18/09/202

Bloc 1 – TP 4

Compte-rendu Java Shifumi

Fait par : Mattéo Mouranchon – Groupe 2

Introduction

Dans le cadre de ce travail pratique, j'ai développé un jeu de **Shifumi** (Pierre-Feuille-Ciseaux) en **Java.** L'objectif était de permettre à un joueur de se mesurer à un adversaire virtuel. Ce projet m'a offert l'occasion de mettre en œuvre les notions fondamentales apprises en cours de programmation : **les variables**, **les boucles**, **les conditions**, ainsi que **la saisie et la gestion des entrées utilisateur**.

I. Conception

Avant de passer à la programmation, j'ai conçu la logique générale du jeu. J'ai pensé à cette logique suivante :

- Demander au joueur combien de points sera nécessaires pour gagner la partie.
- Laisser le joueur choisir entre pierre, feuille ou ciseaux.
- Générer un choix aléatoire pour l'ordinateur.
- Comparer les deux choix et attribuer un point au gagnant de chaque manche.
- Répéter les manches jusqu'à ce qu'un joueur atteigne le nombre de points fixé par le programme.
- Offrir la possibilité de rejouer une nouvelle partie une fois finit.

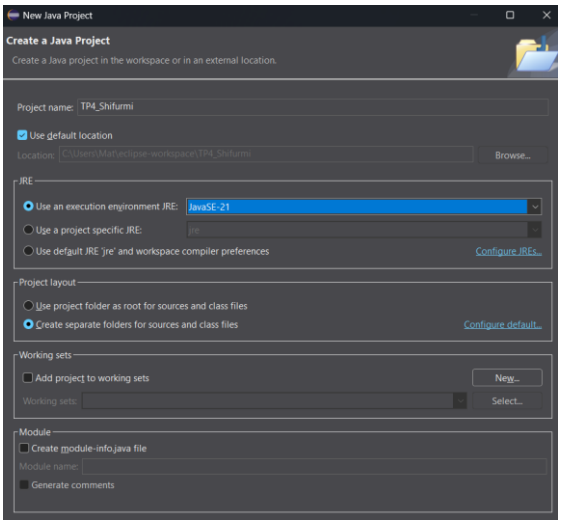
Les prérequis pour faire le programme en Java est le suivant :

- Installé un logiciel pour faire le codage comme Visual Code studio ou IDE éclipse. Dans le cas de se TP, il sera fait sur IDE éclipse for Java.
- Installé oracle java est la base de tout le code java.
- Que la version de java est en 21.

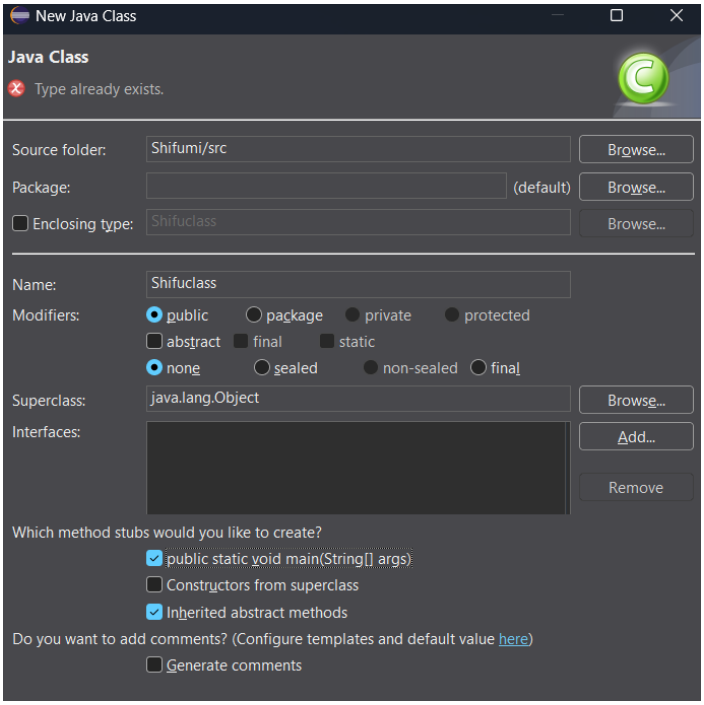
II. Réalisation

Avec toutes les conditions remplies, on va pourvoir commencer le TP.

Pour commencer, on va crée un projet java, nom : TP4_Shifurmi.



Une fois le projet crée, il faut maintenant crée une class. La class va servir pour en comme un bureau pour faire notre programme shifumi



Maintenant que notre dossier et la class crée, on va commencer par les codage, donc au tout début on a ajouté les codes de début on vas faire sur chaque étape donc et puis on a mis les 4 règles principales du jeu et puis juste en bas on voit le résultat :

```java
import java.util.Scanner;

/**
 * Classe principale pour le jeu shifumi (pierre-feuille-ciseau)
 * Ce programme simule un jeu contre l'ordinateur avec des règles clasiques.
 *
 * @author Mat
 * @version 1.0
 * */

public class Shifucalass {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

                //Liste des variables
                int nbpoints = 0;           // nombre de points pour gagner la partie
                char choixJoueur = 0;       // choix du joueur (P, F, C)
                char choixOrdi = 0;         // choix de l'ordi (P, F, C)
                int aleatoire = 0;          // nombre aléatoire pour le choix de l'ordi
                int scoreJoueur = 0;        // score du joueur
                int scoreOrdi = 0;          // score de l'ordi
                boolean rejouer = true;     // pour contrôler la boucle principal
                char reponse = 0;           // réponse du joueur à "rejouer ?"


                // Début
                Scanner sc = new Scanner(System.in);

                System.out.println("Voici les règles du jeu :");
                System.out.println("Feuille bat Pierre");
                System.out.println("Pierre bat Ciseau");
                System.out.println("Ciseau bat Feuille");
                System.out.println("Égalité si 2 éléments similaires");
```

Sur l'image, on a créé une liste de variables qui sera mémoriser dans tout le long du programme selon le nom de variable choisi et on a codé le début du programme qui va montrer les règles de base.

Résultat :

```
Voici les règles du jeu :
Feuille bat Pierre
Pierre bat Ciseau
Ciseau bat Feuille
Égalité si 2 éléments similaires
En combien de points se déroule la partie ? (3, 5 ou 10)
```

Etape 1 : définir le nombre de points

Cette partie va déterminer le

```
// étape1 étape Choix du nombre de points
do {
    System.out.println("En combien de points se déroule la partie ? (3, 5 ou 10)");
    while (!sc.hasNextInt()) {
    System.out.println("Veuillez saisir un nombre valide (3, 5 ou 10) :");
    sc.next();
    }
    nbpoints = sc.nextInt();
    System.out.println("Vous avez saisi : " + nbpoints);
} while (nbpoints != 3 && nbpoints != 5 && nbpoints != 10);
```

Cette étape va déterminé le nombre de partie pour mettre fin. La 1er ligne en-dessous de do va demander le nombre de point entre 3, 5 ou 10, tout autre nombre sera invalide. While retiens le nombre de point fixer selon la variable nbpoint.

Résultats :

```
 voici les régles 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
```

Etape 2 : choix des options Pierre, Ciseau, Feuille

La 2e étape va servir à faire notre choix entre la Pierre, Feuille ou Ciseau

```
//étape 2 : choix du joueur entre pierre(P), ciseau (C) et feuille (F).
do {
    System.out.println("Choisissez Pierre(P), Ciseau (C) et Feuille (F) :");
    choixjoueur=sc.next().charAt(0);
    System.out.println("Vous avez choisi:");
    } while(choixjoueur !='P' && choixjoueur !='F' && choixjoueur != 'C');
```

Sur le code, même chose du do-while. On impose un choix entre ses 3 options et les lettre mémoriser sont P (Pierre), F (Feuille) et C (Ciseau

Résultat :

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

Etape 3 : choix de l'ordi

Cette étape est le même principe du joueur mais à la place c'est les choix pour l'ordi.

```java
//etape 3 : Choix aléatoire de l'ordinateur

// declaration et utilisation de la variable aleatoire

aleatoire= (int)(Math.random()*3)+1;
// Condition d'atribution de P, F et C à partie du random
if (aleatoire==1) {
    choixordi='P';
}
else if (aleatoire==2) {
    choixordi='F';
}
else if (aleatoire==3) {
    choixordi='C';
}
System.out.println("l'ordi a choisi :"+choixordi);
```

A la première ligne, l'ordi va générer un nombre au hasard entre 1 et 3 qui seront définit par les 3 choix actuelle du joueur.

Résultat :

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

Étape 4 : Suspense

Cette étape va servir à ajouter du suspense au joueur pour croire qu'il a réussi ou pas. Pour cela on va ajouter **Thread.sleep(3000)**

```java
//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 problème de la fonction thread
try {
        Thread.sleep(3000);
    } catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    System.out.println("Fin du suspense !");
```

Cela va servir pour ajouter un délai, exemple de 3 secs sur notre code actuel puis il affichera « fin du suspense »

Résultat :

```
Vous avez choisi : P
L'ordi a choisi : C
L'ordinateur réfléchit...
Fin du suspense !
```

Etape 5 : Déterminer le gagnant.

Pour déterminer le gagnant, il faut que notre code connasse les règles et cela va être possible grâce à if/else.

```
// Déterminer le gagnant de la manche
if (choixJoueur == choixOrdi) {
    System.out.println("Égalité ! Voilà 2 gros cerveau !");
} 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++;
}

System.out.println("Score - Vous: " + scoreJoueur + " | Ordi: " + scoreOrdi);
System.out.println("-----------------------------------------------------");
```

L'étape va regarder quel lettre le joueur et l'ordi à poser puis avec les règles fixer. Si Le joueur à réussi, il aura « vous avez gagnez cette manche » ou sinon « L'ordinateur gagne cette manche ». Les deux derniers codes permet de montrer le score du joueur et l'ordi et la ligne sert à séparer les manches pour éviter toute confusion.

Résultat :

Le joueur gagne :

```
Vous avez choisi : P
L'ordi a choisi : C
L'ordinateur réfléchit...
Fin du suspense !
Vous gagnez cette manche !
Score - Vous: 1 | Ordi: 0
-----------------------------------------------------
```

L'ordi gagne :

```
Vous avez choisi : C
L'ordi a choisi : P
L'ordinateur réfléchit...
Fin du suspense !
L'ordinateur gagne cette manche !
Score - Vous: 2 | Ordi: 1
-----------------------------------------------------
```

Egalité :

```
Vous avez choisi : C
L'ordi a choisi : C
L'ordinateur réfléchit...
Fin du suspense !
Égalité ! Voilà 2 gros cerveau !
Score - Vous: 1 | Ordi: 0
-----------------------------------------------------
```

Etape 6 : boucle de la partie

Il est important de faire en sorte que le manche continue en boucle sans à chaque fois redémarrer, alors on va ajouter une boucle while et while continuera et s'arretera quand il aura un gagnant. Soit le joueur soit l'ordi

```
// étape 6 Boucle du jeu
while (scoreJoueur < nbpoints && scoreOrdi < nbpoints) {
```

Résultat :

```
Vous avez choisi : C
L'ordi a choisi : P
L'ordinateur réfléchit...
Fin du suspense !
L'ordinateur gagne cette manche !
Score - Vous: 2 | Ordi: 1
--------------------------------------------------
Choisissez Pierre (P), Ciseau (C) ou Feuille (F) :
```

Etape 7 : fin de partie, et afficher le gagnant. Demander si veut rejouer une partie

Si on veut que cela continue 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.

```
    // 7 étape étape Fin de partie
    if (scoreJoueur == nbpoints) {
        System.out.println("Vous avez gagné ! Vous êtes un génie !");
    } else {
        System.out.println("Vous avez perdu ! vermine !");
    }

    // Demander si on rejoue
    do {
        System.out.println("Voulez-vous rejouer ? (O/N)");
        reponse = sc.next().charAt(0);
    } while (reponse != 'O' && reponse != 'N');

    rejouer = (reponse == 'O');

} while (rejouer);

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

Sur le code, selon l'adversaire, si le joueur gagne, il aura la première ligne ou sinon l'autre ligne que on a perdu puis posera la question si on veut rejouer.

Code du score :

```
// Réinitialisation des scores à chaque nouvelle partie
scoreJoueur = 0;
scoreOrdi = 0;
```

Résultat :

```
Fin du suspense !
Vous gagnez cette manche !
Score - Vous: 3 | Ordi: 0
------------------------------------
Vous avez gagné ! Vous êtes un génie
Voulez-vous rejouer ? (O/N)
```

Retour d'expérience

Ce TP m'a permis de consolider mes bases en programmation Java et de mieux comprendre la logique des boucles et conditions.

L'un des principaux défis a été la **gestion des entrées invalides** : j'ai dû m'assurer que le programme ne plante pas si l'utilisateur entre une valeur incorrecte. J'ai également appris à rendre le jeu plus agréable à jouer en ajoutant un petit délai d'affichage et des messages dynamiques.

Cette expérience m'a montré l'importance d'une **structure claire du code** et de **commentaires explicatifs** pour garder un projet lisible et maintenable.

Conclusion

Ce TP m'a permis de réaliser un **jeu complet, fonctionnel et interactif**. J'ai appliqué les connaissances acquises en cours, notamment sur les variables, les boucles, les conditions et la saisie utilisateur.
Le résultat final est un programme stable, bien organisé et facile à comprendre. Cette réalisation constitue une première étape vers des projets plus complexes dans le monde de Java où la logique et la créativité s'installera petit à petit dans mes compétence s'installera.