

Namespace MotsGlissés

Classes

[Dictionnaire](#)

Représente un dictionnaire, qui est un ensemble de mots triés par ordre alphabétique organisé par ligne dans un fichier texte.

[Extras](#)

Classe contenant des méthodes supplémentaires. Elles sont statiques et ne nécessitent pas d'instanciation.

[Jeu](#)

Instanciation d'un jeu, équivalent à une partie

[Joueur](#)

Instanciation d'un joueur

[Plateau](#)

Classe représentant un plateau de jeu. Stocke aussi les lettres et leur valeur

Structs

[Extras.Position](#)

Représente une position dans un espace bidimensionnel.

Class Dictionnaire

Namespace: [MotsGlissés](#)

Assembly: MotsGlissés.dll

Représente un dictionnaire, qui est un ensemble de mots triés par ordre alphabétique organisé par ligne dans un fichier texte.

```
public class Dictionnaire
```

Inheritance

[object](#) ← Dictionnaire

Inherited Members

[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Constructors

Dictionnaire(string)

Constructeur d'un dictionnaire à partir d'un chemin vers un fichier texte

```
public Dictionnaire(string chemin)
```

Parameters

chemin [string](#)

chemin vers le fichier, utile pour lecture du fichier

Exceptions

[FileNotFoundException](#)

Exception renvoyée si le fichier n'existe pas

Methods

RechDichoRecuratif(string)

Recherche un mot dans le dictionnaire, via un stream, en accédant itérativement à la ligne correspondante au premier caractère du mot Nécessite que le dictionnaire soit trié, et que le format du fichier soit correct et en accord avec le sujet

```
public bool RechDichoRecuratif(string input)
```

Parameters

input [string](#)

Mot recherché

Returns

[bool](#)

bool: Vrai si la mot a été trouvé, faux si le mot n'est pas trouvé, ou est nul, de taille nulle, ou ne contient pas que des lettres

Tri_Fusion()

Effectue un Tri Fusion sur chacune des lignes du dictionnaire Possible de le faire grâce à des streams
Implémentation de Fusion dans Extras.cs

```
public void Tri_Fusion()
```

toString()

Représente le dictionnaire sous forme de chaîne de caractères Affiche la langue du dictionnaire (français par défaut), puis le nombre de mots par lettre Implémenté avec un stream Assume que le dictionnaire soit conforme au format donné dans le sujet

```
public string toString()
```

Returns

[string](#) 

string: la chaîne de caractère

Class Extras

Namespace: [MotsGlissés](#)

Assembly: MotsGlissés.dll








Classe contenant des méthodes supplémentaires. Elles sont statiques et ne nécessitent pas d'instanciation.

```
public class Extras
```

Inheritance

[object](#)  ← Extras

Inherited Members

[object.Equals\(object\)](#)  , [object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  , [object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object, object\)](#)  , [object.ToString\(\)](#) 

Methods

Fusion(string[])

Implémentation simple du Tri fusion

```
public static string[] Fusion(string[] tab)
```

Parameters

tab [string](#)  []

Tableau de string à trier

Returns

[string](#)  []

string[] tableau d'entrée trié

ReadLine(TimeSpan)

Equivalent de ReadLine mais avec un timeout

```
public static string? ReadLine(TimeSpan timeout)
```

Parameters

timeout [TimeSpan](#)

La durée maximale d'attente pour une entrée.

Returns

[string](#)

La chaîne de caractères saisie par l'utilisateur, ou null si aucune saisie n'a été effectuée dans le délai imparti.

Split(string[])

Coupe un tableau de string en deux parties Le polymorphisme n'a pas été implémenté

```
public static (string[], string[]) Split(string[] tab)
```

Parameters

tab [string](#)[]

Tableau à couper

Returns

([string](#)[], [string](#)[])

(string[], string []): les tableaux coupés, dans l'ordre de coupe

Struct Extras.Position

Namespace: [MotsGlissés](#)

Assembly: MotsGlissés.dll

Représente une position dans un espace bidimensionnel.

```
public struct Extras.Position
```

Inherited Members

[ValueType.ToString\(\)](#), [object.Equals\(object, object\)](#), [object.GetType\(\)](#), [object.ReferenceEquals\(object, object\)](#)

Constructors

Position(int, int)

Constructeur de la structure Position

```
public Position(int x, int y)
```

Parameters

x [int](#)

entier représentant la position dans la largeur de la console

y [int](#)

entier représentant la position dans la hauteur de la console

Properties

X

Propriété publique en lecture seule de la position en x

```
public int x { get; }
```

Property Value

[int](#)

this.x

Y

Propriété publique en lecture seule de la position en y

```
public int y { get; }
```

Property Value

[int](#)

this.y

Methods

Equals(object?)

Override de la méthode Equals pour comparer deux positions

```
public override bool Equals(object? obj)
```

Parameters

obj [object](#)

objet que l'on compare

Returns

[bool](#)

GetHashCode()

Calcule un code de hachage pour l'objet actuel en fonction de ses valeurs x et y.

```
public override int GetHashCode()
```

Returns

[int](#)

Un code de hachage pour l'objet actuel.

Operators

operator ==(Position, Position)

Override l'opérateur ==

```
public static bool operator ==(Extras.Position left, Extras.Position right)
```

Parameters

left [Extras.Position](#)

position 1

right [Extras.Position](#)

position 2

Returns

[bool](#)

bool: vrai si les positions sont égales en x et y

operator !=(Position, Position)

Override l'opérateur !=

```
public static bool operator !=(Extras.Position left, Extras.Position right)
```

Parameters

left [Extras.Position](#)

position 1

right [Extras.Position](#)

position 2

Returns

[bool](#)

bool: vrai si les positions sont différentes en x et/ou y

Class Jeu

Namespace: [MotsGlissés](#)

Assembly: MotsGlissés.dll








Instancation d'un jeu, équivalent à une partie

```
public class Jeu
```

Inheritance

[object](#)  ← Jeu

Inherited Members

[object.Equals\(object\)](#)  , [object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  ,
[object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object, object\)](#)  , [object.ToString\(\)](#) 

Constructors

Jeu(Dictionnaire, Plateau, List<Joueur>, TimeSpan, TimeSpan)

Instancie le jeu. Sa fonction principal est de run play()

```
public Jeu(Dictionnaire dico, Plateau plateau, List<Joueur> joueur, TimeSpan tempsJoueur,  
TimeSpan tempsJeu)
```

Parameters

dico [Dictionnaire](#)

Le dictionnaire qui servira à valider les mots

plateau [Plateau](#)

Le plateau

joueur [List](#)  <[Joueur](#)>

Liste des joueurs

tempsJoueur [TimeSpan](#) 

Temps maximal autorisé pour qu'un joueur trouve un mot

tempsJeu [TimeSpan](#)↗

Temps maximal de la partie

Methods

play()

Méthode de jeu

```
public void play()
```

Class Joueur

Namespace: [MotsGlissés](#)

Assembly: MotsGlissés.dll

Instanciation d'un joueur

```
public class Joueur
```

Inheritance

[object](#)  ← Joueur

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

Constructors

Joueur(string)

Constructeur simple d'un joueur

```
public Joueur(string nom)
```

Parameters

nom [string](#) 

Nom assigné au joueur

Joueur(string, int, List<string>)

Constructeur complet d'un joueur, utilisé pour le debugage

```
public Joueur(string nom, int score, List<string> listeMotTrouver)
```

Parameters

nom [string](#)

Nom du joueur

score [int](#)

Score de départ du joueur

listeMotTrouver [List](#) <[string](#)>

Liste des mots à trouver

Properties

Nom

Propriété publique en lecture seule du nom du joueur

```
public string Nom { get; }
```

Property Value

[string](#)

Score

Propriété publique en lecture seule du score du joueur

```
public int Score { get; }
```

Property Value

[int](#)

Methods

Add_Mot(string)

Ajoute un mot à la liste des mots trouvés par le joueur, seulement si ce mot n'a pas déjà été trouvé

```
public void Add_Mot(string mot)
```

Parameters

mot [string](#)

Mot qu'il faut ajouter à la liste des mots trouvés

Add_Score(int)

Ajoute un score au score du joueur Le score est calculé depuis plateau, à partir du poids des lettres et d'un bonus de 5 points pour avoir trouvé un mot

```
public void Add_Score(int val)
```

Parameters

val [int](#)

Valeur à ajouter

Contient(string)

Test si un mot a déjà été trouvé par le joueur

```
public bool Contient(string mot)
```

Parameters

mot [string](#)

Le mot qu'on teste

Returns

[bool](#)

vrai si le mot a déjà été trouvé, faux sinon

toString()

Condense les informations du joueur en une chaîne de caractère

```
public string toString()
```

Returns

[string](#)

La chaîne de caractère représentant le joueur

Class Plateau

Namespace: [MotsGlissés](#)

Assembly: MotsGlissés.dll

Classe représentant un plateau de jeu. Stocke aussi les lettres et leur valeur

```
public class Plateau
```

Inheritance

[object](#)  ← Plateau

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

Constructors

Plateau()

Crée un plateau de maniere aleatoire à partir d'un fichier .txt conforme au sujet Prochaine
implémentations : sélection du fichier et de la taille du plateau

```
public Plateau()
```

Plateau(string)

Constructeur qui crée un plateau à partir d'un fichier .csv Se référer à ToRead

```
public Plateau(string filePath)
```

Parameters

filePath [string](#) 

Properties

NbLettres

Propriété publique en lecture seule du nombres de lettres restantes sur le plateau

```
public int NbLettres { get; }
```

Property Value

[int](#)

int

Methods

GetScore(string)

Récupère le score d'un mot en fonction des valeurs des lettres

```
public int GetScore(string mot)
```

Parameters

mot [string](#)

Mot dont on cherche le scor associé

Returns

[int](#)

int: valeur Scrabble du mot choisi

Maj_Plateau(Stack<Position>)

Mise à jour de la matrice représentant le plateau selon les positions données Remplace les positions par des espaces et fait descendre les lettres Les positions sont données dans une stack pour éviter les modifications intempestives des positions

```
public void Maj_Plateau(Stack<Extras.Position> positions)
```

Parameters

positions [Stack](#) <[Extras.Position](#)>

Pile des positions à modifier

Recherche_Mot(string)

Initie un arbre de recherche sur toutes les cases de la ligne en bas du plateau. Un test est effectué dans la recherche arbre pour savoir si la case correspond à ce que l'on attend. Assez opti

```
public (bool, Stack<Extras.Position>) Recherche_Mot(string mot)
```

Parameters

mot [string](#)

Mot à chercher

Returns

([bool](#), [Stack](#) <[Extras.Position](#)>)

bool : Vrai si le mot a été trouvé, Stack(Position) : représentation du chemin pris, utilisé par Maj_Plateau()

Exceptions

[ArgumentNullException](#)

Exception renvoyée si le mot est nul

ToFile(string)

Sauvegarde l'instance du plateau dans un fichier en respectant la structure précisée

```
public void ToFile(string filePath)
```

Parameters

filePath [string](#)

Chemin du fichier d'écriture

ToRead(string)

Lecture d'un fichier .csv et remplissage du plateau

```
public void ToRead(string filePath)
```

Parameters

filePath [string](#)

Chemin vers le fichier

Exceptions

[FileNotFoundException](#)

Exception renvoyée si le fichier n'est pas trouvé

toString()

Retourne une chaîne de caractères qui représente le plateau

```
public string toString()
```

Returns

[string](#)

Chaîne de caractère représentant le tableau de jeu