

**Xavier Sionnet**

02/10/2025

BTS SIO

## **TP 4 : Shifumi**

---

-Shifumi

-Etape

-Conclusion

## 1) Problématique

L'objectif de ce TP est de mettre en pratique les bases de la programmation acquises durant l'année en réalisant un petit jeu interactif.

Pour cela, nous allons suivre différentes étapes qui nous permettront d'apprendre à gérer les entrées clavier de l'utilisateur, d'utiliser des boucles et conditions, ainsi que d'intégrer des fonctions générant de l'aléatoire.

## 2) Étapes

### Étape 1 :

La première chose à faire est de mettre en place un système de score, afin de définir à combien de points la partie s'arrête.

On crée donc une variable `nbpoints` initialisée à 0.

Un morceau de code est ensuite ajouté pour que le joueur choisisse le nombre de manches (3, 5 ou 10). Si sa saisie est incorrecte, la question lui est reposée.

Enfin, on affiche le choix retenu pour confirmation.

```
// Étape 1 : Définir le nombre de points

do {
    System.out.println("À combien de points souhaitez-vous jouer ? (3, 5 ou 10)");
    nbpoints = sc.nextInt();
    System.out.println("Nombre de points choisi : " + nbpoints);
} while (nbpoints != 3 && nbpoints != 5 && nbpoints != 10);
```

### Étape 2 :

On demande ensuite au joueur de sélectionner entre pierre, feuille ou ciseaux.

Pour cela, on définit une variable de type caractère (`char`) qui correspondra aux initiales P/F/C.

On s'assure que l'utilisateur ne peut saisir que ces trois valeurs et on confirme son choix à l'écran.

```
// Étape 2 : Choix du joueur

do {
    System.out.println("Choisissez votre coup : (p = pierre, f = feuille, c = ciseaux)");
    chjoueur = sc.next().charAt(0);
    chjoueur = Character.toLowerCase(chjoueur);
} while (chjoueur != 'p' && chjoueur != 'f' && chjoueur != 'c');
```

### Étape 3 :

Afin que le joueur puisse affronter l'ordinateur, il faut attribuer une valeur numérique à chaque coup (0 = Pierre, 1 = Feuille, 2 = Ciseau).

En parallèle, on génère une valeur aléatoire qui déterminera le choix de l'ordinateur.

On définit donc deux variables : l'une pour stocker le tirage aléatoire, et l'autre pour représenter le choix final de l'ordinateur.

```
// Étape 3 : Choix aléatoire de l'ordinateur

aleatoire = (int) (Math.random() * 3) + 1; // 1..3
if (aleatoire == 1) {
    chOrdi = 'p'; // pierre
} else if (aleatoire == 2) {
    chOrdi = 'f'; // feuille
} else {
    chOrdi = 'c'; // ciseaux
}
```

### Étape 4 :

Pour ajouter un peu de suspense, on insère un message indiquant que l'ordinateur est en train de choisir.

Puis on fait patienter le joueur trois secondes avant d'afficher le résultat.

```
// Étape 4 : Révélation du suspense

System.out.println("L'ordinateur réfléchit à son choix...");
try {
    Thread.sleep(3000); // pause de 3 secondes (Étape 4)
} catch (InterruptedException e) {
    e.printStackTrace();
}

System.out.println("L'ordinateur a choisi : " + chOrdi);
```

### Étape 5 :

On met en place une fonction qui détermine le vainqueur de chaque manche et qui incrémente le score du gagnant.

Pour cela, deux variables `score` sont créées : une pour le joueur et une pour l'ordinateur.

La fonction compare ensuite les choix respectifs et attribue le point.

```
// Étape 5 : Déterminer le gagnant de la manche et attribuer les points

if (chJoueur == chOrdi) {
    System.out.println("Égalité ! Aucun point attribué.");
} else if ((chJoueur == 'f' && chOrdi == 'p') ||
           (chJoueur == 'p' && chOrdi == 'c') ||
           (chJoueur == 'c' && chOrdi == 'f')) {
    System.out.println("Vous avez gagné ce tour ! +1 point");
    scoreJoueur++;
} else {
    System.out.println("L'ordinateur a gagné ce tour ! +1 point");
    scoreOrdi++;
}
```

### Étape 6 :

À présent, il faut une boucle qui permette au jeu de continuer tant qu'aucun des deux joueurs n'a atteint le nombre de points requis.

On ajoute aussi une variable `gagnant` qui indiquera qui remporte la partie.

```
// Étape 7 : Fin de partie

if (scoreJoueur == nbpoints) {
    System.out.println(" Félicitations ! Vous avez gagné la partie !");
} else {
    System.out.println(" L'ordinateur remporte la partie !");
}
```

### Étape 7 :

Une fois la partie terminée, on affiche le vainqueur et on propose au joueur de relancer une nouvelle partie.

Un système de saisie est donc mis en place avec une variable `char reponse` (O pour oui, N pour non).

En fonction de la réponse, le jeu recommence ou s'arrête.

```
// Demande à rejouer (réponse simple o/n)
System.out.println("Souhaitez-vous rejouer ? (o = oui / n = non)");
char reponse = sc.next().charAt(0);
rejouer = (Character.toLowerCase(reponse) == 'o');

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

### Étape 8 :

Une fois le fonctionnement de base assuré, on peut ajouter quelques options bonus.

Par exemple, on insère des icônes représentant les différents choix afin de rendre l'interface plus attrayante.

### Étapes 9 et 10 :

On ajoute une variante du jeu avec l'option du puits. Celui-ci bat pierre et ciseau, mais perd contre feuille et fait égalité contre lui-même.

Un choix de mode est donc proposé à l'utilisateur, et grâce à un booléen on intègre cette règle supplémentaire dans le code.

```

import java.util.Scanner;

public class accueil_shifumi {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Message d'accueil
        System.out.println("=====");
        System.out.println("      Bienvenue dans le jeu Shifumi !");
        System.out.println("=====");
        System.out.println();
        System.out.println("Souhaitez-vous jouer à la version classique ou avec la variante du puits ?");
        System.out.println("Tapez 'p' pour jouer avec la version puits.");
        System.out.println("Tapez 's' pour jouer à la version standard.");
        System.out.print("Votre choix (p/s) : ");

        char choix = sc.next().charAt(0);
        choix = Character.toLowerCase(choix);

        if (choix == 'p') {
            System.out.println("Lancement de la version avec le puits...");
            Shifumi_puit.main(null); // Appel de la version avec puits
        } else {
            System.out.println("Lancement de la version classique...");
            ShifumiClassique.main(null); // Appel de la version classique
        }

        System.out.println("Merci d'avoir lancé Shifumi !");
        sc.close();
    }
}

```

```

// Détermination du gagnant
if (chJoueur == chOrdi) {
    System.out.println("Égalité !");
} else if ((chJoueur == 'p' && chOrdi == 'c') ||
           (chJoueur == 'f' && (chOrdi == 'p' || chOrdi == 't')) ||
           (chJoueur == 'c' && chOrdi == 'f') ||
           (chJoueur == 't' && (chOrdi == 'p' || chOrdi == 'c')))) {
    System.out.println("Vous avez gagné ce tour !");
    scoreJoueur++;
} else {
    System.out.println("L'ordinateur a gagné ce tour !");
    scoreOrdi++;
}

```

```

// Choix du joueur
do {
    System.out.println("Choisissez votre coup : pierre (p), feuille (f), ciseaux (c), puits (t)");
    chjoueur = Character.toLowerCase(sc.next().charAt(0));
} while (chjoueur != 'p' && chjoueur != 'f' && chjoueur != 'c' && chjoueur != 't');

// Choix de l'ordinateur (aléatoire)
int aleatoire = (int) (Math.random() * 4) + 1;
if (aleatoire == 1)
    chOrdi = 'p';
else if (aleatoire == 2)
    chOrdi = 'f';
else if (aleatoire == 3)
    chOrdi = 't';
else
    chOrdi = 'c';

```

### 3) Conclusion

Ce TP m'a permis de construire progressivement un jeu en Java en utilisant boucles, conditions, fonctions et générateur aléatoire.