

Bloc 1 TP 4

1. Analyse du problème	2
2. La conception de la solution	2
3. Réalisation	2
4. Test	7
5. Retour d'expérience	7

1. Analyse du problème

Le but est de réaliser un Shifumi en Java.

2. La conception de la solution

La règle du jeu consiste à choisir entre une pierre, une feuille et un ciseau.

-La pierre écrase le ciseau.

-La feuille recouvre la pierre.

-Le ciseau coupe la feuille.

Il faut simuler ce jeu sur ordinateur, avec un programme qui choisira aléatoirement l'un des trois objets.

3. Réalisation

Une boucle "restart" est utilisée : elle permet de savoir si le joueur veut rejouer (nous y reviendrons plus tard).

On demande un nombre de rounds (3, 5 ou 10). On récupère la saisie de l'utilisateur. Si la valeur entrée n'est ni 3, ni 5, ni 10, on redemande grâce à une boucle "while" jusqu'à ce qu'elle soit correcte. Un message spécifique est affiché avec "if" pour indiquer si l'utilisateur a entré une mauvaise ou une bonne valeur.

```
public class Shifumi {  
    public static void main(String[] args) {  
        // Déclaration des variables  
        int result = 0;  
        int round = 0;  
        int playerPoint = 0;  
        int botPoint = 0;  
        char playerItem = '0';  
        char botItem = '0';  
        int aleatoire = 0;  
        char restartChoice = '0';  
        boolean restart = true;  
        Scanner sc = new Scanner(System.in);  
  
        //Etape 7 bis  
        while (restart == true) {  
            restart = false;  
            //  
            // Etape 1: Afficher et demander le nombre de points pour la partie  
            while(round != 3 && round != 5 && round != 10) {  
                System.out.println("Combien souhaitez-vous de rounds ? (3, 5 ou 10)");  
                round = sc.nextInt();  
                if (round != 3 && round != 5 && round != 10) {  
                    System.out.println("La valeur entrée est incorrect.");  
                } else  
                    System.out.println("Vous avez rentrez : " + round);  
            }  
        }  
    }  
}
```

Une autre boucle vérifie si les points du joueur ou de l'ordinateur sont égaux au nombre de rounds (nous en reparlerons ensuite).

Ensuite, on demande à l'utilisateur quel objet il souhaite utiliser.

Il peut écrire le mot complet ou simplement la lettre correspondant à l'objet.

On utilise `"sc.next().charAt(0);"` pour récupérer la première lettre de ce qu'il a saisi.

De la même manière, une boucle *"while"* vérifie si la valeur correspond à 'P', 'F' ou 'C'. Sinon, la demande est répétée. Une condition *"if"* permet aussi d'afficher un message personnalisé.

L'objet choisi est ensuite converti en symbole pour un meilleur rendu esthétique côté utilisateur (X, _ >).

```
//Etape 6 bis
while (playerPoint < round && botPoint < round) {
    //
    // Etape 2: Afficher et demander l'item
    while(playerItem != 'P' && playerItem != 'F' && playerItem != 'C' && playerItem != 'p' && playerItem != 'f' && playerItem != 'c') {
        System.out.println("Veuillez choisir entre Pierre (P), Feuille (F) ou Ciseau (C).");
        playerItem = sc.next().charAt(0);

        if (playerItem != 'P' && playerItem != 'F' && playerItem != 'C' && playerItem != 'p' && playerItem != 'f' && playerItem != 'c') {
            System.out.println("L'objet entrée est incorrect.");
        }
    }

    //Etape 8 bis
    if (playerItem == 'P' || playerItem == 'p') {
        playerItem = 'X';
    } else if (playerItem == 'F' || playerItem == 'f') {
        playerItem = '_';
    } else {
        playerItem = '>';
    }
    //
    System.out.println("Vous avez choisie : " + playerItem);

    //Etape 3
    aleatoire = (int) (Math.random()*3)+1;

    if (aleatoire == 1) {
        botItem = 'X';
    } else if (aleatoire == 2) {
        botItem = '_';
    } else {
        botItem = '<';
    }
}
```

L'ordinateur choisit ensuite un nombre aléatoire avec `"(int) (Math.random()*3)+1"` entre 1 et 3. Le chiffre choisi détermine l'objet de l'ordinateur.

-

Un délai de 3 secondes est ajouté pour simuler la réflexion de l'ordinateur, puis celui-ci affiche son choix.

```
aleatoire = (int) (Math.random()*3)+1;

if (aleatoire == 1) {
    botItem = 'X';
} else if (aleatoire == 2) {
    botItem = '_';
} else {
    botItem = '<';
}

//Etape 4

System.out.println("L'ordinateur est en train de choisir...");
try {
    Thread.sleep(3000);
} catch (InterruptedException e) {
    e.printStackTrace();
}

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

Un arbre de possibilités a été créé pour déterminer le gagnant. J'ai pris comme référence "*if (playerItem == x)*" puis comparé avec "*botItem*" afin de définir le résultat : 1 = victoire, 0 = égalité, -1 = défaite.

```
if (playerItem == 'X') {
    switch (botItem) {
        case 'X':
            result = 0;
            break;
        case '_':
            result = -1;
            break;
        case '<':
            result = 1;
            break;
        default:
            break;
    }
} else if (playerItem == '_') {
    switch (botItem) {
        case 'X':
            result = 1;
            break;
        case '_':
            result = 0;
            break;
        case '<':
            result = -1;
            break;
        default:
            break;
    }
} else {
    switch (botItem) {
        case 'X':
            result = -1;
            break;
        case '_':
            result = 1;
            break;
        case '<':
            result = 0;
            break;
        default:
            break;
    }
}
```

Un message affiche ensuite le duel des objets.

Puis, avec les valeurs déterminées, on affiche victoire, défaite ou égalité du round, en ajoutant 1 point au gagnant (aucun point en cas d'égalité).

Ensuite, on réinitialise les objets des participants.

```
System.out.println(playerItem + " vs " + botItem);

if (result == 1) {
    playerPoint = playerPoint + 1;
    System.out.println("Vous avez gagné ! (" + playerPoint + "/" + botPoint + ")");
} else if (result == -1) {
    botPoint = botPoint + 1;
    System.out.println("Vous avez perdu... (" + playerPoint + "/" + botPoint + ")");
} else {
    System.out.println("Égalité ! (" + playerPoint + "/" + botPoint + ")");
}

//Etape 6
playerItem = '0';
botItem = '0';

}
//
```

Une comparaison au début vérifie si le nombre de rounds est atteint par l'un des joueurs. Cela permet de savoir si le match doit continuer ou s'arrêter.

Après cela, on détermine le vainqueur en vérifiant s'il a atteint le nombre de points correspondant aux rounds.

On réinitialise ensuite les rounds, les points et la variable *"restartChoice"*.

On demande au joueur s'il veut recommencer :

-S'il répond Oui, le flag restart est mis à true et le programme recommence.

-Sinon, un message "Merci d'avoir joué !" s'affiche et le programme s'arrête.

```

if (playerPoint == round) {
    System.out.println("Victoire du joueur !");
    System.out.println("Vous avez gagné ! Félicitation !");
} else if (botPoint == round) {
    System.out.println("Victoire de l'ordinateur !");
    System.out.println("Vous avez perdu... Dommage...");
}

round = 0;
playerPoint = 0;
botPoint = 0;
restartChoice = '0';

while (restartChoice != '0' && restartChoice != 'o' && restartChoice != 'N' && restartChoice != 'n') {
    restartChoice = '0';
    System.out.println("Souhaitez-vous rejouer ? (O/N)");

    restartChoice = sc.next().charAt(0);

    if (restartChoice == '0' || restartChoice == 'o') {
        restart = true;
    } else if (restartChoice == 'N' || restartChoice == 'n') {
        System.out.println("Merci d'avoir joué !");
        System.exit(0);
    }
}

```

4. Test

Nous avons testé chaque ligne et étape du code avant de passer à la suite, afin de pouvoir identifier les erreurs dès le début.

5. Retour d'expérience

C'est notre premier programme concret en Java (une vraie interface utilisateur avec un objectif).