



Préambule

Ce travail est à réaliser seul

Il est à rendre au plus tard le lundi 7 janvier à 16h

Vous déposerez votre solution complète Visual Studio sur Devinci-online au format **.zip** sous votre nom.

Un programme ne compilant pas ou ne s'exécutant pas entraîne une note de 0/20.

L'ensemble des codes sera analysé par un système anti-plagiat. Un plagiat entraîne une note de 0/20 au module (pour les 2 protagonistes)

Le sujet proposé a pour objectif de mettre en application tous les concepts vus en TD, c'est pourquoi il est impératif de suivre les énoncés tels qu'ils ont été écrits.

Recommandations générales

Ne pas oublier votre projet Test Unitaire sur au moins le test de 5 fonctions.

Ne pas oublier les commentaires

Ne pas oublier d'écrire les variables et fonctions avec des noms lisibles

Ne pas oublier l'indentation.

Un ensemble de méthodes sont imposées avec des signatures précises.

Vous pouvez rajouter d'autres méthodes si les besoins de votre code l'imposent

Ne pas oublier que l'utilisation de

```
Random r = new Random()
```

ne peut se faire qu'une seule fois. Ensuite `r.Next(..)` peut se faire autant de fois que nécessaire

Présentation du problème

Le jeu commence par le mélange d'un plateau (carré) de 16 dés à 6 faces. Chaque dé possède une lettre différente sur chacune de ses faces. Les dés sont lancés sur le plateau 4 par 4, et seule leur face supérieure est visible. Vous traduirez le lancement de dé par un tirage aléatoire d'une face parmi les 6 d'un dé et ce pour tous les dés. Après cette opération, un compte à rebours de N minutes est lancé qui établit la durée de la partie.

Chaque joueur joue l'un après l'autre pendant un laps de temps de 1 mn.

Chaque joueur cherche des mots pouvant être formés à partir de lettres adjacentes du plateau. Par adjacentes, il est sous-entendu horizontalement, verticalement ou en diagonale. Les mots doivent



être de 3 lettres au minimum, peuvent être au singulier ou au pluriel, conjugués ou non, mais ne doivent pas utiliser plusieurs fois le même dé pour le même mot. Les joueurs saisissent tous les mots qu'ils ont trouvés au clavier. Un score par joueur est mis à jour à chaque mot trouvé et validé.



Le calcul de points se fait de la manière suivante : Un mot n'est accepté qu'une fois au cours du jeu par joueur.

En fonction de la taille du mot les points suivants sont octroyés.

Taille du mot	3	4	5	6	7+
Points	2	3	4	5	11

Votre programme commencera par lire un fichier indiquant pour chaque dé quelles sont les lettres qui sont inscrites sur ses faces. Ensuite, le programme réalisera un lancer des dés et positionnera ce jet sur le plateau de jeu. Le joueur devra saisir les mots qu'il trouve (un mot à la fois !). A chaque saisie, votre programme vérifie que ce mot respecte la contrainte de longueur (au moins 3 caractères de long), que le mot n'a pas encore été proposé (un mot n'est comptabilisé qu'une seule fois, même s'il apparaît plusieurs fois sur le plateau et au fil du jeu), que le mot appartient bien au dictionnaire de mots connus et bien entendu qu'il est possible de former ce mot à partir des faces visibles du plateau. Si tous ces tests sont valides alors le mot sera ajouté à la liste de mots trouvés par le joueur et le score du joueur sera crédité des points correspondants. C'est alors au tour du joueur suivant de jouer.

Voici un extrait du dialogue du jeu dans une interface minimale



```
C'est au tour de DURAND de jouer

O W S I
N V B E
K S E I
A J Y I

Saisissez un nouveau mot trouvé
JASE
Le score de DURAND est de 2 grâce aux mots cités suivants
JASE
Saisissez un nouveau mot trouvé

C'est au tour de DUPOND de jouer

N N E E
B A I R
H I E A
M L S E

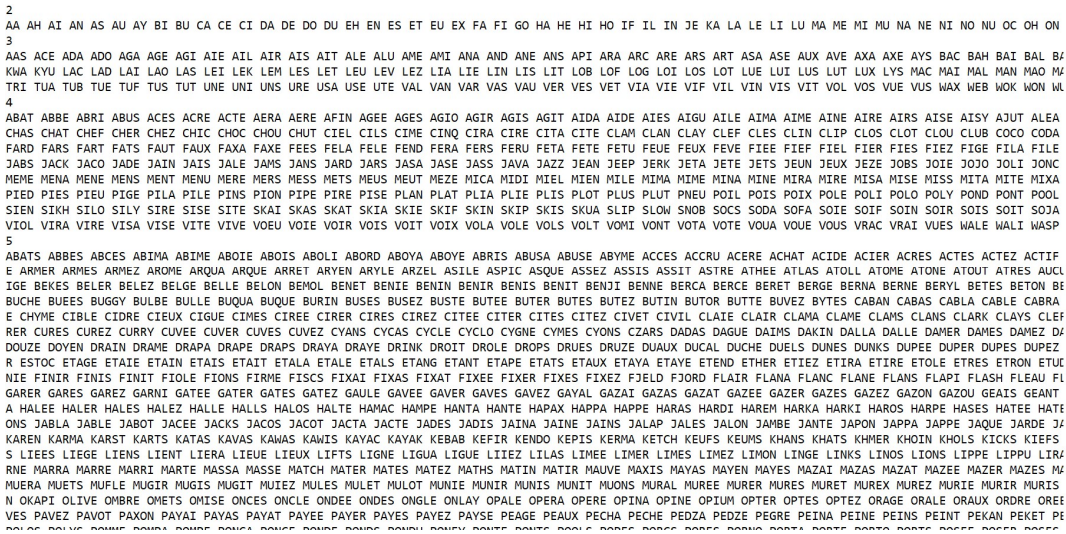
Saisissez un nouveau mot trouvé
RIEN
Le score de DUPOND est de 2 grâce aux mots cités suivants
RIEN
Saisissez un nouveau mot trouvé
LIEE
Le score de DUPOND est de 4 grâce aux mots cités suivants
RIEN LIEE
Saisissez un nouveau mot trouvé
BANIE
Saisissez un nouveau mot trouvé
AIR
Le score de DUPOND est de 5 grâce aux mots cités suivants
RIEN LIEE AIR
Saisissez un nouveau mot trouvé
RASE
Le score de DUPOND est de 7 grâce aux mots cités suivants
RIEN LIEE AIR RASE
Saisissez un nouveau mot trouvé
NEE
Le score de DUPOND est de 8 grâce aux mots cités suivants
RIEN LIEE AIR RASE NEE
C'est au tour de DURAND de jouer

W R A E
Z S N J
A S R A
R G O L

Saisissez un nouveau mot trouvé
```

Avec le tirage au sort des faces du dé, Dupond a joué, sans inspiration, un seul mot a été trouvé dans le temps imparti.

Durand a un tirage plus favorable et trouve 5 mots validés (BANIE n'est pas validé car n'appartient pas au dictionnaire dont un extrait est présenté ci-dessous).



Exercice 1 : Classe Joueur

Les méthodes avec les signatures suivantes sont imposées :

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

Exercice 2 : Classe Dé

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

Exercice 3 : Classe Dictionnaire

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 créerez les propriétés en fonction des besoins de votre programme

Les méthodes avec les signatures suivantes sont imposées :

`public override string ToString()` ou `public string toString()` qui retourne une chaîne de caractères qui décrit un dictionnaire.

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

Exercice 4 : Classe Plateau

Une instance de plateau est définie, en particulier, par les 16 dés (tableau de dés) et leur valeur supérieure

Vous créerez les propriétés en fonction des besoins de votre programme

Les méthodes avec les signatures suivantes sont imposées :

`public override string ToString()` ou `public 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.

Les étudiants qui résoudront ce problème par un algorithme récursif auront un bonus.

Exercice 5 : Classe Jeu (ou Program)

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.