

Compte rendu TP4-Shifumi

Sommaire

<u>Introduction</u>	1
<u>Étape : 1</u>	2
<u>Étape : 2</u>	2
<u>Étape : 3</u>	3
<u>Étape : 4</u>	3
<u>Étape : 5</u>	4
<u>Étape : 6</u>	5
<u>Étape : 7</u>	5
<u>Conclusion</u>	7

Introduction

Dans ce TP, j'ai travaillé sur la création d'un algorithme du jeu shifumi en Java. Ce projet m'a permis de mettre en pratique ce que j'ai appris en programmation et en algorithmique. J'ai pu apprendre à organiser mon code, à utiliser des classes et des méthodes, et à mettre en place la logique du jeu. Dans ce compte-rendu, je vais expliquer les différentes étapes de création de mon projet.

J'ai tout d'abord établi les règles du jeu pour les deux adversaires .

Étape : 1

Définir le nombre de points

```
shifumi.java X
1 import java.util.Scanner;
2
3 /**
4  * la classe principale pour le jeu shifumi (pierre-feuille-ciseaux)
5  * ce programme simule un jeu contre l'ordinateur avec des règles classiques.
6  * Règles du jeu
7  * papier bat la pierre
8  * la pierre bat le ciseaux
9  * le ciseaux bat le papier
10 * si deux items similaires alors égalité
11 *
12 * @author fadi
13 * @version 1.0
14 */
15
16 public class shifumi {
17
18     public static void main(String[] args) {
19         // TODO Auto-generated method stub
20
21         //déclaration de variables
22         int nbpoints = 0;
23
24         //étape 1 : affichez et demandez à l'utilisateur le nombre de points pour la partie en cours
25         while(nbpoints!=3 && nbpoints!=5 && nbpoints!=10) // tant que la saisie n'est pas 3, 5 ou 10
26             System.out.println("En combien de points se déroule la partie ? (3, 5 ou 10)"); // affichez à l'utilisateur
27
28         Scanner scanner = new Scanner(System.in);
29         nbpoints=scanner.nextInt();
30         System.out.println("vous avez saisi " +nbpoints);
31     }
32 }
33
34 }
35
```

Le résultat :

```
Problems Declaration Console X Eclipse IDE for Java Developers 2025-09 Release
<terminated> shifumi [Java Application] C:\Users\eleve\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot
En combien de points se déroule la partie ? (3, 5 ou 10)
3
vous avez saisi 3
```

Étape : 2

Choix du joueur

```
//étape 2 : affichez et demandez à l'utilisateur choisir pierre, feuille ou ciseaux
do {
    System.out.println("choisissez pierre (P), Feuille (F) ou Ciseaux (C):");
    choixjoueur=scanner.next().charAt(0);
    System.out.println("vous avez choisi:" + choixjoueur);
}while(choixjoueur != 'P' && choixjoueur != 'F' && choixjoueur != 'C');
```

Le résultat :

```
En combien de point se deroule la partie ?)(3,5 ou 10
3
vous avez saisi 3
choisissez pierre (P), Feuille (F) ou Ciseaux (C):
```

Étape : 3

Choix aléatoire de l'ordinateur

```
//étape 3: choix aléatoire de l'ordinateur

//déclaration et utilisation de la variable aleatoire
aleatoire= (int)(Math.random()*3)+1;
// condition d'attribution de p,f et c à partir du random
if (aleatoire==1) {
    choixOrdi='P';
}
else if (aleatoire==2) {
    choixOrdi='F';
}
else {
    choixOrdi='C';
}

// System.out.println(l'ordi a choisi : "+choixOrdi);
*
```

Le résultat :

```
En combien de point se deroule la partie ?)(3,5 ou 10
3
vous avez saisi 3
choisissez pierre (P), Feuille (F) ou Ciseaux (C):
P
vous avez choisi:P
L'ordinateur a choisi ...
l'ordi a choisi F
```

Étape : 4

Révélation du suspense

```
// étape 4 : Attente de 3 sec pour le suspense
System.out.println("L'ordinateur a choisi ...");
// try catch permet de continuer le programme en cas de probleme de la fonction thread
try {
    Thread.sleep(3000);
} catch (InterruptedException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

System.out.println("l'ordi a choisi " + choixOrdi);
System.out.println("fin du suspense !");
}
```

Le résultat :

```
choisissez pierre (P), feuille (F) ou ciseaux (C) : P
vous avez choisi:P
L'ordinateur a choisi ...
l'ordi a choisi F
fin du suspense !
```

Étape : 5

Déterminer le gagnant de la manche

```

// étape 5 : déterminer qui gagne la manche
if (choixJoueur == choixOrdi) {
    System.out.println("égalité !");
} else if ((choixJoueur == 'p' && choixOrdi == 'c') ||
           (choixJoueur == 'f' && choixOrdi == 'p') ||
           (choixJoueur == 'c' && choixOrdi == 'f')) {
    // afficher si Joueur gagne la manche
    System.out.println("vous avez gagné la manche !");
    scoreJoueur++;
} else {
    // perdu
    System.out.println("vous avez perdu la manche");
    scoreOrdi++;
}

// Affichage du score
System.out.println("Score : Joueur " + scoreJoueur + " - Score ordi " + scoreOrdi);
}
}

```

Le résultat :

```

L'ordi a choisi : F
Vous avez perdu
scorejoueur 0 - scoreordi : 1

```

Étape : 6

Boucle des manches

```

int scoreOrdi;
//étape 6
//étape 1 : affichez et demander a l'utilisateur le nombre de point pour la partie en cours
while(nbpoints!=3 && nbpoints!=5 && nbpoints!=10) { // tant que la saisie n'est pas n'est pas 3 5 ou 10
    System.out.println("En combien de point se deroule la partie ?)(3,5 ou 10"); // afficher a l'utilisateur
    nbpoints=scanner.nextInt();
    System.out.println("vous avez saisi " +nbpoints);
}

```

Étape : 7

Fin de partie

```
// étape 7 : fin de partie
// déterminer qui a gagné la partie complète
if (scoreJoueur > scoreOrdi) {
    System.out.println("Félicitations, vous avez gagné la partie !");
} else if (scoreJoueur < scoreOrdi) {
    System.out.println("Vous avez perdu la partie...");
} else {
    System.out.println("Égalité parfaite !");
}

// affichage du score final
System.out.println("Score final : Joueur " + scoreJoueur + " - Ordinateur " + scoreOrdi);
System.out.println("Merci d'avoir joué !");
```

Cas 1 — le joueur gagne :

cela affichera :

Félicitations, vous avez gagné la partie !

Score final : Joueur 3 - Ordinateur 1

Merci d'avoir joué !

Cas 2 — le joueur perd :

cela affichera :

Vous avez perdu la partie...

Score final : Joueur 2 - Ordinateur 3

Merci d'avoir joué !

Cas 3 — égalité :

cela affichera :

Égalité parfaite !

Score final : Joueur 2 - Ordinateur 2

Merci d'avoir joué !

Conclusion

Ce projet de création d'un algorithme de jeu en Java m'a permis de mieux comprendre la logique de développement d'un programme complet.

J'ai appris à résoudre certains problèmes rencontrés pendant la réalisation du jeu malgré mes difficultés .