

Jeu de Boggle

1

Généré par Doxygen 1.8.1.2

Mardi Avril 30 2013 10 :07 :23

Table des matières

1	Index des structures de données	1
1.1	Structures de données	1
2	Index des fichiers	3
2.1	Liste des fichiers	3
3	Documentation des structures de données	5
3.1	Référence de la structure Case	5
3.2	Référence de la structure Dico	5
3.3	Référence de la structure Jeu	5
3.4	Référence de la structure Plateau	5
3.5	Référence de la structure Solution	6
4	Documentation des fichiers	7
4.1	Référence du fichier dictionnaire.c	7
4.1.1	Description détaillée	7
4.1.2	Documentation des fonctions	7
4.1.2.1	dictionnaire_nouveau	7
4.1.2.2	dictionnaire_rechercheDichotomique	8
4.1.2.3	dictionnaire_motDansDico	8
4.2	Référence du fichier interfaceNcurses.c	8
4.2.1	Description détaillée	8
4.3	Référence du fichier jeu.c	9
4.3.1	Description détaillée	9
4.3.2	Documentation des fonctions	9
4.3.2.1	jeu_compteurClaque	9
4.3.2.2	jeu_lancer	9
4.3.2.3	jeu_lancerModeTexte	10
4.3.2.4	jeu_nouveau	10
4.3.2.5	jeu_proposerMot	10
4.3.2.6	jeu_stopper	10
4.3.2.7	jeu_tempsRestant	10

4.4	Référence du fichier plateau.c	11
4.4.1	Description détaillée	11
4.4.2	Documentation des fonctions	11
4.4.2.1	plateau_choisirLettre	11
4.4.2.2	plateau_detruire	11
4.4.2.3	plateau_nouveau	11
4.4.2.4	plateau_probaLettre	12
4.4.2.5	plateau_remplirGrilleAleatoire	12
4.4.2.6	plateau_remplirGrillePredefinie	12
4.5	Référence du fichier resolveur.c	12
4.5.1	Description détaillée	12
4.5.2	Documentation des fonctions	13
4.5.2.1	recurse	13
4.5.2.2	resolveur	13
4.6	Référence du fichier util.c	13
4.6.1	Description détaillée	14
4.6.2	Documentation des fonctions	14
4.6.2.1	util_affichageDebug	14
4.6.2.2	util_conversionTemps	14
4.6.2.3	util_deplacerCurseurDunMot	14
4.6.2.4	util_echanger	14
4.6.2.5	util_isInArray	14
4.6.2.6	util_nbAleatoire	15
4.6.2.7	util_quickSort	15
4.6.2.8	util_substr	15
4.6.2.9	util_supprimerAccents	15
4.6.2.10	util_uppercase	16

Chapitre 1

Index des structures de données

1.1 Structures de données

Liste des structures de données avec une brève description :

Case	5
Dico	5
Jeu	5
Plateau	5
Solution	6

Chapitre 2

Index des fichiers

2.1 Liste des fichiers

Liste de tous les fichiers documentés avec une brève description :

case.h	??
dictionnaire.c	
Gestion du dictionnaire Fonctions de gestion du dictionnaire, le dictionnaire doit être dans un fichier texte avec un mot différent par ligne	7
dictionnaire.h	??
interfaceNcurses.c	
Affichage avec la bibliothèque Ncurses Affiche et permet de jouer au jeu de Boggle avec un affichage utilisant la bibliothèque Ncurses	8
interfaceNcurses.h	??
interfaceTexte.h	??
jeu.c	
Gestion du jeu de Boggle Les fonctions permettant de gérer le Jeu de Boggle	9
jeu.h	??
plateau.c	
Gestion du plateau de Jeu Fonctions se rapportant à un plateau de jeu, sera sert à gérer la grille de Boggle	11
plateau.h	??
resolveur.c	
Gestion du résolveur Permet de résoudre une grille de Boggle, une fois résolue, la solution se trouve dans le module Solution	12
resolveur.h	??
solution.h	??
util.c	
Fonctions utiles à l'ensemble du projet Toutes les fonctions de bases utiles au projet, ces fonctions travaillent sur des types de base et ne sont pas spécifiques au projet, ce module permet de mieux organiser le code	13
util.h	??

Chapitre 3

Documentation des structures de données

3.1 Référence de la structure Case

Champs de données

- unsigned char **i**
- unsigned char **j**

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

- case.h

3.2 Référence de la structure Dico

Champs de données

- FILE * **dico**
- long int **marqueurs** [32]

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

- dictionnaire.h

3.3 Référence de la structure Jeu

Champs de données

- [Plateau](#) **plateau**
- [Dico](#) **dico**
- time_t **timestampDepart**
- [Solution](#) **solutionUtilisateur**

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

- jeu.h

3.4 Référence de la structure Plateau

Champs de données

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

La documentation de cette structure a été générée à partir du fichier suivant :
– plateau.h

3.5 Référence de la structure Solution

Champs de données

- char ** **mots**
- unsigned int **nbMots**
- unsigned int **nbPointsTotal**

La documentation de cette structure a été générée à partir du fichier suivant :
– solution.h

Chapitre 4

Documentation des fichiers

4.1 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 "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.

4.1.1 Description détaillée

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

4.1.2 Documentation des fonctions

4.1.2.1 [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

4.1.2.2 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 <i>pMotLePlusProche</i> == <i>pMotAChercher</i> , il est présent dans le dictionnaire

4.1.2.3 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.

4.2 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 <stdlib.h>
#include <string.h>
#include "interfaceNcurses.h"
#include "plateau.h"
#include "jeu.h"
#include "util.h"
```

Fonctions

- void **interfaceNcurses_afficherSolution** (const [Solution](#) pSolution)
- WINDOW * **interfaceNcurses_initialiser** (void)
- void **interfaceNcurses_afficherGrille** (const [Plateau](#) pPlateau, const [Case](#) pSelectedCase, const [Case](#) *pUsed-Case, const int pLgUsedCase)
- void **interfaceNcurses_terminer** (WINDOW *fenetre)
- void **jeu_lancerModeNcurses** ([Jeu](#) pJeu)

4.2.1 Description détaillée

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

4.3 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 unsigned char pTaillePlateau)`
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.
- `void jeu_lancerModeTexte (Jeu pJeu)`
Lance le jeu en mode texte.

4.3.1 Description détaillée

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

Voir également

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

4.3.2 Documentation des fonctions

4.3.2.1 `bool jeu_compteurClaque (const Jeu pJeu)`

Retourne vrai si le temps est écoulé faux sinon.

Paramètres

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

Renvoie

Vrai si le temps est écoulé faux sinon

4.3.2.2 `void jeu_lancer (Jeu * pJeu)`

Lance le jeu.

Paramètres

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

4.3.2.3 void jeu_lancerModeTexte (Jeu *pJeu*)

Lance le jeu en mode texte.

Paramètres

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

4.3.2.4 Jeu jeu_nouveau (const char * *pNomDico*, const unsigned char *pTaillePlateau*)

Créer un nouveau jeu.

Paramètres

<i>pNomDico</i>	Le dictionnaire utilisé dans le Jeu
<i>pTaillePlateau</i>	La taille du plateau à créer

Renvoie

Le nouveau [Jeu](#)

4.3.2.5 _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

4.3.2.6 void jeu_stopper (Jeu *pJeu*)

Fin du jeu.

Paramètres

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

4.3.2.7 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

4.4 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 unsigned char 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.

4.4.1 Description détaillée

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

4.4.2 Documentation des fonctions

4.4.2.1 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

4.4.2.2 void plateau_detruire (Plateau * pPlateau)

Détruit le plateau pPlateau.

Paramètres

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

4.4.2.3 Plateau plateau_nouveau (const unsigned char pTailleGrille)

Créer un nouveau plateau.

Paramètres

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

Renvoie

Le nouveau plateau

4.4.2.4 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

4.4.2.5 void plateau_remplirGrilleAleatoire (Plateau * pPlateau)

Rempli une grille aléatoirement dans le plateau.

Paramètres

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

4.4.2.6 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
-----------------	----------------------

4.5 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 "dictionnaire.h"
#include "resolveur.h"
#include "solution.h"
```

Fonctions

- void [recurse](#) (Plateau *pPlateau, int x, int y, int depth, char *choices, [Dico](#) pDico)
Fonction privée récursive permettant de résoudre une grille de Boggle.
- void [resolveur](#) (Plateau *pPlateau, [Dico](#) pDico)
Résoud une grille de boggle.

4.5.1 Description détaillée

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

4.5.2 Documentation des fonctions

4.5.2.1 void recurse (Plateau * pPlateau, int x, int y, int depth, char * choices, Dico pDico)

Fonction privée récursive permettant de résoudre une grille de Boggle.

Paramètres

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

4.5.2.2 void resolveur (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

4.6 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 <stdlib.h>
#include <stdbool.h>
#include "util.h"
#include "jeu.h"
#include "case.h"
```

Fonctions

- void [util_affichageDebug](#) (const char *pNomFonction, const char *pChaine)
Affiche une chaine 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 chaine de caratère.
- void [util_uppercase](#) (char *pChaine)
Modifie la chaine 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 *chaine, int debut, int fin, char *result)
Créer une sous-chaine de caractère de chaine depuis début jusqu'à fin.
- void [util_echanger](#) (char **tableau, int a, int b)
Echange deux variables dans un tableau de chaine de caractères.
- void [util_quickSort](#) (char **tableau, int debut, int fin)
Effectue un tri sur une porition d'un tableau de chaine 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 `Case` *pTableau, const int pTaille, const `Case` 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.

4.6.1 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.

4.6.2 Documentation des fonctions

4.6.2.1 `void util_affichageDebug (const char * pNomFonction, const char * pChaine) [inline]`

Affiche une chaîne de caractère sur la console uniquement si le define `MODE_DEBUG` est vrai.

Paramètres

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

4.6.2.2 `void util_conversionTemps (const time_t pTimestamp, int * pMinutes, int * pSecondes)`

Convertis un nombre de secondes en minutes et secondes.

Paramètres

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

4.6.2.3 `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

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

4.6.2.4 `void util_echanger (char ** tableau, int a, int b)`

Echange deux variables dans un tableau de chaîne de caractères.

Paramètres

<code>tableau</code>	Le tableau sur lequel on effectue l'échange
<code>a</code>	La première valeur à échanger
<code>b</code>	La seconde valeur

4.6.2.5 `_Bool util_isInArray (const Case * pTableau, const int pTaille, const Case pCase)`

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

Paramètres

<i>pTableau</i>	Le tableau sur lequelle 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.

4.6.2.6 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

4.6.2.7 void util_quickSort (char ** *tableau*, int *debut*, int *fin*)

Effectue un tri sur une porition d'un tableau de chaine de caractère s.

Ce tri utilise l'algorithme du tri rapide.

Paramètres

<i>tableau</i>	Le tableau à trier
<i>debut</i>	Le début du tableau à trier
<i>fin</i>	La fin du tableau à trier

4.6.2.8 int util_substr (const char * *chaine*, int *debut*, int *fin*, char * *result*)

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

Paramètres

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

Renvoie

La taille de la chaine résultat

4.6.2.9 char* util_supprimerAccents (const char * *pChaine*)

Supprime les accents d'une chaine de caratère.

Paramètres

<i>pChaine</i>	La chaine de caractère avec les accents
----------------	---

Renvoie

La nouvelle chaîne de caractère sans accents

4.6.2.10 void util_uppercase (char * *pChaine*)

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

Paramètres

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

Index

Case, [5](#)

Dico, [5](#)

dictionnaire.c, [7](#)

 dictionnaire_nouveau, [7](#)

 dictionnaire_rechercheDichotomique, [8](#)

 dictionnaire_motDansDico, [8](#)

dictionnaire_nouveau

 dictionnaire.c, [7](#)

dictionnaire_rechercheDichotomique

 dictionnaire.c, [8](#)

dictionnaire_motDansDico

 dictionnaire.c, [8](#)

interfaceNcurses.c, [8](#)

Jeu, [5](#)

jeu.c, [9](#)

 jeu_compteurClaque, [9](#)

 jeu_lancer, [9](#)

 jeu_lancerModeTexte, [10](#)

 jeu_nouveau, [10](#)

 jeu_proposerMot, [10](#)

 jeu_stopper, [10](#)

 jeu_tempsRestant, [10](#)

jeu_compteurClaque

 jeu.c, [9](#)

jeu_lancer

 jeu.c, [9](#)

jeu_lancerModeTexte

 jeu.c, [10](#)

jeu_nouveau

 jeu.c, [10](#)

jeu_proposerMot

 jeu.c, [10](#)

jeu_stopper

 jeu.c, [10](#)

jeu_tempsRestant

 jeu.c, [10](#)

Plateau, [5](#)

plateau.c, [11](#)

 plateau_choisirLettre, [11](#)

 plateau_detruire, [11](#)

 plateau_nouveau, [11](#)

 plateau_probaLettre, [12](#)

 plateau_remplirGrilleAleatoire, [12](#)

 plateau_remplirGrillePredefinie, [12](#)

plateau_choisirLettre

 plateau.c, [11](#)

plateau_detruire

 plateau.c, [11](#)

plateau_nouveau

 plateau.c, [11](#)

plateau_probaLettre

 plateau.c, [12](#)

plateau_remplirGrilleAleatoire

 plateau.c, [12](#)

plateau_remplirGrillePredefinie

 plateau.c, [12](#)

recurse

 resolveur.c, [13](#)

resolveur

 resolveur.c, [13](#)

resolveur.c, [12](#)

 recurse, [13](#)

 resolveur, [13](#)

Solution, [6](#)

util.c, [13](#)

 util_affichageDebug, [14](#)

 util_conversionTemps, [14](#)

 util_deplacerCurseurDunMot, [14](#)

 util_echanger, [14](#)

 util_isInArray, [14](#)

 util_nbAleatoire, [15](#)

 util_quickSort, [15](#)

 util_substr, [15](#)

 util_supprimerAccents, [15](#)

 util_uppercase, [16](#)

util_affichageDebug

 util.c, [14](#)

util_conversionTemps

 util.c, [14](#)

util_deplacerCurseurDunMot

 util.c, [14](#)

util_echanger

 util.c, [14](#)

util_isInArray

 util.c, [14](#)

util_nbAleatoire

 util.c, [15](#)

util_quickSort

 util.c, [15](#)

util_substr

 util.c, [15](#)

util_supprimerAccents

 util.c, [15](#)

util_uppercase
 util.c, [16](#)