

Csuper - Compteur de Score Universel Permettant l'Exemption de Reflexion

2.1.8

Generated by Doxygen 1.8.6

Thu Apr 3 2014 21:54:50

Contents

1	Data Structure Index	1
1.1	Data Structures	1
2	File Index	3
2.1	File List	3
3	Data Structure Documentation	5
3.1	Fichier_Jeu Struct Reference	5
3.1.1	Detailed Description	5
3.1.2	Field Documentation	5
3.1.2.1	annee	5
3.1.2.2	config	5
3.1.2.3	distribue	5
3.1.2.4	jour	6
3.1.2.5	mois	6
3.1.2.6	nb_joueur	6
3.1.2.7	nb_tour	6
3.1.2.8	nom_joueur	6
3.1.2.9	point	6
3.1.2.10	point_tot	6
3.1.2.11	position	6
3.1.2.12	taille_max_nom	6
3.1.2.13	version	6
3.2	game_config Struct Reference	6
3.2.1	Detailed Description	7
3.2.2	Field Documentation	7
3.2.2.1	begin_score	7
3.2.2.2	max	7
3.2.2.3	name	7
3.2.2.4	nb_max	7
3.2.2.5	number_after_comma	7
3.2.2.6	sens_premier	7

3.2.2.7	tour_par_tour	7
3.2.2.8	use_distributor	7
3.3	list_game_config Struct Reference	8
3.3.1	Field Documentation	8
3.3.1.1	name_game_config	8
3.3.1.2	nb_config	8
4	File Documentation	9
4.1	argument_main.c File Reference	9
4.1.1	Detailed Description	9
4.1.2	Function Documentation	9
4.1.2.1	searchArgument	9
4.2	argument_main.h File Reference	10
4.2.1	Detailed Description	10
4.2.2	Macro Definition Documentation	10
4.2.2.1	CHAINE_LECTURE_FICHER	10
4.2.2.2	CHAINE_LECTURE_FICHER_RED	10
4.2.2.3	CHAINE_OUVERTURE_FICHER	11
4.2.2.4	CHAINE_OUVERTURE_FICHER_RED	11
4.2.2.5	LECTURE_FICHER	11
4.2.2.6	OUVERTURE_FICHER	11
4.2.3	Function Documentation	11
4.2.3.1	searchArgument	11
4.3	config.c File Reference	11
4.3.1	Detailed Description	12
4.3.2	Function Documentation	12
4.3.2.1	addConfigListFile	12
4.3.2.2	freeListGameConfig	12
4.3.2.3	makeConfigFile	13
4.3.2.4	makeConfigListFile	13
4.3.2.5	makeListGameConfig	13
4.3.2.6	readConfigFile	14
4.3.2.7	readConfigListFile	14
4.3.2.8	removeConfigFile	15
4.3.2.9	removeConfigListFile	15
4.4	config.h File Reference	16
4.4.1	Detailed Description	17
4.4.2	Macro Definition Documentation	17
4.4.2.1	NOM_DOSSIER_CONFIG	17
4.4.2.2	NOM_FICHER_CONFIG	17

4.4.3	Function Documentation	17
4.4.3.1	addConfigListFile	17
4.4.3.2	freeListGameConfig	18
4.4.3.3	makeConfigFile	18
4.4.3.4	makeConfigListFile	18
4.4.3.5	makeListGameConfig	18
4.4.3.6	readConfigFile	19
4.4.3.7	readConfigListFile	19
4.4.3.8	removeConfigFile	20
4.4.3.9	removeConfigListFile	20
4.5	emplacement_fichier.c File Reference	21
4.5.1	Detailed Description	21
4.5.2	Function Documentation	22
4.5.2.1	changerCheminFichier	22
4.5.2.2	creationPreferences	22
4.5.2.3	lectureCheminFichier	22
4.5.2.4	lecturePreferences	23
4.6	emplacement_fichier.h File Reference	24
4.6.1	Detailed Description	24
4.6.2	Macro Definition Documentation	24
4.6.2.1	NOM_DOSSIER_CSUPER	24
4.6.2.2	NOM_FICHER_PREFERENCES	24
4.6.3	Function Documentation	24
4.6.3.1	changerCheminFichier	24
4.6.3.2	creationPreferences	25
4.6.3.3	lectureCheminFichier	25
4.6.3.4	lecturePreferences	26
4.7	fichier.c File Reference	27
4.7.1	Detailed Description	27
4.7.2	Function Documentation	27
4.7.2.1	ecrireFichier	27
4.7.2.2	lireFichier	28
4.7.2.3	nouveauScore	28
4.7.2.4	ouvrirFichierExtension	29
4.7.2.5	renommerFichier	30
4.7.2.6	supprimerFichier	31
4.8	fichier.h File Reference	31
4.8.1	Detailed Description	32
4.8.2	Macro Definition Documentation	32
4.8.2.1	EXTENSION_FICHER	32

4.8.2.2	TAILLE_MAX_NOM_FICHIER	32
4.8.2.3	TYPE_FICHIER	32
4.8.3	Function Documentation	32
4.8.3.1	ecrireFichier	32
4.8.3.2	lireFichier	33
4.8.3.3	nouveauScore	34
4.8.3.4	ouvrirFichierExtension	34
4.8.3.5	renommerFichier	35
4.8.3.6	supprimerFichier	35
4.9	fonction.c File Reference	35
4.9.1	Detailed Description	36
4.9.2	Function Documentation	36
4.9.2.1	ajoutExtension	36
4.9.2.2	compareFlottantCroissant	36
4.9.2.3	compareFlottantDecroissant	36
4.9.2.4	fermerFichier	37
4.9.2.5	lireTailleFichier	37
4.9.2.6	mauvais_choix	37
4.9.2.7	myAlloc	37
4.9.2.8	myRealloc	37
4.9.2.9	ouvrirFichier	37
4.9.2.10	systemEfface	38
4.10	fonction.h File Reference	38
4.10.1	Detailed Description	38
4.10.2	Macro Definition Documentation	39
4.10.2.1	FAUX	39
4.10.2.2	VRAI	39
4.10.3	Function Documentation	39
4.10.3.1	ajoutExtension	39
4.10.3.2	compareFlottantCroissant	39
4.10.3.3	compareFlottantDecroissant	39
4.10.3.4	fermerFichier	39
4.10.3.5	lireTailleFichier	40
4.10.3.6	mauvais_choix	41
4.10.3.7	myAlloc	41
4.10.3.8	myRealloc	41
4.10.3.9	ouvrirFichier	41
4.10.3.10	systemEfface	41
4.11	structure.c File Reference	41
4.11.1	Detailed Description	42

4.11.2	Function Documentation	42
4.11.2.1	ajoutDistribueStruct	42
4.11.2.2	calculPosition	42
4.11.2.3	creerFichierStruct	43
4.11.2.4	debNouvTour	43
4.11.2.5	depScoreMax	44
4.11.2.6	fermeeFichierStruct	45
4.11.2.7	finNouvTour	45
4.11.2.8	maxNbTour	45
4.11.2.9	rechercheNumJoueur	45
4.12	structure.h File Reference	46
4.12.1	Detailed Description	46
4.12.2	Macro Definition Documentation	47
4.12.2.1	TAILLE_MAX_NOM	47
4.12.2.2	VERSION	47
4.12.3	Function Documentation	47
4.12.3.1	ajoutDistribueStruct	47
4.12.3.2	calculPosition	47
4.12.3.3	creerFichierStruct	48
4.12.3.4	debNouvTour	48
4.12.3.5	depScoreMax	48
4.12.3.6	fermeeFichierStruct	49
4.12.3.7	finNouvTour	49
4.12.3.8	maxNbTour	49
4.12.3.9	rechercheNumJoueur	49
Index		51

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Fichier_Jeu	5
game_config	6
list_game_config	8

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

argument_main.c		
	Lancement du programme	9
argument_main.h		
	Lancement du programme	10
config.c		
	Fonction des fichiers lies a la config	11
config.h		
	Prototypes des fonction des fichiers lies a la config	16
emplacement_fichier.c		
	Fonctions qui l'emplacement des fichiers sauvegardes	21
emplacement_fichier.h		
	Prototypes des fonctions qui l'emplacement des fichiers sauvegardes	24
fichier.c		
	Fonction de gestion des fichiers	27
fichier.h		
	Prototypes des fonction de gestion des fichiers	31
fonction.c		
	Fonctions essentielles au fonctionnement du programme	35
fonction.h		
	Prototypes des fonctions essentielles au fonctionnement du programme	38
structure.c		
	Fonction de gestion des fichiers de la struction contenant les informations	41
structure.h		
	Prototypes des fonction de gestion des fichiers de la struction contenant les informations	46

Chapter 3

Data Structure Documentation

3.1 Fichier_Jeu Struct Reference

```
#include <structure.h>
```

Data Fields

- float [version](#)
- float [taille_max_nom](#)
- float [jour](#)
- float [mois](#)
- float [annee](#)
- float [nb_joueur](#)
- [game_config](#) config
- char ** [nom_joueur](#)
- float * [point_tot](#)
- float * [position](#)
- float * [nb_tour](#)
- float [distribue](#)
- float ** [point](#)

3.1.1 Detailed Description

Type representant un fichier .jeu

3.1.2 Field Documentation

3.1.2.1 float annee

Annee de creation de la structure.

3.1.2.2 game_config config

La configuration de la partie

3.1.2.3 float distribue

Numero de la personne qui doit distribuer.

3.1.2.4 float jour

Jour de creation de la structure.

3.1.2.5 float mois

Mois de creation de la structure.

3.1.2.6 float nb_joueur

Nombre de joueurs.

3.1.2.7 float* nb_tour

Nombre de tour dans le jeu par joueur.

3.1.2.8 char** nom_joueur

Tableau contenant tout les noms de joueurs.

3.1.2.9 float** point

Tableau contenant les points de chaque joueur a chaque tour.

3.1.2.10 float* point_tot

Tableau contenant tout les points totaux des joueurs.

3.1.2.11 float* position

Tableau contenant la position des joueurs.

3.1.2.12 float taille_max_nom

Taille maximum que peut prendre un nom de joueur.

3.1.2.13 float version

Version de la structure.

The documentation for this struct was generated from the following file:

- [structure.h](#)

3.2 game_config Struct Reference

```
#include <structure.h>
```

Data Fields

- float [nb_max](#)
- char [sens_premier](#)
- char [tour_par_tour](#)
- char [use_distributeur](#)
- char [number_after_comma](#)
- char [max](#)
- char [name](#) [TAILLE_MAX_NOM]
- float [begin_score](#)

3.2.1 Detailed Description

Type representant une configuration de jeu

3.2.2 Field Documentation

3.2.2.1 float begin_score

L score de chacun des joueurs au debut de la partie

3.2.2.2 char max

Vaut 1 si on utilise un maximum, 0 si c'est un minimum

3.2.2.3 char name[TAILLE_MAX_NOM]

Le nom de la configuration de jeu

3.2.2.4 float nb_max

Nombre maximum que peut prendre un joueur.

3.2.2.5 char number_after_comma

Le nombre de chiffres apres la virgule dans l'affichage

3.2.2.6 char sens_premier

Vaut 1 si le premier est celui qui a le plus de points, -1 sinon

3.2.2.7 char tour_par_tour

Vaut 1 si on joue en tour par tour, 0 sinon

3.2.2.8 char use_distributeur

Vaut 1 si on utilise un distributeur, 0 sinon

The documentation for this struct was generated from the following file:

- [structure.h](#)

3.3 list_game_config Struct Reference

```
#include <config.h>
```

Data Fields

- int [nb_config](#)
- char ** [name_game_config](#)

3.3.1 Field Documentation

3.3.1.1 char** name_game_config

3.3.1.2 int nb_config

The documentation for this struct was generated from the following file:

- [config.h](#)

Chapter 4

File Documentation

4.1 argument_main.c File Reference

Lancement du programme.

```
#include "argument_main.h"
```

Functions

- int [searchArgument](#) (int argc, char *argv[], int *fonction, int *emplacement_fichier)

4.1.1 Detailed Description

Lancement du programme.

Author

Remi BERTHO

Date

21/03/14

Version

2.1.0

4.1.2 Function Documentation

4.1.2.1 int searchArgument (int *argc*, char * *argv*[], int * *fonction*, int * *emplacement_fichier*)

Lance le programme

Parameters

in	<i>argc</i>	le nombre d'argument
----	-------------	----------------------

in	<i>argv</i>	le tableau des arguments
in	<i>fonction</i>	entier determinant quelle fonction lancer
in	<i>emplacement_ - fichier</i>	entier donnant l'emplacement du fichier a ouvrir

Returns

VRAI si la fonction a trouve un argument, FAUX sinon

4.2 argument_main.h File Reference

Lancement du programme.

```
#include "fonction.h"
```

Macros

- `#define CHAINE_LECTURE_FICHIER "--read"`
- `#define CHAINE_LECTURE_FICHIER_RED "-r"`
- `#define LECTURE_FICHIER 0`
- `#define CHAINE_OUVERTURE_FICHIER "--open"`
- `#define CHAINE_OUVERTURE_FICHIER_RED "-o"`
- `#define OUVERTURE_FICHIER 1`

Functions

- `int searchArgument (int argc, char *argv[], int *fonction, int *emplacement_fichier)`

4.2.1 Detailed Description

Lancement du programme.

Author

Remi BERTHO

Date

21/03/14

Version

2.1.0

4.2.2 Macro Definition Documentation

4.2.2.1 `#define CHAINE_LECTURE_FICHIER "--read"`

Definit l'appel a la lecture du fichier a "--read"

4.2.2.2 `#define CHAINE_LECTURE_FICHIER_RED "-r"`

Definit l'appel a la lecture du fichier a "-r"

4.2.2.3 #define CHAINE_OUVERTURE_FICHIER "--open"

Definit l'appel a l'ouverture du fichier a "--open"

4.2.2.4 #define CHAINE_OUVERTURE_FICHIER_RED "-o"

Definit l'appel a l'ouverture du fichier a "-o"

4.2.2.5 #define LECTURE_FICHIER 0

Definit l'appel a la lecture du fichier a 0

4.2.2.6 #define OUVERTURE_FICHIER 1

Definit l'appel a l'ouverture du fichier a 1

4.2.3 Function Documentation

4.2.3.1 int searchArgument (int argc, char * argv[], int * fonction, int * emplacement_fichier)

Lance le programme

Parameters

in	argc	le nombre d'argument
in	argv	le tableau des arguments
in	fonction	entier determinant quelle fonction lancer
in	emplacement_ - fichier	entier donnant l'emplacement du fichier a ouvrir

Returns

VRAI si la fonction a trouve un argument, FAUX sinon

4.3 config.c File Reference

Fonction des fichiers lies a la config.

```
#include "config.h"
```

Functions

- [list_game_config](#) * [makeListGameConfig](#) (int nb_config)
- void [freeListGameConfig](#) ([list_game_config](#) *ptr_list_config)
- int [makeConfigListFile](#) ()
- [list_game_config](#) * [readConfigListFile](#) ()
- int [addConfigListFile](#) (char *new_config_name)
- int [removeConfigListFile](#) (int num_suppr, [list_game_config](#) *ptr_list_config)
- int [makeConfigFile](#) ([game_config](#) config)
- int [removeConfigFile](#) (char *config_name)
- int [readConfigFile](#) (int num_read, [list_game_config](#) *ptr_list_config, [game_config](#) *ptr_config)

4.3.1 Detailed Description

Fonction des fichiers lies a la config.

Author

Remi BERTHO

Date

27/03/14

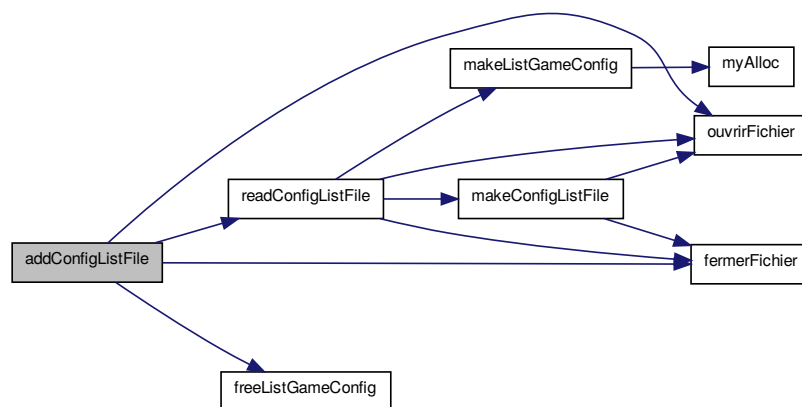
Version

2.2.0

4.3.2 Function Documentation

4.3.2.1 `int addConfigListFile (char * new_config_name)`

Here is the call graph for this function:

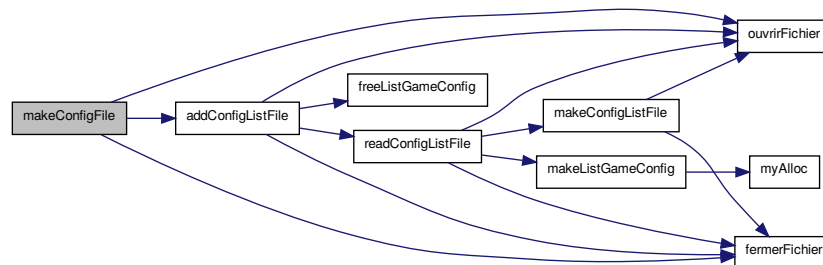


4.3.2.2 `void freeListGameConfig (list_game_config * ptr_list_config)`

Desalloue une structure `list_game_config` [in] `ptr_list_config` un pointeur sur un `list_game_config`

4.3.2.3 int makeConfigFile (game_config config)

Here is the call graph for this function:



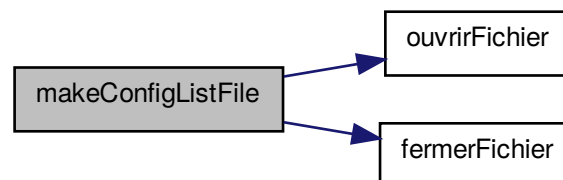
4.3.2.4 int makeConfigListFile ()

Cree les dossier et le fichier qui va contenir les configs

Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



4.3.2.5 list_game_config * makeListGameConfig (int nb_config)

Cree une structure [list_game_config](#) [in] nb_config le nombre de configuration disponible

Returns

une [list_game_config](#)

Here is the call graph for this function:



4.3.2.6 `int readConfigFile (int num_read, list_game_config * ptr_list_config, game_config * ptr_config)`

Lis un fichier de configuration de jeu et ferme la liste de configuration de jeu.

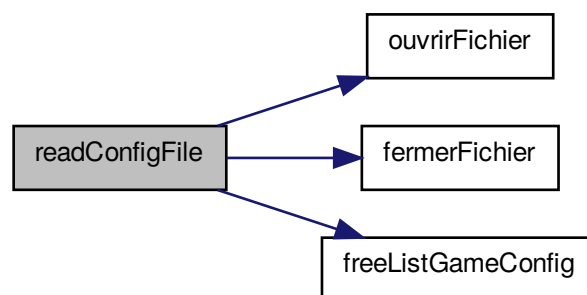
Parameters

in	<i>num_read</i>	le numero de la liste de config a lire
in	<i>ptr_list_config</i>	un pointeur sur une liste de configuration de jeu
in	<i>ptr_config</i>	un pointeur sur une configuration de jeu

Returns

une [list_game_config](#)

Here is the call graph for this function:



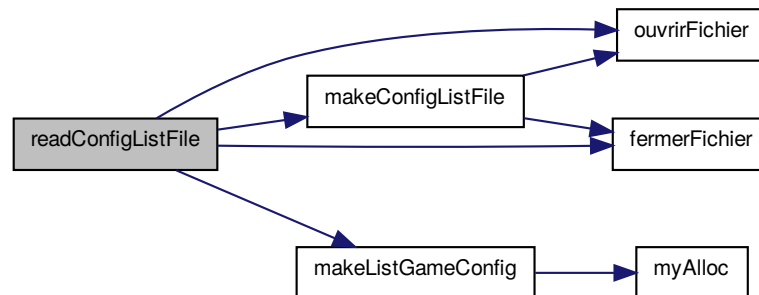
4.3.2.7 `list_game_config * readConfigListFile ()`

Cree une structure [list_game_config](#) a partir du fichier de configuration

Returns

une [list_game_config](#)

Here is the call graph for this function:



4.3.2.8 int removeConfigFile (char * config_name)

Supprime un fichier de configs de jeu

Parameters

in	<i>config_name</i>	le nom de la configuration de jeu
----	--------------------	-----------------------------------

Returns

VRAI si tout s'est bien passe, FAUX sinon

4.3.2.9 int removeConfigListFile (int num_suppr, list_game_config * ptr_list_config)

Enleve une configuration du fichier qui les liste

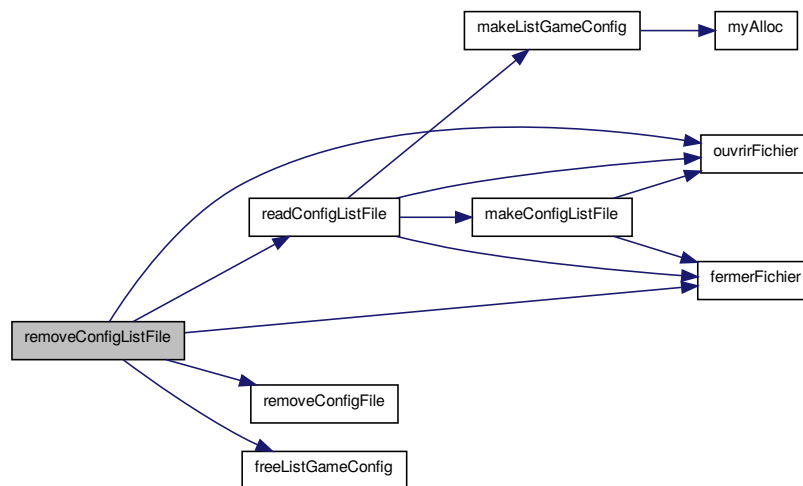
Parameters

in	<i>num_sppr</i>	le numero de la config a supprimer
----	-----------------	------------------------------------

Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



4.4 config.h File Reference

Prototypes des fonction des fichiers lies a la config.

```
#include <math.h>
#include "fonction.h"
#include "structure.h"
#include "fichier.h"
#include "emplacement_fichier.h"
```

Data Structures

- struct [list_game_config](#)

Macros

- #define [NOM_DOSSIER_CONFIG](#) "config"
- #define [NOM_FICHIER_CONFIG](#) "configuration"

Functions

- [list_game_config](#) * [makeListGameConfig](#) (int nb_config)
- void [freeListGameConfig](#) ([list_game_config](#) *ptr_list_config)
- int [makeConfigListFile](#) ()
- [list_game_config](#) * [readConfigListFile](#) ()
- int [addConfigListFile](#) (char *new_config_name)
- int [removeConfigListFile](#) (int num_suppr, [list_game_config](#) *ptr_list_config)

- int [makeConfigFile](#) (game_config config)
- int [removeConfigFile](#) (char *config_name)
- int [readConfigFile](#) (int num_read, list_game_config *ptr_list_config, game_config *ptr_config)

4.4.1 Detailed Description

Prototypes des fonction des fichiers lies a la config.

Author

Remi BERTHO

Date

27/03/14

Version

2.2.0

4.4.2 Macro Definition Documentation

4.4.2.1 #define NOM_DOSSIER_CONFIG "config"

Definit NOM_DOSSIER_CONFIG a "config"

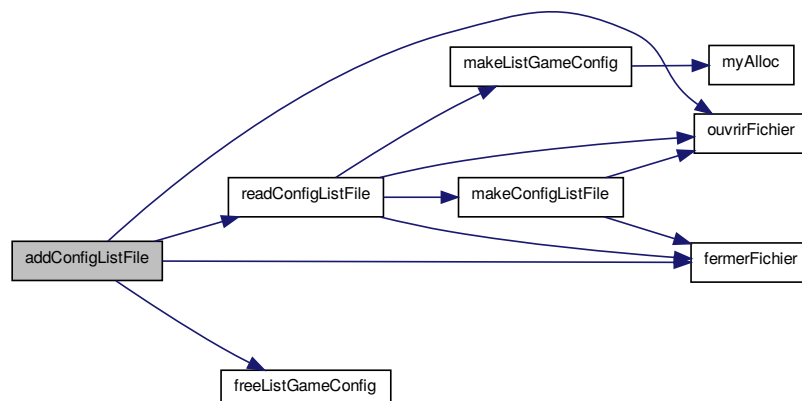
4.4.2.2 #define NOM_FICHIER_CONFIG "configuration"

Definit NOM_FICHIER_CONFIG a "configuratio"

4.4.3 Function Documentation

4.4.3.1 int addConfigListFile (char * new_config_name)

Here is the call graph for this function:

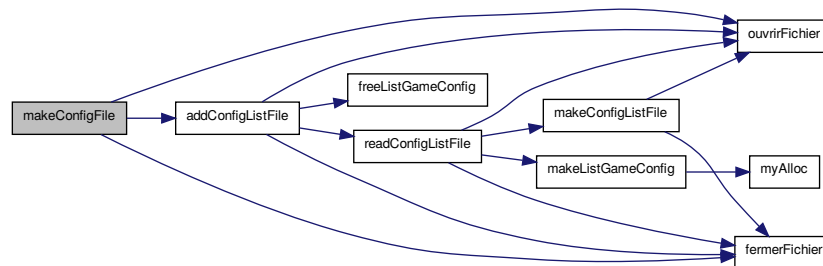


4.4.3.2 void freeListGameConfig (list_game_config * ptr_list_config)

Desalloue une structure [list_game_config](#) [in] ptr_list_config un pointeur sur un [list_game_config](#)

4.4.3.3 int makeConfigFile (game_config config)

Here is the call graph for this function:



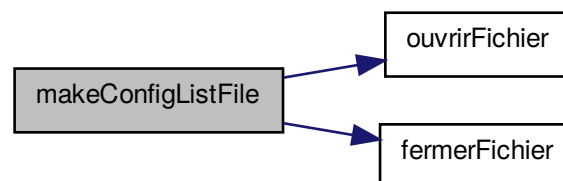
4.4.3.4 int makeConfigListFile ()

Cree les dossier et le fichier qui va contenir les configs

Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



4.4.3.5 list_game_config* makeListGameConfig (int nb_config)

Cree une structure [list_game_config](#) [in] nb_config le nombre de configuration disponible

Returns

une [list_game_config](#)

Here is the call graph for this function:



4.4.3.6 int readConfigFile (int num_read, list_game_config * ptr_list_config, game_config * ptr_config)

Lis un fichier de configuration de jeu et ferme la liste de configuration de jeu.

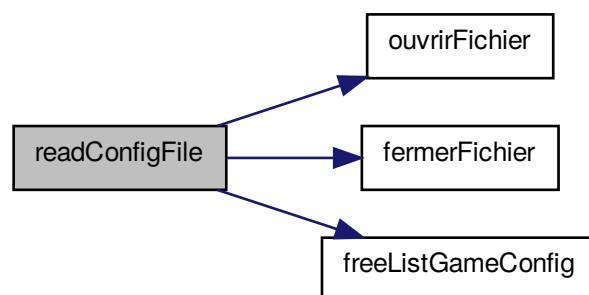
Parameters

in	<i>num_read</i>	le numero de la liste de config a lire
in	<i>ptr_list_config</i>	un pointeur sur une liste de configuration de jeu
in	<i>ptr_config</i>	un pointeur sur une configuration de jeu

Returns

une [list_game_config](#)

Here is the call graph for this function:



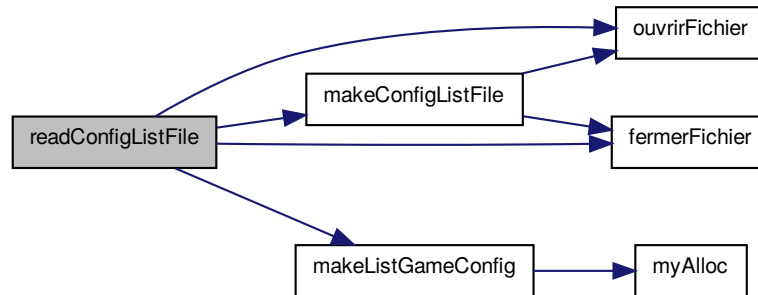
4.4.3.7 list_game_config* readConfigListFile ()

Cree une structure [list_game_config](#) a partir du fichier de configuration

Returns

une [list_game_config](#)

Here is the call graph for this function:



4.4.3.8 int removeConfigFile (char * config_name)

Supprime un fichier de configs de jeu

Parameters

in	<i>config_name</i>	le nom de la configuration de jeu
----	--------------------	-----------------------------------

Returns

VRAI si tout s'est bien passe, FAUX sinon

4.4.3.9 int removeConfigListFile (int num_suppr, list_game_config * ptr_list_config)

Enleve une configuration du fichier qui les liste

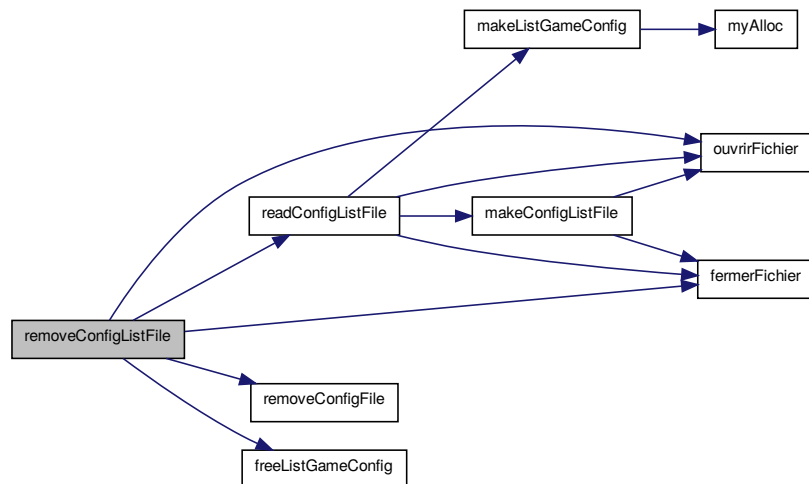
Parameters

in	<i>num_suppr</i>	le numero de la config a supprimer
----	------------------	------------------------------------

Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



4.5 emplacement_fichier.c File Reference

Fonctions qui l'emplacement des fichiers sauvegardes.

```
#include "emplacement_fichier.h"
```

Functions

- int [creationPreferences](#) ()
- int [lecturePreferences](#) (char *nom_fichier)
- int [lectureCheminFichier](#) (char *nom_fichier)
- int [changerCheminFichier](#) (char *nouveauChemin)

4.5.1 Detailed Description

Fonctions qui l'emplacement des fichiers sauvegardes.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.5.2 Function Documentation

4.5.2.1 int changerCheminFichier (char * *nouveauChemin*)

Changer le chemin de sauvegarde des fichier par nouveauChemin

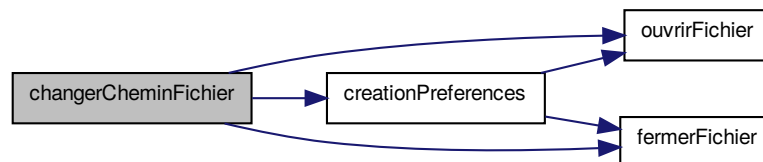
Parameters

in, out	* <i>nouveau-Chemin</i>	le nouveau chemin
---------	-------------------------	-------------------

Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



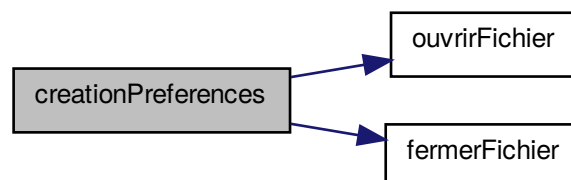
4.5.2.2 void creationPreferences ()

Cree le dossier et le fichier qui va contenir les preferences

Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



4.5.2.3 int lectureCheminFichier (char * *nom_fichier*)

Ajoute le chemin du fichier dans le nom du fichier a partir des preferences, si le fichier de preference n'existe pas il le cree Attention ne fait quelque chose uniquement si la constant PORTABLE n'est pas defini

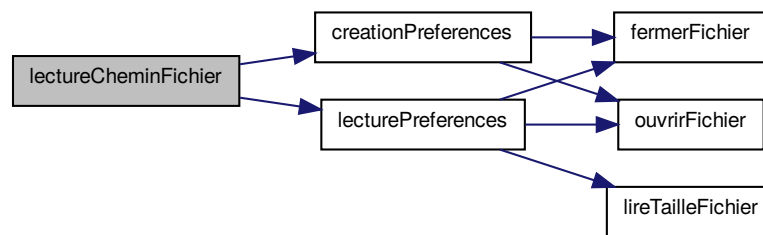
Parameters

in, out	* <i>nom_fichier</i>	le nom du fichier a qui on va ajouter son chemin
---------	----------------------	--

Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:

4.5.2.4 int lecturePreferences (char * *nom_fichier*)

lis les preferences du fichier et ajoute le chemin lu dans le nom de fichier passe en parametre

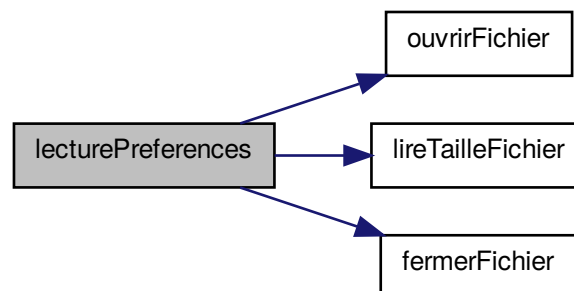
Parameters

in, out	* <i>nom_fichier</i>	le nom du fichier a qui on va ajouter son chemin
---------	----------------------	--

Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



4.6 emplacement_fichier.h File Reference

Prototypes des fonctions qui l'emplacement des fichiers sauvegardes.

```
#include <sys/stat.h>
#include <sys/types.h>
#include "fichier.h"
```

Macros

- `#define NOM_FICHIER_PREFERENCES` "preferences.txt"
- `#define NOM_DOSSIER_CSUPER` ".csuper"

Functions

- `int creationPreferences ()`
- `int lecturePreferences (char *nom_fichier)`
- `int lectureCheminFichier (char *nom_fichier)`
- `int changerCheminFichier (char *nouveauChemin)`

4.6.1 Detailed Description

Prototypes des fonctions qui l'emplacement des fichiers sauvegardes.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.6.2 Macro Definition Documentation

4.6.2.1 `#define NOM_DOSSIER_CSUPER ".csuper"`

Definit NOM_DOSSIER_CSUPER

4.6.2.2 `#define NOM_FICHIER_PREFERENCES "preferences.txt"`

Definit NOM_FICHIER_PREFERENCES a "preferences.txt"

4.6.3 Function Documentation

4.6.3.1 `int changerCheminFichier (char * nouveauChemin)`

Changer le chemin de sauvegarde des fichier par nouveauChemin

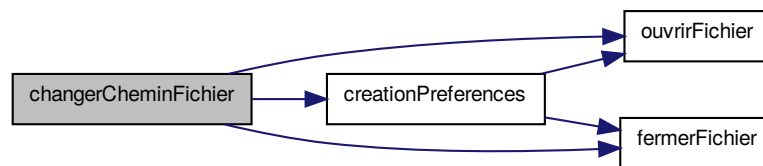
Parameters

<code>in, out</code>	<code>*nouveau-Chemin</code>	le nouveau chemin
----------------------	------------------------------	-------------------

Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:

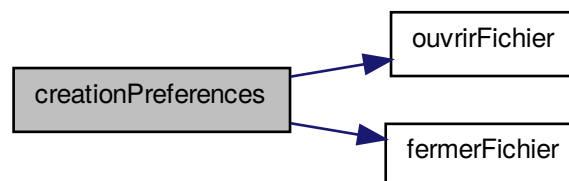
4.6.3.2 `int creationPreferences ()`

Cree le dossier et le fichier qui va contenir les preferences

Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:

4.6.3.3 `int lectureCheminFichier (char * nom_fichier)`

Ajoute le chemin du fichier dans le nom du fichier a partir des preferences, si le fichier de preference n'existe pas il le cree Attention ne fait quelque chose uniquement si la constant PORTABLE n'est pas defini

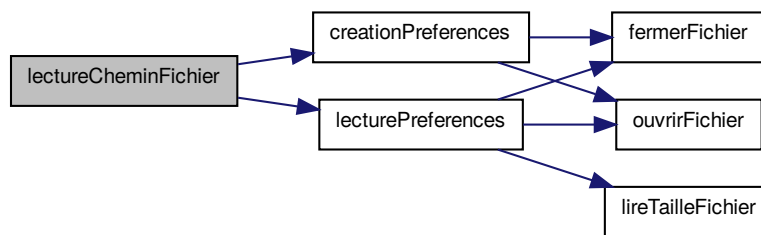
Parameters

<i>in, out</i>	<i>*nom_fichier</i>	le nom du fichier a qui on va ajouter son chemin
----------------	---------------------	--

Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



4.6.3.4 int lecturePreferences (char * nom_fichier)

lis les preferences du fichier et ajoute le chemin lu dans le nom de fichier passe en parametre

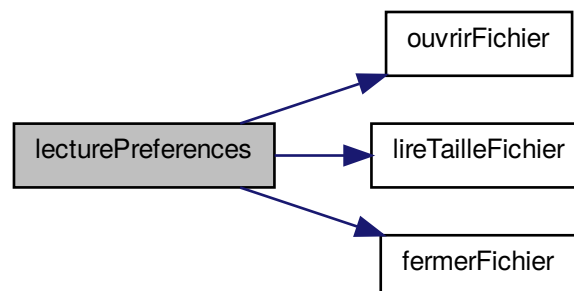
Parameters

<i>in, out</i>	<i>*nom_fichier</i>	le nom du fichier a qui on va ajouter son chemin
----------------	---------------------	--

Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



4.7 fichier.c File Reference

Fonction de gestion des fichiers.

```
#include "fichier.h"
```

Functions

- FILE * [ouvrirFichierExtension](#) (char nom[], char mode[])
- [Fichier_Jeu](#) * [lireFichier](#) (char *nom)
- int [ecrireFichier](#) (char *nom, [Fichier_Jeu](#) *ptr_struct_fichier)
- int [nouveauScore](#) (char *nom, [Fichier_Jeu](#) *ptr_struct_fichier)
- int [supprimerFichier](#) (char *nom)
- int [renommerFichier](#) (char *nom_ancien, char *nom_nouveau)

4.7.1 Detailed Description

Fonction de gestion des fichiers.

Author

Remi BERTHO

Date

09/03/14

Version

2.1.0

4.7.2 Function Documentation

4.7.2.1 int [ecrireFichier](#) (char * *nom*, [Fichier_Jeu](#) * *ptr_struct_fichier*)

Cree un fichier .jeu qui contient toutes les informations de la structure [Fichier_Jeu](#) mis en parametre

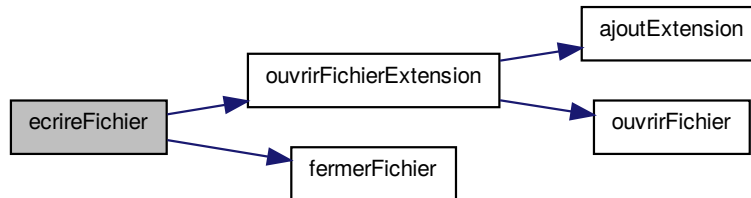
Parameters

in	<i>*nom</i>	le nom du fichier
in	<i>*ptr_struct_fichier</i>	la structure du fichier avec lequel on veut cree le fichier

Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:

**4.7.2.2 Fichier_Jeu * lireFichier (char * nom)**

Lis ce qu'il y a dans le fichier avec le nom donne en parametre et le met dans une structure [Fichier_Jeu](#) rendu par la fonction

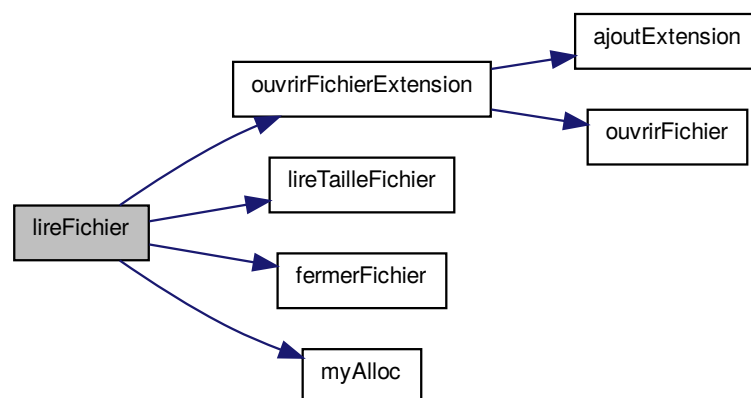
Parameters

in	<i>nom[]</i>	le nom du fichier
----	--------------	-------------------

Returns

un pointeur sur la structure [Fichier_Jeu](#) cree, NULL s'il y a un probleme d'ouverture du fichier

Here is the call graph for this function:

**4.7.2.3 void nouveauScore (char * nom, Fichier_Jeu * ptr_struct_fichier)**

Mets a jour le fichier avec les nouveaux scores

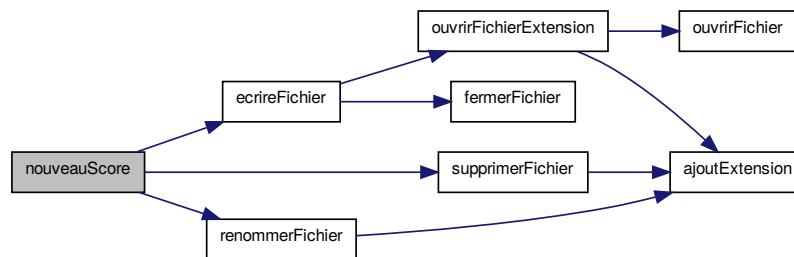
Parameters

in	<i>*nom</i>	le nom du fichier
in	<i>*ptr_struct_ - fichier</i>	la structure du fichier avec lequel on veut mettre un nouveau score

Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



4.7.2.4 FILE * ouvrirFichierExtension (char nom[], char mode[])

Ouvre un fichier a partir de son nom et du mode voulu en y ajouter l'extension du fichier si necessaire

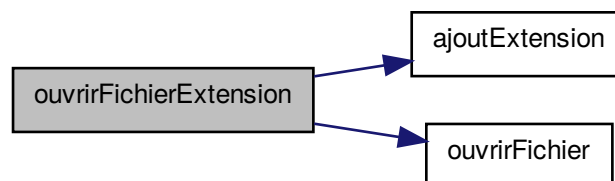
Parameters

in	<i>nom[]</i>	le nom du fichier
in	<i>mode[]</i>	le mode voulu

Returns

un pointeur sur le fichier ouvert, NULL s'il y a eut un probleme

Here is the call graph for this function:



4.7.2.5 `int renommerFichier (char * nom_ancien, char * nom_nouveau)`

Renomme le fichier dont le nom est en parametre

Parameters

in	<i>*nom_ancien</i>	l'ancien nom du fichier
in	<i>*nom_nouveau</i>	le nouveau nom du fichier

Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



4.7.2.6 int supprimerFichier (char * nom)

Supprime le fichier dont le nom est en parametre

Parameters

in	<i>*nom</i>	le nom du fichier
----	-------------	-------------------

Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



4.8 fichier.h File Reference

Prototypes des fonction de gestion des fichiers.

```
#include "structure.h"
#include <unistd.h>
```

Macros

- #define `TAILLE_MAX_NOM_FICHIER` 250
- #define `EXTENSION_FICHIER` "csu"
- #define `TYPE_FICHIER` "CompteurScoreUniversel"

Functions

- FILE * `ouvrirFichierExtension` (char nome[], char mode[])
- `Fichier_Jeu` * `lireFichier` (char *nom)
- int `ecrireFichier` (char *nom, `Fichier_Jeu` *ptr_struct_fichier)
- int `nouveauScore` (char *nom, `Fichier_Jeu` *ptr_struct_fichier)
- int `supprimerFichier` (char *nom)
- int `renommerFichier` (char *nom_ancien, char *nom_nouveau)

4.8.1 Detailed Description

Prototypes des fonction de gestion des fichiers.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.8.2 Macro Definition Documentation

4.8.2.1 #define `EXTENSION_FICHIER` "csu"

Definit l'extension du fichier a "csu"

4.8.2.2 #define `TAILLE_MAX_NOM_FICHIER` 250

Definit la taille max d'un nom a 250

4.8.2.3 #define `TYPE_FICHIER` "CompteurScoreUniversel"

Definit la chaine de caractere permettant de verifier le type de fichier a "CompteurScoreUniversel"

4.8.3 Function Documentation

4.8.3.1 int `ecrireFichier` (char * *nom*, `Fichier_Jeu` * *ptr_struct_fichier*)

Cree un fichier .jeu qui contient toutes les informations de la structure `Fichier_Jeu` mis en parametre

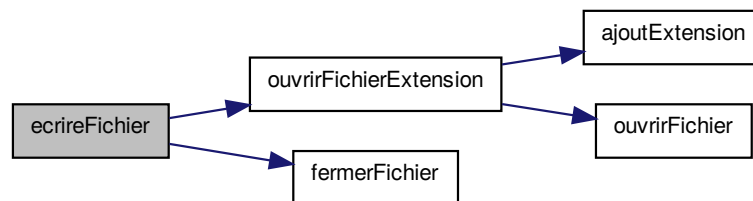
Parameters

in	<i>*nom</i>	le nom du fichier
in	<i>*ptr_struct_ - fichier</i>	la structure du fichier avec lequel on veut cree le fichier

Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



4.8.3.2 Fichier_Jeu* lireFichier (char * nom)

Lis ce qu'il y a dans le fichier avec le nom donne en parametre et le met dans une structure `Fichier_Jeu` rendu par la fonction

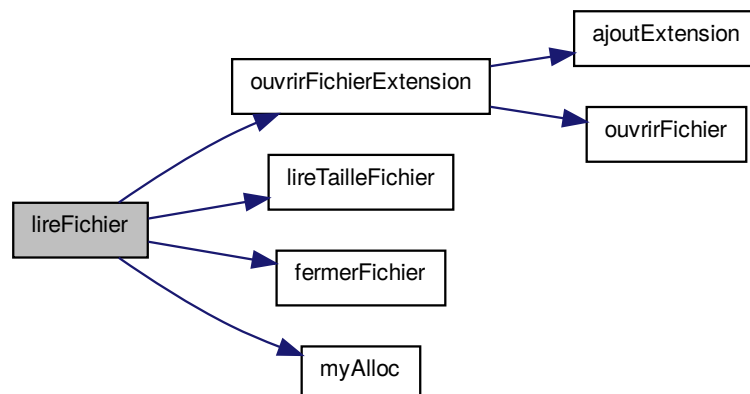
Parameters

in	<i>nom[]</i>	le nom du fichier
----	--------------	-------------------

Returns

un pointeur sur la structure `Fichier_Jeu` cree, NULL s'il y a un probleme d'ouverture du fichier

Here is the call graph for this function:



4.8.3.3 int nouveauScore (char * nom, Fichier_Jeu * ptr_struct_fichier)

Mets a jour le fichier avec les nouveaux scores

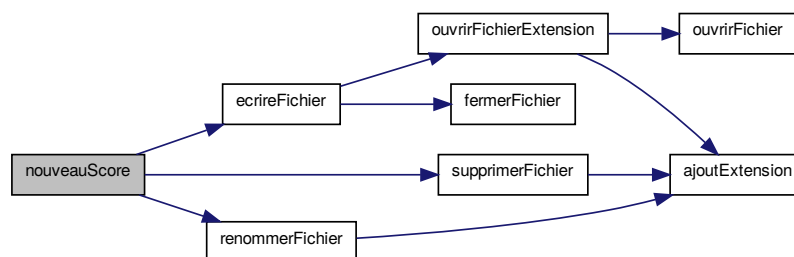
Parameters

in	<i>*nom</i>	le nom du fichier
in	<i>*ptr_struct_fichier</i>	la structure du fichier avec lequel on veut mettre un nouveau score

Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



4.8.3.4 FILE* ouvrirFichierExtension (char nom[], char mode[])

Ouvre un fichier a partir de son nom et du mode voulu en y ajouter l'extension du fichier si necessaire

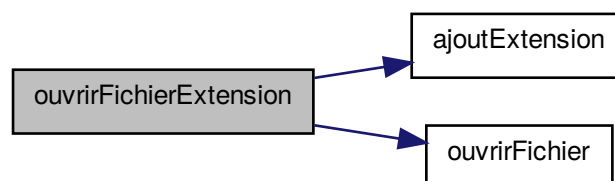
Parameters

in	<i>nom[]</i>	le nom du fichier
in	<i>mode[]</i>	le mode voulu

Returns

un pointeur sur le fichier ouvert, NULL s'il y a eut un probleme

Here is the call graph for this function:



4.8.3.5 int renommerFichier (char * *nom_ancien*, char * *nom_nouveau*)

Renomme le fichier dont le nom est en parametre

Parameters

in	* <i>nom_ancien</i>	l'ancien nom du fichier
in	* <i>nom_nouveau</i>	le nouveau nom du fichier

Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



4.8.3.6 int supprimerFichier (char * *nom*)

Supprime le fichier dont le nom est en parametre

Parameters

in	* <i>nom</i>	le nom du fichier
----	--------------	-------------------

Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



4.9 fonction.c File Reference

Fonctions essentielles au fonctionnement du programme.

```
#include "fonction.h"
#include "fichier.h"
```

Functions

- void [mauvais_choix](#) ()
- void [systemEfface](#) ()
- int [compareFlottantCroissant](#) (void const *a, void const *b)
- int [compareFlottantDecroissant](#) (void const *a, void const *b)
- FILE * [ouvrirFichier](#) (char nom[], char mode[])
- int [fermerFichier](#) (FILE *ptr_fichier)
- int [lireTailleFichier](#) (FILE *ptr_fichier)
- void * [myAlloc](#) (int taille_alloue)
- void [myRealloc](#) (void **ptr, int taille_alloue)
- void [ajoutExtension](#) (char *nom_fichier)

4.9.1 Detailed Description

Fonctions essentielles au fonctionnement du programme.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.9.2 Function Documentation

4.9.2.1 void ajoutExtension (char * *nom_fichier*)

Ajoute l'extension du fichier si elle n'y est pas

Parameters

in	<i>nom_fichier</i>	le nom de fichier
----	--------------------	-------------------

4.9.2.2 int compareFlottantCroissant (void const * *a*, void const * *b*)

Compare 2 Flottant, renvoie 1 si $a > b$, 0 si $a = b$ et -1 si $a < b$

Parameters

in	<i>*a</i>	un pointeur sur un flottant
in	<i>*b</i>	un pointeur sur un flottant

4.9.2.3 int compareFlottantDecroissant (void const * *a*, void const * *b*)

Compare 2 Flottant, renvoie 1 si $a < b$, 0 si $a = b$ et -1 si $a > b$

Parameters

in	<i>*a</i>	un pointeur sur un flottant
in	<i>*b</i>	un pointeur sur un flottant

4.9.2.4 int fermerFichier (FILE * *ptr_fichier*)

Ferme le fichier

Parameters

in	<i>*ptr_fichier</i>	le fichier
----	---------------------	------------

Returns

entier 0 si tout s'est bien passe, 1 sinon

4.9.2.5 int lireTailleFichier (FILE * *ptr_fichier*)

Lis la taille du fichier

Parameters

in	<i>*ptr_fichier</i>	le fichier
----	---------------------	------------

Returns

entier ayant la taille du fichier

4.9.2.6 void mauvais_choix ()

Affiche un message d'erreur.

4.9.2.7 void * myAlloc (int *taille_alloue*)

Alloue un bloc memoire et verifie que ca s'est bien alloue

Parameters

in	<i>taille_alloue</i>	la taille à allouer
----	----------------------	---------------------

Returns

un pointeur sur la structure alloué

4.9.2.8 void myRealloc (void ** *ptr*, int *taille_alloue*)4.9.2.9 FILE * ouvrirFichier (char *nom*[], char *mode*[])

Ouvre un fichier a partir de son nom (*nom*[]) et du mode voulu (*mode*[])

Parameters

in	<i>nom[]</i>	le nom du fichier
in	<i>mode[]</i>	le mode voulu

Returns

un pointeur sur le fichier ouvert, NULL s'il y a eut un probleme

4.9.2.10 void systemEfface ()

Efface la console de l'utilisateur.

4.10 fonction.h File Reference

Prototypes des fonctions essentielles au fonctionnement du programme.

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
```

Macros

- `#define VRAI 1`
- `#define FAUX 0`

Functions

- void [mauvais_choix](#) ()
- void [systemEfface](#) ()
- int [compareFlottantDecroissant](#) (void const *a, void const *b)
- int [compareFlottantCroissant](#) (void const *a, void const *b)
- FILE * [ouvrirFichier](#) (char nome[], char mode[])
- int [fermerFichier](#) (FILE *ptr_fichier)
- int [lireTailleFichier](#) (FILE *ptr_fichier)
- void * [myAlloc](#) (int taille_alloue)
- void [myRealloc](#) (void **ptr, int taille_alloue)
- void [ajoutExtension](#) (char *nom_fichier)

4.10.1 Detailed Description

Prototypes des fonctions essentielles au fonctionnement du programme.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.10.2 Macro Definition Documentation

4.10.2.1 #define FAUX 0

Definit FAUX a 0

4.10.2.2 #define VRAI 1

Definit VRAI a 1

4.10.3 Function Documentation

4.10.3.1 void ajoutExtension (char * *nom_fichier*)

Ajoute l'extension du fichier si elle n'y est pas

Parameters

in	<i>nom_fichier</i>	le nom de fichier
----	--------------------	-------------------

4.10.3.2 int compareFlottantCroissant (void const * *a*, void const * *b*)

Compare 2 Flottant, renvoie 1 si $a > b$, 0 si $a = b$ et -1 si $a < b$

Parameters

in	<i>*a</i>	un pointeur sur un flottant
in	<i>*b</i>	un pointeur sur un flottant

4.10.3.3 int compareFlottantDecroissant (void const * *a*, void const * *b*)

Compare 2 Flottant, renvoie 1 si $a < b$, 0 si $a = b$ et -1 si $a > b$

Parameters

in	<i>*a</i>	un pointeur sur un flottant
in	<i>*b</i>	un pointeur sur un flottant

4.10.3.4 int fermerFichier (FILE * *ptr_fichier*)

Ferme le fichier

Parameters

in	<i>*ptr_fichier</i>	le fichier
----	---------------------	------------

Returns

entier 0 si tout s'est bien passe, 1 sinon

4.10.3.5 int lireTailleFichier (FILE * *ptr_fichier*)

Lis la taille du fichier

Parameters

<code>in</code>	<code>*ptr_fichier</code>	le fichier
-----------------	---------------------------	------------

Returns

entier ayant la taille du fichier

4.10.3.6 void mauvais_choix ()

Affiche un message d'erreur.

4.10.3.7 void* myAlloc (int *taille_alloue*)

Alloue un bloc memoire et verifie que ca s'est bien alloue

Parameters

<code>in</code>	<code>taille_alloue</code>	la taille à allouer
-----------------	----------------------------	---------------------

Returns

un pointeur sur la structure alloué

4.10.3.8 void myRealloc (void ** *ptr*, int *taille_alloue*)4.10.3.9 FILE* ouvrirFichier (char *nom*[], char *mode*[])

Ouvre un fichier a partir de son nom (*nom*[]) et du mode voulu (*mode*[])

Parameters

<code>in</code>	<code>nom[]</code>	le nom du fichier
<code>in</code>	<code>mode[]</code>	le mode voulu

Returns

un pointeur sur le fichier ouvert, NULL s'il y a eut un probleme

4.10.3.10 void systemEfface ()

Efface la console de l'utilisateur.

4.11 structure.c File Reference

Fonction de gestion des fichiers de la struction contenant les informations.

```
#include "structure.h"
```

Functions

- [Fichier_Jeu * creerFichierStruct](#) (float nb_joueur, [game_config](#) config)
- void [fermeeFichierStruct](#) ([Fichier_Jeu](#) *ptr_struct_fichier)
- void [debNouvTour](#) ([Fichier_Jeu](#) *ptr_struct_fichier, int num_joueur)
- void [finNouvTour](#) ([Fichier_Jeu](#) *ptr_struct_fichier, int num_joueur)
- void [calculPosition](#) ([Fichier_Jeu](#) *ptr_struct_fichier)
- void [ajoutDistribueStruct](#) ([Fichier_Jeu](#) *ptr_struct_fichier, char *nom_distribue)
- int [depScoreMax](#) ([Fichier_Jeu](#) *ptr_struct_fichier)
- int [maxNbTour](#) ([Fichier_Jeu](#) *ptr_struct_fichier)
- int [rechercheNumJoueur](#) ([Fichier_Jeu](#) *ptr_struct_fichier, char *nom_personne)

4.11.1 Detailed Description

Fonction de gestion des fichiers de la struction contenant les informations.

Author

Remi BERTHO

Date

09/03/14

Version

2.1.0

4.11.2 Function Documentation

4.11.2.1 void ajoutDistribueStruct ([Fichier_Jeu](#) * *ptr_struct_fichier*, char * *nom_distribue*)

Ajoute la personne qui distribue dans la structure.

Parameters

in	<i>*nom_distribue</i>	le nom de la personne qui distribue
in	<i>*ptr_struct_fichier</i>	la structure du fichier avec lequel on veut mettre la personne qui distribue

Here is the call graph for this function:



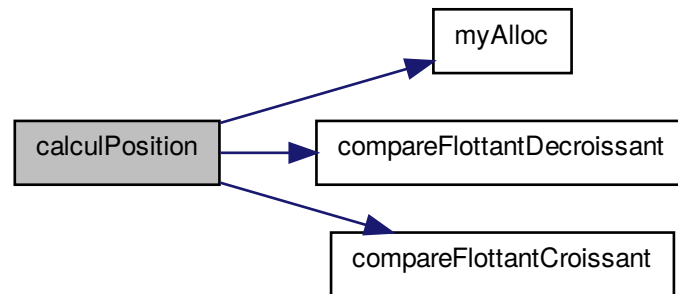
4.11.2.2 void calculPosition ([Fichier_Jeu](#) * *ptr_struct_fichier*)

Calcule les positions des joueurs

Parameters

in, out	<i>*ptr_struct_ - fichier</i>	un pointeur sur la structure Fichier_Jeu
---------	-----------------------------------	--

Here is the call graph for this function:



4.11.2.3 `Fichier_Jeu* creerFichierStruct (float nb_joueur, game_config config)`

Here is the call graph for this function:



4.11.2.4 `void debNouvTour (Fichier_Jeu * ptr_struct_fichier, int num_joueur)`

Here is the call graph for this function:



4.11.2.5 `int depScoreMax (Fichier_Jeu * ptr_struct_fichier)`

Verifie si quelqu'un a depasse le score maximum

Parameters

in	<i>*ptr_struct_fichier</i>	la structure du fichier
----	----------------------------	-------------------------

Returns

VRAI s'il y a un depassement, FAUX sinon

4.11.2.6 void fermeeFichierStruct (Fichier_Jeu * ptr_struct_fichier)

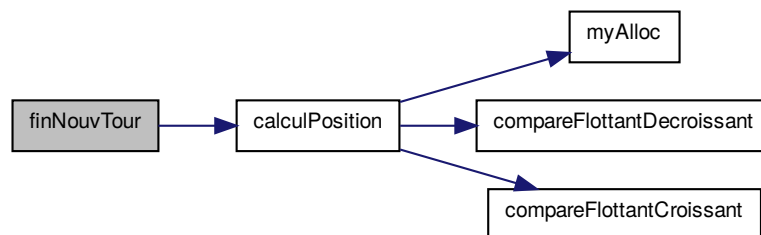
Desalloue la memoire attribuee a la structure [Fichier_Jeu](#) mis en parametre

Parameters

in, out	<i>*ptr_struct_fichier</i>	un pointeur sur la structure Fichier_Jeu a fermer
---------	----------------------------	---

4.11.2.7 void finNouvTour (Fichier_Jeu * ptr_struct_fichier, int num_joueur)

Here is the call graph for this function:



4.11.2.8 int maxNbTour (Fichier_Jeu * ptr_struct_fichier)

Cherche le maximum de nombre de tour

Parameters

in	<i>*ptr_struct_fichier</i>	la structure du fichier
----	----------------------------	-------------------------

Returns

le nombre maximum de nombre de tour

4.11.2.9 int rechercheNumJoueur (Fichier_Jeu * ptr_struct_fichier, char * nom_personne)

Cherche l'indice du tableau dans lequel est stocke une personne.

Parameters

in	<i>*nom_distribue</i>	le nom de la personne
in	<i>*ptr_struct_ - fichier</i>	la structure du fichier

Returns

l'indice du tableau, -1 si non trouve

4.12 structure.h File Reference

Prototypes des fonction de gestion des fichiers de la struction contenant les informations.

```
#include "fonction.h"  
#include <time.h>  
#include <float.h>
```

Data Structures

- struct [game_config](#)
- struct [Fichier_Jeu](#)

Macros

- #define [TAILLE_MAX_NOM](#) 30
- #define [VERSION](#) 1.4

Functions

- [Fichier_Jeu * creerFichierStruct](#) (float nb_joueur, [game_config](#) config)
- void [fermeeFichierStruct](#) ([Fichier_Jeu](#) *ptr_struct_fichier)
- void [debNouvTour](#) ([Fichier_Jeu](#) *ptr_struct_fichier, int num_joueur)
- void [finNouvTour](#) ([Fichier_Jeu](#) *ptr_struct_fichier, int num_joueur)
- void [calculPosition](#) ([Fichier_Jeu](#) *ptr_struct_fichier)
- void [ajoutDistribueStruct](#) ([Fichier_Jeu](#) *ptr_struct_fichier, char *nom_distribue)
- int [depScoreMax](#) ([Fichier_Jeu](#) *ptr_struct_fichier)
- int [maxNbTour](#) ([Fichier_Jeu](#) *ptr_struct_fichier)
- int [rechercheNumJoueur](#) ([Fichier_Jeu](#) *ptr_struct_fichier, char *nom_personne)

4.12.1 Detailed Description

Prototypes des fonction de gestion des fichiers de la struction contenant les informations.

Author

Remi BERTHO

Date

09/03/14

Version

2.1.0

4.12.2 Macro Definition Documentation

4.12.2.1 #define TAILLE_MAX_NOM 30

Definit la taille max d'un nom a 30

4.12.2.2 #define VERSION 1.4

Definit la version a 1.4

4.12.3 Function Documentation

4.12.3.1 void ajoutDistribueStruct (Fichier_Jeu * ptr_struct_fichier, char * nom_distribue)

Ajoute la personne qui distribue dans la structure.

Parameters

in	* <i>nom_distribue</i>	le nom de la personne qui distribue
in	* <i>ptr_struct_fichier</i>	la structure du fichier avec lequel on veut mettre la personne qui distribue

Here is the call graph for this function:



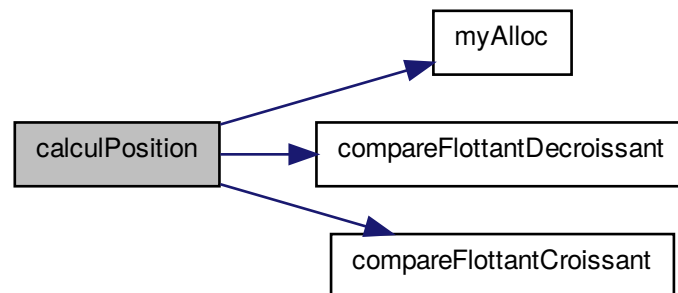
4.12.3.2 void calculPosition (Fichier_Jeu * ptr_struct_fichier)

Calcule les positions des joueurs

Parameters

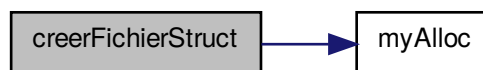
in, out	* <i>ptr_struct_fichier</i>	un pointeur sur la structure Fichier_Jeu
---------	-----------------------------	--

Here is the call graph for this function:



4.12.3.3 `Fichier_Jeu* creerFichierStruct (float nb_joueur, game_config config)`

Here is the call graph for this function:



4.12.3.4 `void debNouvTour (Fichier_Jeu * ptr_struct_fichier, int num_joueur)`

Here is the call graph for this function:



4.12.3.5 `int depScoreMax (Fichier_Jeu * ptr_struct_fichier)`

Verifie si quelqu'un a depasse le score maximum

Parameters

in	<i>*ptr_struct_fichier</i>	la structure du fichier
----	----------------------------	-------------------------

Returns

VRAI s'il y a un depassement, FAUX sinon

4.12.3.6 void fermeeFichierStruct (Fichier_Jeu * ptr_struct_fichier)

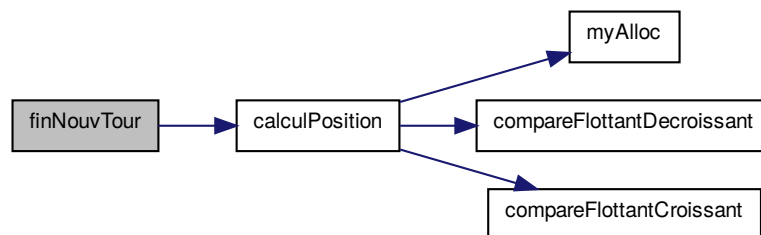
Desalloue la memoire attribuee a la structure [Fichier_Jeu](#) mis en parametre

Parameters

in, out	<i>*ptr_struct_fichier</i>	un pointeur sur la structure Fichier_Jeu a fermer
---------	----------------------------	---

4.12.3.7 void finNouvTour (Fichier_Jeu * ptr_struct_fichier, int num_joueur)

Here is the call graph for this function:



4.12.3.8 int maxNbTour (Fichier_Jeu * ptr_struct_fichier)

Cherche le maximum de nombre de tour

Parameters

in	<i>*ptr_struct_fichier</i>	la structure du fichier
----	----------------------------	-------------------------

Returns

le nombre maximum de nombre de tour

4.12.3.9 int rechercheNumJoueur (Fichier_Jeu * ptr_struct_fichier, char * nom_personne)

Cherche l'indice du tableau dans lequel est stocke une personne.

Parameters

in	<i>*nom_distribue</i>	le nom de la personne
in	<i>*ptr_struct_ - fichier</i>	la structure du fichier

Returns

l'indice du tableau, -1 si non trouve

Index

- addConfigListFile
 - config.c, [12](#)
 - config.h, [17](#)
- ajoutDistribueStruct
 - structure.c, [42](#)
 - structure.h, [47](#)
- ajoutExtension
 - fonction.c, [36](#)
 - fonction.h, [39](#)
- annee
 - Fichier_Jeu, [5](#)
- argument_main.c, [9](#)
 - searchArgument, [9](#)
- argument_main.h, [10](#)
 - LECTURE_FICHER, [11](#)
 - OUVERTURE_FICHER, [11](#)
 - searchArgument, [11](#)
- begin_score
 - game_config, [7](#)
- calculPosition
 - structure.c, [42](#)
 - structure.h, [47](#)
- changerCheminFichier
 - emplacement_fichier.c, [22](#)
 - emplacement_fichier.h, [24](#)
- compareFlottantCroissant
 - fonction.c, [36](#)
 - fonction.h, [39](#)
- compareFlottantDecroissant
 - fonction.c, [36](#)
 - fonction.h, [39](#)
- config
 - Fichier_Jeu, [5](#)
- config.c, [11](#)
 - addConfigListFile, [12](#)
 - freeListGameConfig, [12](#)
 - makeConfigFile, [12](#)
 - makeConfigListFile, [13](#)
 - makeListGameConfig, [13](#)
 - readConfigFile, [14](#)
 - readConfigListFile, [14](#)
 - removeConfigFile, [15](#)
 - removeConfigListFile, [15](#)
- config.h, [16](#)
 - addConfigListFile, [17](#)
 - freeListGameConfig, [17](#)
 - makeConfigFile, [18](#)
 - makeConfigListFile, [18](#)
 - makeListGameConfig, [18](#)
 - NOM_DOSSIER_CONFIG, [17](#)
 - NOM_FICHER_CONFIG, [17](#)
 - readConfigFile, [19](#)
 - readConfigListFile, [19](#)
 - removeConfigFile, [20](#)
 - removeConfigListFile, [20](#)
- creationPreferences
 - emplacement_fichier.c, [22](#)
 - emplacement_fichier.h, [25](#)
- creerFichierStruct
 - structure.c, [43](#)
 - structure.h, [48](#)
- debNouvTour
 - structure.c, [43](#)
 - structure.h, [48](#)
- depScoreMax
 - structure.c, [43](#)
 - structure.h, [48](#)
- distribue
 - Fichier_Jeu, [5](#)
- EXTENSION_FICHER
 - fichier.h, [32](#)
- ecrireFichier
 - fichier.c, [27](#)
 - fichier.h, [32](#)
- emplacement_fichier.c, [21](#)
 - changerCheminFichier, [22](#)
 - creationPreferences, [22](#)
 - lectureCheminFichier, [22](#)
 - lecturePreferences, [23](#)
- emplacement_fichier.h, [24](#)
 - changerCheminFichier, [24](#)
 - creationPreferences, [25](#)
 - lectureCheminFichier, [25](#)
 - lecturePreferences, [26](#)
- FAUX
 - fonction.h, [39](#)
- fermeeFichierStruct
 - structure.c, [45](#)
 - structure.h, [49](#)
- fermerFichier
 - fonction.c, [37](#)
 - fonction.h, [39](#)
- fichier.c, [27](#)
 - ecrireFichier, [27](#)
 - lireFichier, [28](#)

- nouveauScore, 28
- ouvrirFichierExtension, 29
- renommerFichier, 29
- supprimerFichier, 31
- fichier.h, 31
 - EXTENSION_FICHIER, 32
 - ecrireFichier, 32
 - lireFichier, 33
 - nouveauScore, 34
 - ouvrirFichierExtension, 34
 - renommerFichier, 34
 - supprimerFichier, 35
 - TYPE_FICHIER, 32
- Fichier_Jeu, 5
 - annee, 5
 - config, 5
 - distribue, 5
 - jour, 5
 - mois, 6
 - nb_joueur, 6
 - nb_tour, 6
 - nom_joueur, 6
 - point, 6
 - point_tot, 6
 - position, 6
 - taille_max_nom, 6
 - version, 6
- finNouvTour
 - structure.c, 45
 - structure.h, 49
- fonction.c, 35
 - ajoutExtension, 36
 - compareFlottantCroissant, 36
 - compareFlottantDecroissant, 36
 - fermerFichier, 37
 - lireTailleFichier, 37
 - mauvais_choix, 37
 - myAlloc, 37
 - myRealloc, 37
 - ouvrirFichier, 37
 - systemEfface, 38
- fonction.h, 38
 - ajoutExtension, 39
 - compareFlottantCroissant, 39
 - compareFlottantDecroissant, 39
 - FAUX, 39
 - fermerFichier, 39
 - lireTailleFichier, 39
 - mauvais_choix, 41
 - myAlloc, 41
 - myRealloc, 41
 - ouvrirFichier, 41
 - systemEfface, 41
 - VRAI, 39
- freeListGameConfig
 - config.c, 12
 - config.h, 17
- game_config, 6
 - begin_score, 7
 - max, 7
 - name, 7
 - nb_max, 7
 - number_after_comma, 7
 - sens_premier, 7
 - tour_par_tour, 7
 - use_distributor, 7
- jour
 - Fichier_Jeu, 5
- LECTURE_FICHIER
 - argument_main.h, 11
- lectureCheminFichier
 - emplacement_fichier.c, 22
 - emplacement_fichier.h, 25
- lecturePreferences
 - emplacement_fichier.c, 23
 - emplacement_fichier.h, 26
- lireFichier
 - fichier.c, 28
 - fichier.h, 33
- lireTailleFichier
 - fonction.c, 37
 - fonction.h, 39
- list_game_config, 8
 - name_game_config, 8
 - nb_config, 8
- makeConfigFile
 - config.c, 12
 - config.h, 18
- makeConfigListFile
 - config.c, 13
 - config.h, 18
- makeListGameConfig
 - config.c, 13
 - config.h, 18
- mauvais_choix
 - fonction.c, 37
 - fonction.h, 41
- max
 - game_config, 7
- maxNbTour
 - structure.c, 45
 - structure.h, 49
- mois
 - Fichier_Jeu, 6
- myAlloc
 - fonction.c, 37
 - fonction.h, 41
- myRealloc
 - fonction.c, 37
 - fonction.h, 41
- NOM_DOSSIER_CONFIG
 - config.h, 17
- NOM_DOSSIER_CSUPER

- emplacement_fichier.h, 24
- NOM_FICHER_CONFIG
 - config.h, 17
- name
 - game_config, 7
- name_game_config
 - list_game_config, 8
- nb_config
 - list_game_config, 8
- nb_joueur
 - Fichier_Jeu, 6
- nb_max
 - game_config, 7
- nb_tour
 - Fichier_Jeu, 6
- nom_joueur
 - Fichier_Jeu, 6
- nouveauScore
 - fichier.c, 28
 - fichier.h, 34
- number_after_comma
 - game_config, 7
- OUVERTURE_FICHER
 - argument_main.h, 11
- ouvrirFichier
 - fonction.c, 37
 - fonction.h, 41
- ouvrirFichierExtension
 - fichier.c, 29
 - fichier.h, 34
- point
 - Fichier_Jeu, 6
- point_tot
 - Fichier_Jeu, 6
- position
 - Fichier_Jeu, 6
- readConfigFile
 - config.c, 14
 - config.h, 19
- readConfigListFile
 - config.c, 14
 - config.h, 19
- rechercheNumJoueur
 - structure.c, 45
 - structure.h, 49
- removeConfigFile
 - config.c, 15
 - config.h, 20
- removeConfigListFile
 - config.c, 15
 - config.h, 20
- renommerFichier
 - fichier.c, 29
 - fichier.h, 34
- searchArgument
 - argument_main.c, 9
 - argument_main.h, 11
- sens_premier
 - game_config, 7
- structure.c, 41
 - ajoutDistribueStruct, 42
 - calculPosition, 42
 - creerFichierStruct, 43
 - debNouvTour, 43
 - depScoreMax, 43
 - fermeeFichierStruct, 45
 - finNouvTour, 45
 - maxNbTour, 45
 - rechercheNumJoueur, 45
- structure.h, 46
 - ajoutDistribueStruct, 47
 - calculPosition, 47
 - creerFichierStruct, 48
 - debNouvTour, 48
 - depScoreMax, 48
 - fermeeFichierStruct, 49
 - finNouvTour, 49
 - maxNbTour, 49
 - rechercheNumJoueur, 49
 - TAILLE_MAX_NOM, 47
 - VERSION, 47
- supprimerFichier
 - fichier.c, 31
 - fichier.h, 35
- systemEfface
 - fonction.c, 38
 - fonction.h, 41
- TAILLE_MAX_NOM
 - structure.h, 47
- TYPE_FICHER
 - fichier.h, 32
- taille_max_nom
 - Fichier_Jeu, 6
- tour_par_tour
 - game_config, 7
- use_distributor
 - game_config, 7
- VERSION
 - structure.h, 47
- VRAI
 - fonction.h, 39
- version
 - Fichier_Jeu, 6