

# Jeu de Boggle

1

Généré par Doxygen 1.8.1.2

Mardi Mai 7 2013 18 :45 :02



# Table des matières

<b>1</b>	<b>Jeu de Boggle -- UE Projet</b>	<b>1</b>
1.1	Exécution du programme . . . . .	1
1.2	Version 1 : Le résolveur . . . . .	1
1.3	Version 2 : Le mode texte . . . . .	1
1.4	Version 3 : L'interface Ncurses . . . . .	2
<b>2</b>	<b>Index des structures de données</b>	<b>3</b>
2.1	Structures de données . . . . .	3
<b>3</b>	<b>Index des fichiers</b>	<b>5</b>
3.1	Liste des fichiers . . . . .	5
<b>4</b>	<b>Documentation des structures de données</b>	<b>7</b>
4.1	Référence de la structure Couple . . . . .	7
4.1.1	Description détaillée . . . . .	7
4.1.2	Documentation des champs . . . . .	7
4.1.2.1	x . . . . .	7
4.1.2.2	y . . . . .	7
4.2	Référence de la structure Dico . . . . .	7
4.2.1	Description détaillée . . . . .	8
4.2.2	Documentation des champs . . . . .	8
4.2.2.1	dico . . . . .	8
4.2.2.2	marqueurs . . . . .	8
4.3	Référence de la structure Jeu . . . . .	8
4.3.1	Description détaillée . . . . .	8
4.3.2	Documentation des champs . . . . .	8
4.3.2.1	dico . . . . .	8
4.3.2.2	plateau . . . . .	8
4.3.2.3	solutionUtilisateur . . . . .	9
4.3.2.4	timestampDepart . . . . .	9
4.4	Référence de la structure Plateau . . . . .	9
4.4.1	Description détaillée . . . . .	9

4.4.2	Documentation des champs	9
4.4.2.1	grille	9
4.4.2.2	solution	9
4.4.2.3	tailleGrille	9
4.5	Référence de la structure Solution	9
4.5.1	Description détaillée	10
4.5.2	Documentation des champs	10
4.5.2.1	mots	10
4.5.2.2	nbMots	10
<b>5</b>	<b>Documentation des fichiers</b>	<b>11</b>
5.1	Référence du fichier couple.h	11
5.1.1	Description détaillée	11
5.2	Référence du fichier dictionnaire.c	11
5.2.1	Description détaillée	11
5.2.2	Documentation des fonctions	12
5.2.2.1	dictionnaire_nouveau	12
5.2.2.2	dictionnaire_rechercheDichotomique	12
5.2.2.3	dictionnaire_motDansDico	12
5.3	Référence du fichier dictionnaire.h	12
5.3.1	Description détaillée	13
5.3.2	Documentation des fonctions	13
5.3.2.1	dictionnaire_nouveau	13
5.3.2.2	dictionnaire_rechercheDichotomique	13
5.3.2.3	dictionnaire_motDansDico	13
5.4	Référence du fichier interfaceNcurses.c	13
5.4.1	Description détaillée	14
5.4.2	Documentation des fonctions	14
5.4.2.1	fctMenu	14
5.4.2.2	interfaceNcurses_afficherBoiteDialogue	15
5.4.2.3	interfaceNcurses_afficherFenetreJeu	15
5.4.2.4	interfaceNcurses_afficherFenetreSolution	15
5.4.2.5	interfaceNcurses_afficherGrille	15
5.4.2.6	interfaceNcurses_afficherSolution	15
5.4.2.7	interfaceNcurses_afficherTitre	15
5.4.2.8	interfaceNcurses_initialiser	16
5.4.2.9	interfaceNcurses_menu	16
5.4.2.10	interfaceNcurses_terminer	16
5.4.2.11	jeu_lancerModeNcurses	16
5.5	Référence du fichier interfaceNcurses.h	16

5.5.1	Description détaillée . . . . .	17
5.5.2	Documentation des fonctions . . . . .	17
5.5.2.1	interfaceNcurses_afficherBoiteDialogue . . . . .	17
5.5.2.2	interfaceNcurses_afficherFenetreJeu . . . . .	17
5.5.2.3	interfaceNcurses_afficherFenetreSolution . . . . .	17
5.5.2.4	interfaceNcurses_afficherGrille . . . . .	18
5.5.2.5	interfaceNcurses_afficherSolution . . . . .	18
5.5.2.6	interfaceNcurses_afficherTitre . . . . .	18
5.5.2.7	interfaceNcurses_initialiser . . . . .	18
5.5.2.8	interfaceNcurses_menu . . . . .	18
5.5.2.9	interfaceNcurses_terminer . . . . .	18
5.5.2.10	jeu_lancerModeNcurses . . . . .	19
5.6	Référence du fichier interfaceTexte.c . . . . .	19
5.6.1	Description détaillée . . . . .	19
5.6.2	Documentation des fonctions . . . . .	19
5.6.2.1	interfaceTexte_afficherSolution . . . . .	19
5.6.2.2	jeu_lancerModeTexte . . . . .	19
5.7	Référence du fichier interfaceTexte.h . . . . .	20
5.7.1	Description détaillée . . . . .	20
5.7.2	Documentation des fonctions . . . . .	20
5.7.2.1	jeu_lancerModeTexte . . . . .	20
5.8	Référence du fichier jeu.c . . . . .	20
5.8.1	Description détaillée . . . . .	21
5.8.2	Documentation des fonctions . . . . .	21
5.8.2.1	jeu_compteurClaque . . . . .	21
5.8.2.2	jeu_lancer . . . . .	21
5.8.2.3	jeu_nouveau . . . . .	21
5.8.2.4	jeu_proposerMot . . . . .	21
5.8.2.5	jeu_stopper . . . . .	22
5.8.2.6	jeu_tempsRestant . . . . .	22
5.9	Référence du fichier jeu.h . . . . .	22
5.9.1	Description détaillée . . . . .	22
5.9.2	Documentation des fonctions . . . . .	23
5.9.2.1	jeu_compteurClaque . . . . .	23
5.9.2.2	jeu_lancer . . . . .	23
5.9.2.3	jeu_nouveau . . . . .	23
5.9.2.4	jeu_proposerMot . . . . .	23
5.9.2.5	jeu_stopper . . . . .	23
5.9.2.6	jeu_tempsRestant . . . . .	24
5.10	Référence du fichier main.c . . . . .	24

5.10.1	Description détaillée	24
5.10.2	Documentation des fonctions	24
5.10.2.1	main	24
5.10.2.2	mauvaisParametre	25
5.11	Référence du fichier plateau.c	25
5.11.1	Description détaillée	25
5.11.2	Documentation des fonctions	25
5.11.2.1	plateau_choisirLettre	25
5.11.2.2	plateau_detruire	26
5.11.2.3	plateau_nouveau	26
5.11.2.4	plateau_probaLettre	26
5.11.2.5	plateau_remplirGrilleAleatoire	26
5.11.2.6	plateau_remplirGrillePredefinie	26
5.12	Référence du fichier plateau.h	26
5.12.1	Description détaillée	27
5.12.2	Documentation des fonctions	27
5.12.2.1	plateau_choisirLettre	27
5.12.2.2	plateau_detruire	27
5.12.2.3	plateau_nouveau	27
5.12.2.4	plateau_probaLettre	28
5.12.2.5	plateau_remplirGrilleAleatoire	28
5.12.2.6	plateau_remplirGrillePredefinie	28
5.13	Référence du fichier resolveur.c	28
5.13.1	Description détaillée	29
5.13.2	Documentation des fonctions	29
5.13.2.1	jeu_lancerModeSolveur	29
5.13.2.2	recurse	29
5.13.2.3	resolveTouteLaGrille	29
5.14	Référence du fichier resolveur.h	29
5.14.1	Description détaillée	30
5.14.2	Documentation des fonctions	30
5.14.2.1	jeu_lancerModeSolveur	30
5.14.2.2	resolveTouteLaGrille	30
5.15	Référence du fichier solution.c	30
5.15.1	Description détaillée	30
5.15.2	Documentation des fonctions	31
5.15.2.1	solution_ajouterMot	31
5.15.2.2	solution_creer	31
5.15.2.3	solution_detruire	31
5.15.2.4	solution_getNbMotsDebut	31

5.15.2.5	<a href="#">solution_motEstPresent</a>	31
5.15.2.6	<a href="#">solution_nbPoints</a>	32
5.16	Référence du fichier <a href="#">solution.h</a>	32
5.16.1	Description détaillée	32
5.16.2	Documentation des fonctions	32
5.16.2.1	<a href="#">solution_ajouterMot</a>	32
5.16.2.2	<a href="#">solution_creer</a>	33
5.16.2.3	<a href="#">solution_detruire</a>	33
5.16.2.4	<a href="#">solution_getNbMotsDebut</a>	33
5.16.2.5	<a href="#">solution_motEstPresent</a>	33
5.16.2.6	<a href="#">solution_nbPoints</a>	33
5.17	Référence du fichier <a href="#">util.c</a>	34
5.17.1	Description détaillée	34
5.17.2	Documentation des fonctions	34
5.17.2.1	<a href="#">util_affichageDebug</a>	35
5.17.2.2	<a href="#">util_conversionTemps</a>	35
5.17.2.3	<a href="#">util_deplacerCurseurDunMot</a>	35
5.17.2.4	<a href="#">util_echanger</a>	35
5.17.2.5	<a href="#">util_isInArray</a>	35
5.17.2.6	<a href="#">util_nbAleatoire</a>	35
5.17.2.7	<a href="#">util_quickSort</a>	36
5.17.2.8	<a href="#">util_substr</a>	36
5.17.2.9	<a href="#">util_supprimerAccents</a>	36
5.17.2.10	<a href="#">util_uppercase</a>	36
5.18	Référence du fichier <a href="#">util.h</a>	37
5.18.1	Description détaillée	37
5.18.2	Documentation des fonctions	37
5.18.2.1	<a href="#">util_affichageDebug</a>	37
5.18.2.2	<a href="#">util_conversionTemps</a>	38
5.18.2.3	<a href="#">util_deplacerCurseurDunMot</a>	38
5.18.2.4	<a href="#">util_echanger</a>	38
5.18.2.5	<a href="#">util_isInArray</a>	38
5.18.2.6	<a href="#">util_nbAleatoire</a>	38
5.18.2.7	<a href="#">util_quickSort</a>	39
5.18.2.8	<a href="#">util_substr</a>	39
5.18.2.9	<a href="#">util_supprimerAccents</a>	39





# Chapitre 1

## Jeu de Boggle -- UE Projet

### Auteur

Antoine de Roquemaurel (G2.2)  
Fabrice Valleix (G2.2)

### Date

07/05/13 à 18 :20 :03

### 1.1 Exécution du programme

Afin d'exécuter le programme, un fichier de test fonctionnel a été créé, celui-ci génère un exécutable du nom de `./boggle`, celui-ci est placé à la racine du projet.

Il doit être exécuté avec des arguments :

- `./boggle --solveur`  
Lance un solveur, c'est-à-dire la version 1,
- `./boggle --text`  
Lance le jeu en mode text, soit la version 2
- `./boggle --ncurses`  
Lance le jeu avec le mode ncurses, la version 3

### 1.2 Version 1 : Le résolveur

Afin d'appeler la version 1 de l'application, l'exécutable doit être appelé à l'aide de l'argument `--solveur`

Dans cette version, une grille carrée de la taille demandée par l'utilisateur est générée, en tenant compte de la fréquence des lettres dans la langue Française. Une fois la grille générée, la position d'une case est demandée à l'utilisateur, l'utilisateur entre donc les deux coordonnées, et tous les mots commençant par cette case seront affichés à l'écran.

Attention, les coordonnées de la grille commencent à zéro.

### 1.3 Version 2 : Le mode texte

Afin d'appeler la version 2, l'exécutable doit être appelé à l'aide de l'argument `--text`

Cette version fait appel à la version 1, en effet, au lancement de l'application, il est de nouveau demandé la taille de la grille, ensuite l'intégralité de la grille générée est résolue. Une fois cette étape franchie, l'utilisateur a 3 minutes pour entrer le plus de mots possibles, l'application lui signalant si le mot est accepté ou non, une fois ce temps imparti, la solution est affichée, puis le nombre de points obtenu par le joueur.

## 1.4 Version 3 : L'interface Ncurses

Afin d'appeler la version 3, l'exécutable doit être appelé à l'aide de l'argument `--ncurses`

Cette version suit le même principe que la version précédente, à la différence près qu'elle utilise la bibliothèque Ncurses. Ainsi, la saisie des mots se fait dorénavant avec les touches fléchées du clavier, et espace pour ajouter une lettre au mot. Pour proposer le mot surligné, la touche entrée doit être appuyée. Il est également possible de demander le nombre de mots commençant par la case sélectionnée à l'aide de la touche h.

Une fois les 3 minutes écoulées, les mots proposés par l'utilisateur et le nombre de points obtenus sont affichés, il est proposé à l'utilisateur d'afficher la solution complète.

## Chapitre 2

# Index des structures de données

### 2.1 Structures de données

Liste des structures de données avec une brève description :

<a href="#">Couple</a>	Un couple de valeurs entières . . . . .	7
<a href="#">Dico</a>	Dictionnaire Contient le dictionnaire du jeu . . . . .	7
<a href="#">Jeu</a>	Structure permettant de jouer au Boggle Structure contenant toutes les informations nécessaire au bon déroulement du jeu . . . . .	8
<a href="#">Plateau</a>	Grille du jeu Contient une grille permettant de jouer au Boggle . . . . .	9
<a href="#">Solution</a>	<a href="#">Solution</a> du jeu Contient une solution pour une grille de Boggle . . . . .	9



## Chapitre 3

# Index des fichiers

### 3.1 Liste des fichiers

Liste de tous les fichiers documentés avec une brève description :

<a href="#">couple.h</a>	Structure de données d'un couple Structure de données d'un couple de valeurs entières . . .	11
<a href="#">dictionnaire.c</a>	Gestion du dictionnaire Fonctions de gestion du dictionnaire, le dictionnaire doit être dans un fichier texte avec un mot différent par ligne . . . . .	11
<a href="#">dictionnaire.h</a>	Gestion du dictionnaire Fonctions de gestion du dictionnaire, le dictionnaire doit être dans un fichier texte avec un mot différent par ligne . . . . .	12
<a href="#">interfaceNcurses.c</a>	Affichage avec la bibliothèque Ncurses Affiche et permet de jouer au jeu de Boggle avec un affichage utilisant la bibliothèque Ncurses . . . . .	13
<a href="#">interfaceNcurses.h</a>	Affichage avec la bibliothèque Ncurses Affiche et permet de jouer au jeu de Boggle avec un affichage utilisant la bibliothèque Ncurses . . . . .	16
<a href="#">interfaceTexte.c</a>	Tous les affichages en mode texte seulement Permet l'affichage du jeu en mode texte uniquement, les saisies se font au clavier . . . . .	19
<a href="#">interfaceTexte.h</a>	Tous les affichages en mode texte seulement Permet l'affichage du jeu en mode texte uniquement, les saisies se font au clavier . . . . .	20
<a href="#">jeu.c</a>	Gestion du jeu de Boggle Les fonctions permettant de gérer le <a href="#">Jeu</a> de Boggle . . . . .	20
<a href="#">jeu.h</a>	Gestion du jeu de Boggle Les fonctions permettant de gérer le <a href="#">Jeu</a> de Boggle . . . . .	22
<a href="#">main.c</a>	. . . . .	24
<a href="#">plateau.c</a>	Gestion du plateau de <a href="#">Jeu</a> Fonctions se rapportant à un plateau de jeu, sera sert à gérer la grille de Boggle . . . . .	25
<a href="#">plateau.h</a>	Gestion du plateau de <a href="#">Jeu</a> Fonctions se rapportant à un plateau de jeu, sera sert à gérer la grille de Boggle . . . . .	26
<a href="#">resolveur.c</a>	Gestion du résolveur Permet de résoudre une grille de Boggle, une fois résolue, la solution se trouve dans le module <a href="#">Solution</a> . . . . .	28
<a href="#">resolveur.h</a>	Gestion du résolveur Permet de résoudre une grille de Boggle, une fois résolue, la solution se trouve dans le module <a href="#">Solution</a> . . . . .	29

<a href="#">solution.c</a>	Gère la solution d'une grille de boggle Toutes les fonctions se rapportant à la solution d'une grille	30
<a href="#">solution.h</a>	Gère la solution d'une grille de boggle Toutes les fonctions se rapportant à la solution d'une grille	32
<a href="#">util.c</a>	Fonctions utiles à l'ensemble du projet Toutes les fonctions de bases utiles au projet, ces fonctions travaillent sur des types de bae et ne sont pas spécifiques au projet, ce module permet de mieux organiser le code . . . . .	34
<a href="#">util.h</a>	Fonctions utiles à l'ensemble du projet Toutes les fonctions de bases utiles au projet, ces fonctions travaillent sur des types de bae et ne sont pas spécifiques au projet, ce module permet de mieux organiser le code . . . . .	37

## Chapitre 4

# Documentation des structures de données

### 4.1 Référence de la structure Couple

Un couple de valeurs entières.

```
#include <couple.h>
```

#### Champs de données

- unsigned char [x](#)
- unsigned char [y](#)

#### 4.1.1 Description détaillée

Un couple de valeurs entières.

#### 4.1.2 Documentation des champs

##### 4.1.2.1 unsigned char Couple : :x

Premier nombre du couple

##### 4.1.2.2 unsigned char Couple : :y

Second nombre du couple

La documentation de cette structure a été générée à partir du fichier suivant :

- [couple.h](#)

### 4.2 Référence de la structure Dico

Dictionnaire Contient le dictionnaire du jeu.

```
#include <dictionnaire.h>
```

#### Champs de données

- FILE \* [dico](#)
- long int [marqueurs](#) [32]

### 4.2.1 Description détaillée

Dictionnaire Contient le dictionnaire du jeu.

### 4.2.2 Documentation des champs

#### 4.2.2.1 FILE\* Dico : :dico

Fichier contenant le dictionnaire, celui-ci doit être un fichier texte avec un mot par ligne

#### 4.2.2.2 long int Dico : :marqueurs[32]

Contient les positions de chacune des lettres de l'alphabet dans le fichier

La documentation de cette structure a été générée à partir du fichier suivant :

– [dictionnaire.h](#)

## 4.3 Référence de la structure Jeu

Structure permettant de jouer au Boggle Structure contenant toutes les informations nécessaire au bon déroulement du jeu.

```
#include <jeu.h>
```

### Champs de données

- [Plateau](#) plateau
- [Dico](#) dico
- [time\\_t](#) timestampDepart
- [Solution](#) solutionUtilisateur
- unsigned int **lgTimer**

### 4.3.1 Description détaillée

Structure permettant de jouer au Boggle Structure contenant toutes les informations nécessaire au bon déroulement du jeu.

Voir également

[Dico](#), [Plateau](#), [Solution](#)

### 4.3.2 Documentation des champs

#### 4.3.2.1 Dico Jeu : :dico

Le dictionnaire

#### 4.3.2.2 Plateau Jeu : :plateau

Le plateau de [Jeu](#)



#### 4.3.2.3 Solution Jeu : :solutionUtilisateur

La solution que propose l'utilisateur, celle-ci peut ne pas être complète, mais tous les mots présents dans cette solution sont présent dans la grille

#### 4.3.2.4 time\_t Jeu : :timestampDepart

Le timestamp auquel on a commencé à jouer

La documentation de cette structure a été générée à partir du fichier suivant :

– [jeu.h](#)

## 4.4 Référence de la structure Plateau

Grille du jeu Contient une grille permettant de jouer au Boggle.

```
#include <plateau.h>
```

### Champs de données

- char \*\* [grille](#)
- [Couple](#) [tailleGrille](#)
- int \*\* [grid](#)
- [Solution](#) [solution](#)

#### 4.4.1 Description détaillée

Grille du jeu Contient une grille permettant de jouer au Boggle.

#### 4.4.2 Documentation des champs

##### 4.4.2.1 char\*\* Plateau : :grille

La grille de boggle

##### 4.4.2.2 Solution Plateau : :solution

La solution complète de cette grille de Boggle

##### 4.4.2.3 Couple Plateau : :tailleGrille

La taille de la grille

La documentation de cette structure a été générée à partir du fichier suivant :

– [plateau.h](#)

## 4.5 Référence de la structure Solution

[Solution](#) du jeu Contient une solution pour une grille de Boggle.

```
#include <solution.h>
```

## Champs de données

- char \*\* [mots](#)
- unsigned int [nbMots](#)

### 4.5.1 Description détaillée

[Solution](#) du jeu Contient une solution pour une grille de Boggle.

### 4.5.2 Documentation des champs

#### 4.5.2.1 char\*\* Solution : :mots

Les mots présents dans la solution

#### 4.5.2.2 unsigned int Solution : :nbMots

Le nombre de mots que contient la solution

La documentation de cette structure a été générée à partir du fichier suivant :

- [solution.h](#)

## Chapitre 5

# Documentation des fichiers

### 5.1 Référence du fichier couple.h

Structure de données d'un couple Structure de données d'un couple de valeurs entières.

#### Structures de données

- struct [Couple](#)  
*Un couple de valeurs entières.*

#### 5.1.1 Description détaillée

Structure de données d'un couple Structure de données d'un couple de valeurs entières.

### 5.2 Référence du fichier dictionnaire.c

Gestion du dictionnaire Fonctions de gestion du dictionnaire, le dictionnaire doit être dans un fichier texte avec un mot différent par ligne.

```
#include <string.h>
#include <stdlib.h>
#include "dictionnaire.h"
#include "util.h"
```

#### Fonctions

- [Dico dictionnaire\\_nouveau](#) (const char \*pNomFichier)  
*Créer un nouveau dictionnaire à partir d'un fichier.*
- int [dictionnaire\\_motDansDico](#) ([Dico](#) pDictionnaire, char \*pMot)  
*Permet de savoir si un mot est présent dans le dictionnaire.*
- void [dictionnaire\\_rechercheDichotomique](#) ([Dico](#) pDictionnaire, char \*pMotAChercher, char \*pMotLePlusProche)  
*Effectue une recherche dichotomique de pMotAChercher dans le dictionnaire.*

#### 5.2.1 Description détaillée

Gestion du dictionnaire Fonctions de gestion du dictionnaire, le dictionnaire doit être dans un fichier texte avec un mot différent par ligne.

## 5.2.2 Documentation des fonctions

### 5.2.2.1 Dico dictionnaire\_nouveau ( const char \* pNomFichier )

Créer un nouveau dictionnaire à partir d'un fichier.

#### Paramètres

<i>pNomFichier</i>	Le fichier contenant les mots du dictionnaire.
--------------------	--

#### Renvoie

Le nouveau dictionnaire

### 5.2.2.2 void dictionnaire\_rechercheDichotomique ( Dico pDictionnaire, char \* pMotAChercher, char \* pMotLePlusProche )

Effectue une recherche dichotomique de pMotAChercher dans le dictionnaire.

#### Paramètres

<i>pDictionnaire</i>	Le dictionnaire dans lequel chercher
<i>pMotAChercher</i>	Le mot à chercher
<i>pMotLePlus-Proche</i>	Le mot le plus proche trouvé, si pMotLePlusProche == pMotAChercher, il est présent dans le dictionnaire

### 5.2.2.3 int dictionnaire\_motDansDico ( Dico pDictionnaire, char \* pMot )

Permet de savoir si un mot est présent dans le dictionnaire.

#### Paramètres

<i>pDictionnaire</i>	Le dictionnaire dans lequel chercher
<i>pMot</i>	Le mot à chercher

#### Renvoie

0 si le mot est absent, 1 si des mots commencent par pMot et 10 si le mot exact est trouvé.

## 5.3 Référence du fichier dictionnaire.h

Gestion du dictionnaire Fonctions de gestion du dictionnaire, le dictionnaire doit être dans un fichier texte avec un mot différent par ligne.

```
#include <stdio.h>
#include <stdbool.h>
```

### Structures de données

- struct [Dico](#)  
*Dictionnaire Contient le dictionnaire du jeu.*

### Fonctions

- [Dico dictionnaire\\_nouveau](#) (const char \*pNomFichier)  
*Créer un nouveau dictionnaire à partir d'un fichier.*
- [Dico dictionnaire\\_positionnerMarqueurs](#) ([Dico](#) \*pDico)

- int `dictionnaire_motDansDico` (Dico pDictionnaire, char \*pMot)  
*Permet de savoir si un mot est présent dans le dictionnaire.*
- void `dictionnaire_rechercheDichotomique` (Dico pDictionnaire, char \*pMotAChercher, char \*pMotLePlusProche)  
*Effectue une recherche dichotomique de pMotAChercher dans le dictionnaire.*

### 5.3.1 Description détaillée

Gestion du dictionnaire Fonctions de gestion du dictionnaire, le dictionnaire doit être dans un fichier texte avec un mot différent par ligne.

### 5.3.2 Documentation des fonctions

#### 5.3.2.1 Dico dictionnaire\_nouveau ( const char \* pNomFichier )

Créer un nouveau dictionnaire à partir d'un fichier.

##### Paramètres

<i>pNomFichier</i>	Le fichier contenant les mots du dictionnaire.
--------------------	--

##### Renvoie

Le nouveau dictionnaire

#### 5.3.2.2 void dictionnaire\_rechercheDichotomique ( Dico pDictionnaire, char \* pMotAChercher, char \* pMotLePlusProche )

Effectue une recherche dichotomique de pMotAChercher dans le dictionnaire.

##### Paramètres

<i>pDictionnaire</i>	Le dictionnaire dans lequel chercher
<i>pMotAChercher</i>	Le mot à chercher
<i>pMotLePlusProche</i>	Le mot le plus proche trouvé, si pMotLePlusProche == pMotAChercher, il est présent dans le dictionnaire

#### 5.3.2.3 int dictionnaire\_motDansDico ( Dico pDictionnaire, char \* pMot )

Permet de savoir si un mot est présent dans le dictionnaire.

##### Paramètres

<i>pDictionnaire</i>	Le dictionnaire dans lequel chercher
<i>pMot</i>	Le mot à chercher

##### Renvoie

0 si le mot est absent, 1 si des mots commencent par pMot et 10 si le mot exact est trouvé.

## 5.4 Référence du fichier interfaceNcurses.c

Affichage avec la bibliothèque Ncurses Affiche et permet de jouer au jeu de Boggle avec un affichage utilisant la bibliothèque Ncurses.

```
#include <ncurses.h>
#include <panel.h>
#include <menu.h>
#include <stdlib.h>
#include <string.h>
#include "interfaceNcurses.h"
#include "plateau.h"
#include "jeu.h"
#include "util.h"
#include "resolveur.h"
```

## Fonctions

- int **fctMenu** (char \*name)  
*Retourne le numéro de l'item appelé*
- void **interfaceNcurses\_afficherTitre** (const char \*pTitle)  
*Afficher un titre à la fenêtre.*
- void **interfaceNcurses\_afficherSolution** (const **Solution** pSolution)  
*Affiche la solution passée en paramètre.*
- WINDOW \* **interfaceNcurses\_initialiser** (void)  
*Initialise l'interface Ncurses.*
- void **interfaceNcurses\_afficherGrille** (const **Plateau** pPlateau, const **Couple** pSelectedCase, const **Couple** \*pUsedCase, const int pLgUsedCase)  
*Affiche la grille, composée de couleurs en fonctions des différentes cases.*
- void **interfaceNcurses\_terminer** (WINDOW \*fenetre, **Jeu** pJeu)  
*Termine le jeu.*
- void **jeu\_lancerModeNcurses** (**Jeu** pJeu)  
*Lance le mode Ncurses.*
- void **interfaceNcurses\_afficherBoiteDialogue** (const **Jeu** pJeu)  
*Affiche une boite de dialogue de fin de Jeu.*
- void **interfaceNcurses\_afficherFenetreJeu** (const **Jeu** pJeu, char \*pMot, const **Couple** pSelectedCase, **Couple** \*pUsedCase)  
*Affiche la fenêtre permettant de jouer.*
- void **win\_show** (WINDOW \*win, char \*label, **Jeu** pJeu)
- void **interfaceNcurses\_menu** (WINDOW \*pDialogBoxWin, const **Jeu** pJeu)  
*Affiche un menu permettant de choisir entre quitter et afficher la solution.*
- void **interfaceNcurses\_afficherFenetreSolution** (const **Jeu** pJeu)  
*Affiche la fenêtre avec la solution complète du jeu.*
- void **print\_in\_middle** (WINDOW \*win, int starty, int startx, int width, char \*string, **Jeu** pJeu)

### 5.4.1 Description détaillée

Affichage avec la bibliothèque Ncurses Affiche et permet de jouer au jeu de Boggle avec un affichage utilisant la bibliothèque Ncurses.

### 5.4.2 Documentation des fonctions

#### 5.4.2.1 int fctMenu ( char \* name )

Retourne le numéro de l'item appelé

#### Paramètres

Le	nom de l'item
----	---------------

#### Renvoie

Le numéro de l'item appelé

5.4.2.2 void interfaceNcurses\_afficherBoiteDialogue ( const Jeu *pJeu* )

Affiche une boite de dialogue de fin de [Jeu](#).

## Paramètres

<i>pJeu</i>	Le jeu
-------------	--------

5.4.2.3 void interfaceNcurses\_afficherFenetreJeu ( const Jeu *pJeu*, char \* *pMot*, const Couple *pSelectedCase*, Couple \* *pUsedCase* )

Affiche la fenêtre permettant de jouer.

## Paramètres

<i>pJeu</i>	Le jeu
<i>pMot</i>	Le mot courant
<i>pSelectedCase</i>	La case sélectionnée
<i>pUsedCase</i>	Les cases utilisés pour créer le mot <i>pMot</i>

5.4.2.4 void interfaceNcurses\_afficherFenetreSolution ( const Jeu *pJeu* )

Affiche la fenêtre avec la solution complète du jeu.

## Paramètres

<i>pJeu</i>	Le jeu
-------------	--------

5.4.2.5 void interfaceNcurses\_afficherGrille ( const Plateau *pPlateau*, const Couple *pSelectedCase*, const Couple \* *pUsedCase*, const int *pLgUsedCase* )

Affiche la grille, composée de couleurs en fonctions des différentes cases.

## Paramètres

<i>pPlateau</i>	La grille à afficher
<i>pSelectedCase</i>	La case sélectionnée
<i>pUsedCase</i>	Les cases utilisés pour le mot courant
<i>pLgUsedCase</i>	La longueur du mot courant

5.4.2.6 void interfaceNcurses\_afficherSolution ( const Solution *pSolution* )

Affiche la solution passée en paramètre.

## Paramètres

<i>pSolution</i>	La solution à afficher
------------------	------------------------

5.4.2.7 void interfaceNcurses\_afficherTitre ( const char \* *pTitle* )

Afficher un titre à la fenêtre.

## Paramètres

<i>pTitle</i>	Le titre à afficher
---------------	---------------------

#### 5.4.2.8 WINDOW\* interfaceNcurses\_initialiser ( void )

Initialise l'interface Ncurses.

Renvoie

La fenêtre créée

#### 5.4.2.9 void interfaceNcurses\_menu ( WINDOW \* pDialogBoxWin, const Jeu pJeu )

Affiche un menu permettant de choisir entre quitter et afficher la solution.

Paramètres

<i>pDialogBoxWin</i>	La fenêtre danslequel s'affiche le menu
<i>pJeu</i>	Le jeu

Renvoie

#### 5.4.2.10 void interfaceNcurses\_terminer ( WINDOW \* fenetre, Jeu pJeu )

Termine le jeu.

Paramètres

<i>fenetre</i>	La fenêtre à détruire
<i>pJeu</i>	le jeu à terminer

#### 5.4.2.11 void jeu\_lancerModeNcurses ( Jeu pJeu )

Lance le mode Ncurses.

Paramètres

<i>pJeu</i>	Le jeu à lancer
-------------	-----------------

## 5.5 Référence du fichier interfaceNcurses.h

Affichage avec la bibliothèque Ncurses Affiche et permet de jouer au jeu de Boggle avec un affichage utilisant la bibliothèque Ncurses.

```
#include <ncurses.h>
#include "solution.h"
#include "couple.h"
#include "plateau.h"
#include "jeu.h"
```

### Macros

```
- #define COULEUR_SELECTION_CASE 1
- #define COULEUR_CASE_CHOISI 2
- #define NB_LIGNES 25
```



## Fonctions

- void `interfaceNcurses_afficherTitre` (const char \*pTitle)  
*Afficher un titre à la fenêtre.*
- void `interfaceNcurses_afficherSolution` (const `Solution` pSolution)  
*Affiche la solution passée en paramètre.*
- void `jeu_lancerModeNcurses` (`Jeu` pJeu)  
*Lance le mode Ncurses.*
- WINDOW \* `interfaceNcurses_initialiser` (void)  
*Initialise l'interface Ncurses.*
- void `interfaceNcurses_afficherGrille` (const `Plateau` pPlateau, const `Couple` pSelectedCase, const `Couple` \*pUsedCase, const int pLgUsedCase)  
*Affiche la grille, composée de couleurs en fonctions des différentes cases.*
- void `interfaceNcurses_terminer` (WINDOW \*fenetre, `Jeu` pJeu)  
*Termine le jeu.*
- void `interfaceNcurses_afficherFenetreJeu` (const `Jeu` pJeu, char \*pMot, const `Couple` pSelectedCase, `Couple` \*pUsedCase)  
*Affiche la fenêtre permettant de jouer.*
- void `interfaceNcurses_afficherBoiteDialogue` (const `Jeu` pJeu)  
*Affiche une boite de dialogue de fin de Jeu.*
- void `interfaceNcurses_afficherFenetreSolution` (const `Jeu` pJeu)  
*Affiche la fenêtre avec la solution complète du jeu.*
- void `interfaceNcurses_menu` (WINDOW \*pDialogBoxWin, const `Jeu` pJeu)  
*Affiche un menu permettant de choisir entre quitter et afficher la solution.*

### 5.5.1 Description détaillée

Affichage avec la bibliothèque Ncurses Affiche et permet de jouer au jeu de Boggle avec un affichage utilisant la bibliothèque Ncurses.

### 5.5.2 Documentation des fonctions

#### 5.5.2.1 void interfaceNcurses\_afficherBoiteDialogue ( const Jeu pJeu )

Affiche une boite de dialogue de fin de `Jeu`.

##### Paramètres

<code>pJeu</code>	Le jeu
-------------------	--------

#### 5.5.2.2 void interfaceNcurses\_afficherFenetreJeu ( const Jeu pJeu, char \* pMot, const Couple pSelectedCase, Couple \* pUsedCase )

Affiche la fenêtre permettant de jouer.

##### Paramètres

<code>pJeu</code>	Le jeu
<code>pMot</code>	Le mot courant
<code>pSelectedCase</code>	La case sélectionnée
<code>pUsedCase</code>	Les cases utilisés pour créer le mot pMot

#### 5.5.2.3 void interfaceNcurses\_afficherFenetreSolution ( const Jeu pJeu )

Affiche la fenêtre avec la solution complète du jeu.

##### Paramètres

<code>pJeu</code>	Le jeu
-------------------	--------

**5.5.2.4** void interfaceNcurses.afficherGrille ( const Plateau *pPlateau*, const Couple *pSelectedCase*, const Couple \* *pUsedCase*, const int *pLgUsedCase* )

Affiche la grille, composée de couleurs en fonctions des différentes cases.

**Paramètres**

<i>pPlateau</i>	La grille à afficher
<i>pSelectedCase</i>	La case sélectionnée
<i>pUsedCase</i>	Les cases utilisés pour le mot courant
<i>pLgUsedCase</i>	La longueur du mot courant

**5.5.2.5** void interfaceNcurses.afficherSolution ( const Solution *pSolution* )

Affiche la solution passée en paramètre.

**Paramètres**

<i>pSolution</i>	La solution à afficher
------------------	------------------------

**5.5.2.6** void interfaceNcurses.afficherTitre ( const char \* *pTitle* )

Afficher un titre à la fenêtre.

**Paramètres**

<i>pTitle</i>	Le titre à afficher
---------------	---------------------

**5.5.2.7** WINDOW\* interfaceNcurses.initialiser ( void )

Initialise l'interface Ncurses.

**Renvoie**

La fenêtre créée

**5.5.2.8** void interfaceNcurses.menu ( WINDOW \* *pDialogBoxWin*, const Jeu *pJeu* )

Affiche un menu permettant de choisir entre quitter et afficher la solution.

**Paramètres**

<i>pDialogBoxWin</i>	La fenêtre danslequel s'affiche le menu
<i>pJeu</i>	Le jeu

**Renvoie**

**5.5.2.9** void interfaceNcurses.terminer ( WINDOW \* *fenetre*, Jeu *pJeu* )

Termine le jeu.

**Paramètres**

<i>fenetre</i>	La fenêtre à détruire
<i>pJeu</i>	le jeu à terminer

## 5.5.2.10 void jeu\_lancerModeNcurses ( Jeu pJeu )

Lance le mode Ncurses.

## Paramètres

<i>pJeu</i>	Le jeu à lancer
-------------	-----------------

## 5.6 Référence du fichier interfaceTexte.c

Tous les affichages en mode texte seulement Permet l'affichage du jeu en mode texte uniquement, les saisies se font au clavier.

```
#include <stdio.h>
#include <string.h>
#include "solution.h"
#include "jeu.h"
#include "plateau.h"
#include "interfaceTexte.h"
#include "util.h"
```

## Fonctions

- void **interfaceTexte\_afficherPlateau** (const [Plateau](#) pPlateau)
- void **interfaceTexte\_afficherSolution** (const [Solution](#) pSolution)  
*Affiche la solution en mode texte.*
- void **interfaceTexte\_afficherPlateauEtSolution** (const [Plateau](#) pPlateau, const [Solution](#) pSolution)
- void **jeu\_lancerModeTexte** ([Jeu](#) pJeu)  
*Lance le jeu en mode texte.*

## 5.6.1 Description détaillée

Tous les affichages en mode texte seulement Permet l'affichage du jeu en mode texte uniquement, les saisies se font au clavier.

## 5.6.2 Documentation des fonctions

5.6.2.1 void interfaceTexte\_afficherSolution ( const [Solution](#) pSolution )

Affiche la solution en mode texte.

## Paramètres

<i>pSolution</i>	la solution à afficher
------------------	------------------------

## 5.6.2.2 void jeu\_lancerModeTexte ( Jeu pJeu )

Lance le jeu en mode texte.

## Paramètres

<i>pJeu</i>	le jeu à lancer
-------------	-----------------

## 5.7 Référence du fichier interfaceTexte.h

Tous les affichages en mode texte seulement Permet l’affichage du jeu en mode texte uniquement, les saisies se font au clavier.

```
#include "plateau.h"
```

### Fonctions

- void **interfaceTexte\_afficherPlateau** (const [Plateau](#) pPlateau)
- void **jeu\_lancerModeTexte** ([Jeu](#) pJeu)  
*Lance le jeu en mode texte.*

#### 5.7.1 Description détaillée

Tous les affichages en mode texte seulement Permet l’affichage du jeu en mode texte uniquement, les saisies se font au clavier.

#### 5.7.2 Documentation des fonctions

##### 5.7.2.1 void jeu\_lancerModeTexte ( [Jeu](#) pJeu )

Lance le jeu en mode texte.

##### Paramètres

<i>pJeu</i>	le jeu à lancer
-------------	-----------------

## 5.8 Référence du fichier jeu.c

Gestion du jeu de Boggle Les fonctions permettant de gérer le [Jeu](#) de Boggle.

```
#include <stdlib.h>
#include <string.h>
#include "util.h"
#include "resolveur.h"
#include "jeu.h"
#include "plateau.h"
#include "dictionnaire.h"
#include "interfaceTexte.h"
```

### Fonctions

- time\_t **jeu\_tempsRestant** (const [Jeu](#) pJeu)  
*Retourne le nombre de secondes restantes avant la fin du jeu.*
- bool **jeu\_compteurClaque** (const [Jeu](#) pJeu)  
*Retourne vrai si le temps est écoulé faux sinon.*
- [Jeu](#) **jeu\_nouveau** (const char \*pNomDico, const [Couple](#) pTaillePlateau, const unsigned int pLgTimer)  
*Créer un nouveau jeu.*
- void **jeu\_lancer** ([Jeu](#) \*pJeu)  
*Lance le jeu.*
- \_Bool **jeu\_proposerMot** ([Jeu](#) \*pJeu, const char \*pMot)  
*Proposer un mot.*
- void **jeu\_stopper** ([Jeu](#) pJeu)  
*Fin du jeu.*

### 5.8.1 Description détaillée

Gestion du jeu de Boggle Les fonctions permettant de gérer le [Jeu](#) de Boggle.

Voir également

[Plateau](#), [Resolveur](#), [Dico](#), [Solution](#)

### 5.8.2 Documentation des fonctions

#### 5.8.2.1 bool jeu\_compteurClaque ( const Jeu *pJeu* )

Retourne vrai si le temps est écoulé faux sinon.

Paramètres

<i>pJeu</i>	Le jeu pour lequel on veut savoir si le temps est écoulé
-------------	--

Renvoie

Vrai si le temps est écoulé faux sinon

#### 5.8.2.2 void jeu\_lancer ( Jeu \* *pJeu* )

Lance le jeu.

Paramètres

<i>pJeu</i>	Le jeu à lancer
-------------	-----------------

#### 5.8.2.3 Jeu jeu\_nouveau ( const char \* *pNomDico*, const Couple *pTaillePlateau*, const unsigned int *pLgTimer* )

Créer un nouveau jeu.

Paramètres

<i>pNomDico</i>	Le dictionnaire utilisé dans le <a href="#">Jeu</a>
<i>pTaillePlateau</i>	La taille du plateau à créer
<i>pLgTimer</i>	La longueur du timer en secondes

Renvoie

Le nouveau [Jeu](#)

#### 5.8.2.4 \_Bool jeu\_proposerMot ( Jeu \* *pJeu*, const char \* *pMot* )

Proposer un mot.

Paramètres

<i>pJeu</i>	Le jeu pour lequel on veut proposer un mot
<i>pMot</i>	Le mot proposé

Renvoie

Vrai si le mot est validé

### 5.8.2.5 void jeu\_stopper ( Jeu pJeu )

Fin du jeu.

#### Paramètres

<i>pJeu</i>	Le jeu à terminer
-------------	-------------------

### 5.8.2.6 time\_t jeu\_tempsRestant ( const Jeu pJeu )

Retourne le nombre de secondes restantes avant la fin du jeu.

#### Paramètres

<i>pJeu</i>	Le jeu pour lequel on veut savoir le temps restant
-------------	--

#### Renvoie

Le nombre de secondes restantes

## 5.9 Référence du fichier jeu.h

Gestion du jeu de Boggle Les fonctions permettant de gérer le [Jeu](#) de Boggle.

```
#include <time.h>
#include "plateau.h"
#include "dictionnaire.h"
```

### Structures de données

- struct [Jeu](#)  
*Structure permettant de jouer au Boggle Structure contenant toutes les informations nécessaire au bon déroulement du jeu.*

### Fonctions

- [Jeu jeu\\_nouveau](#) (const char \*pNomDico, const [Couple](#) pTaillePlateau, const unsigned int pLgTimer)  
*Créer un nouveau jeu.*
- void [jeu\\_lancer](#) (Jeu \*pJeu)  
*Lance le jeu.*
- \_Bool [jeu\\_proposerMot](#) (Jeu \*pJeu, const char \*pMot)  
*Proposer un mot.*
- bool [jeu\\_compteurClaque](#) (const [Jeu](#) pJeu)  
*Retourne vrai si le temps est écoulé faux sinon.*
- time\_t [jeu\\_tempsRestant](#) (const [Jeu](#) pJeu)  
*Retourne le nombre de secondes restantes avant la fin du jeu.*
- void [jeu\\_stopper](#) (Jeu pJeu)  
*Fin du jeu.*

### 5.9.1 Description détaillée

Gestion du jeu de Boggle Les fonctions permettant de gérer le [Jeu](#) de Boggle.

#### Voir également

[Plateau](#), [Resolveur](#), [Dico](#), [Solution](#)

## 5.9.2 Documentation des fonctions

### 5.9.2.1 bool jeu\_compteurClaque ( const Jeu *pJeu* )

Retourne vrai si le temps est écoulé faux sinon.

#### Paramètres

<i>pJeu</i>	Le jeu pour lequel on veut savoir si le temps est écoulé
-------------	--

#### Renvoie

Vrai si le temps est écoulé faux sinon

### 5.9.2.2 void jeu\_lancer ( Jeu \* *pJeu* )

Lance le jeu.

#### Paramètres

<i>pJeu</i>	Le jeu à lancer
-------------	-----------------

### 5.9.2.3 Jeu jeu\_nouveau ( const char \* *pNomDico*, const Couple *pTaillePlateau*, const unsigned int *pLgTimer* )

Créer un nouveau jeu.

#### Paramètres

<i>pNomDico</i>	Le dictionnaire utilisé dans le <a href="#">Jeu</a>
<i>pTaillePlateau</i>	La taille du plateau à créer
<i>pLgTimer</i>	La longueur du timer en secondes

#### Renvoie

Le nouveau [Jeu](#)

### 5.9.2.4 \_Bool jeu\_proposerMot ( Jeu \* *pJeu*, const char \* *pMot* )

Proposer un mot.

#### Paramètres

<i>pJeu</i>	Le jeu pour lequel on veut proposer un mot
<i>pMot</i>	Le mot proposé

#### Renvoie

Vrai si le mot est validé

### 5.9.2.5 void jeu\_stopper ( Jeu *pJeu* )

Fin du jeu.

#### Paramètres

<i>pJeu</i>	Le jeu à terminer
-------------	-------------------

### 5.9.2.6 time\_t jeu\_tempsRestant ( const Jeu pJeu )

Retourne le nombre de secondes restantes avant la fin du jeu.

#### Paramètres

<i>pJeu</i>	Le jeu pour lequel on veut savoir le temps restant
-------------	--

#### Renvoie

Le nombre de secondes restantes

## 5.10 Référence du fichier main.c

```
#include <stdio.h>
#include <stdlib.h>
#include "dictionnaire.h"
#include "util.h"
#include "plateau.h"
#include "resolveur.h"
#include "jeu.h"
#include "interfaceNcurses.h"
#include "interfaceTexte.h"
```

### Énumérations

- enum [TypeJeu](#) { **TEXTE**, **SOLVEUR**, **NCURSES** }  
*Enumeration contenant le type de [Jeu](#) possible.*

### Fonctions

- int [mauvaisParametre](#) (char \*pNomExe)  
*Affiche une erreur de paramètres.*
- int [main](#) (int argc, char \*\*argv)  
*Lance le jeu de Boggle en fonction des paramètres.*

### 5.10.1 Description détaillée

#### Voir également

Dictionnaire, Util, [Plateau](#), [Resolveur](#), [Jeu](#), [InterfaceNcurses](#), [interfaceTexte](#)

### 5.10.2 Documentation des fonctions

#### 5.10.2.1 int main ( int argc, char \*\* argv )

Lance le jeu de Boggle en fonction des paramètres.

#### Paramètres

<i>argc</i>	Nombre de paramètres
<i>argv</i>	Tableau de paramètres



**Renvoie**

Le code de retour du programme.

**5.10.2.2 int mauvaisParametre ( char \* pNomExe )**

Affiche une erreur de paramètres.

**Paramètres**

<i>pNomExe</i>	Le nom de l'exécutable
----------------	------------------------

**Renvoie**

Code de retour du programme

**5.11 Référence du fichier plateau.c**

Gestion du plateau de [Jeu Fonctions](#) se rapportant à un plateau de jeu, sera sert à gérer la grille de Boggle.

```
#include <stdlib.h>
#include "solution.h"
#include "plateau.h"
#include "util.h"
```

**Fonctions**

- [Plateau plateau\\_nouveau](#) (const [Couple](#) pTailleGrille)  
*Créer un nouveau plateau.*
- void [plateau\\_detruire](#) ([Plateau](#) \*pPlateau)  
*Détruit le plateau pPlateau.*
- void [plateau\\_remplirGrilleAleatoire](#) ([Plateau](#) \*pPlateau)  
*Rempli une grille aléatoirement dans le plateau.*
- void [plateau\\_remplirGrillePredefinie](#) ([Plateau](#) \*pPlateau)  
*Rempli le plateau avec une grille prédéfinie de taille 4\*4 E D R C A N V C I R Q A E B R U.*
- char [plateau\\_choisirLettre](#) (void)  
*Choisi une lettre avec la probabilité des lettres en fonctions de leurs importance d'apparition dans la langue française.*
- double [plateau\\_probaLettre](#) (const char pLettre)  
*Retourne la probabilité d'apparition de la lettre passée en paramètre.*

**5.11.1 Description détaillée**

Gestion du plateau de [Jeu Fonctions](#) se rapportant à un plateau de jeu, sera sert à gérer la grille de Boggle.

**5.11.2 Documentation des fonctions****5.11.2.1 char plateau\_choisirLettre ( void )**

Choisi une lettre avec la probabilité des lettres en fonctions de leurs importance d'apparition dans la langue française.

**Renvoie**

La lettre choisie

#### 5.11.2.2 void plateau\_detruire ( Plateau \* pPlateau )

Détruit le plateau pPlateau.

##### Paramètres

<i>pPlateau</i>	Le plateau à détruire
-----------------	-----------------------

#### 5.11.2.3 Plateau plateau\_nouveau ( const Couple pTailleGrille )

Créer un nouveau plateau.

##### Paramètres

<i>pTailleGrille</i>	La taille de la grille à créer
----------------------	--------------------------------

##### Renvoie

Le nouveau plateau

#### 5.11.2.4 double plateau\_probaLettre ( const char pLettre )

Retourne la probabilité d'apparition de la lettre passée en paramètre.

##### Paramètres

<i>pLettre</i>	La lettre pour laquelle retourner la probabilité
----------------	--

##### Renvoie

La probabilité

#### 5.11.2.5 void plateau\_remplirGrilleAleatoire ( Plateau \* pPlateau )

Rempli une grille aléatoirement dans le plateau.

##### Paramètres

<i>pPlateau</i>	Le plateau à remplir
-----------------	----------------------

#### 5.11.2.6 void plateau\_remplirGrillePredefinie ( Plateau \* pPlateau )

Rempli le plateau avec une grille prédéfinie de taille 4\*4 E D R C A N V C I R Q A E B R U.

##### Paramètres

<i>pPlateau</i>	Le plateau à remplir
-----------------	----------------------

## 5.12 Référence du fichier plateau.h

Gestion du plateau de [Jeu](#) Fonctions se rapportant à un plateau de jeu, sera sert à gérer la grille de Boggle.

```
#include "solution.h"
#include "couple.h"
```

## Structures de données

- struct [Plateau](#)  
*Grille du jeu Contient une grille permettant de jouer au Boggle.*

## Macros

- #define **TAILLE\_MOT\_MIN** 3
- #define **TAILLE\_MOT\_MAX** 32

## Fonctions

- [Plateau](#) [plateau\\_nouveau](#) (const [Couple](#) pTailleGrille)  
*Créer un nouveau plateau.*
- void [plateau\\_remplirGrilleAleatoire](#) ([Plateau](#) \*pPlateau)  
*Rempli une grille aléatoirement dans le plateau.*
- void [plateau\\_remplirGrillePredefinie](#) ([Plateau](#) \*pPlateau)  
*Rempli le plateau avec une grille prédéfinie de taille 4\*4 E D R C A N V C I R Q A E B R U.*
- double [plateau\\_probaLettre](#) (const char pLettre)  
*Retourne la probabilité d'apparition de la lettre passée en paramètre.*
- char [plateau\\_choisirLettre](#) (void)  
*Choisi une lettre avec la probabilité des lettres en fonctions de leurs importance d'apparition dans la langue française.*
- void [plateau\\_detruire](#) ([Plateau](#) \*pPlateau)  
*Détruit le plateau pPlateau.*

### 5.12.1 Description détaillée

Gestion du plateau de [Jeu](#) Fonctions se rapportant à un plateau de jeu, sera sert à gérer la grille de Boggle.

### 5.12.2 Documentation des fonctions

#### 5.12.2.1 char [plateau\\_choisirLettre](#) ( void )

Choisi une lettre avec la probabilité des lettres en fonctions de leurs importance d'apparition dans la langue française.

#### Renvoie

La lettre choisie

#### 5.12.2.2 void [plateau\\_detruire](#) ( [Plateau](#) \* [pPlateau](#) )

Détruit le plateau pPlateau.

#### Paramètres

<a href="#">pPlateau</a>	Le plateau à détruire
--------------------------	-----------------------

#### 5.12.2.3 [Plateau](#) [plateau\\_nouveau](#) ( const [Couple](#) [pTailleGrille](#) )

Créer un nouveau plateau.

#### Paramètres

<a href="#">pTailleGrille</a>	La taille de la grille à créer
-------------------------------	--------------------------------

**Renvoie**

Le nouveau plateau

**5.12.2.4 double plateau\_probaLettre ( const char *pLettre* )**

Retourne la probabilité d'apparition de la lettre passée en paramètre.

**Paramètres**

<i>pLettre</i>	La lettre pour laquelle retourner la probabilité
----------------	--

**Renvoie**

La probabilité

**5.12.2.5 void plateau\_remplirGrilleAleatoire ( Plateau \* *pPlateau* )**

Rempli une grille aléatoirement dans le plateau.

**Paramètres**

<i>pPlateau</i>	Le plateau à remplir
-----------------	----------------------

**5.12.2.6 void plateau\_remplirGrillePredefinie ( Plateau \* *pPlateau* )**

Rempli le plateau avec une grille prédéfinie de taille 4\*4 E D R C A N V C I R Q A E B R U.

**Paramètres**

<i>pPlateau</i>	Le plateau à remplir
-----------------	----------------------

**5.13 Référence du fichier `resolveur.c`**

Gestion du résolveur Permet de résoudre une grille de Boggle, une fois résolue, la solution se trouve dans le module [Solution](#).

```
#include <string.h>
#include "util.h"
#include "dictionnaire.h"
#include "resolveur.h"
#include "solution.h"
#include "interfaceTexte.h"
#include "jeu.h"
```

**Fonctions**

- void [recurse](#) (Plateau \*pPlateau, [Solution](#) \*pSolution, int x, int y, int depth, char \*choices, [Dico](#) pDico)  
*Fonction privée récursive permettant de résoudre une grille de Boggle.*
- [Solution](#) [resolveur\\_resolveUneCase](#) (Plateau pPlateau, [Dico](#) pDico, const [Couple](#) pCase)
- void [resolveTouteLaGrille](#) (Plateau \*pPlateau, [Dico](#) pDico)  
*Résoud une grille de boggle.*
- void [jeu\\_lancerModeSolveur](#) (Jeu pJeu)  
*Lance le jeu en mode solveur.*

### 5.13.1 Description détaillée

Gestion du résolveur Permet de résoudre une grille de Boggle, une fois résolue, la solution se trouve dans le module [Solution](#).

### 5.13.2 Documentation des fonctions

#### 5.13.2.1 `void jeu_lancerModeSolveur ( Jeu pJeu )`

Lance le jeu en mode solveur.

##### Paramètres

<code>pJeu</code>	le jeu à lancer
-------------------	-----------------

#### 5.13.2.2 `void recurse ( Plateau * pPlateau, Solution * pSolution, int x, int y, int depth, char * choices, Dico pDico )`

Fonction privée récursive permettant de résoudre une grille de Boggle.

##### Paramètres

<code>pPlateau</code>	Le plateau à résoudre
<code>x</code>	L'abscisse de la lettre pour laquelle on part
<code>y</code>	L'ordonnée de la lettre pour laquelle on part
<code>depth</code>	La profondeur à laquelle on se trouve
<code>choices</code>	Le mot formé actuellement
<code>pDico</code>	Le dictionnaire dans lequel chercher les mots

#### 5.13.2.3 `void resolveTouteLaGrille ( Plateau * pPlateau, Dico pDico )`

Résoud une grille de boggle.

##### Paramètres

<code>pPlateau</code>	Le plateau à résoudre
<code>choices</code>	
<code>pDico</code>	le dictionnaire

## 5.14 Référence du fichier `resolveur.h`

Gestion du résolveur Permet de résoudre une grille de Boggle, une fois résolue, la solution se trouve dans le module [Solution](#).

```
#include "plateau.h"
#include "dictionnaire.h"
#include "jeu.h"
```

### Fonctions

- `void resolveTouteLaGrille (Plateau *pPlateau, Dico pDico)`  
*Résoud une grille de boggle.*
- `Solution resolveur_resolveUneCase (Plateau pPlateau, Dico pDico, Couple pCase)`
- `void jeu_lancerModeSolveur (Jeu pJeu)`  
*Lance le jeu en mode solveur.*

### 5.14.1 Description détaillée

Gestion du résolveur Permet de résoudre une grille de Boggle, une fois résolue, la solution se trouve dans le module [Solution](#).

### 5.14.2 Documentation des fonctions

#### 5.14.2.1 void jeu\_lancerModeSolveur ( Jeu pJeu )

Lance le jeu en mode solveur.

Paramètres

<i>pJeu</i>	le jeu à lancer
-------------	-----------------

#### 5.14.2.2 void resolveTouteLaGrille ( Plateau \* pPlateau, Dico pDico )

Résoud une grille de boggle.

Paramètres

<i>pPlateau</i>	Le plateau à résoudre
<i>choices</i>	
<i>pDico</i>	le dictionnaire

## 5.15 Référence du fichier solution.c

Gère la solution d'une grille de boggle Toutes les fonctions se rapportant à la solution d'une grille.

```
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
#include "solution.h"
```

### Fonctions

- [Solution solution\\_creer](#) (void)  
*Créer la structure de données la solution.*
- [\\_Bool solution\\_motEstPresent](#) (const [Solution](#) pSolution, const char \*pMot)  
*Retourne vrai si le mot pMot est présent dans la solution.*
- [int solution\\_getNbMotsDebut](#) ([Solution](#) pSolution, char \*pDebut)  
*Retourne le nombre de mots présents dans la solution commençant pas la chaine de caractère pDebut.*
- [void solution\\_ajouterMot](#) ([Solution](#) \*pSolution, const char \*mot)  
*Ajoute un mot dans la solution pSolution.*
- [void solution\\_detruire](#) ([Solution](#) \*pSolution)  
*Détruit la solution.*
- [int solution\\_nbPoints](#) ([Solution](#) pSolution)  
*Retourn le nombre de points pouvant être obtenus avec la solution pSolution.*

### 5.15.1 Description détaillée

Gère la solution d'une grille de boggle Toutes les fonctions se rapportant à la solution d'une grille.

Voir également

plateau

## 5.15.2 Documentation des fonctions

### 5.15.2.1 void solution\_ajouterMot ( Solution \* pSolution, const char \* mot )

Ajoute un mot dans la solution pSolution.

Paramètres

<i>pSolution</i>	La solution dans laquelle ajouter le mot
<i>mot</i>	Le mot à ajouter

### 5.15.2.2 Solution solution\_cree ( void )

Créer la structure de données la solution.

Renvoie

La nouvelle solution

### 5.15.2.3 void solution\_detruire ( Solution \* pSolution )

Détruit la solution.

Paramètres

<i>pSolution</i>	La solution à détruire
------------------	------------------------

### 5.15.2.4 int solution\_getNbMotsDebut ( Solution pSolution, char \* pDebut )

Retourne le nombre de mots présents dans la solution commençant par la chaîne de caractère pDebut.

Paramètres

<i>pSolution</i>	La solution
<i>pDebut</i>	Le début de la chaîne

Renvoie

Le nombre de mots commençant par pDebut

### 5.15.2.5 Bool solution\_motEstPresent ( const Solution pSolution, const char \* pMot )

Retourne vrai si le mot pMot est présent dans la solution.

Paramètres

<i>pSolution</i>	La solution dans laquelle chercher
<i>pMot</i>	Le mot à chercher

Renvoie

Vrai si pMot est présent dans pSolution

#### 5.15.2.6 int solution\_nbPoints ( Solution pSolution )

Retourn le nombre de points pouvant être obtenus avec la solution pSolution.

##### Paramètres

<i>pSolution</i>	la solution pour laquelle compter le nombre de points
------------------	---

##### Renvoie

Le nombres de points à compter

## 5.16 Référence du fichier solution.h

Gère la solution d'une grille de boggle Toutes les fonctions se rapportant à la solution d'une grille.

### Structures de données

- struct [Solution](#)  
*Solution du jeu Contient une solution pour une grille de Boggle.*

### Fonctions

- [Solution solution\\_creer](#) (void)  
*Créer la structure de données la solution.*
- void [solution\\_ajouterMot](#) (Solution \*pSolution, const char \*mot)  
*Ajoute un mot dans la solution pSolution.*
- void [solution\\_afficher](#) (const Solution pSolution)
- void [solution\\_detruire](#) (Solution \*pSolution)  
*Détruit la solution.*
- \_Bool [solution\\_motEstPresent](#) (const Solution pSolution, const char \*pMot)  
*Retourne vrai si le mot pMot est présent dans la solution.*
- int [solution\\_nbPoints](#) (Solution pSolution)  
*Retourn le nombre de points pouvant être obtenus avec la solution pSolution.*
- int [solution\\_getNbMotsDebut](#) (Solution pSolution, char \*pDebut)  
*Retourne le nombre de mots présents dans la solution commençant pas la chaine de caractère pDebut.*

#### 5.16.1 Description détaillée

Gère la solution d'une grille de boggle Toutes les fonctions se rapportant à la solution d'une grille.

##### Voir également

plateau

## 5.16.2 Documentation des fonctions

#### 5.16.2.1 void solution\_ajouterMot ( Solution \* pSolution, const char \* mot )

Ajoute un mot dans la solution pSolution.

##### Paramètres

<i>pSolution</i>	La solution dans laquelle ajouter le mot
<i>mot</i>	Le mot à ajouter



**5.16.2.2 Solution solution\_creer ( void )**

Créer la structure de données la solution.

Renvoie

La nouvelle solution

**5.16.2.3 void solution\_detruire ( Solution \* pSolution )**

Détruit la solution.

Paramètres

<i>pSolution</i>	La solution à détruire
------------------	------------------------

**5.16.2.4 int solution\_getNbMotsDebut ( Solution pSolution, char \* pDebut )**

Retourne le nombre de mots présents dans la solution commençant pas la chaine de caractère pDebut.

Paramètres

<i>pSolution</i>	La solution
<i>pDebut</i>	Le début de la chaine

Renvoie

Le nombre de mots commençant par pDebut

**5.16.2.5 \_Bool solution\_motEstPresent ( const Solution pSolution, const char \* pMot )**

Retourne vrai si le mot pMot est présent dans la solution.

Paramètres

<i>pSolution</i>	La solution dans laquelle chercher
<i>pMot</i>	Le mot à chercher

Renvoie

Vrai si pMot est présent dans pSolution

**5.16.2.6 int solution\_nbPoints ( Solution pSolution )**

Retourn le nombre de points pouvant être obtenus avec la solution pSolution.

Paramètres

<i>pSolution</i>	la solution pour laquelle compter le nombre de points
------------------	---

## Renvoie

Le nombres de points à compter

## 5.17 Référence du fichier util.c

Fonctions utiles à l'ensemble du projet Toutes les fonctions de bases utiles au projet, ces fonctions travaillent sur des types de bae et ne sont pas spécifiques au projet, ce module permet de mieux organiser le code.

```
#include <string.h>
#include <wchar.h>
#include <time.h>
#include <stdlib.h>
#include <stdbool.h>
#include <locale.h>
#include "util.h"
#include "jeu.h"
#include "couple.h"
```

### Fonctions

- void **util\_affichageDebug** (const char \*pNomFonction, const char \*pChaine)  
*Affiche une chaîne de caractère sur la console uniquement si le define MODE\_DEBUG est vrai.*
- void **util\_afficherTableAscii** (void)  
*Affiche la tableau ASCII.*
- char \* **util\_supprimerAccents** (const char \*pChaine)  
*Supprime les accents d'une chaîne de caractère.*
- void **util\_uppercase** (char \*pChaine)  
*Modifie la chaîne de caractère afin qu'elle soit en majuscule.*
- char **util\_nbAleatoire** (const char pDebut, const char pFin)  
*Retourne un nombre aléatoire entre pDebut et pFin.*
- int **util\_substr** (const char \*chaîne, int debut, int fin, char \*result)  
*Crée une sous-chaîne de caractère de chaîne depuis début jusqu'à fin.*
- void **util\_echanger** (void \*\*tableau, int a, int b)  
*Echange deux variables dans un tableau de chaîne de caractères.*
- void **util\_quickSort** (void \*\*tableau, int debut, int fin, int(\*fctCmp)(void \*, void \*))  
*Effectue un tri sur une portion d'un tableau de chaîne de caractère s.*
- int **util\_strcmp** (void \*str1, void \*str2)
- void **util\_deplacerCurseurDunMot** (FILE \*pFichier, const int pSens)  
*Déplace le curseur dans le fichier au début du mot précédent ou suivant en fonction de pSens.*
- \_Bool **util\_isInArray** (const **Couple** \*pTableau, const int pTaille, const **Couple** pCase)  
*Retourne vrai si la case pCase est présente dans le tableau.*
- void **util\_conversionTemps** (const time\_t pTimestamp, int \*pMinutes, int \*pSecondes)  
*Convertis un nombre de secondes en minutes et secondes.*
- int **util\_cherchePremiereOccurrenceDansTableau** (char \*\*pTableau, const int pTaille, char \*pRecherche)
- void **util\_nettoyerConsole** (void)  
*Nettoie toute la console.*
- void **util\_viderBuffer** (void)
- int **util\_lireChaine** (char \*chaîne, int longueur)

### 5.17.1 Description détaillée

Fonctions utiles à l'ensemble du projet Toutes les fonctions de bases utiles au projet, ces fonctions travaillent sur des types de bae et ne sont pas spécifiques au projet, ce module permet de mieux organiser le code.

### 5.17.2 Documentation des fonctions

5.17.2.1 void util\_affichageDebug ( const char \* *pNomFonction*, const char \* *pChaine* )

Affiche une chaîne de caractère sur la console uniquement si le define MODE\_DEBUG est vrai.

## Paramètres

<i>pNomFonction</i>	Le nom de la fonction de laquelle est appelée cette fonction
<i>pChaine</i>	La chaîne de caractère à afficher

5.17.2.2 void util\_conversionTemps ( const time\_t *pTimestamp*, int \* *pMinutes*, int \* *pSecondes* )

Convertis un nombre de secondes en minutes et secondes.

## Paramètres

<i>pTimestamp</i>	Le nombre de secondes à convertir
<i>pMinutes</i>	Le nombre de minutes résultat
<i>pSecondes</i>	Le nombre de secondes résultat

5.17.2.3 void util\_deplacerCurseurDunMot ( FILE \* *pFichier*, const int *pSens* )

Déplace le curseur dans le fichier au début du mot précédent ou suivant en fonction de *pSens*.

## Paramètres

<i>pFichier</i>	Le fichier sur lequel s'applique le curseur. Celui-ci doit être ouvert en lecture
<i>pSens</i>	le sens du déplacement MOT_PRECEDENT ou MOT_SUIVANT

5.17.2.4 void util\_echanger ( void \*\* *tableau*, int *a*, int *b* )

Echange deux variables dans un tableau de chaîne de caractères.

## Paramètres

<i>tableau</i>	Le tableau sur lequel on effectue l'échange, tableau de n'importe quel type d'éléments.
<i>a</i>	La première valeur à échanger
<i>b</i>	La seconde valeur

5.17.2.5 Bool util\_isInArray ( const Couple \* *pTableau*, const int *pTaille*, const Couple *pCase* )

Retourne vrai si la case *pCase* est présente dans le tableau.

## Paramètres

<i>pTableau</i>	Le tableau sur lequel s'effectue le test
<i>pTaille</i>	La taille du tableau
<i>pCase</i>	La case à chercher

## Renvoie

Vrai si la case existe dans le tableau.

5.17.2.6 char util\_nbAleatoire ( const char *pDebut*, const char *pFin* )

Retourne un nombre aléatoire entre *pDebut* et *pFin*.

## Paramètres

<i>pDebut</i>	la borne inférieur
<i>pFin</i>	la borne supérieur

## Renvoie

Le nombre généré

5.17.2.7 void util\_quickSort ( void \*\* *tableau*, int *debut*, int *fin*, int(\*) (void \*, void \*) *fctCmp* )

Effectue un tri sur une portion d'un tableau de chaîne de caractère s.

Ce tri s'effectue en fonction du critère de la fonction de comparaison *fctCmp*. Ce tri utilise l'algorithme du tri rapide.

## Paramètres

<i>tableau</i>	Le tableau à trier. Les éléments sont de n'importe quel type.
<i>debut</i>	Le début du tableau à trier
<i>fin</i>	La fin du tableau à trier
<i>fctCmp</i>	Pointeurs de fonction pour la comparaison de deux éléments du tableau. retourne un nombre inférieur à 0 si param 1 doit être avant 2, supérieur à 0 si après ou égal à 0 si les deux paramètres sont égaux dans le classement

5.17.2.8 int util\_substr ( const char \* *chaîne*, int *debut*, int *fin*, char \* *result* )

Créer une sous-chaîne de caractère de chaîne depuis début jusqu'à fin.

## Paramètres

<i>chaîne</i>	La chaîne sur laquelle s'applique la fonction
<i>debut</i>	Le début du découpage
<i>fin</i>	La fin du découpage
<i>result</i>	La chaîne résultat

## Renvoie

La taille de la chaîne résultat

5.17.2.9 char\* util\_supprimerAccents ( const char \* *pChaîne* )

Supprime les accents d'une chaîne de caractère.

## Paramètres

<i>pChaîne</i>	La chaîne de caractère avec les accents
----------------	---

## Renvoie

La nouvelle chaîne de caractère sans accents

5.17.2.10 void util\_uppercase ( char \* *pChaîne* )

Modifie la chaîne de caractère afin qu'elle soit en majuscule.

## Paramètres

<i>pChaîne</i>	La chaîne de caractère à modifier
----------------	-----------------------------------

## 5.18 Référence du fichier util.h

Fonctions utiles à l'ensemble du projet Toutes les fonctions de bases utiles au projet, ces fonctions travaillent sur des types de bae et ne sont pas spécifiques au projet, ce module permet de mieux organiser le code.

```
#include <stdio.h>
#include <time.h>
#include "couple.h"
```

### Macros

```
- #define MODE_DEBUG 1
- #define MOT_PRECEDENT 1
- #define MOT_SUIVANT 1
```

### Fonctions

- void **util\_affichageDebug** (const char \*pNomFonction, const char \*pChaine)  
*Affiche une chaîne de caractère sur la console uniquement si le define MODE\_DEBUG est vrai.*
- char \* **util\_supprimerAccents** (const char \*pChaine)  
*Supprime les accents d'une chaîne de caractère.*
- char **util\_nbAleatoire** (const char pDebut, const char pFin)  
*Retourne un nombre aléatoire entre pDebut et pFin.*
- int **util\_substr** (const char \*chaîne, int debut, int fin, char \*result)  
*Crée une sous-chaîne de caractère de chaîne depuis début jusqu'à fin.*
- void **util\_echanger** (void \*\*tableau, int a, int b)  
*Echange deux variables dans un tableau de chaîne de caractères.*
- void **util\_quickSort** (void \*\*tableau, int debut, int fin, int(\*fctCmp)(void \*, void \*))  
*Effectue un tri sur une portion d'un tableau de chaîne de caractère s.*
- void **util\_deplacerCurseurDunMot** (FILE \*pFichier, const int pSens)  
*Déplace le curseur dans le fichier au début du mot précédent ou suivant en fonction de pSens.*
- \_Bool **util\_isInArray** (const **Couple** \*pTableau, const int pTaille, const **Couple** pCase)  
*Retourne vrai si la case pCase est présente dans le tableau.*
- void **util\_conversionTemps** (const time\_t pTimestamp, int \*pMinutes, int \*pSecondes)  
*Convertit un nombre de secondes en minutes et secondes.*
- int **util\_strcmp** (void \*str1, void \*str2)
- int **util\_cherchePremiereOccurrenceDansTableau** (char \*\*pTableau, const int pTaille, char \*pRecherche)
- void **util\_nettoyerConsole** (void)  
*Nettoie toute la console.*
- int **util\_lireChaine** (char \*chaîne, int longueur)

### 5.18.1 Description détaillée

Fonctions utiles à l'ensemble du projet Toutes les fonctions de bases utiles au projet, ces fonctions travaillent sur des types de bae et ne sont pas spécifiques au projet, ce module permet de mieux organiser le code.

### 5.18.2 Documentation des fonctions

#### 5.18.2.1 void util\_affichageDebug ( const char \* pNomFonction, const char \* pChaine )

Affiche une chaîne de caractère sur la console uniquement si le define MODE\_DEBUG est vrai.

#### Paramètres

<i>pNomFonction</i>	Le nom de la fonction de laquelle est appelée cette fonction
<i>pChaine</i>	La chaîne de caractère à afficher

#### 5.18.2.2 void util\_conversionTemps ( const time\_t *pTimestamp*, int \* *pMinutes*, int \* *pSecondes* )

Convertis un nombre de secondes en minutes et secondes.

##### Paramètres

<i>pTimestamp</i>	Le nombre de secondes à convertir
<i>pMinutes</i>	Le nombre de minutes résultat
<i>pSecondes</i>	Le nombre de secondes résultat

#### 5.18.2.3 void util\_deplacerCurseurDunMot ( FILE \* *pFichier*, const int *pSens* )

Déplace le curseur dans le fichier au début du mot précédent ou suivant en fonction de *pSens*.

##### Paramètres

<i>pFichier</i>	Le fichier sur lequel s'applique le curseur. Celui-ci doit être ouvert en lecture
<i>pSens</i>	le sens du déplacement MOT_PRECEDENT ou MOT_SUIVANT

#### 5.18.2.4 void util\_echanger ( void \*\* *tableau*, int *a*, int *b* )

Echange deux variables dans un tableau de chaîne de caractères.

##### Paramètres

<i>tableau</i>	Le tableau sur lequel on effectue l'échange, tableau de n'importe quel type d'éléments.
<i>a</i>	La première valeur à échanger
<i>b</i>	La seconde valeur

#### 5.18.2.5 \_Bool util\_isInArray ( const Couple \* *pTableau*, const int *pTaille*, const Couple *pCase* )

Retourne vrai si la case *pCase* est présente dans le tableau.

##### Paramètres

<i>pTableau</i>	Le tableau sur lequel s'effectue le test
<i>pTaille</i>	La taille du tableau
<i>pCase</i>	La case à chercher

##### Renvoie

Vrai si la case existe dans le tableau.

#### 5.18.2.6 char util\_nbAleatoire ( const char *pDebut*, const char *pFin* )

Retourne un nombre aléatoire entre *pDebut* et *pFin*.

##### Paramètres

<i>pDebut</i>	la borne inférieure
<i>pFin</i>	la borne supérieure

##### Renvoie

Le nombre généré

**5.18.2.7 void util\_quickSort ( void \*\* *tableau*, int *debut*, int *fin*, int(\*) (void \*, void \*) *fctCmp* )**

Effectue un tri sur une portion d'un tableau de chaîne de caractère s.

Ce tri s'effectue en fonction du critère de la fonction de comparaison *fctCmp*. Ce tri utilise l'algorithme du tri rapide.

**Paramètres**

<i>tableau</i>	Le tableau à trier. Les éléments sont de n'importe quel type.
<i>debut</i>	Le début du tableau à trier
<i>fin</i>	La fin du tableau à trier
<i>fctCmp</i>	Pointeurs de fonction pour la comparaison de deux éléments du tableau. retourne un nombre inférieur à 0 si param 1 doit être avant 2, supérieur à 0 si après ou égal à 0 si les deux paramètres sont égaux dans le classement

**5.18.2.8 int util\_substr ( const char \* *chaîne*, int *debut*, int *fin*, char \* *result* )**

Créer une sous-chaîne de caractère de chaîne depuis début jusqu'à fin.

**Paramètres**

<i>chaîne</i>	La chaîne sur laquelle s'applique la fonction
<i>debut</i>	Le début du découpage
<i>fin</i>	La fin du découpage
<i>result</i>	La chaîne résultat

**Renvoie**

La taille de la chaîne résultat

**5.18.2.9 char\* util\_supprimerAccents ( const char \* *pChaîne* )**

Supprime les accents d'une chaîne de caractère.

**Paramètres**

<i>pChaîne</i>	La chaîne de caractère avec les accents
----------------	---

**Renvoie**

La nouvelle chaîne de caractère sans accents

# Index

- Couple, 7
  - x, 7
  - y, 7
- couple.h, 11
- Dico, 7
  - dico, 8
  - marqueurs, 8
- dico
  - Dico, 8
  - Jeu, 8
- dictionnaire.c, 11
  - dictionnaire\_nouveau, 12
  - dictionnaire\_rechercheDichotomique, 12
  - dictionnaire\_motDansDico, 12
- dictionnaire.h, 12
  - dictionnaire\_nouveau, 13
  - dictionnaire\_rechercheDichotomique, 13
  - dictionnaire\_motDansDico, 13
- dictionnaire\_nouveau
  - dictionnaire.c, 12
  - dictionnaire.h, 13
- dictionnaire\_rechercheDichotomique
  - dictionnaire.c, 12
  - dictionnaire.h, 13
- dictionnaire\_motDansDico
  - dictionnaire.c, 12
  - dictionnaire.h, 13
- fctMenu
  - interfaceNcurses.c, 14
- grille
  - Plateau, 9
- interfaceNcurses.c, 13
  - fctMenu, 14
  - interfaceNcurses\_afficherBoiteDialogue, 14
  - interfaceNcurses\_afficherFenetreJeu, 15
  - interfaceNcurses\_afficherFenetreSolution, 15
  - interfaceNcurses\_afficherGrille, 15
  - interfaceNcurses\_afficherSolution, 15
  - interfaceNcurses\_afficherTitre, 15
  - interfaceNcurses\_initialiser, 15
  - interfaceNcurses\_menu, 16
  - interfaceNcurses\_terminer, 16
  - jeu\_lancerModeNcurses, 16
- interfaceNcurses.h, 16
  - interfaceNcurses\_afficherBoiteDialogue, 17
  - interfaceNcurses\_afficherFenetreJeu, 17
  - interfaceNcurses\_afficherFenetreSolution, 17
  - interfaceNcurses\_afficherGrille, 17
  - interfaceNcurses\_afficherSolution, 18
  - interfaceNcurses\_afficherTitre, 18
  - interfaceNcurses\_initialiser, 18
  - interfaceNcurses\_menu, 18
  - interfaceNcurses\_terminer, 18
  - jeu\_lancerModeNcurses, 19
- interfaceNcurses\_afficherBoiteDialogue
  - interfaceNcurses.c, 14
  - interfaceNcurses.h, 17
- interfaceNcurses\_afficherFenetreJeu
  - interfaceNcurses.c, 15
  - interfaceNcurses.h, 17
- interfaceNcurses\_afficherFenetreSolution
  - interfaceNcurses.c, 15
  - interfaceNcurses.h, 17
- interfaceNcurses\_afficherGrille
  - interfaceNcurses.c, 15
  - interfaceNcurses.h, 17
- interfaceNcurses\_afficherSolution
  - interfaceNcurses.c, 15
  - interfaceNcurses.h, 18
- interfaceNcurses\_afficherTitre
  - interfaceNcurses.c, 15
  - interfaceNcurses.h, 18
- interfaceNcurses\_initialiser
  - interfaceNcurses.c, 15
  - interfaceNcurses.h, 18
- interfaceNcurses\_menu
  - interfaceNcurses.c, 16
  - interfaceNcurses.h, 18
- interfaceNcurses\_terminer
  - interfaceNcurses.c, 16
  - interfaceNcurses.h, 18
- interfaceTexte.c, 19
  - interfaceTexte\_afficherSolution, 19
  - jeu\_lancerModeTexte, 19
- interfaceTexte.h, 20
  - jeu\_lancerModeTexte, 20
- interfaceTexte\_afficherSolution
  - interfaceTexte.c, 19
- Jeu, 8
  - dico, 8
  - plateau, 8
  - solutionUtilisateur, 8
  - timestampDepart, 9
- jeu.c, 20
  - jeu\_compteurClaque, 21



- jeu\_lancer, 21
- jeu\_nouveau, 21
- jeu\_proposerMot, 21
- jeu\_stopper, 21
- jeu\_tempsRestant, 22
- jeu.h, 22
  - jeu\_compteurClaque, 23
  - jeu\_lancer, 23
  - jeu\_nouveau, 23
  - jeu\_proposerMot, 23
  - jeu\_stopper, 23
  - jeu\_tempsRestant, 23
- jeu\_compteurClaque
  - jeu.c, 21
  - jeu.h, 23
- jeu\_lancer
  - jeu.c, 21
  - jeu.h, 23
- jeu\_lancerModeNcurses
  - interfaceNcurses.c, 16
  - interfaceNcurses.h, 19
- jeu\_lancerModeSolveur
  - resolveur.c, 29
  - resolveur.h, 30
- jeu\_lancerModeTexte
  - interfaceTexte.c, 19
  - interfaceTexte.h, 20
- jeu\_nouveau
  - jeu.c, 21
  - jeu.h, 23
- jeu\_proposerMot
  - jeu.c, 21
  - jeu.h, 23
- jeu\_stopper
  - jeu.c, 21
  - jeu.h, 23
- jeu\_tempsRestant
  - jeu.c, 22
  - jeu.h, 23
- main
  - main.c, 24
- main.c, 24
  - main, 24
  - mauvaisParametre, 25
- marqueurs
  - Dico, 8
- mauvaisParametre
  - main.c, 25
- mots
  - Solution, 10
- nbMots
  - Solution, 10
- Plateau, 9
  - grille, 9
  - solution, 9
  - tailleGrille, 9
- plateau
  - Jeu, 8
- plateau.c, 25
  - plateau\_choisirLettre, 25
  - plateau\_detruire, 25
  - plateau\_nouveau, 26
  - plateau\_probaLettre, 26
  - plateau\_remplirGrilleAleatoire, 26
  - plateau\_remplirGrillePredefinie, 26
- plateau.h, 26
  - plateau\_choisirLettre, 27
  - plateau\_detruire, 27
  - plateau\_nouveau, 27
  - plateau\_probaLettre, 28
  - plateau\_remplirGrilleAleatoire, 28
  - plateau\_remplirGrillePredefinie, 28
- plateau\_choisirLettre
  - plateau.c, 25
  - plateau.h, 27
- plateau\_detruire
  - plateau.c, 25
  - plateau.h, 27
- plateau\_nouveau
  - plateau.c, 26
  - plateau.h, 27
- plateau\_probaLettre
  - plateau.c, 26
  - plateau.h, 28
- plateau\_remplirGrilleAleatoire
  - plateau.c, 26
  - plateau.h, 28
- plateau\_remplirGrillePredefinie
  - plateau.c, 26
  - plateau.h, 28
- recurse
  - resolveur.c, 29
- resolveTouteLaGrille
  - resolveur.c, 29
  - resolveur.h, 30
- resolveur.c, 28
  - jeu\_lancerModeSolveur, 29
  - recurse, 29
  - resolveTouteLaGrille, 29
- resolveur.h, 29
  - jeu\_lancerModeSolveur, 30
  - resolveTouteLaGrille, 30
- Solution, 9
  - mots, 10
  - nbMots, 10
- solution
  - Plateau, 9
- solution.c, 30
  - solution\_ajouterMot, 31
  - solution\_creer, 31
  - solution\_detruire, 31
  - solution\_getNbMotsDebut, 31
  - solution\_motEstPresent, 31

- solution\_nbPoints, 31
- solution.h, 32
  - solution\_ajouterMot, 32
  - solution\_creer, 32
  - solution\_detruire, 33
  - solution\_getNbMotsDebut, 33
  - solution\_motEstPresent, 33
  - solution\_nbPoints, 33
- solution\_ajouterMot
  - solution.c, 31
  - solution.h, 32
- solution\_creer
  - solution.c, 31
  - solution.h, 32
- solution\_detruire
  - solution.c, 31
  - solution.h, 33
- solution\_getNbMotsDebut
  - solution.c, 31
  - solution.h, 33
- solution\_motEstPresent
  - solution.c, 31
  - solution.h, 33
- solution\_nbPoints
  - solution.c, 31
  - solution.h, 33
- solutionUtilisateur
  - Jeu, 8
- tailleGrille
  - Plateau, 9
- timestampDepart
  - Jeu, 9
- util.c, 34
  - util\_affichageDebug, 34
  - util\_conversionTemps, 35
  - util\_deplacerCurseurDunMot, 35
  - util\_echanger, 35
  - util\_isInArray, 35
  - util\_nbAleatoire, 35
  - util\_quickSort, 36
  - util\_substr, 36
  - util\_supprimerAccents, 36
  - util\_uppercase, 36
- util.h, 37
  - util\_affichageDebug, 37
  - util\_conversionTemps, 37
  - util\_deplacerCurseurDunMot, 38
  - util\_echanger, 38
  - util\_isInArray, 38
  - util\_nbAleatoire, 38
  - util\_quickSort, 38
  - util\_substr, 39
  - util\_supprimerAccents, 39
- util\_affichageDebug
  - util.c, 34
  - util.h, 37
- util\_conversionTemps
  - util.c, 35
  - util.h, 37
- util\_deplacerCurseurDunMot
  - util.c, 35
  - util.h, 38
- util\_echanger
  - util.c, 35
  - util.h, 38
- util\_isInArray
  - util.c, 35
  - util.h, 38
- util\_nbAleatoire
  - util.c, 35
  - util.h, 38
- util\_quickSort
  - util.c, 36
  - util.h, 38
- util\_substr
  - util.c, 36
  - util.h, 39
- util\_supprimerAccents
  - util.c, 36
  - util.h, 39
- util\_uppercase
  - util.c, 36
- x
  - Couple, 7
- y
  - Couple, 7