

Groupe 1

Bloc 1 TP 7

Étape 1	2
Étape 2	2
Étape 3	3
Étape 4	4
Étape 5	5
Étape 6	6
Étape 7	7
Étape 8	8
Étape 9	8
Conclusion	8

Étape 1

Le programme commence par utiliser une boucle “**for**” pour passer sur chaque ligne des deux tableaux de jeu, “**tabJoueur**” et “**tabOrdi**”, et utilise “**Arrays.fill()**” pour remplir chacune de ces lignes avec des “**0**” afin de garantir que les grilles sont vides.

```
//Etape 1
for (int[] row : tabJoueur) {
    Arrays.fill(row, val: 0);
}

for (int[] row : tabOrdi) {
    Arrays.fill(row, val: 0);
```

Étape 2

Une boucle “**while (*i* < 5)**” force le joueur à placer 5 pions en utilisant d'autres boucles “**while**” pour valider que les coordonnées entrées (1-5) sont correctes, puis un “**if**” vérifie que la case n'est pas déjà prise “**(== 1)**”, sinon “**(else)**” le pion est placé et le compteur “**i**” augmente.

```
// Etape 2
while (i < 5) {
    valueLigneJ = 0;
    valueColonneJ = 0;

    while (valueLigneJ < 1 || valueLigneJ > 5) {
        System.out.println("[" + (i + 1) + "] Sur quel ligne souhaitez vous placer votre pion ? (1-5)");
        valueLigneJ = sc.nextInt();

        if (valueLigneJ < 1 || valueLigneJ > 5)
            System.out.println("Valeur incorrecte");
    }

    while (valueColonneJ < 1 || valueColonneJ > 5) {
        System.out.println("[" + (i + 1) + "] Sur quel colonne souhaitez vous placer votre pion ? (1-5)");
        valueColonneJ = sc.nextInt();
        [1] sur quel ligne souhaitez vous placer votre pion ? (1-5)
        2
        [1] sur quel colonne souhaitez vous placer votre pion ? (1-5)
        6
        Valeur incorrecte
        [1] sur quel colonne souhaitez vous placer votre pion ? (1-5)
        4
        [2] sur quel ligne souhaitez vous placer votre pion ? (1-5)

        if (tabJoueur[valueLigneJ - 1][valueColonneJ - 1] == 1) {
            System.out.println("[" + (i + 1) + "] Vous avez déjà placer un pion a cette endroit.");
        } else {
            tabJoueur[valueLigneJ - 1][valueColonneJ - 1] = 1;
            i++;
        }
    }
}
```

Étape 3

L'ordinateur utilise la même boucle “**while (*i* < 5)**” pour placer ses 5 pions, mais il génère des coordonnées aléatoires et utilise une condition “**if**” pour vérifier si la case est libre, si elle est prise, la boucle recommence sans incrémenter “*i*”, forçant un nouveau tirage.

```
//Etape 3
while (_i < 5) {
    valueLigne0 = (int)(Math.random() * 5) + 1;
    valueColonne0 = (int)(Math.random() * 5) + 1;

    if (tabOrdi[valueLigne0 - 1][valueColonne0 - 1] == 1) {
        // return

    } else {
        tabOrdi[valueLigne0 - 1][valueColonne0 - 1] = 1;
        _i++;
    }
}
System.out.println("\nL'ordinateur a placé ses pions.");
```

Étape 4

Cette procédure d'affichage "visible" parcourt le tableau donné avec deux boucles "**while**" imbriquées et utilise un "**if**" pour afficher "**o**" si la case vaut "**1**" (pion) ou "**2**" (touché), sinon elle affiche "**~**" pour l'eau.

```
//Etape 4
static void affichageTabJ (int tabJoueur[][], int nbcase) {

    System.out.println("    1 2 3 4 5");

    int i = 0;
    while (i < nbcase) {
        System.out.print((i + 1) + "  ");

        int j = 0;
        while (j < nbcase) {

            if (tabJoueur[i][j] == 1 || tabJoueur[i][j] == 2) {
                System.out.print("o ");
            } else {
                System.out.print("~ ");
            }
            j++;
        }
        System.out.println();
        i++;
    }
    // Etape 4 bis
    nbcase = tabJoueur.length;
    System.out.println("\nVotre grille :");
    affichageTabJ(tabJoueur, nbcase);
    //
    i = 0;
```

Étape 5

Cette procédure d'affichage "cachée" parcourt également le tableau, mais utilise une instruction “**switch**” pour afficher “?” si la case est “0” ou “1” (inconnue), “o” si la case est “2” (touché), et “x” si la case est “3” (raté).

```
//Etape 5
static void affichageTabBataille (int tab[][], int nbcase) {

    System.out.println("    1 2 3 4 5");

    int i = 0;
    while (i < nbcase) {
        System.out.print((i + 1) + " ");

        int j = 0;
        while (j < nbcase) {
            switch (tab[i][j]) {
                case 0: // VIDE
                case 1: // PION
                    System.out.print("? ");
                    break;
                case 2: // TOUCHE
                    System.out.print("o ");
                    break;
                case 3: // RATE
                    System.out.print("x ");
                    break;
            }
            j++;
        }
        System.out.println();
        i++;
    }
}
```

Étape 6

Le tour du joueur affiche d'abord la grille ennemie (Etape 5), puis utilise une boucle “**while(true)**” pour forcer une saisie valide en vérifiant que la case visée n'est ni “**2**” ni “**3**”, avant de finalement analyser le tir avec un “**if**” pour changer la case à “**2**” (Touché) ou “**3**” (Raté) et augmenter le score si besoin.

```
// Etape 8 (Boucle de jeu)
while (nbPionTrouverJoueur < 5 && nbPionTrouveOrdi < 5) {

    // Etape 6
    System.out.println("\nA vous de jouer !");
    affichageTabBataille(tabOrdi, nbcase);

    while (true) {
        choixL = 0;
        choixC = 0;

        while (choixL < 1 || choixL > 5) {
            System.out.println("Sur quel ligne souhaitez vous tirer ? (1-5)");
            choixL = sc.nextInt();

            if (choixL < 1 || choixL > 5)
                System.out.println("Zone incorrecte");
        }

        while (choixC < 1 || choixC > 5) {
            System.out.println("Sur quel colonne souhaitez vous tirer ? (1-5)");
            choixC = sc.nextInt();

            if (choixC < 1 || choixC > 5)
                System.out.println("Zone incorrecte");
        }

        if (tabOrdi[choixL - 1][choixC - 1] == 2 || tabOrdi[choixL - 1][choixC - 1] == 3) {
            System.out.println("Vous avez déjà tiré sur cette case. Réessayez.");
        } else {
            break;
        }
    }

    System.out.println("Tir en cours...");
    Thread.sleep( millis: 2 * 1000 );

    if (tabOrdi[(choixL - 1)][(choixC - 1)] == 1) {
        tabOrdi[(choixL - 1)][(choixC - 1)] = 2; // TOUCHÉ
        nbPionTrouverJoueur++;
        System.out.println("Touché ! (" + nbPionTrouverJoueur + "/5)");
    }
    else { // C'était 0 (VIDE)
        tabOrdi[(choixL - 1)][(choixC - 1)] = 3; // RATE
        System.out.println("Raté !");
    }

    affichageTabBataille(tabOrdi, nbcase); // Ré-affiche la grille

    if (nbPionTrouverJoueur == 5) {
        break;
    }

    tabOrdi[(choixL - 1)][(choixC - 1)] = 3; // RATE
    System.out.println("Raté !");
}

affichageTabBataille(tabOrdi, nbcase); // Ré-affiche la grille

if (nbPionTrouverJoueur == 5) {
    break;
}
```

Étape 7

Le tour de l'ordinateur utilise une boucle “**while(true)**” pour générer des coordonnées aléatoires jusqu'à ce qu'il en trouve une qui soit valide (une case “**0**” ou “**1**” sur la grille du joueur), puis il met à jour cette case à “**2**” (Touché) ou “**3**” (Raté) et affiche la grille du joueur mise à jour.

```
// Etape 7
System.out.println("\nAu tour de l'ordinateur...");
Thread.sleep( millis: 2 * 1000);

while (true) {
    choixL0rdi = (int)(Math.random() * 5); // 0-4
    choixC0rdi = (int)(Math.random() * 5); // 0-4

    if (tabJoueur[choixL0rdi][choixC0rdi] == 0 || tabJoueur[choixL0rdi][choixC0rdi] == 1) {
        break;
    }
}

System.out.println("L'ordinateur tire en [" + (choixL0rdi + 1) + "," + (choixC0rdi + 1) + "]...");
Thread.sleep( millis: 2 * 1000);

if (tabJoueur[choixL0rdi][choixC0rdi] == 1) {
    tabJoueur[choixL0rdi][choixC0rdi] = 2; // TOUCHE
    nbPionTrouve0rdi++;
    System.out.println("L'ordinateur a touché un de vos pions ! (" + nbPionTrouve0rdi + "/5)");
} else {
    tabJoueur[choixL0rdi][choixC0rdi] = 3; // RATE
    System.out.println("L'ordinateur a raté !");
}

System.out.println("\nVotre grille :");
affichageTabJ(tabJoueur, nbcase);

}
```

Étape 8

Le cœur du jeu est une grande boucle “**while**” qui continue de s'exécuter tant que le score du joueur ET le score de l'ordinateur sont tous les deux inférieurs à 5, faisant s'enchaîner l'Étape 6 (tour joueur) et l'Étape 7 (tour ordinateur).

```
// Etape 8 (Boucle de jeu)
while (nbPionTrouverJoueur < 5 && nbPionTrouveOrdi < 5) {
```

Étape 9

Une fois la boucle principale terminée, un simple bloc “**if/else**” vérifie si c'est le “**nbPionTrouverJoueur**” qui a atteint 5 pour déclarer le joueur vainqueur, sinon il déclare l'ordinateur vainqueur et affiche sa grille en mode visible.

```
// Etape 9
System.out.println("\nPARTIE TERMINEE");
if (nbPionTrouverJoueur == 5) {
    System.out.println("Vous avez gagné !");
} else {
    System.out.println("L'ordinateur a gagné.");
    System.out.println("Voici où étaient ses pions :");
    affichageTabJ(tabOrdi, nbcase);
}

sc.close();
```

Conclusion

Approfondissement de la gestion des tableaux à deux dimensions pour stocker des états (touché, raté, etc.) et de la création de procédures pour séparer l'affichage (ça rend le code principal plus lisible).