Jeu de Boggle

1

Généré par Doxygen 1.8.1.2

Mardi Avril 30 2013 10 :29 :28

Table des matières

Chapitre 1

Index des structures de données

1.1 Structures de données

Liste des structures de données avec une brève description :

Chapitre 2

Index des fichiers

2.1 Liste des fichiers

Liste de tous les fichiers documentés avec une brève description :

Chapitre 3

Documentation des structures de données

3.1 Référence de la structure Case

Une case de la grille.

`#include <case.h>`

Champs de données

- unsigned char i
- unsigned char j

3.1.1 Description détaillée

Une case de la grille.

3.1.2 Documentation des champs

3.1.2.1 unsigned char Case : :i

Abscisse

3.1.2.2 unsigned char Case : :j

ordonnée

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

3.2 Référence de la structure Dico

Dictionnaire Contient le dictionnaire du jeu.

`#include <dictionnaire.h>`

Champs de données

- FILE * dico
- long int marqueurs [32]

3.2.1 Description détaillée

Dictionnaire Contient le dictionnaire du jeu.

3.2.2 Documentation des champs

3.2.2.1 FILE* Dico : :dico

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

3.2.2.2 long int Dico : :marqueurs[32]

Contient les positions de chacunes des lettres de l'alphabets dans le fichier

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

3.3 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

3.3.1 Description détaillée

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

Voir également

Dico, Plateau, Solution

3.3.2 Documentation des champs

3.3.2.1 Dico Jeu : :dico

Le dictionnaire

3.3.2.2 Plateau Jeu : :plateau

Le plateau de Jeu

3.3.2.3 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

3.3.2.4 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

3.4 Référence de la structure Plateau

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

`#include <plateau.h>`

Champs de données

– char ** grille
– unsigned int tailleGrille
– int ** **grid**
– Solution solution

3.4.1 Description détaillée

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

3.4.2 Documentation des champs

3.4.2.1 char** Plateau : :grille

La grille de boggle

3.4.2.2 Solution Plateau : :solution

La solution complète de cette grille de Boggle

3.4.2.3 unsigned int Plateau : :tailleGrille

La taille de la grille

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

3.5 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

3.5.1 Description détaillée

Solution du jeu Contient une solution pour une grille de Boggle.

3.5.2 Documentation des champs

3.5.2.1 char** Solution : :mots

Les mots présents dans la solution

3.5.2.2 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 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 dictonnaire_motDansDico (Dico pDictionnaire, char *pMot)
  Permet de savoir si un mot est présent dans le dictionnaire.
- void dictionnaire_rechercheDichotomique (Dico pDictionnaire, char *pMotAChercher, char *pMotLePlus-Proche)
  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

pNomFichier	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

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

4.1.2.3 int dictonnaire_motDansDico (Dico *pDictionnaire,* char * *pMot*)

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

Paramètres

pDictionnaire	Le dictionnaire dans lequel chercher
pMot	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, Dico, Solution

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

pJeu	Le jeu pourlequel on veut saoir 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

pJeu	Le jeu à lancer

4.3.2.3 void jeu_lancerModeTexte (Jeu *pJeu*)

Lance le jeu en mode texte.

Paramètres

pJeu	le jeu à lancer

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

Créer un nouveau jeu.

Paramètres

pNomDico	Le dictionnaire utilisé dans le Jeu
pTaillePlateau	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

pJeu	Le jeu pour lequel on veut proposer un mot
pMot	Le mot proposé

Renvoie

Vrai si le mot est validé

4.3.2.6 void jeu_stopper (Jeu *pJeu*)

Fin du jeu.

Paramètres

pJeu	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

pJeu	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 letre 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 letre 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

pPlateau	Le plateau à détruire

4.4.2.3 Plateau plateau_nouveau (const unsigned char *pTailleGrille*)

Créer un nouveau plateau.

Paramètres

pTailleGrille	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

pLettre	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

pPlateau	Le plateau à remplire

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

pPlateau	Le plateau à remplire

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

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

4.5.2.2 void resolveur (Plateau * *pPlateau,* Dico *pDico*)

Résoud une grille de boggle.

Paramètres

pPlateau	Le plateau à résoudre
choices	
pDico	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 spécfiques au projet, ce module permet de meux 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 curseurdans 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 spécfiques au projet, ce module permet de meux 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 chaine de caractère sur la console uniquement si le define MODE_DEBUG est vrai.

Paramètres

pNomFonction	Le nom de la fonction de laquelle est appellée cette fonction
pChaine	La chaine 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

pTimestamp	Le nombre de secondes à converti
pMinutes	Le nombre de minutes résultat
pSecondes	Le nombre de secondes résultat

4.6.2.3 void util_deplacerCurseurDunMot (FILE ∗ *pFichier,* const int *pSens*)

Déplace le curseurdans le fichier au début du mot précédent ou suivant en fonction de pSens.

Paramètres

pFichier	Le fichier surlequelle s'appliqeu le curseur. Celui-ci doit être ouvert en lecture
pSens	le sens du déplacement MOT_PRECEDENT ou MOT_SUIVANT

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

Echange deux variables dans un tableau de chaine de caractères.

Paramètres

tableau	Le tableau sur laquelle on effectue l'échange
a	La première valeure à échanger
b	La seconde valeure

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

pTableau	Le tableau sur lequele s'effectue le test
pTaille	La taille du tableau
pCase	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

pDebut	la borne inférieur
pFin	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

tableau	Le tableau à trier
debut	Le début du tableau à trier
fin	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

chaine	La chaine sur laquelle s'applique la fonction
debut	Le début du découpage
fin	La fin du découpage
result	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

pChaine	La chaine de caractère avec les accents

Renvoie

La nouvelle chaine de caractère sans accents

4.6.2.10 void util_uppercase (char * *pChaine*)

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

Paramètres

pChaine	La chaine de caractère à modifier

Index