveOrdi++;
// compteur de pion trouvée = +1
} else {
// pion non découvert + aucune action sur la case -> ?
    System.out.println("Tir à blanc !");
}
// affiche tir à blanc pour les autres alternatives
}

// Affichage du plateau ordinateur après le tir
System.out.println("Tableau du joueur après le tir :");
affichageAvecEtats(tabjoueur, 5);

if(pionTrouve == 5) {
    System.out.print("Le Vainqueur est le joueur !");
}
if(pionTrouveOrdi == 5) {
    System.out.print("Le Vainqueur est l'ordinateur !");
}
System.out.print("Souhaitez-vous refaire une partie? O/N ");
int réponse = sc.nextInt();

}

```

```
}
```

```
}
```

```
}
```

Retour expérience

Ce TP était instructif et assez technique reflétant tout ce que nous avions déjà abordé.

Quelques remarques

Sans remarques