

II.1102 – Algorithmique et Programmation

TP noté 1

Patrick Wang

Vendredi 11 octobre 2019

Table des matières

1	Consignes générales	1
2	Exercices	2
2.1	Décodeur de message	2
2.1.1	Encodage de caractères en Java	2
2.1.2	Présentation de l'exercice	2
2.1.3	Questions	3
2.2	Puissance 4	3
2.2.1	Description du jeu	3
2.2.2	Description du programme de départ	4
2.2.3	Logique du jeu	4
2.2.4	Questions	5

1 Consignes générales

- Aucun document n'est autorisé.
- Ce TP est à réaliser **individuellement**.
- Prenez garde à bien nommer votre TP en précisant votre groupe d'APP, prénom et nom : `TP1_G1_PRENOM_NOM`.
- L'ensemble du TP sera à réaliser à l'intérieur d'une seule et unique classe `Main`. Merci de bien respecter ce nom lors de la création de votre projet.
- Une boîte de dépôt est créée sur Moodle afin de déposer vos projets. Celle-ci se fermera 15min après la fin du temps réglementaire.
- Pour déposer votre projet, créez une archive au format ZIP reprenant le nom de votre projet. Les formats 7z ou RAR ne seront pas acceptés.
- Prenez garde à bien lire l'énoncé dans son ensemble avant de démarrer les exercices. Un fichier de départ vous est fourni pour ce TP et cet énoncé décrit avec détails le contenu de ce fichier.
- Ne copiez/collez pas bêtement l'ensemble du fichier `Main.java` qui vous est fourni. Des erreurs d'importation de *package* pourront vous être retournées.
- Toute tentative de triche sera sanctionnée d'un 0.
- Un point sur 20 est attribué à la lisibilité de votre code. Cela concerne l'indentation de votre code, le choix des noms de variables, et la présence de commentaires pertinents.

2 Exercices

2.1 Décodeur de message

2.1.1 Encodage de caractères en Java

Le système d'encodage des caractères utilisé en Java permet de convertir simplement des variables `char` en `int`. La conversion se fait de la manière suivante pour les lettres en minuscule :

- `a` → 97;
- `b` → 98;
- `c` → 99;
- `d` → 100;
- ...
- `w` → 119;
- `x` → 120;
- `y` → 121;
- `z` → 122.

En Java, les variables de type `char` sont équivalentes à des variables de type `int`. Il est donc possible de passer facilement d'un type à l'autre. Les instructions suivantes vous indiquent comment réaliser ces conversions :

```
char a = 'a';
int aEntier = a; // aEntier <- 97
int wEntier = 119;
char w = (char) wEntier; // On force le type en un char
```

2.1.2 Présentation de l'exercice

Pour discuter avec vos amis, vous avez décidé d'utiliser un langage secret. Ce langage repose sur un algorithme pour chiffrer et déchiffrer les messages. L'un de vos amis a écrit un programme pour encoder les messages mais a du mal à concevoir celui pour les décoder.

Votre rôle est donc de concevoir l'algorithme de décodage permettant de déchiffrer les messages reçus. Cet algorithme va respecter les règles suivantes :

- Si la lettre est représentée par un entier pair, alors on la remplace par la cinquième lettre située à sa droite;
- Si la lettre est représentée par un entier impair, alors on la remplace par la troisième lettre située à sa gauche;
- Les caractères `<espace>` restent inchangés.

Remarque 1 : On a donc les exemples suivants, où l'on peut noter le passage du `a` au `x` et celui du `z` au `e` :

- La lettre `a` (97) devient la lettre `x` (120);
- La lettre `b` (98) devient la lettre `g` (103);
- La lettre `y` (121) devient la lettre `v` (119);
- La lettre `z` (122) devient la lettre `e` (101).

Remarque 2 : On fera attention à ne pas utiliser d'accent, de lettre majuscule, ni de symbole de ponctuation lorsque l'on testera cet exercice. Les seuls symboles tolérés sont ceux des lettres minuscules sans accent et des espaces.

2.1.3 Questions

Le code fourni déclare la fonction `decodeurMessage(String message)`. Complétez cette fonction afin de décoder le `message` fourni en paramètre puis de retourner le message décodé. Pour cet exercice, vous pourrez avoir besoin d'utiliser les instructions suivantes :

```
String message = "un message";  
int tailleMessage = message.length(); // tailleMessage <- 10  
char unCarctere = message.charAt(4); // unCaractere <- 'e'
```

Remarque : La classe `Main` déclare deux variables : `messagesChiffres` et `messagesDechiffres` qui vous permettront de vérifier que votre algorithme est correct. Par exemple, le premier élément de `messagesChiffres` correspond au premier élément de `messagesDechiffres`.

2.2 Puissance 4

2.2.1 Description du jeu

D'après Wikipédia, le Puissance 4 est un jeu dont "le but est d'aligner une suite de 4 pions de même couleur sur un grille comptant 6 rangées et 7 colonnes".

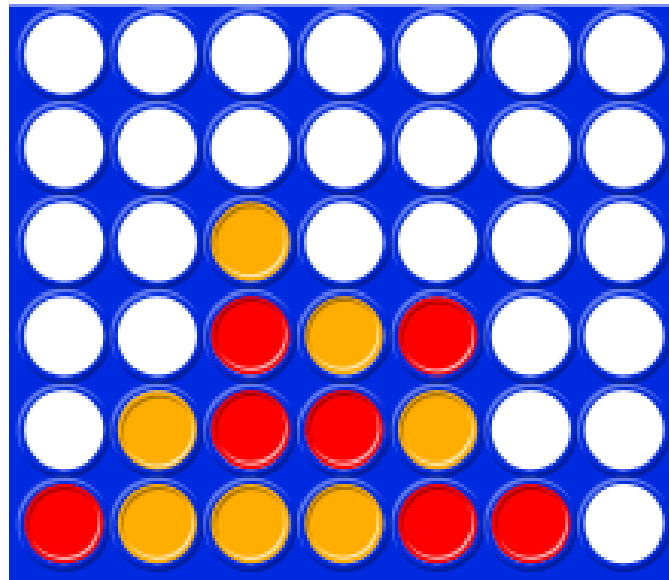


FIGURE 1 – Illustration du jeu du Puissance 4.

Dans cet exercice, nous allons implémenter une série de fonctions qui nous permettront de programmer le jeu du Puissance 4 dans la console Java. Les règles utilisées sont les règles du jeu standard :

- Le plateau utilisé est un plateau de 6 lignes et 7 colonnes ;
- Plutôt que d'utiliser des couleurs, nous utiliserons des 0 pour le Joueur 1 et des X pour le Joueur 2 ;
- Le premier joueur à aligner quatre pions gagne la partie ;
- La partie est déclarée nulle si le plateau est rempli sans qu'aucun alignement de quatre pions n'ait été créé.

2.2.2 Description du programme de départ

Le programme vous est fourni avec quelques éléments de départ. **Prenez bien le temps de lire ce qui suit pour comprendre l'architecture du programme.**

On note dans un premier temps la déclaration de variables dans le `Main` :

- `char[] [] plateau` qui représente le plateau 6×7 ;
- `char joueur1, joueur2` qui stockent les symboles de `O` et `X` des deux joueurs ;
- `boolean estJoueur1` qui vaut `true` si c'est au Joueur 1 de jouer ;
- `Scanner scanner` qui vous permettra de récupérer des saisies d'utilisateur sans avoir à créer une nouvelle variable `Scanner` à chaque fois.

On note dans un second temps la déclaration de plusieurs fonctions :

- `void puissance4()` qui est la fonction principale qui va implémenter la logique du jeu et appeler un bon nombre des autres fonctions ;
- `void initialiserPlateau()` qui est la fonction qui va initialiser le contenu de `plateau` avec des caractères `<espace>` ;
- `void afficherPlateau()` qui est la fonction qui va afficher le contenu du plateau. Un cadre délimitant le plateau est aussi affiché ;
- `int saisieJoueur()` qui demande au joueur en cours de saisir un entier puis le renvoie. Cet entier correspond à l'indice de colonne ;
- `boolean saisieValide(int indiceColonne)` qui vérifie que la saisie de l'utilisateur est valide et que la colonne peut contenir un pion supplémentaire. Cette fonction renvoie `true` si la saisie est valide, `false` sinon.
- `void updatePlateau(int saisie)` qui est la fonction qui va mettre à jour le contenu du tableau avec la `saisie` du joueur en cours ;
- `boolean finDuJeu()` qui renvoie `true` si le jeu est terminé, `false` sinon. Le jeu peut se terminer de deux façons : soit le plateau est rempli, soit un joueur a créé un alignement de quatre pions sur le plateau. Il y a trois types d'alignements possibles : un alignement sur une ligne, un alignement sur une colonne, ou un alignement sur une diagonale ;
- `boolean plateauEstComplet()` qui renvoie `true` si le plateau est complet, `false` sinon ;
- `boolean alignementGagnant()` qui renvoie `true` si un joueur a créé un alignement gagnant sur une ligne, une colonne, ou une diagonale.

2.2.3 Logique du jeu

L'implémentation du jeu suit la logique suivante :

1. On affiche le menu du jeu ;
2. Si les joueurs veulent débiter une partie :
 - (a) On initialise le plateau du jeu ;
 - (b) On affiche le plateau du jeu ;
 - (c) On demande au joueur de saisir son coup (cette saisie doit être validée) ;
 - (d) On met à jour le plateau du jeu avec cette nouvelle saisie ;
 - (e) On vérifie si le jeu est fini ou non :
 - i. Si le jeu est fini, c'est la fin de la partie ;
 - ii. Sinon, on passe la main à l'autre joueur.
 - (f) On recommence à partir de l'étape (b).
3. On recommence à partir de l'étape 1.

2.2.4 Questions

Complétez le programme fourni pour implémenter le jeu du Puissance 4. Appuyez vous sur la description qui a été faite en Section 2.2.2 et en Section 2.2.3. Appuyez-vous également sur les commentaires présents dans le programme de départ.