Le jeu du Boogle

Préambule

Ce travail est à réaliser seul ou en binôme du même groupe

Il est à rendre au plus tard à la fin de votre dernière session

Vous déposerez votre solution Visual Studio sur DVO au format .zip sous votre ou vos noms

Attention à ne pas déposer une partie de votre solution, vérifiez que vous avez intégré tous les fichiers nécessaires.

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.

La dernière séance donnera lieu à une revue de code. Un étudiant doit être en mesure de décrire n'importe quelle partie de code (y compris une portion écrite par son binôme).

Je précise que la note du problème ne sera définitive qu'à l'issue de l'examen. En effet, tout écart trop important entre la note du problème et celle de l'examen donnera lieu à une khôlle individuelle en janvier.

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 pour faciliter une correction précise

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

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 adjacente, 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. Ce barème est une proposition, vous pouvez changer le poids en fonction de la taille des mots

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.

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).

```
2
AA AH AI AN AS AU AY BI BU CA CE CI DA DE DO DU EH EN ES ET EU EX FA FI GO HA HE HI HO IF IL IN JE KA LA LE LI LU MA ME MI MU NA NE NI NO NU OC OH ON
3
AAS ACE ADA ADO AGA AGE AGI AIE AIL AIR AIS AIT ALE ALU AME AMI ANA AND ANE ANS API ARA ARC ARE ARS ART ASA ASE AUX AVE AXA AXE AYS BAC BAH BAI BAL BA
KWA KYU LAC LAD LAI LAO LAS LEI LEK LEM LES LET LEU LEV LEZ LIA LIE LIN LIS LIT LOB LOF LOG LOI LOS LOT LUE LUI LUS LUT LUX LYS MAC MAI MAL MAN MAO MA
TRI TUA TUB TUE TUF TUS TUT UNE UNI UNS URE USA USE UTE VAL VAN VAR VAS VAU VER VES VET VIA VIE VIF VIL VIN VIS VIT VOL VOS VUE VUS WAX WEB WOK WON WU
4
ABAT ABBE ABRI ABUS ACES ACRE ACTE AERA AERE AFIN AGEE AGES AGIO AGIR AGIS AGIT AIDA AIDE AIES AIGU AILE AIMA AIME AINE AIRE AIRS AISE AISY AJUT ALEA
CHAS CHAT CHEF CHER CHEZ CHIC CHOC CHOU CHUT CIEL CILS CIME CINQ CIRA CIRE CITA CITE CLAM CLAN CLAY CLEF CLES CLIN CLIP CLOS CLOT CLOU CLUB COCO CODA
FARD FARS FART FATS FAUT FAUX FAXA FAXE FEES FELA FELE FEND FERA FERS FERU FETA FETE FETU FEUE FEUX FEVE FIEE FIEF FIEL FIER FIES FIEZ FIGE FILA FILE
JABS JACK JACO JADE JAIN JAIS JALE JAMS JANS JARD JARS JASA JASE JASS JAVA JAZZ JEAN JEEP JERK JETA JETE JETS JEUN JEUX JEZE JOBS JOIE JOJO JOLI JONC
MEME MENA MENE MENS MENT MENU MERE MERS MESS METS MEUS MEUT MEZE MICA MIDI MIEL MIEN MILE MIMA MIME MINA MINE MIRA MIRE MISA MISE MISS MITA MITE MIXA
PIED PIES PIEU PIGE PILA PILE PINS PION PIPE PIRE PISE PLAN PLAT PLIA PLIE PLIS PLOT PLUS PLUT PNEU POIL POIS POIX POLE POLI POLO POLY POND PONT POOL
SIEN SIKH SILO SILY SIRE SISE SITE SKAI SKAS SKAT SKIA SKIE SKIF SKIN SKIP SKIS SKUA SLIP SLOW SNOB SOCS SODA SOFA SOIE SOIF SOIN SOIR SOIS SOIT SOJA
VIOL VIRA VIRE VISA VISE VITE VIVE VOEU VOIE VOIR VOIS VOIT VOIX VOLA VOLE VOLS VOLT VOMI VONT VOTA VOTE VOUA VOUE VOUS VRAC VRAI VUES WALE WALI WASP
5
ABATS ABBES ABCES ABIMA ABIME ABOIE ABOIS ABOLI ABORD ABOYA ABOYE ABRIS ABUSA ABUSE ABYME ACCES ACCRU ACERE ACHAT ACIDE ACIER ACRES ACTES ACTEZ ACTIF
E ARMER ARMES ARMEZ AROME ARQUA ARQUE ARRET ARYEN ARYLE ARZEL ASILE ASPIC ASQUE ASSEZ ASSIS ASSIT ASTRE ATHEE ATLAS ATOLL ATOME ATONE ATOUT ATRES AUCU
IGE BEKES BELER BELEZ BELGE BELLE BELON BEMOL BENET BENIE BENIN BENIR BENIS BENIT BENJI BENNE BERCA BERCE BERET BERGE BERNA BERNE BERYL BETES BETON BE
BUCHE BUEES BUGGY BULBE BULLE BUQUA BUQUE BURIN BUSES BUSEZ BUSTE BUTEE BUTER BUTES BUTEZ BUTIN BUTOR BUTTE BUVEZ BYTES CABAN CABAS CABLA CABLE CABRA
E CHYME CIBLE CIDRE CIEUX CIGUE CIMES CIREE CIRER CIRES CIREZ CITEE CITER CITES CITEZ CIVET CIVIL CLAIE CLAIR CLAMA CLAME CLAMS CLANS CLARK CLAYS CLEF
RER CURES CUREZ CURRY CUVEE CUVER CUVES CUVEZ CYANS CYCAS CYCLE CYCLO CYGNE CYMES CYONS CZARS DADAS DAGUE DAIMS DAKIN DALLA DALLE DAMER DAMES DAMEZ DA
DOUZE DOYEN DRAIN DRAME DRAPA DRAPE DRAPS DRAYA DRAYE DRINK DROIT DROLE DROPS DRUES DRUZE DUAUX DUCAL DUCHE DUELS DUNES DUNKS DUPEE DUPER DUPES DUPEZ
R ESTOC ETAGE ETAIE ETAIN ETAIS ETAIT ETALA ETALE ETALS ETANG ETANT ETAPE ETATS ETAUX ETAYA ETAYE ETEND ETHER ETIEZ ETIRA ETIRE ETOLE ETRES ETRON ETUD
NIE FINIR FINIS FINIT FIOLE FIONS FIRME FISCS FIXAI FIXAS FIXAT FIXEE FIXER FIXES FIXEZ FJELD FJORD FLAIR FLANA FLANC FLANE FLANS FLAPI FLASH FLEAU FL
GARER GARES GAREZ GARNI GATEE GATER GATES GATEZ GAULE GAVEE GAVER GAVES GAVEZ GAYAL GAZAI GAZAS GAZAT GAZEE GAZER GAZES GAZEZ GAZON GAZOU GEAIS GEANT
A HALEE HALER HALES HALEZ HALLE HALLS HALOS HALTE HAMAC HAMPE HANTA HANTE HAPAX HAPPA HAPPE HARAS HARDI HAREM HARKA HARKI HAROS HARPE HASES HATEE HATE
ONS JABLA JABLE JABOT JACEE JACKS JACOS JACOT JACTA JACTE JADES JADIS JAINA JAINE JAINS JALAP JALES JALON JAMBE JANTE JAPON JAPPA JAPPE JAQUE JARDE JA
KAREN KARMA KARST KARTS KATAS KAVAS KAWAS KAWIS KAYAC KAYAK KEBAB KEFIR KENDO KEPIS KERMA KETCH KEUFS KEUMS KHANS KHATS KHMER KHOIN KHOLS KICKS KIEFS
S LIEES LIEGE LIENS LIENT LIERA LIEUE LIEUX LIFTS LIGNE LIGUA LIGUE LIIEZ LILAS LIMEE LIMER LIMES LIMEZ LIMON LINGE LINKS LINOS LIONS LIPPE LIPPU LIRA
RNE MARRA MARRE MARRI MARTE MASSA MASSE MATCH MATER MATES MATEZ MATHS MATIN MATIR MAUVE MAXIS MAYAS MAYEN MAYES MAZAI MAZAS MAZAT MAZEE MAZER MAZES MA
MUERA MUETS MUFLE MUGIR MUGIS MUGIT MUIEZ MULES MULET MULOT MUNIE MUNIR MUNIS MUNIT MUONS MURAL MUREE MURER MURES MURET MUREX MUREZ MURIE MURIR MURIS
N OKAPI OLIVE OMBRE OMETS OMISE ONCES ONCLE ONDEE ONDES ONGLE ONLAY OPALE OPERA OPERE OPINA OPINE OPIUM OPTER OPTES OPTEZ ORAGE ORALE ORAUX ORDRE OREE
VES PAVEZ PAVOT PAXON PAYAI PAYAS PAYAT PAYEE PAYER PAYES PAYEZ PAYSE PEAGE PEAUX PECHA PECHE PEDZA PEDZE PEGRE PEINA PEINE PEINS PEINT PEKAN PEKET PE
```

Vos algorithmes seront basés sur la programmation objet et pour cela vous créerez au moins 4 classes : Joueur, De, Plateau, Dictionnaire et la classe Program, (vous pourrez la renommez Jeu) modélisera le jeu.

Exercice 1 : Classe Joueur (à réaliser dans un fichier Joueur.cs)

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 suivantes sont imposées : (les signatures peuvent être ajustées en fonction de votre code)

`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 bool Mot_Cite(string mot) teste si le mot passé en paramètre a déjà été cité`

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

Exercice 2 : Classe Dé (à réaliser dans un fichier De.cs)

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 suivantes sont imposées : : (les signatures peuvent être ajustées en fonction de votre code)

`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 (à réaliser dans un fichier Dictionnaire.cs)

Une instance de dictionnaire associe un ensemble de mots avec une longueur déterminée ainsi qu'une langue.

Les méthodes sont imposées : : (les signatures peuvent être ajustées en fonction de votre code)

`public override string ToString() ou public string toString() qui retourne une chaîne de caractères qui décrit le dictionnaire à savoir ici le nombre de mots par longueur et la langue`

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

Exercice 4 : Classe Plateau (à réaliser dans un fichier Plateau.cs)

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

Vous créerez les propriétés en fonction des besoins de votre programme

Les méthodes sont imposées : : (les signatures peuvent être ajustées en fonction de votre code)

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

`public bool Test_Plateau(string 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.`

`Un algorithme récursif sera plus apprécié.`

Exercice 5 : Classe Jeu (à réaliser dans un fichier Jeu .cs)

La classe Jeu possède 2 attributs : Un tableau de Dictionnaire ''mondico'' et un plateau ''monplateau''

- ''mondico'' sera instancié par la lecture du fichier donné en pièce jointe : MotspossiblesFrancais.txt. Prenez le temps d'analyser la construction de ce fichier
- ''monplateau'' sera instancié à partir de 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.

Au bout d'un laps de temps prévu au départ du jeu (6mn par exemple), vous affichez les scores des joueurs et indiquez le gagnant.

Exercice 6 : BONUS

Le joueur est une 'IA' : Vous devez donc explorer toutes les positions du plateau à la recherche des mots qui sont dans le dictionnaire.

Il est important de limiter la recherche au maximum afin de s'assurer que la tâche sera achevée dans un temps raisonnable.

L'une des stratégies les plus importantes consiste à déterminer lorsque la recherche s'engage sur une voie morte pour l'abandonner au plus vite. Par exemple, si la recherche a construit un chemin menant au préfixe 'zx', vous pouvez utiliser votre dictionnaire pour déterminer qu'il n'y a pas de mot qui commence par ce préfixe. Et donc, vous pouvez arrêter cette recherche pour continuer sur une voie meilleure. Si vous n'appliquez pas cette optimisation, votre ordinateur passera un temps non négligeable à vérifier la construction de mots qui n'existent pas tel que 'zxgub'.