

## Rapport de TP : TP Shifumi

### Sommaire :

<b>1. Introduction</b>	<b>1</b>
<b>2. Conception de la solution</b>	<b>2</b>
<b>3. Réalisation</b>	<b>3</b>
Étape 1 : Choix du nombre de points	3
Étape 2 : Choix du joueur	3
Étape 3 – Choix aléatoire de l'ordinateur	4
Étape 4 – Révélation du suspense	4
Étape 5 – Déterminer le gagnant de la manche	5
Étape 6 – Boucle des manches	6
Étape 7 – Fin de partie	7
Étape 8 – Mise en forme	8
<b>4. Conclusion</b>	<b>9</b>

## 1. Introduction

- **Objectif :**

L'objectif de ce TP est de me familiariser avec la programmation en Java en développant un petit jeu interactif, le Shifumi, permettant de manipuler des variables, des boucles, des conditions et des saisies utilisateur.

- **Mise en situation :**

En tant qu'étudiant en BTS SIO, je dois créer un programme qui simule un jeu entre un joueur et l'ordinateur, en gérant le choix des points, la saisie utilisateur, le choix aléatoire de l'ordinateur et le calcul du score.

- **Résolution :**

J'ai travaillé depuis le compilateur en ligne [www.jdoodle.com](http://www.jdoodle.com). J'ai testé le programme directement dans la console, permettant de vérifier le fonctionnement des boucles, conditions et affichages, et de simuler une partie complète de Shifumi sur mon ordinateur.

## 2. Conception de la solution

La réalisation de ce TP peut se décomposer en plusieurs sous-problèmes :

### **Étape 1 – Choix du nombre de points :**

Il me faut demander à l'utilisateur combien de points il souhaite pour la partie. J'ai utilisé une boucle **while** pour forcer une saisie correcte (3, 5 ou 10) et afficher le nombre choisi.

### **Étape 2 – Choix du joueur :**

Je dois récupérer le choix du joueur entre pierre (p), feuille (f) ou ciseau (c). Pour cela, j'ai utilisé `next().charAt(0)` et ajouté une boucle **while** pour vérifier la validité de la saisie.

### **Étape 3 – Choix aléatoire de l'ordinateur :**

Pour l'ordinateur, j'ai utilisé `Math.random()` pour générer un nombre aléatoire de 1 à 3, correspondant aux trois choix du jeu.

### **Étape 4 – Révélation du suspense :**

J'ai créé un effet de suspense avant de dévoiler le choix de l'ordinateur avec `Thread.sleep(3000)`.

### **Étape 5 – Déterminer le gagnant de la manche :**

J'ai comparé les choix du joueur et de l'ordinateur avec des conditions **if** / **else if** / **else** pour déterminer le vainqueur de chaque manche. J'ai dû déclarer les variables `scoreJoueur` et `scoreOrdi` avant de les utiliser.

### **Étape 6 – Boucle des manches :**

J'ai utilisé une boucle **while** qui répète les étapes précédentes jusqu'à ce qu'un joueur atteigne le nombre de points choisi, permettant de gérer automatiquement plusieurs manches.

### **Étape 7 – Fin de partie :**

Quand un joueur atteint le nombre de points, le programme affiche le gagnant final avec des emojis, puis demande si le joueur souhaite rejouer.

## Étape 8 – Mise en forme :

Pour rendre le jeu plus ludique, j'ai remplacé les lettres par des symboles Pierre = α

Feuille = \_\_

Ciseau =>

## 3. Réalisation

### Étape 1 : Choix du nombre de points

Pour commencer, j'ai demandé à l'utilisateur combien de points il voulait pour la partie (3, 5 ou 10).

J'ai utilisé une boucle `while` pour forcer une saisie correcte : tant que la valeur entrée n'est pas 3, 5 ou 10, le programme redemande la saisie. J'ai affiché ensuite le nombre choisi.

```
// étape 1: afficher et demander à l'utilisateur le nombre de points de la partie
Scanner Scanner = new Scanner(System.in);

while(nbpoints!=3 && nbpoints!= 5 && nbpoints!=10) { // tant que la saisie n'est pas 3, 5
    System.out.println("en combien de points doit se dérouler la partie.(3, 5 ou 10)");
    nbpoints=Scanner.nextInt();
    System.out.println("vous avez saisi: "+nbpoints);
}
```

Résultat :

```
en combien de points doit se dérouler la partie.(3, 5 ou 10)
2
vous avez saisi: 2
en combien de points doit se dérouler la partie.(3, 5 ou 10)
3
vous avez saisi: 3
choisissez un outil (p, f ou c)
```

## Étape 2 : Choix du joueur

Ensuite, j'ai demandé au joueur de choisir entre **pierre (p)**, **feuille (f)** ou **ciseau (c)**.

J'ai utilisé la méthode `next().charAt(0)` pour récupérer la première lettre tapée par le joueur.

J'ai aussi ajouté une boucle `while` pour vérifier que la saisie est correcte.

```
// étape 2: choisir l'outil de jeu
System.out.println("Choisissez un outil (p, f ou c)");
choixJoueur = Scanner.next().charAt(0);
while (choixJoueur != 'p' && choixJoueur != 'f' && choixJoueur != 'c') {
    System.out.println("Erreur. Tapez p, f ou c :");
    choixJoueur = Scanner.next().charAt(0);
}
System.out.println("Vous avez choisi : " + choixJoueur);
```

Résultat :

```
Choisissez un outil (p, f ou c)
m
Erreur. Tapez p, f ou c :
p
Vous avez choisi : p
```

## Étape 3 – Choix aléatoire de l'ordinateur

Pour le choix de l'ordinateur, j'ai utilisé la fonction `Math.random()` pour tirer un nombre aléatoire entre 1 et 3. Chaque nombre correspond à un choix :

1=pierre

2=feuille

3=ciseau

```
// étape 3 : choix de l'ordinateur
aleatoire = (int)(Math.random() * 3) + 1;
if (aleatoire == 1) choixOrdi = 'p';
else if (aleatoire == 2) choixOrdi = 'f';
else choixOrdi = 'c';
```

## Étape 4 – Révélation du suspense

Ensuite, nous devons créer un petit effet de suspense avant d'afficher le choix de l'ordinateur. J'ai donc ajouté une pause avec la fonction `Thread.sleep(3000)` (3 secondes).

```
// étape 4 : effet de suspens
try {
    Thread.sleep(3000);
} catch (InterruptedException e) {
    e.printStackTrace();
}
System.out.println("L'ordinateur a choisi : " + choixOrdi);
```

Résultat:

```
Vous avez choisi : p
L'ordinateur a choisi : c
```

La saisie de l'ordinateur s'est bien affichée au bout de 3 secondes.

## Étape 5 – Déterminer le gagnant de la manche

Dans cette étape, j'ai dû comparer le choix du joueur avec celui de l'ordinateur pour savoir qui remportait la manche.

Pour cela, j'ai utilisé plusieurs conditions `if`, `else if` et `else` pour vérifier toutes les possibilités selon les règles du jeu Shifumi.

Au début, quand j'ai voulu écrire : `scoreJoueur++` et `scoreOrdi++`;

le programme m'a affiché une **erreur de compilation** indiquant :  
`cannot find symbol : variable scoreJoueur`.

J'ai compris que cette erreur venait du fait que **je n'avais jamais déclaré ces deux variables** avant de les utiliser. J'ai donc ajouté en haut du code deux nouvelles variables pour compter les points :

```
int scoreJoueur = 0;
int scoreOrdi = 0;
```

Une fois ces variables créées, j'ai pu écrire la comparaison des choix comme ceci :

```
// étape 5 : comparaison
if (choixJoueur == choixOrdi) {
    System.out.println("Égalité !");
} else if ((choixJoueur == 'p' && choixOrdi == 'c') ||
           (choixJoueur == 'f' && choixOrdi == 'p') ||
           (choixJoueur == 'c' && choixOrdi == 'f')) {
    System.out.println("Vous gagnez cette manche !");
    scoreJoueur++;
} else {
    System.out.println("L'ordinateur gagne cette manche !");
    scoreOrdi++;
}
}
```

Résultat :

```
En combien de points doit se dérouler la
3
Vous avez saisi: 3
Choisissez un outil (p, f ou c)
p
Vous avez choisi : p
L'ordinateur a choisi : c
Vous gagnez cette manche !
Score joueur : 1 | Score ordi : 0
```

## Étape 6 – Boucle des manches

J'ai mis une boucle **while** qui répète les étapes 2 à 5 jusqu'à ce que l'un des deux joueurs atteigne le nombre de points choisi au départ. Cela permet de rejouer plusieurs manches automatiquement.


```
// boucle de manches
while (scoreJoueur < nbpoints && scoreOrdi < nbpoints) { ←
    // étape 2: choix du joueur
    System.out.println("Choisissez un outil (p, f ou c)");
    choixJoueur = Scanner.next().charAt(0);
    while (choixJoueur != 'p' && choixJoueur != 'f' && choixJoueur != 'c') {
        System.out.println("Erreur. Tapez p, f ou c :");
        choixJoueur = Scanner.next().charAt(0);
    }
    System.out.println("Vous avez choisi : " + choixJoueur);
}
```

Résultat :

```

En combien de points doit se dérouler la partie (3, 5 ou 10) ?
3
Vous avez saisi: 3
Choisissez un outil (p, f ou c)
p
Vous avez choisi : p
L'ordinateur a choisi : c
Vous gagnez cette manche !
Score joueur : 1 | Score ordi : 0
Choisissez un outil (p, f ou c)

```



## Étape 7 – Fin de partie

Quand un joueur atteint le nombre de points, j'affiche le gagnant final avec des emojis que j'ai tapés via le raccourci emoji.

```

// étape 7 : fin de partie
if (scoreJoueur == nbpoints) {
    System.out.println("🎉 Félicitations, vous avez gagné la partie !");
} else {
    System.out.println("😞 L'ordinateur a gagné la partie !");
}

```

Résultat lorsque le joueur a gagné la partie par exemple :

```

Vous avez choisi : p
L'ordinateur a choisi : c
Vous gagnez cette manche !
Score joueur : 3 | Score ordi : 2
🎉 Félicitations, vous avez gagné la partie !


```

J'ai ensuite ajouté une question pour savoir si le joueur veut rejouer :

```

// étape 7 : fin de partie
if (scoreJoueur == nbpoints) {
    System.out.println("🎉 Félicitations, vous avez gagné la partie !");
} else {
    System.out.println("😞 L'ordinateur a gagné la partie !");
}
System.out.println("Voulez-vous rejouer ? (O/N)");
char rejouer = Scanner.next().charAt(0);

```



Résultat :

```

Vous avez choisi : p
L'ordinateur a choisi : f
L'ordinateur gagne cette manche !
Score joueur : 2 | Score ordi : 3
😞 L'ordinateur a gagné la partie !
Voulez-vous rejouer ? (O/N)

```

## Étape 8 – Mise en forme

Pour rendre le jeu plus agréable, j'ai remplacé les lettres par des symboles :

Pierre → ✎

Feuille → \_\_

Ciseau → >

```
// affichage stylisé du choix du joueur étape 8 ←
if (choixJoueur == 'p') System.out.println("Pierre : ✎");
else if (choixJoueur == 'f') System.out.println("Feuille : __");
else System.out.println("Ciseau : >");
```

```
// étape 3 : choix de l'ordinateur
aleatoire = (int)(Math.random() * 3) + 1;
if (aleatoire == 1) choixOrdi = 'p';
else if (aleatoire == 2) choixOrdi = 'f';
else choixOrdi = 'c';
```

```
// étape 4 : effet de suspense
try {
    Thread.sleep(3000);
} catch (InterruptedException e) {
    e.printStackTrace();
}
```

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

```
// affichage stylisé du choix de l'ordinateur étape 8 ←
if (choixOrdi == 'p') System.out.println("Pierre : ✎");
else if (choixOrdi == 'f') System.out.println("Feuille : __");
else System.out.println("Ciseau : >");
```

```
// etape 7 : fin de partie
if (scoreJoueur == nbpoints) {
    System.out.println("🎉 Félicitations, vous avez gagné la partie !");
} else {
    System.out.println("😞 L'ordinateur a gagné la partie !");
}
System.out.println("Voulez-vous rejouer ? (O/N)");
char rejouer = Scanner.next().charAt(0);
```

```
// étape 8 : mise en forme
System.out.println("\nMerci d'avoir joué au Shifumi !");
System.out.println("Voici la légende des symboles :");
System.out.println("Pierre = ✎   Feuille = __   Ciseau = >");
```

Résultat :



```
En combien de points doit se dérouler la partie (3, 5 ou 10) ?
3
Vous avez saisi: 3
Choisissez un outil (p, f ou c)
p
Vous avez choisi : p
Pierre : ✎
```

On a bien un symbole à la place d'un simple "p", rendant donc le jeu un petit peu plus amusant.

## 4. Conclusion

Ainsi, ce TP m'a permis de consolider mes connaissances en Java en manipulant variables, boucles, conditions et entrées utilisateur. J'ai apprécié coder ce petit jeu Shifumi et y consacrer environ 2 heures pour concevoir, tester et améliorer la mise en forme du programme. Cette première expérience de jeu interactif m'a permis de lier la théorie (algorithmes) à la pratique (code exécutable) de manière ludique.