

Projet « Jeu de morpion » 2013

Le but de ce projet découpé sous forme de plusieurs exercices à trous est de programmer un jeu de morpion.

Chacun des exercices devra être réalisé de manière individuelle.

Les trous à remplir dans un questionnaire en ligne seront référencés par un numéro souligné, sous la forme suivante : {-1-}.

Quand ce numéro suit les doubles slash de commentaires (sous la forme // {-1-}), cela signifie qu'il faut commenter le code qui suit, c'est-à-dire donner une brève explication de ce qui est réalisé par l'algorithme.

Règles du jeu

Le jeu se joue à deux joueurs. Le plateau de jeu est un carré de 9 cases, réparties en trois lignes et trois colonnes. A tour de rôle, chaque joueur pose un objet dans une case vide : un rond pour le premier joueur, une croix pour le second joueur. Le premier joueur qui parvient à poser trois objets alignés (verticalement, horizontalement ou en diagonale) a gagné.

Exercice 1

La copie d'écran ci-contre montre l'exécution d'une implémentation du jeu du morpion. On va dans cet exercice séparer différentes étapes du jeu, afin de définir les diverses fonctions qui seront programmées dans les exercices suivants, et appelées tout au long de l'algorithme.

```
C:\Windows\system32\cmd.exe - java Morpion
C:\>java Morpion
Voici le plateau de jeu :
+--+--+
| | |
+--+--+
| | |
+--+--+
| | |
+--+--+
Joueur 1, quelle case souhaitez-vous remplir ?
+--+--+
|0|1|2|
+--+--+
|3|4|5|
+--+--+
|6|7|8|
+--+--+
4
Voici le plateau de jeu :
+--+--+
| | |x|
+--+--+
| |o| |
+--+--+
| | |
+--+--+
Joueur 2, quelle case souhaitez-vous remplir ?
+--+--+
|0|1|2|
+--+--+
|3|o|5|
+--+--+
|6|7|8|
+--+--+
2
Voici le plateau de jeu :
+--+--+
| | |x|
+--+--+
|3|o|5|
+--+--+
|6|7|8|
+--+--+
Joueur 1, quelle case souhaitez-vous remplir ?
```

Le plateau de jeu est représenté par un tableau d'entiers de 9 cases, qui contient un 0 si la case correspondante du plateau est vide, un 1 si la case correspondante du plateau contient un rond, et un 2 si elle contient une croix. La numérotation des cases du plateau est indiquée dans la figure ci-dessus : ainsi la case en haut à gauche est la case numéro 0 (en Java) du tableau d'entiers représentant le plateau, la case en haut au milieu est la case numéro 1, etc.

La fonction **FaitJouer** prend en entrée le numéro du joueur (1 ou 2) ainsi que le tableau représentant le plateau de jeu, demande au joueur concerné la case sur laquelle il veut jouer, et renvoie le numéro de case choisie.

La fonction **AffichePlateau** prend en entrée le tableau représentant le plateau de jeu, ainsi qu'une variable d'entrée *aRemplir* et elle affiche le tableau de jeu. Si cette variable *aRemplir* vaut *false*, on affichera le numéro de chaque case qui n'est pas encore remplie (comme dans la deuxième et la quatrième case où le plateau s'affiche, dans la copie d'écran ci-dessus). Si elle vaut vrai, en revanche, on affichera une case vide (comme dans le premier et le troisième affichage du plateau de jeu, dans la copie d'écran ci-dessus).

La fonction **AfficheLigne** prend en entrée le tableau représentant le plateau de jeu, ainsi qu'une variable *numLigne* qui indique le numéro de ligne à afficher, et une variable *aRemplir* qui indique comme précédemment comment traiter les cases vides. Cette fonction affiche uniquement une ligne du plateau de jeu. Par exemple, l'appel de fonction servant à afficher la première ligne du premier affichage du plateau, dans la copie d'écran ci-dessus, va afficher :

```
|||
```

et l'appel de fonction affichant la première ligne du deuxième affichage du plateau va afficher :

```
|0|1|2|
```

Précisons que cette fonction ne renvoie rien.

Enfin, la fonction **VerificationJeuEnCours** prend en entrée le tableau représentant le plateau de jeu, et renvoie *true* si le jeu continue, et *false* sinon. Dans ce dernier cas, elle informera le joueur concerné de sa victoire, ou du match nul.

Dans ce premier exercice, il faut compléter les trous dans les déclarations de ces fonctions :

```
//demande au joueur joueur la case qu'il choisit et renvoie le resultat
public static     {-1-}     FaitJouer (    {-2-}    ) {
    //...
}
```

Les ... représentent le code qui sera ajouté dans les prochains exercices, on ne vous demande pas de remplir le code à l'intérieur de chaque fonction, pour le moment, juste d'écrire les bonnes déclarations et commentaires dans les trous.

```

//renvoie true si le jeu continue, affiche les messages de fin de jeu
//et renvoie false sinon.
public static _____{-3-} VerificationJeuEnCours(_____{-4-}){
    //...
}

//_____{-5-}
public static _____{-6-} AffichePlateau(_____{-7-}){
    //...
}

//_____{-8-}
public static _____{-9-} AfficheLigne(_____{-10-}){
    //...
}

```

Attention ! Les trous ne contiennent pas qu'un seul mot. Il n'y a aucune limite sur le nombre de caractères à utiliser pour les remplir.

Adresse du questionnaire à remplir avant le 9 février 23h59 (de préférence avant le 8 février) :
<https://docs.google.com/spreadsheets/viewform?formkey=dHFoZUYyZUJCSkg0eVNrWEczWndFRnc6MQ>

Exercice 2

L'objectif de ce deuxième exercice est d'obtenir un code qui compile correctement, enregistré sous le nom de fichier *Morpion.java*, de créer le tableau correspondant au plateau de jeu, entièrement vide au début du jeu, et de créer la fonction qui affiche le plateau de jeu (mais sans écrire encore celle qui affiche une ligne du plateau de jeu, qui sera complétée plus tard).

Peu importe la valeur renvoyée par les fonctions **VerificationJeuEnCours** et **FaitJouer**, on souhaite juste que le programme compile correctement.

```

import java.io.*;
public class _____{-1-}{
    public static void main(String[] arg){
        //creation du tableau plateau
        _____{-2-}
        //affichage du plateau de jeu avec des cases vides (sans numéros)
        _____{-3-}
    }

    //renvoie true si le jeu continue, affiche les messages de fin de jeu
    //et renvoie false sinon.
    public static boolean VerificationJeuEnCours(int[] plateau){
        return _____{-4-} ;
    }

    //demande au joueur joueur la case qu'il choisit et renvoie le resultat
    public static int FaitJouer(int joueur,int[] plateau){
        return _____{-5-} ;
    }

    //affiche le plateau de jeu, en indiquant le numero des cases vides
    //si aRemplir est fixe a false.
    public static void AffichePlateau(int[] plateau,boolean aRemplir){
        _____{-6-}
        System.out.println("+-+--+");
        while(_____{-7-}){
            AfficheLigne(_____{-8-});
            System.out.println("_____{-9-}");
            _____{-10-};
        }
    }

    //affiche chacune des trois colonnes de la ligne ligne du tableau plateau
    public static void AfficheLigne(int[] plateau,int numLigne,boolean aRemplir){
    }
}

```

Adresse du questionnaire à remplir avant le 14 février 23h59 :
<https://docs.google.com/forms/d/1hbV5oM6Xv-Cvi3xW7TC23zIacRAAzjO8gMOBb0d6O3k/viewform>

Exercice 3

L'objectif de ce troisième exercice est de commencer à écrire la fonction `AfficheLigne` qui affiche une ligne du plateau de jeu. Pour l'instant, on ne préoccupera pas de la valeur du booléen `aRemplir` fourni en entrée de la fonction, et on fera comme s'il valait `true`, c'est-à-dire qu'on n'affichera pas le numéro des cases vides.

```
import java.io.*;
public class Morpion{
    public static void main(String[] arg){
        //creation du tableau plateau
        int[] plateau = {0,0,0,0,0,0,0,0,0};
        //affichage du plateau de jeu avec des cases vides (sans numéros)
        AffichePlateau(plateau,true);
    }

    //renvoie true si le jeu continue, affiche les messages de fin de jeu
    //et renvoie false sinon.
    public static boolean VerificationJeuEnCours(int[] plateau){
        return true;
    }

    //demande au joueur joueur la case qu'il choisit et renvoie le resultat
    public static int FaitJouer(int joueur,int[] plateau){
        return 1 ;
    }

    //affiche le plateau de jeu, en indiquant le numero des cases vides
    //si aRemplir est fixe a false.
    public static void AffichePlateau(int[] plateau,boolean aRemplir){
        int i=0;
        System.out.println("+--+--+");
        while(i<3){
            AfficheLigne(plateau,i,aRemplir);
            System.out.println("+--+--+");
            i++;
        }
    }

    //affiche chacune des trois colonnes de la ligne ligne du tableau plateau
    public static void AfficheLigne(int[] plateau,int numLigne,boolean aRemplir){
        //Création de la chaîne de caractères à afficher :
        String chaine _____{-1-}_____;

        //Création du compteur :
        _____{-2-}_____;

        //Affichage de chacune des trois colonnes de la ligne du plateau :
        while(_____{-3-}_____) {
            if(_____{-4-}_____==0){
                chaine=chaine+" ";
            }
            if(_____{-5-}_____==1){
                chaine=chaine+"o";
            }
            if(_____{-6-}_____==2){
                chaine=chaine_____{-7-}_____;
            }
            chaine=chaine+_____{-8-}_____;
            _____{-9-}_____
        }
        System.out.println(_____{-10-}_____);
    }
}
```

Adresse du questionnaire à remplir avant le 26 février 23h59 :

https://docs.google.com/forms/d/16fqZ5klCWZQXIDuqwGsAU6Hk_MfPdsoRPI5F2da289U/viewform

Proposition de solution pour l'exercice 3

```
//affiche chacune des trois colonnes de la ligne ligne du tableau plateau
public static void AfficheLigne(int[] plateau,int numLigne,boolean aRemplir){
    //Création de la chaîne de caractères à afficher :
    String chaine = "|";

    //Création du compteur :
    int i=0;

    //Affichage de chacune des trois colonnes de la ligne du plateau :
    while(i<3){
        if(plateau[3*numLigne+i]==0){
            chaine=chaine+" ";
        }
        if(plateau[3*numLigne+i]==1){
            chaine=chaine+"o";
        }
        if(plateau[3*numLigne+i]==2){
            chaine=chaine+"x";
        }
        chaine=chaine+"|";
        i++;
    }
    System.out.println(chaine);
}
```

Exercice 4

L'objectif de ce quatrième exercice est d'écrire la fonction `FaitJouer` qui demande au joueur dont le numéro `joueur` est fourni en entrée, la case du plateau `plateau` (également fourni en entrée) qu'il choisit, et renvoie le numéro de la case. Le joueur est relancé tant qu'il donne un numéro invalide.

Soyez particulièrement attentifs à ne rien oublier, ou mettre en trop, dans les trous (par exemple, les « ; » ou « (» ou «) »).

```
public static int FaitJouer(int joueur,int[] plateau){
    //question au joueur sur le numero de case qu'il veut remplir
    System.out.println("Joueur "+joueur+____{-1-}____

    //affichage du plateau avec numeros de cases
    AffichePlateau(____{-2-}____);

    ____{-3-}____;
    int caseChoisie=0;

    while (reponseDonnee==false){
        //choix d'une case par le joueur
        ____{-4-}____
        caseChoisie = lecture.nextInt();
        System.out.println("");
        if (caseChoisie____{-5-}____){
            //numero de case trop petit ou trop grand
            System.out.println("Numero de case invalide.");
            System.out.println("Merci de choisir un numero de case non remplie");
            System.out.println("entre 0 et 8.");
        } else {
            if (____{-6-}____){
                //case deja remplie
                System.out.println("____{-7-}____");
                System.out.println("Merci de choisir ____{-8-}____");
            } else {
                reponseDonnee=____{-9-}____;
            }
        }
    }

    return ____{-10-}____;
}
```

Adresse du questionnaire à remplir avant le 2 avril 23h59 :

https://docs.google.com/forms/d/1c9p9b2clkgF6T25S6y3bZjOpS29_pEaWDSZ2_V8KUuo/viewform

Proposition de solution pour l'exercice 4

```
public static int FaitJouer(int joueur,int[] plateau){
    //question au joueur sur le numero de case qu'il veut remplir
    System.out.println("Joueur "+joueur+", quelle case souhaitez-vous remplir ?");

    //affichage du plateau avec numeros de cases
    AffichePlateau(plateau,true);

    boolean reponseDonnee=false;
    int caseChoisie=0;

    while (reponseDonnee==false){
        //choix d'une case par le joueur
        Scanner lecture = new Scanner(System.in);
        caseChoisie = lecture.nextInt();
        System.out.println("");
        if (caseChoisie<0||caseChoisie>8){
            //numero de case trop petit ou trop grand
            System.out.println("Numero de case invalide.");
            System.out.println("Merci de choisir un numero de case non remplie");
            System.out.println("entre 0 et 8.");
        } else {
            if (plateau[caseChoisie]>0){
                //case deja remplie
                System.out.println("Case deja remplie.");
                System.out.println("Merci de choisir un numero de case pas encore remplie");
            } else {
                reponseDonnee=true;
            }
        }
    }

    return caseChoisie;
}
```

Exercice 5

L'objectif de ce cinquième exercice est de commencer à écrire la fonction `VerificationJeuEnCours` de manière à détecter s'il reste encore des cases vides dans le plateau de jeu. Ainsi, cette fonction renverra `true` s'il reste encore des cases vides et `false` sinon. **Vous penserez à bien commenter votre code (c'est-à-dire insérer des lignes précédées par // où vous décrivez les opérations que l'algorithme effectue).**

Nous modifierons le code de cette fonction dans le prochain exercice pour prendre en compte le cas où un des deux joueurs a gagné.

```
//renvoie true si le jeu continue, affiche les messages de fin de jeu
//et renvoie false sinon.
public static boolean VerificationJeuEnCours(int[] plateau){
    _____{-1-}_____
}
```

Adresse du questionnaire à remplir avant le 9 avril 23h59 :

<https://docs.google.com/forms/d/1s4RHijEhyscYamQBkn92eC7sDvxyvu7uHkXzRPyKGAA/viewform>

Proposition de solution pour l'exercice 5

```
//renvoie true si le jeu continue, affiche les messages de fin de jeu
//et renvoie false sinon.
public static boolean VerificationJeuEnCours(int[] plateau){
    boolean resultat=true;

    //comptage des cases vides
    int nbCasesVides=0;
    int i=0;
    while(i<plateau.length){
        if(plateau[i]==0){
            nbCasesVides=nbCasesVides+1;
        }
        i++;
    }

    //verification qu'il reste encore des cases vides
    if(nbCasesVides<=0){
        resultat=false;
    }
    return resultat;
}
```

Exercice 6

L'objectif de ce cinquième exercice est de finir d'écrire la fonction `VerificationJeuEnCours` de manière à détecter si un des deux joueurs a gagné. Pour cela, on va tester, pour chaque alignement possible (horizontal, vertical, diagonal), si les trois cases de l'alignement ont la même valeur. Si c'est le cas, on affichera le numéro du joueur qui a gagné et on renverra `false`. Si ce n'est pas le cas, on affiche qu'il y a eu match nul et on renvoie `false` s'il n'y a plus de case vide. S'il reste des cases vides, on renvoie `true`.

Vous penserez à bien commenter votre code (c'est-à-dire insérer des lignes précédées par `//` où vous décrivez les opérations que l'algorithme effectue).

Il est très conseillé de commencer à réunir tous les précédents extraits de codes afin de créer un fichier compilable et testable.

Les trous de cet exercice contiendront probablement plusieurs lignes

```
//renvoie true si le jeu continue, affiche les messages de fin de jeu
//et renvoie false sinon.
public static boolean VerificationJeuEnCours(int[] plateau){

    //comptage des cases vides
    int nbCasesVides=0;
    int i=0;
    while(i<plateau.length){
        if(plateau[i]==0){
            nbCasesVides=nbCasesVides+1;
        }
        i++;
    }

    //Debut de la detection de la victoire d'un des deux joueurs
    boolean resultat=false;
    int numeroDuGagnant=0;

    //detection des 3 alignements horizontaux
    _____
    {-1-}

    //detection des 3 alignements verticaux
    _____
    {-2-}

    //detection des 2 alignements diagonaux
    _____
    {-3-}

    //message en cas de victoire d'un des deux joueurs :
    _____
    {-4-}

    if(numeroDuGagnant==0){
        //aucun joueur ne gagne
        if(nbCasesVides<=0){
            //message en cas de match nul :
            _____
            {-5-}
        } else {
            //le jeu peut continuer :
            _____
            {-6-}
        }
    }
    return resultat;
}
```

Adresse du questionnaire à remplir avant le 14 mai 23h59 :

<https://docs.google.com/forms/d/1QWA-cxgCuZo3VY7AB49SH7LjeyL9pYIkdzYUPCc5Bvo/viewform>