

TP SHIFUMI

Objectif: Réaliser le jeu "pierre feuille ciseau" contre un ordinateur en Java.

Introduction

Dans ce TP, j'ai créé un jeu de SHIFUMI en Java. Le but est de jouer contre l'ordinateur. Ce projet m'a permis d'utiliser ce que j'ai appris en cours : les variables, les boucles, les conditions et la saisie utilisateur.

Outils utilisés

Pour faire ce projet, j'ai utilisé :

- Le langage Java avec le JDK 17
- L'IDE Eclipse pour écrire mon code
- Un projet nommé tp4_shifumi
- Une classe principale appelée shifumi

Étape 1 : Définir le nombre de points, et ajouter des commentaire sur chaque étape pour bien comprendre,	6
Étape 2 : Choix du joueur,	7
Étape 3 : Choix ordinateur,	7
Étape 4 : Suspense,	8
Étape 5 : Déterminer le gagnant	8
J'ai ajouté une boucle while avant l'étape 2 pour répéter le jeu jusqu'à ce qu'un joueur gagne la partie	
Étape 7: fin de partie, et afficher le gagnant. Et demander de rejouer une partie?	9
CONCLUSION	11

- Création de **projets** sur **Eclipse**, pour cela il faut aller sur file → New → Java Project)

The screenshot shows the 'New Java Project' dialog box in Eclipse. The title bar reads 'New Java Project'. The main heading is 'Create a Java Project' with a subtitle 'Create a Java project in the workspace or in an external location.' and a folder icon. The 'Project name' field contains 'tp4_shifumi'. The 'Use default location' checkbox is checked, and the 'Location' field shows 'C:\Users\F A H A D\eclipse-workspace\tp4_shifumi' with a 'Browse...' button. The 'JRE' section has three radio buttons: 'Use an execution environment JRE:' (selected) with a dropdown showing 'JavaSE-21', 'Use a project specific JRE:' with a dropdown showing 'jre', and 'Use default JRE 'jre' and workspace compiler preferences'. A 'Configure JREs...' link is to the right. The 'Project layout' section has two radio buttons: 'Use project folder as root for sources and class files' and 'Create separate folders for sources and class files' (selected). A 'Configure default...' link is to the right. The 'Working sets' section has an 'Add project to working sets' checkbox and a 'New...' button. Below is a 'Working sets:' dropdown and a 'Select...' button. The 'Module' section has a 'Create module-info.java file' checkbox, a 'Module name:' field, and a 'Generate comments' checkbox. At the bottom are a help icon, '< Back' button, 'Next >' button, 'Finish' button, and 'Cancel' button.

New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: tp4_shifumi

☒ Use default location

Location: C:\Users\F A H A D\eclipse-workspace\tp4_shifumi [Browse...](#)

JRE

☒ Use an execution environment JRE: JavaSE-21 [Configure JREs...](#)

☐ Use a project specific JRE: jre

☐ Use default JRE 'jre' and workspace compiler preferences

Project layout

☐ Use project folder as root for sources and class files

☒ Create separate folders for sources and class files [Configure default...](#)

Working sets

☐ Add project to working sets [New...](#)

Working sets: [Select...](#)

Module

☐ Create module-info.java file

Module name:

☐ Generate comments

? < Back Next > Finish Cancel

- Création de **classe**, pour cela il faut New > Class > NOM: tp4_shifumi

New Java Class

Java Class

✖ Type already exists.

Source folder:

Package:

☐ Enclosing type:

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass:

Interfaces:

Which method stubs would you like to create?

☒ `public static void main(String[] args)`

☐ Constructors from superclass

☐ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

☐ Generate comments

- Et puis on commence par les **codage**, donc au tout début on a ajouter des commentaire de ce que on vas faire sur chaque etape donc et puis on a mit les 4 règles principales du jeu et puis juste en bas on voit le **résultat** :

```
shifumi.java x
1 package tp4_shifumi;
2 /**
3  * class principale pour le jeu shifumi (pierre-feuille-ciseau)
4  * ce programme simule un jeu contre l'ordinateur avec des règles classique
5  */
6 /** author Mme Beauvallet
7  * @version 1.0
8  */
9
10 public class shifumi {
11
12     public static void main(String[] args) {
13         // TODO Auto-generated method stub
14
15     }
16
17 }
18
```

```
shifumi.java x
1 package tp4_shifumi;
2 /**
3  * class principale pour le jeu shifumi (pierre-feuille-ciseau)
4  * ce programme simule un jeu contre l'ordinateur avec des règles classique
5  */
6 /** author Mme Beauvallet
7  * @version 1.0
8  */
9
10 public class shifumi {
11
12     public static void main(String[] args) {
13         // TODO Auto-generated method stub
14
15         System.out.println(" voici les règles du jeu : ");
16         System.out.println(" feuille bat pierre : ");
17         System.out.println(" Pierre bat ciseau : ");
18         System.out.println(" ciseau bat feuille : ");
19         System.out.println(" égalite si 2 éléments similaire : ");
20     }
21
22 }
23
24
```

le résultat :

```
Problems Javadoc Declaration Console x
terminated> shifumi [Java Application] C:\Users\F A H A D\.p2\p
voici les règles du jeu :
feuille bat pierre :
Pierre bat ciseau :
ciseau bat feuille :
égalité si 2 éléments similaire :
```

Étape 1 : Définir le nombre de points, et ajouter des commentaires sur chaque étape pour bien comprendre,

```
50 /**
6  * class principale pour le jeu shifumi (pierre-feuille-ciseau)
7  * ce programme simule un jeu contre l'ordinateur avec des règles classique
8  * @author Mme Beauvallet
9  * @version 1.0
10 */
11
12 public class shifumi {
13
14 public static void main(String[] args) {
15 // TODO Auto-generated method stub
16
17 System.out.println(" voici les règles du jeu : ");
18 System.out.println(" feuille bat pierre : ");
19 System.out.println(" Pierre bat ciseau : ");
20 System.out.println(" ciseau bat feuille : ");
21 System.out.println(" égalite si 2 éléments similaire : ");
22
23 //variables
24 int nbpoints;
25
26 java.util.Scanner sc = new Scanner(System.in);
27
28 // étape 1 : définir le nombre de points
29 do {
30 System.out.println("En combien de points se déroule la partie ? (3.5 ou 10)");
31 nbpoints= sc.nextInt();
32 System.out.println("vous avez saisie " + nbpoints);
33 } while(nbpoints!=3 && nbpoints!=5 && nbpoints!=10);
34
35 }
36
37 }
```

Une boucle do-while demande 3, 5 ou 10 points. La boucle se répète tant que la valeur n'est pas valide.

Et voici le **résultat**,

```
voici les règles du jeu :
feuille bat pierre :
Pierre bat ciseau :
ciseau bat feuille :
égalite si 2 éléments similaire :
En combien de points se déroule la partie ? (3.5 ou 10 )
```

```
égalite si 2 éléments similaire :
En combien de points se déroule la partie ? (3.5 ou 10 )
6
vous avez saisie 6
En combien de points se déroule la partie ? (3.5 ou 10 )
5
vous avez saisie 5
```

Étape 2 : Choix du joueur,

Une boucle do-while demande p, f ou c. J'utilise charAt(0) pour récupérer le caractère saisi.

```
36 //étape 2: Choisir l'outil pour le jeu P,F,C
37
38 do {
39     System.out.println(" choisissez pierre (p), feuille (f) ou ciseau (c):");
40     choix=sc.next().charAt(0);
41     System.out.println("vous avez choisi: "+ choix);
42 }while(choix !='p' && choix != 'f' && choix != 'c');
43
44 }
45
46
```

Et voici le **résultat**:

```
En combien de points se déroule la partie ? (3.5 ou 10 )
5
vous avez saisi 5
 choisissez pierre (p), feuille (f) ou ciseau (c):
p
vous avez choisi: P
 choisissez pierre (p), feuille (f) ou ciseau (c):
```

Étape 3 : Choix ordinateur,

Je génère un nombre aléatoire entre 1 et 3 avec Math.random(). Des conditions if/else attribuent p, f ou c selon le nombre.

```
5 //étape 3 : Choix aléatoire de l'ordinateur
6
7 char choixordi = 0;
8
9 //Déclaration et utilisation de la variable aléatoire
10
11 int aleatoire = (int)(Math.random() * 3) + 1;
12
13 //Condition d'attribution de p, F et C à partir du random
14
15 if(aleatoire == 1) {
16     choixordi = 'p';}
17
18 else if (aleatoire == 2) {
19     choixordi = 'f';}
20
21 else if(aleatoire == 3) {
22     choixordi = 'c';}
23
24 System.out.println("L'ordinateur a choisi: " +choixordi);
25
26 }
27
28 }
```

Et voici le **résultat**,

```
En combien de points se déroule la partie ? (3.5 ou 10 )
5
vous avez saisi 5
  choisissez pierre (p), feuille (f) ou ciseau (c):
f
vous avez choisi: f
L'ordinateur a choisi: p
```

Étape 4 : Suspense,

J'ai ajouté **Thread.sleep(3000)** dans un try-catch pour une pause de 3 secondes avant d'afficher le résultat.

```
//étape 4 : Attente 2,4 pour le suspense

System.out.println("l'ordinateur a choisi");
// try catch permet de continuer le programme en cas de la fonction thread
try {
    Thread.sleep(3000);
} catch (InterruptedException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
System.out.println("fin du suspense");
```

Et voici le **résultat**,

```
vous avez choisi: p
L'ordinateur a choisi: p
l'ordinateur a choisi
```

Étape 5 : Déterminer le gagnant

Je compare les choix avec des conditions **if/else**. Si égalité, j'affiche "égalité !". Si le joueur gagne, j'augmente son score. Sinon, j'augmente le score de l'ordinateur. J'affiche ensuite les scores.

```
2 if (choix == choixordi) {
3     System.out.println("égalité !");
4 } else if ((choix=='p' && choixordi=='c') ||
5     (choix=='f' && choixordi=='p') ||
6     (choix=='c' && choixordi=='f')) {
7
8     //afficher si joueur gagne ou perd
9     System.out.println("vous avez gagné la manche");
10    scorejoueur++;
11 } else {
12     System.out.println("l'ordinateur a gagné la manche");
13     scoreordi++;
14 }
```

Voici le **résultat**,

```
vous avez choisi: c
L'ordinateur a choisi: c
l'ordinateur a choisi
fin du suspense
égalité !
score : joueur 0 - score ordi : 0
choisissez pierre (p), feuille (f) ou ciseau (c):
```

Etape 6: Boucle des manches,

J'ai ajouté une boucle while avant l'étape 2 pour répéter le jeu jusqu'à ce qu'un joueur gagne la partie.

```
//etape 6: boucle des manches
//faire une boucle qui permet de relancer le choix entre P, F et C jusqu'a ce que le score
// d'un joueur soit égale au nombre de points de la manche choisi par le joueur
while (scorejoueur<nbpoints && scoreordi<nbpoints) {
```

```
vous avez choisi: f
l'ordinateur a choisi
L'ordinateur a choisi: p
fin du suspense
vous avez gagné la manche
score : joueur 1 - score ordi : 1
choisissez pierre (p), feuille (f) ou ciseau (c):
```

Étape 7: fin de partie, et afficher le gagnant. Et demander de rejouer une partie?

J'ai créé une boucle do-while qui englobe tout le jeu (de l'étape 1 à l'étape 6). Cette boucle commence avant l'étape 1 et se termine après avoir demandé si le joueur veut rejouer.

```
//étape 7:
// REJOUER LA PARTIE
do {
scorejoueur = 0;
scoreordi = 0;
```

```
// Afficher le gagnant final
if(scorejoueur == nbpoints) {
    System.out.println("Félicitations! Vous avez gagné la partie!");
} else {
    System.out.println("L'ordinateur a gagné la partie !");
}

// Demander si on rejoue
System.out.println(" voulez vous rejouer ? (o/n)");
reponse = sc.next().charAt(0);
rejouer = (reponse == 'o' || reponse == 'O');

} while(rejouer);

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


Et voici le **résultat**,

```
l'ordinateur a choisi
L'ordinateur a choisi: Ciseaux
fin du suspense
vous avez gagné la manche ! Pierre bat Ciseaux
score : joueur 5 - score ordi : 1
Félicitations! Vous avez gagné la partie!
voulez vous rejouer ? (o/n)
```

Etape 8 : J'ai ajouté la conversion des lettres en noms complets. Après la saisie du joueur, des conditions if/else transforment 'p' en "Pierre", 'f' en "Feuille" et 'c' en "Ciseaux". Cette conversion rend l'affichage plus clair,

```
String symboleJoueur = "";
String symboleOrdi = "";
```

```
//étape 8: Mise en forme - conversion du choix en texte
    if (choix == 'p') {
        symboleJoueur = "Pierre";
    } else if (choix == 'f') {
        symboleJoueur = "Feuille";
    } else if (choix == 'c') {
        symboleJoueur = "Ciseaux";
    }
}
```

```
if (choix == choixordi) {
    System.out.println("égalité ! " + symboleJoueur + " vs " + symboleOrdi);
} else if ((choix=='p' && choixordi=='c') ||
           (choix=='f' && choixordi=='p') ||
           (choix=='c' && choixordi=='f')) {
```

CONCLUSION

J'ai réussi à développer un jeu de Shifumi complet et fonctionnel. Le programme permet de choisir le nombre de points, joue autant de manches que nécessaire jusqu'à ce qu'un joueur gagne, affiche les résultats de manière claire avec les noms complets des objets au lieu de simples lettres, et permet de rejouer autant de fois qu'on le souhaite.

Ce TP m'a permis de mettre en pratique toutes les notions vues en cours : les variables, les différents types de boucles (do-while et while), les conditions if-else, et la gestion des entrées utilisateur avec Scanner. Le code est bien structuré avec des commentaires qui expliquent chaque étape, ce qui facilite la compréhension et les modifications futures.