



Suite TD2 Partie 1

Pré-requis

Programme A2 (récursivité, matrice, lecture de fichiers)

Exercice 1 : Classe Joueur

Un joueur est caractérisé par son nom, son score et par les mots trouvés au cours de la partie

La création d'un joueur n'est possible que si celui-ci a un nom.

Vous créerez les propriétés en fonction des besoins de votre programme

Les méthodes avec les signatures suivantes sont imposées :

`public bool Contain(string mot)` qui teste si le mot passé appartient déjà aux mots trouvés par le joueur pendant la partie

`public void Add_Mot (string mot)` ajoute le mot dans la liste des mots déjà trouvés par le joueur au cours de la partie

`public string toString()` qui retourne une chaîne de caractères qui décrit un joueur.

Exercice 2 : Classe Dé

Un dé est caractérisé par un ensemble de lettres et une lettre tirée au hasard parmi ces 6 lettres pour représenter la face visible du plateau

La création d'un dé n'est possible que si les 6 valeurs des faces sont fournies

Vous créerez les propriétés en fonction des besoins de votre programme

Les méthodes avec les signatures suivantes sont imposées :

`public void Lance(Random r)` : cette méthode permet de tirer au hasard une lettre parmi les 6

`public string toString()` qui retourne une chaîne de caractères qui décrit un dé.

Exercice 3 : Classe Plateau

Une instance de plateau est définie, en particulier, par les 16 dés (tableau de dés) et leur valeur supérieure. Un fichier de 16 dés est joint au sujet.

La construction de l'instance de plateau passera par la lecture du fichier de dés. (fichier ci-joint)



Vous créez les propriétés en fonction des besoins de votre programme.

Les méthodes avec les signatures suivantes sont imposées :

`string toString()` qui retourne une chaîne de caractères qui décrit un plateau.

`public bool Test_Plateau(char[] mot)` qui teste si le mot passé en paramètre est un mot éligible c'est-à-dire qu'il respecte la contrainte d'adjacence décrite ci-dessus.

Pour cela on vous demande de réfléchir à un algorithme récursif si possible.

Exercice 4 : Classe Dictionnaire (supplémentaire)

Une instance de dictionnaire va contenir l'ensemble des mots du fichier des mots. Vous réfléchirez à une structure de données qui permet de conserver la structure du fichier de mots, à savoir conserver les mots par groupe en fonction du nombre de lettres.

Vous créez les propriétés en fonction des besoins de votre programme

Les méthodes avec les signatures suivantes sont imposées :

`string toString()` qui retourne une chaîne de caractères qui décrit un dictionnaire.

`public bool RechDichoRécursif(int debut, int fin, string mot)` qui teste que le mot appartient bien au tableau de mots de ce dictionnaire

Exercice 5 : Classe Jeu (version 1 sans la classe dictionnaire)

La classe Jeu possède a minima 2 attributs :

- Un plateau
- Un joueur courant

Le programme principal Main va donc à tour de rôle donner la main à un joueur puis à un autre (dans notre cas 2 joueurs suffisent)

Le joueur saisit un mot. Après chaque saisie, vous testez si ce mot est éligible à savoir si la longueur du mot ≥ 3 et s'il respecte les contraintes d'adjacence

A la fin du nombre de tours (10 par exemple), vous afficherez à la fin de la partie les scores des joueurs et indiquerez le gagnant.



Exercice 6 : Classe Jeu (version 2 avec la classe dictionnaire et une gestion des joueurs différente) (supplémentaire)

La classe Jeu possède a minima 2 attributs :

- Un dictionnaire : Le dictionnaire sera instancié par la lecture du fichier donné en pièce jointe : Motspossibles.txt. Prenez le temps d'analyser la construction de ce fichier
- Un plateau : Le plateau sera instancié par la lecture du fichier donné en pièce jointe : Des.txt

Le programme principal Main va donc à tour de rôle donner la main à un joueur puis à un autre (dans notre cas 2 joueurs suffisent) pour un laps de temps à définir (6mn par exemple)

Chaque joueur a une minute (voir la classe DateTime et TimeSpan) pour trouver les mots sur une nouvelle instance de plateau.

Le joueur saisit un mot après l'autre. Après chaque saisie, vous testez si ce mot est éligible à savoir s'il appartient au dictionnaire (algorithme récursif demandé), si la longueur du mot ≥ 3 et s'il respecte les contraintes d'adjacence

A la fin du temps imparti, l'autre joueur joue et ce qu'aux 6 minutes de la partie.

Vous afficherez à la fin de la partie les scores des joueurs et indiquerez le gagnant.