

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

2.1.7

Generated by Doxygen 1.8.6

Tue Apr 1 2014 20:33:28

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	6
3.1.2	Field Documentation	6
3.1.2.1	annee	6
3.1.2.2	config	6
3.1.2.3	distribue	6
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	7
3.1.2.13	version	7
3.2	game_config Struct Reference	7
3.2.1	Detailed Description	7
3.2.2	Field Documentation	7
3.2.2.1	name	7
3.2.2.2	nb_max	7
3.2.2.3	number_after_comma	7
3.2.2.4	sens_premier	7
3.2.2.5	tour_par_tour	7
3.2.2.6	use_distributeur	8

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	affichage.c File Reference	9
4.1.1	Detailed Description	10
4.1.2	Function Documentation	10
4.1.2.1	afficherChaineTroisTab	10
4.1.2.2	afficherDistribue	10
4.1.2.3	afficherEnTete	10
4.1.2.4	afficherLicense	11
4.1.2.5	afficherLigne	11
4.1.2.6	afficherNom	11
4.1.2.7	afficherPartieFinie	11
4.1.2.8	afficherPosition	12
4.1.2.9	afficherScore	12
4.1.2.10	afficherScoreEntier	13
4.1.2.11	afficherScoreTotal	13
4.1.2.12	afficherStruct	13
4.1.2.13	printGameConfig	14
4.2	affichage.h File Reference	14
4.2.1	Detailed Description	15
4.2.2	Function Documentation	16
4.2.2.1	afficherChaineTroisTab	16
4.2.2.2	afficherDistribue	16
4.2.2.3	afficherEnTete	16
4.2.2.4	afficherLicense	16
4.2.2.5	afficherLigne	16
4.2.2.6	afficherNom	17
4.2.2.7	afficherPartieFinie	17
4.2.2.8	afficherPosition	17
4.2.2.9	afficherScore	17
4.2.2.10	afficherScoreEntier	18
4.2.2.11	afficherScoreTotal	18
4.2.2.12	afficherStruct	19
4.2.2.13	printGameConfig	19
4.3	argument_main.c File Reference	19
4.3.1	Detailed Description	20

4.3.2	Function Documentation	20
4.3.2.1	searchArgument	20
4.4	argument_main.h File Reference	21
4.4.1	Detailed Description	22
4.4.2	Macro Definition Documentation	22
4.4.2.1	CHAINE_LECTURE_FICHER	22
4.4.2.2	CHAINE_LECTURE_FICHER_RED	22
4.4.2.3	CHAINE_OUVERTURE_FICHER	22
4.4.2.4	CHAINE_OUVERTURE_FICHER_RED	22
4.4.2.5	LECTURE_FICHER	22
4.4.2.6	OUVERTURE_FICHER	22
4.4.3	Function Documentation	22
4.4.3.1	searchArgument	22
4.5	config.c File Reference	23
4.5.1	Detailed Description	24
4.5.2	Function Documentation	24
4.5.2.1	addConfigListFile	24
4.5.2.2	freeListGameConfig	24
4.5.2.3	makeConfigFile	25
4.5.2.4	makeConfigListFile	25
4.5.2.5	makeListGameConfig	25
4.5.2.6	readConfigFile	26
4.5.2.7	readConfigListFile	26
4.5.2.8	removeConfigFile	27
4.5.2.9	removeConfigListFile	27
4.6	config.h File Reference	28
4.6.1	Detailed Description	29
4.6.2	Macro Definition Documentation	30
4.6.2.1	NOM_DOSSIER_CONFIG	30
4.6.2.2	NOM_FICHER_CONFIG	30
4.6.3	Function Documentation	30
4.6.3.1	addConfigListFile	30
4.6.3.2	freeListGameConfig	30
4.6.3.3	makeConfigFile	31
4.6.3.4	makeConfigListFile	31
4.6.3.5	makeListGameConfig	31
4.6.3.6	readConfigFile	32
4.6.3.7	readConfigListFile	32
4.6.3.8	removeConfigFile	33
4.6.3.9	removeConfigListFile	33

4.7	emplacement_fichier.c File Reference	34
4.7.1	Detailed Description	35
4.7.2	Function Documentation	36
4.7.2.1	changerCheminFichier	36
4.7.2.2	creationPreferences	36
4.7.2.3	lectureCheminFichier	37
4.7.2.4	lecturePreferences	37
4.8	emplacement_fichier.h File Reference	38
4.8.1	Detailed Description	39
4.8.2	Macro Definition Documentation	40
4.8.2.1	NOM_DOSSIER_CSUPER	40
4.8.2.2	NOM_FICHER_PREFERENCES	40
4.8.3	Function Documentation	40
4.8.3.1	changerCheminFichier	40
4.8.3.2	creationPreferences	40
4.8.3.3	lectureCheminFichier	41
4.8.3.4	lecturePreferences	41
4.9	fichier.c File Reference	42
4.9.1	Detailed Description	43
4.9.2	Function Documentation	44
4.9.2.1	ecrireFichier	44
4.9.2.2	lireFichier	44
4.9.2.3	nouveauScore	45
4.9.2.4	ouvrirFichierExtension	46
4.9.2.5	renommerFichier	47
4.9.2.6	supprimerFichier	47
4.10	fichier.h File Reference	48
4.10.1	Detailed Description	49
4.10.2	Macro Definition Documentation	49
4.10.2.1	EXTENSION_FICHER	49
4.10.2.2	TAILLE_MAX_NOM_FICHER	49
4.10.2.3	TYPE_FICHER	49
4.10.3	Function Documentation	49
4.10.3.1	ecrireFichier	49
4.10.3.2	lireFichier	50
4.10.3.3	nouveauScore	51
4.10.3.4	ouvrirFichierExtension	51
4.10.3.5	renommerFichier	52
4.10.3.6	supprimerFichier	52
4.11	fonction.c File Reference	52

4.11.1	Detailed Description	53
4.11.2	Function Documentation	54
4.11.2.1	ajoutExtension	54
4.11.2.2	compareFlottantCroissant	54
4.11.2.3	compareFlottantDecroissant	54
4.11.2.4	fermerFichier	54
4.11.2.5	lireTailleFichier	54
4.11.2.6	mauvais_choix	55
4.11.2.7	myAlloc	55
4.11.2.8	myRealloc	55
4.11.2.9	ouvrirFichier	55
4.11.2.10	systemEfface	55
4.12	fonction.h File Reference	55
4.12.1	Detailed Description	56
4.12.2	Macro Definition Documentation	57
4.12.2.1	FAUX	57
4.12.2.2	VRAI	57
4.12.3	Function Documentation	57
4.12.3.1	ajoutExtension	57
4.12.3.2	compareFlottantCroissant	57
4.12.3.3	compareFlottantDecroissant	57
4.12.3.4	fermerFichier	57
4.12.3.5	lireTailleFichier	57
4.12.3.6	mauvais_choix	58
4.12.3.7	myAlloc	58
4.12.3.8	myRealloc	58
4.12.3.9	ouvrirFichier	58
4.12.3.10	systemEfface	58
4.13	interface.c File Reference	58
4.13.1	Detailed Description	59
4.13.2	Function Documentation	60
4.13.2.1	afficheFichier	60
4.13.2.2	afficheFichierLocale	61
4.13.2.3	chargerPartie	61
4.13.2.4	chargerPartieLocale	62
4.13.2.5	jouer	63
4.13.2.6	lireCheminFichier	64
4.13.2.7	listerFichier	65
4.13.2.8	menuPreferences	65
4.13.2.9	menuPrincipal	65

4.13.2.10 newGameConfig	66
4.13.2.11 nouveauCheminFichier	67
4.13.2.12 nouvellePartie	67
4.13.2.13 printGameConfigFile	69
4.13.2.14 printListGameConfig	69
4.13.2.15 removeGameConfig	69
4.13.2.16 supprimerFichierNom	70
4.14 interface.h File Reference	70
4.14.1 Detailed Description	72
4.14.2 Enumeration Type Documentation	72
4.14.2.1 MenuPreferences	72
4.14.2.2 MenuPrincipal	72
4.14.3 Function Documentation	72
4.14.3.1 afficheFichier	72
4.14.3.2 afficheFichierLocale	73
4.14.3.3 chargerPartie	74
4.14.3.4 chargerPartieLocale	75
4.14.3.5 jouer	76
4.14.3.6 lireCheminFichier	77
4.14.3.7 listerFichier	78
4.14.3.8 menuPreferences	78
4.14.3.9 menuPrincipal	78
4.14.3.10 newGameConfig	79
4.14.3.11 nouveauCheminFichier	80
4.14.3.12 nouvellePartie	80
4.14.3.13 printGameConfigFile	82
4.14.3.14 printListGameConfig	82
4.14.3.15 removeGameConfig	82
4.14.3.16 supprimerFichierNom	83
4.15 main.c File Reference	83
4.15.1 Detailed Description	84
4.15.2 Function Documentation	85
4.15.2.1 main	85
4.16 main.h File Reference	86
4.16.1 Detailed Description	87
4.16.2 Function Documentation	87
4.16.2.1 main	88
4.17 menu.c File Reference	90
4.17.1 Detailed Description	90
4.17.2 Function Documentation	91

4.17.2.1	menuContinuer	91
4.17.2.2	menuDebutPartie	91
4.17.2.3	menuDistribue	92
4.17.2.4	menuGameConfig	92
4.17.2.5	menuNomFichier	93
4.17.2.6	menuNomJoueur	93
4.17.2.7	menuNouveauChemin	94
4.17.2.8	menuNumJoueur	95
4.17.2.9	menuPointsJoueur	95
4.17.2.10	menuSupprimer	96
4.18	menu.h File Reference	96
4.18.1	Detailed Description	97
4.18.2	Function Documentation	98
4.18.2.1	menuContinuer	98
4.18.2.2	menuDebutPartie	98
4.18.2.3	menuDistribue	98
4.18.2.4	menuGameConfig	99
4.18.2.5	menuNomFichier	99
4.18.2.6	menuNomJoueur	100
4.18.2.7	menuNouveauChemin	100
4.18.2.8	menuNumJoueur	101
4.18.2.9	menuPointsJoueur	101
4.18.2.10	menuSupprimer	102
4.19	saisie_clavier.c File Reference	102
4.19.1	Detailed Description	103
4.19.2	Function Documentation	103
4.19.2.1	clean_stdin	103
4.19.2.2	saisieClavierCaractere	104
4.19.2.3	saisieClavierChaine	105
4.19.2.4	saisieClavierDouble	105
4.19.2.5	saisieClavierEntier	106
4.19.2.6	saisieClavierFlottant	106
4.19.2.7	saisieClavierFlottantSansVirgule	106
4.19.2.8	systemPause	107
4.20	saisie_clavier.h File Reference	107
4.20.1	Detailed Description	108
4.20.2	Macro Definition Documentation	108
4.20.2.1	NB_CHARACTER_DOUB	108
4.20.2.2	NB_CHARACTER_FLOT	108
4.20.2.3	NB_CHARACTER_INT	108

4.20.3	Function Documentation	108
4.20.3.1	clean_stdin	108
4.20.3.2	saisieClavierCaractere	108
4.20.3.3	saisieClavierChaine	109
4.20.3.4	saisieClavierDouble	109
4.20.3.5	saisieClavierEntier	110
4.20.3.6	saisieClavierFlottant	111
4.20.3.7	saisieClavierFlottantSansVirgule	111
4.20.3.8	systemPause	111
4.21	structure.c File Reference	112
4.21.1	Detailed Description	113
4.21.2	Function Documentation	113
4.21.2.1	ajoutDistribueStruct	113
4.21.2.2	calculPosition	113
4.21.2.3	creerFichierStruct	114
4.21.2.4	debNouvTour	114
4.21.2.5	depScoreMax	114
4.21.2.6	fermeeFichierStruct	115
4.21.2.7	finNouvTour	115
4.21.2.8	maxNbTour	115
4.21.2.9	rechercheNumJoueur	115
4.22	structure.h File Reference	116
4.22.1	Detailed Description	117
4.22.2	Macro Definition Documentation	117
4.22.2.1	TAILLE_MAX_NOM	117
4.22.2.2	VERSION	117
4.22.3	Function Documentation	117
4.22.3.1	ajoutDistribueStruct	117
4.22.3.2	calculPosition	118
4.22.3.3	creerFichierStruct	119
4.22.3.4	debNouvTour	119
4.22.3.5	depScoreMax	120
4.22.3.6	fermeeFichierStruct	121
4.22.3.7	finNouvTour	121
4.22.3.8	maxNbTour	121
4.22.3.9	rechercheNumJoueur	121
Index		123

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Fichier_Jeu	5
game_config	7
list_game_config	8

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

affichage.c	Fonctions qui gerent l'affichage des scores du logiciel	9
affichage.h	Prototypes des fonctions qui gerent l'affichage des scores du logiciel	14
argument_main.c	Lancement du programme	19
argument_main.h	Lancement du programme	21
config.c	Fonction des fichiers lies a la config	23
config.h	Prototypes des fonction des fichiers lies a la config	28
emplacement_fichier.c	Fonctions qui l'emplacement des fichiers sauvegardes	34
emplacement_fichier.h	Prototypes des fonctions qui l'emplacement des fichiers sauvegardes	38
fichier.c	Fonction de gestion des fichiers	42
fichier.h	Prototypes des fonction de gestion des fichiers	48
fonction.c	Fonctions essentielles au fonctionnement du programme	52
fonction.h	Prototypes des fonctions essentielles au fonctionnement du programme	55
interface.c	Fonctions qui gerent l'interface graphique du logiciel	58
interface.h	Prototypes des fonctions qui gerent l'interface graphique du logiciel	70
main.c	Lancement du programme	83
main.h	Lancement du programme	86
menu.c	Fonctions qui gerent les menus demandant de rentrer des valeurs du logiciel	90
menu.h	Prototypes des fonctions qui gerent les menus demandant de rentrer des valeurs du logiciel	96
saisie_clavier.c	Fonctions de saisie clavier	102

saisie_clavier.h	
Prototypes des fonctions de saisie clavier	107
structure.c	
Fonction de gestion des fichiers de la struction contenant les informations	112
structure.h	
Prototypes des fonction de gestion des fichiers de la struction contenant les informations . . .	116

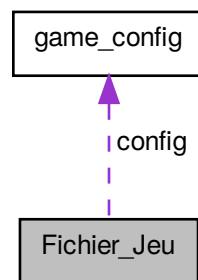
Chapter 3

Data Structure Documentation

3.1 Fichier_Jeu Struct Reference

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

Collaboration diagram for Fichier_Jeu:



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 contenat 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_distributor](#)
- char [number_after_comma](#)
- char [name](#) [TAILLE_MAX_NOM]

3.2.1 Detailed Description

Type representant une configuration de jeu

3.2.2 Field Documentation

3.2.2.1 char name[TAILLE_MAX_NOM]

Le nom de la configuration de jeu

3.2.2.2 float nb_max

Nombre maximum que peut prendre un joueur.

3.2.2.3 char number_after_comma

Le nombre de chiffres apres la virgule dans l'affichage

3.2.2.4 char sens_premier

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

3.2.2.5 char tour_par_tour

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

3.2.2.6 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

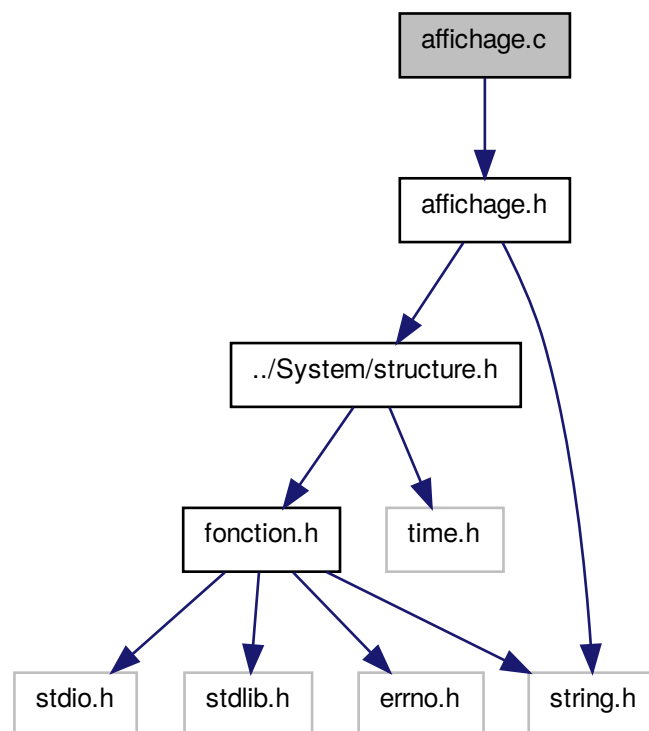
File Documentation

4.1 affichage.c File Reference

Fonctions qui gerent l'affichage des scores du logiciel.

```
#include "affichage.h"
```

Include dependency graph for affichage.c:



Functions

- void `afficherNom` (`Fichier_Jeu` *ptr_struct_fichier, int *ptr_taille_ligne)

- void [afficherLigne](#) (int taille_ligne)
- void [afficherScoreTotal](#) ([Fichier_Jeu](#) *ptr_struct_fichier)
- void [afficherDistribue](#) ([Fichier_Jeu](#) *ptr_struct_fichier)
- void [afficherEnTete](#) ([Fichier_Jeu](#) *ptr_struct_fichier)
- void [afficherScoreEntier](#) ([Fichier_Jeu](#) *ptr_struct_fichier)
- void [afficherPosition](#) ([Fichier_Jeu](#) *ptr_struct_fichier)
- void [afficherScore](#) ([Fichier_Jeu](#) *ptr_struct_fichier)
- void [afficherStruct](#) ([Fichier_Jeu](#) *ptr_struct_fichier)
- void [afficherPartieFinie](#) ([Fichier_Jeu](#) *ptr_struct_fichier)
- void [afficherChaineTroisTab](#) (char *chaine)
- void [afficherLicense](#) ()
- void [printGameConfig](#) (game_config config)

4.1.1 Detailed Description

Fonctions qui gerent l'affichage des scores du logiciel.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.1.2 Function Documentation

4.1.2.1 void [afficherChaineTroisTab](#) (char * *chaine*)

Affiche la chaine de caractere passe en parametre centrer sur un espace de trois tabulations

Parameters

in	* <i>chaine</i>	une chaine de caractere
----	-----------------	-------------------------

4.1.2.2 void [afficherDistribue](#) ([Fichier_Jeu](#) * *ptr_struct_fichier*)

Affiche la personne devant distribuer si l'on utilise un distributeur

Parameters

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

4.1.2.3 void [afficherEnTete](#) ([Fichier_Jeu](#) * *ptr_struct_fichier*)

Affiche l'en tete de la structure

Parameters

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

Here is the call graph for this function:



4.1.2.4 void afficherLicense ()

Affiche la license

4.1.2.5 void afficherLigne (int *taille_ligne*)

Affiche *taille_ligne* - apres une tabulation

Parameters

in	<i>taille_ligne</i>	la taille de la ligne
----	---------------------	-----------------------

4.1.2.6 void afficherNom ([Fichier_Jeu](#) * *ptr_struct_fichier*, int * *ptr_taille_ligne*)

Affiche les noms des joueurs et calcule la taille de la ligne

Parameters

in	<i>*ptr_struct_fichier</i>	un pointeur sur la structure Fichier_Jeu
in, out	<i>*ptr_taille_ligne</i>	un pointeur sur la taille de la ligne a modifier

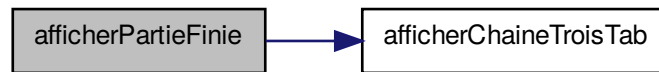
4.1.2.7 void afficherPartieFinie ([Fichier_Jeu](#) * *ptr_struct_fichier*)

Affiche un podium des resultats

Parameters

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

Here is the call graph for this function:



4.1.2.8 void afficherPosition (Fichier_Jeu * ptr_struct_fichier)

Affiche les positions des joueurs

Parameters

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

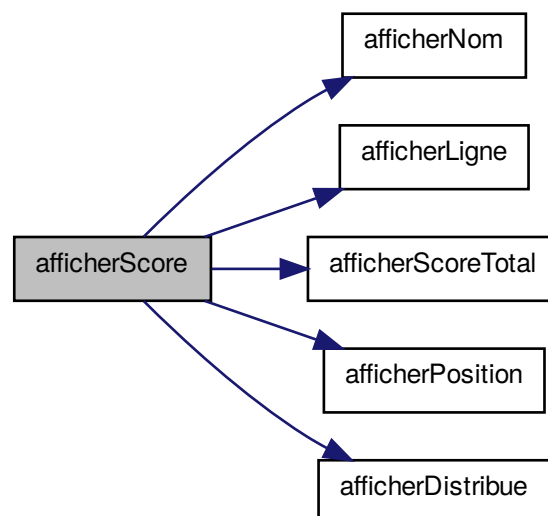
4.1.2.9 void afficherScore (Fichier_Jeu * ptr_struct_fichier)

Affiche les noms des personnes avec leurs scores totaux ainsi que la personne devant distribuer a partir d'une structure [Fichier_Jeu](#)

Parameters

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

Here is the call graph for this function:



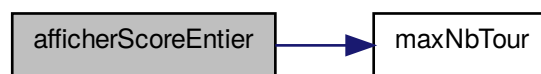
4.1.2.10 void afficherScoreEntier (Fichier_Jeu * ptr_struct_fichier)

Affiche tout les scores de tous les joueurs a chaque tour.

Parameters

in	*ptr_struct_fichier	un pointeur sur la structure Fichier_Jeu
----	---------------------	--

Here is the call graph for this function:



4.1.2.11 void afficherScoreTotal (Fichier_Jeu * ptr_struct_fichier)

Affiche le score total des joueurs

Parameters

in	*ptr_struct_fichier	un pointeur sur la structure Fichier_Jeu
----	---------------------	--

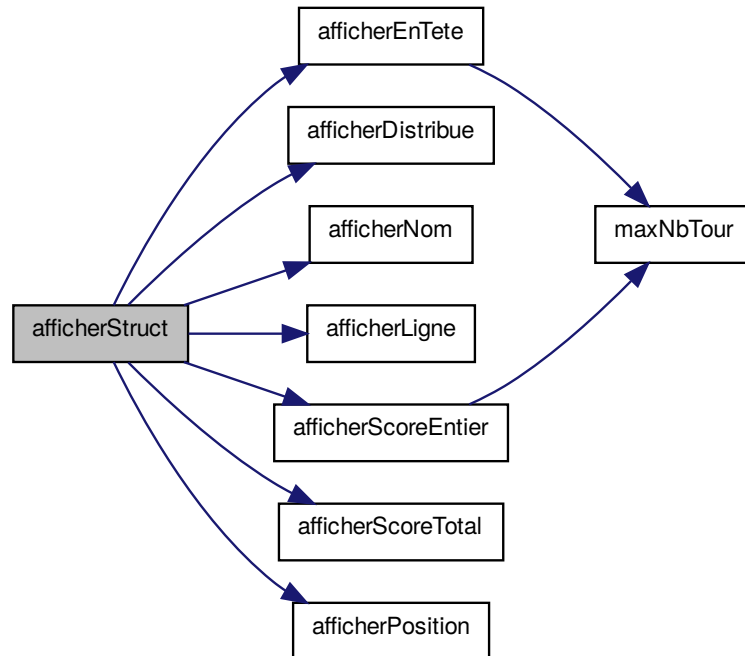
4.1.2.12 void afficherStruct (Fichier_Jeu * ptr_struct_fichier)

Affiche toutes les donnes d'une structure [Fichier_Jeu](#)

Parameters

in	*ptr_struct_fichier	un pointeur sur la structure Fichier_Jeu
----	---------------------	--

Here is the call graph for this function:



4.1.2.13 void printGameConfig (game_config config)

Print the game config

Parameters

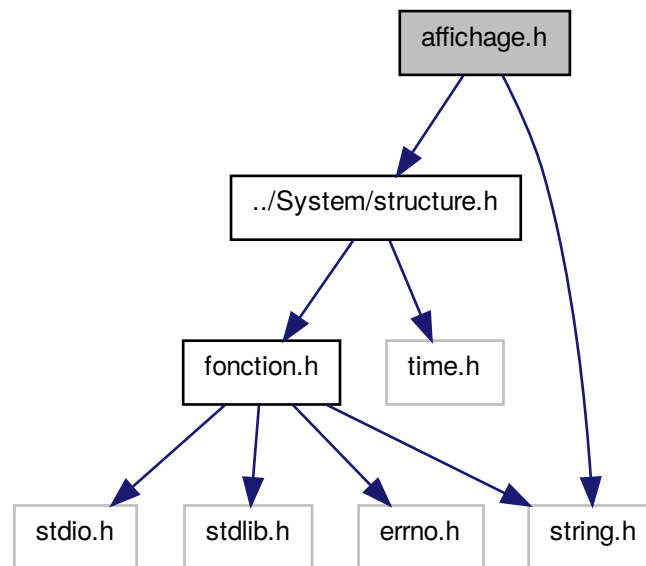
in	config	a game config
----	--------	---------------

4.2 affichage.h File Reference

Prototypes des fonctions qui gerent l'affichage des scores du logiciel.

```
#include "../System/structure.h"
#include <string.h>
```


Include dependency graph for affichage.h:



Functions

- void `afficherNom` (`Fichier_Jeu *ptr_struct_fichier`, `int *ptr_taille_ligne`)
- void `afficherLigne` (`int taille_ligne`)
- void `afficherScoreTotal` (`Fichier_Jeu *ptr_struct_fichier`)
- void `afficherDistribue` (`Fichier_Jeu *ptr_struct_fichier`)
- void `afficherEnTete` (`Fichier_Jeu *ptr_struct_fichier`)
- void `afficherScoreEntier` (`Fichier_Jeu *ptr_struct_fichier`)
- void `afficherPosition` (`Fichier_Jeu *ptr_struct_fichier`)
- void `afficherScore` (`Fichier_Jeu *ptr_struct_fichier`)
- void `afficherStruct` (`Fichier_Jeu *ptr_struct_fichier`)
- void `afficherPartieFinie` (`Fichier_Jeu *ptr_struct_fichier`)
- void `afficherChaineTroisTab` (`char *chaine`)
- void `afficherLicense` ()
- void `printGameConfig` (`game_config config`)

4.2.1 Detailed Description

Prototypes des fonctions qui gerent l'affichage des scores du logiciel.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.2.2 Function Documentation

4.2.2.1 void afficherChaineTroisTab (char * *chaine*)

Affiche la chaine de caractere passe en parametre centrer sur un espace de trois tabulations

Parameters

in	* <i>chaine</i>	une chaine de caractere
----	-----------------	-------------------------

4.2.2.2 void afficherDistribue (Fichier_Jeu * *ptr_struct_fichier*)

Affiche la personne devant distribuer si l'on utilise un distributeur

Parameters

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

4.2.2.3 void afficherEnTete (Fichier_Jeu * *ptr_struct_fichier*)

Affiche l'en tete de la structure

Parameters

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

Here is the call graph for this function:



4.2.2.4 void afficherLicense ()

Affiche la license

4.2.2.5 void afficherLigne (int *taille_ligne*)

Affiche *taille_ligne* - apres une tabulation

Parameters

in	<i>taille_ligne</i>	la taille de la ligne
----	---------------------	-----------------------

4.2.2.6 void afficherNom (Fichier_Jeu * ptr_struct_fichier, int * ptr_taille_ligne)

Affiche les noms des joueurs et calcule la taille de la ligne

Parameters

in	<i>*ptr_struct_fichier</i>	un pointeur sur la structure Fichier_Jeu
in, out	<i>*ptr_taille_ligne</i>	un pointeur sur la taille de la ligne a modifier

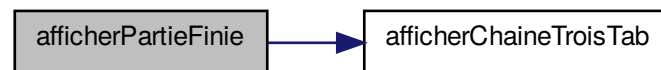
4.2.2.7 void afficherPartieFinie (Fichier_Jeu * ptr_struct_fichier)

Affiche un podium des resultats

Parameters

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

Here is the call graph for this function:

**4.2.2.8 void afficherPosition (Fichier_Jeu * ptr_struct_fichier)**

Affiche les positions des joueurs

Parameters

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

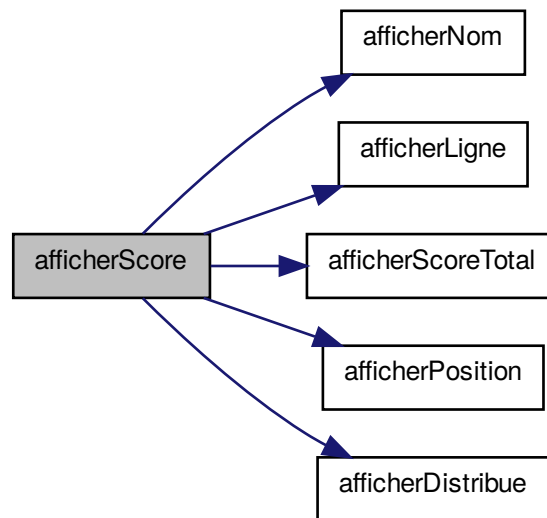
4.2.2.9 void afficherScore (Fichier_Jeu * ptr_struct_fichier)

Affiche les noms des personnes avec leurs scores totaux ainsi que la personne devant distribuer a partir d'une structure [Fichier_Jeu](#)

Parameters

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

Here is the call graph for this function:



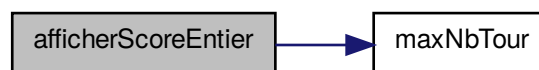
4.2.2.10 void afficherScoreEntier (Fichier_Jeu * ptr_struct_fichier)

Affiche tout les scores de tous les joueurs a chaque tour.

Parameters

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

Here is the call graph for this function:



4.2.2.11 void afficherScoreTotal (Fichier_Jeu * ptr_struct_fichier)

Affiche le score total des joueurs

Parameters

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

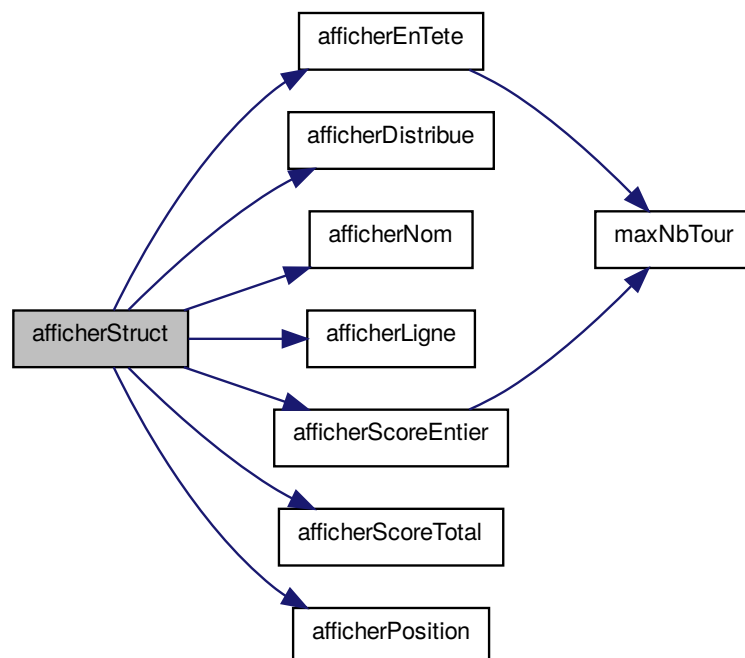
4.2.2.12 void afficherStruct (Fichier_Jeu * ptr_struct_fichier)

Affiche toutes les donnes d'une structure [Fichier_Jeu](#)

Parameters

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

Here is the call graph for this function:



4.2.2.13 void printGameConfig (game_config config)

Print the game config

Parameters

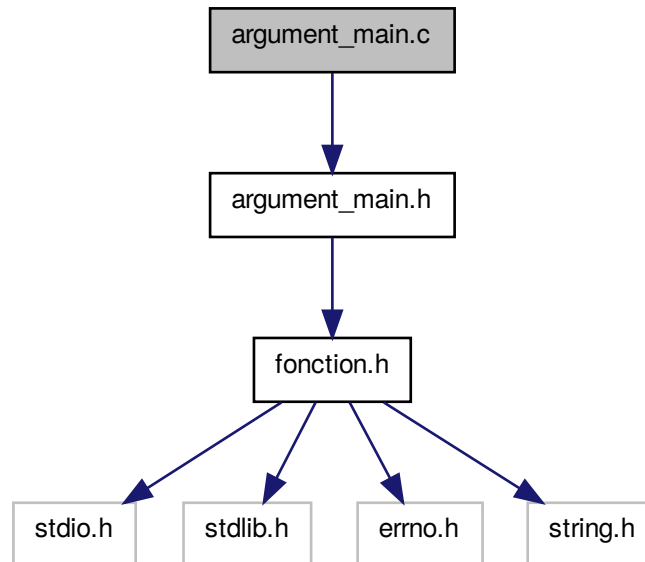
in	<i>config</i>	a game config
----	---------------	---------------

4.3 argument_main.c File Reference

Lancement du programme.

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

Include dependency graph for argument_main.c:



Functions

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

4.3.1 Detailed Description

Lancement du programme.

Author

Remi BERTHO

Date

21/03/14

Version

2.1.0

4.3.2 Function Documentation

4.3.2.1 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

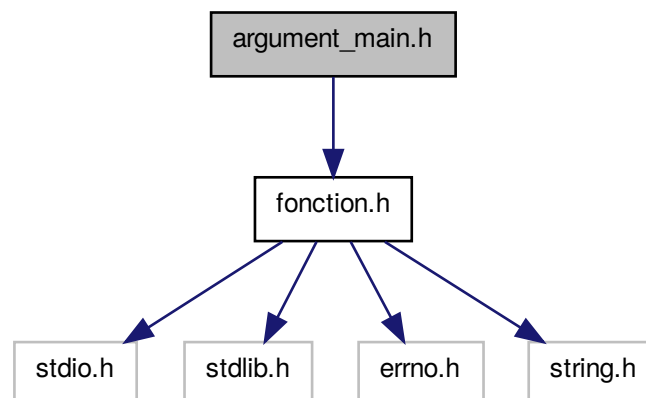
VRAI si la fonction a trouve un argument, FAUX sinon

4.4 argument_main.h File Reference

Lancement du programme.

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

Include dependency graph for argument_main.h:



Macros

- `#define CHAINE_LECTURE_FICHER "--read"`
- `#define CHAINE_LECTURE_FICHER_RED "-r"`
- `#define LECTURE_FICHER 0`
- `#define CHAINE_OUVERTURE_FICHER "--open"`
- `#define CHAINE_OUVERTURE_FICHER_RED "-o"`
- `#define OUVERTURE_FICHER 1`

Functions

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

4.4.1 Detailed Description

Lancement du programme.

Author

Remi BERTHO

Date

21/03/14

Version

2.1.0

4.4.2 Macro Definition Documentation

4.4.2.1 #define CHAINE_LECTURE_FICHIER "--read"

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

4.4.2.2 #define CHAINE_LECTURE_FICHIER_RED "-r"

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

4.4.2.3 #define CHAINE_OUVERTURE_FICHIER "--open"

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

4.4.2.4 #define CHAINE_OUVERTURE_FICHIER_RED "-o"

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

4.4.2.5 #define LECTURE_FICHIER 0

Definit l'appel a la lecture du fichier a 0

4.4.2.6 #define OUVERTURE_FICHIER 1

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

4.4.3 Function Documentation

4.4.3.1 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

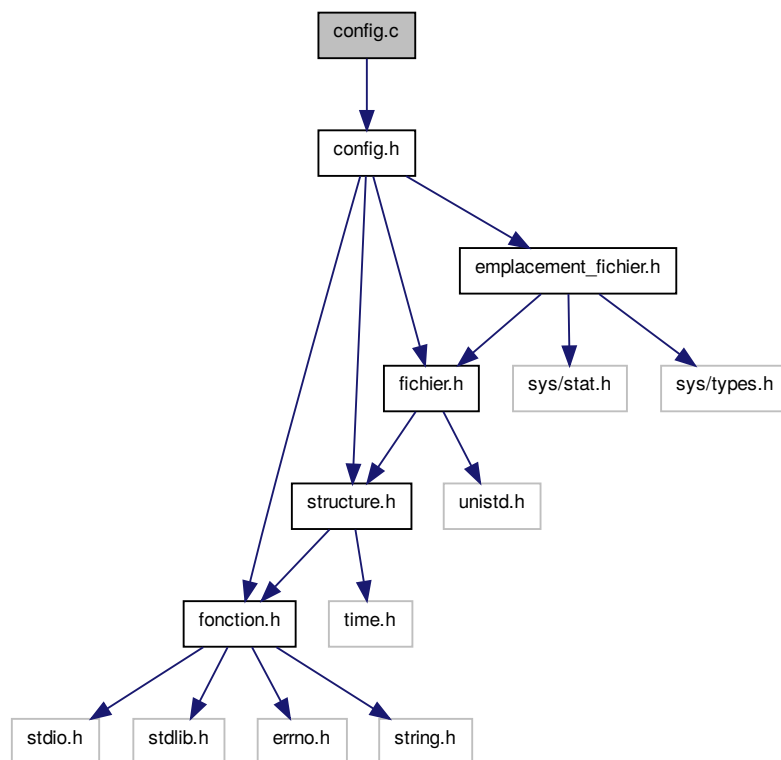
VRAI si la fonction a trouve un argument, FAUX sinon

4.5 config.c File Reference

Fonction des fichiers lies a la config.

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

Include dependency graph for config.c:



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)

4.5.1 Detailed Description

Fonction des fichiers lies a la config.

Author

Remi BERTHO

Date

27/03/14

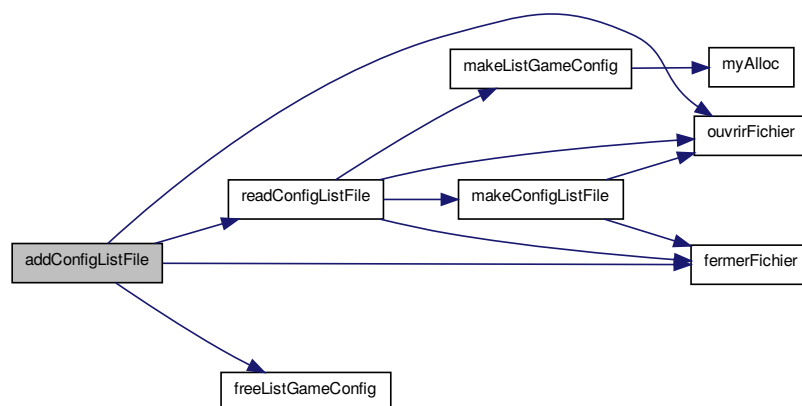
Version

2.2.0

4.5.2 Function Documentation

4.5.2.1 int addConfigListFile (char * new_config_name)

Here is the call graph for this function:

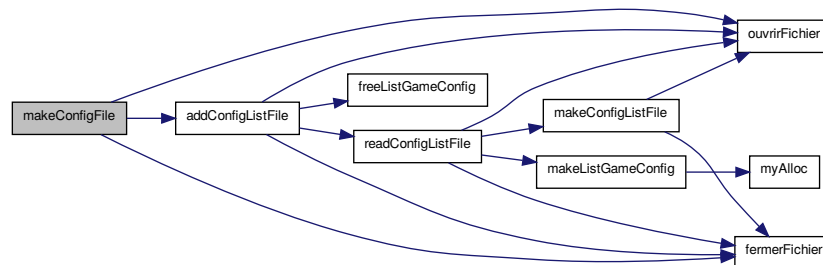


4.5.2.2 void freeListGameConfig (list_game_config * ptr_list_config)

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

4.5.2.3 int makeConfigFile (game_config config)

Here is the call graph for this function:



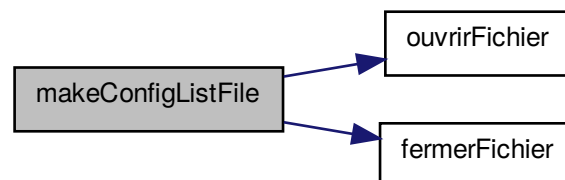
4.5.2.4 int makeConfigListFile ()

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:



4.5.2.5 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:



4.5.2.6 `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.

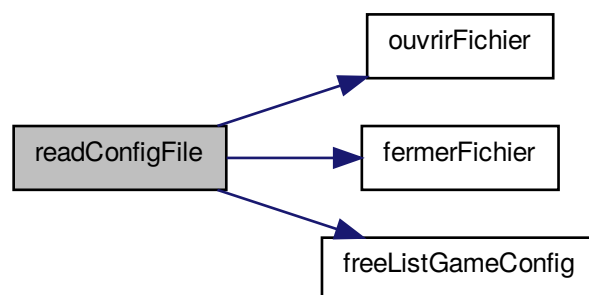
Parameters

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:



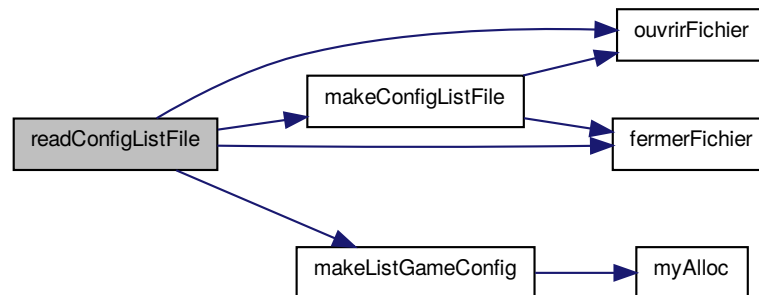
4.5.2.7 `list_game_config * readConfigListFile ()`

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:



4.5.2.8 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

4.5.2.9 int removeConfigListFile (int num_suppr, list_game_config * ptr_list_config)

Enleve une configuration du fichier qui les liste

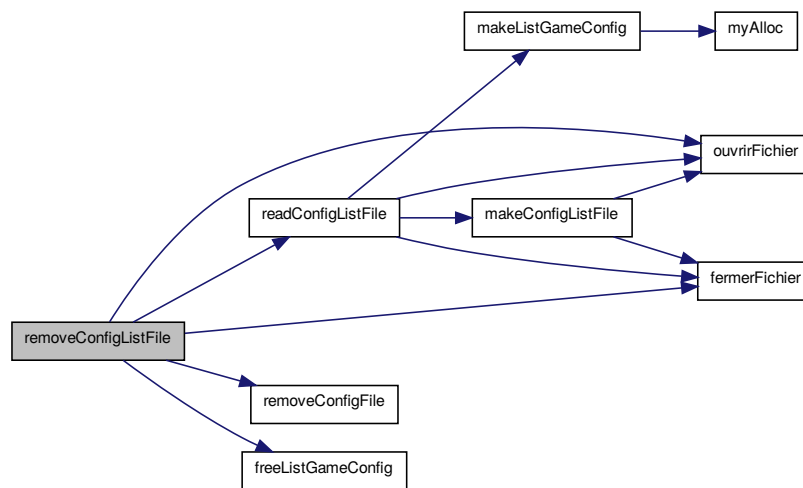
Parameters

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:

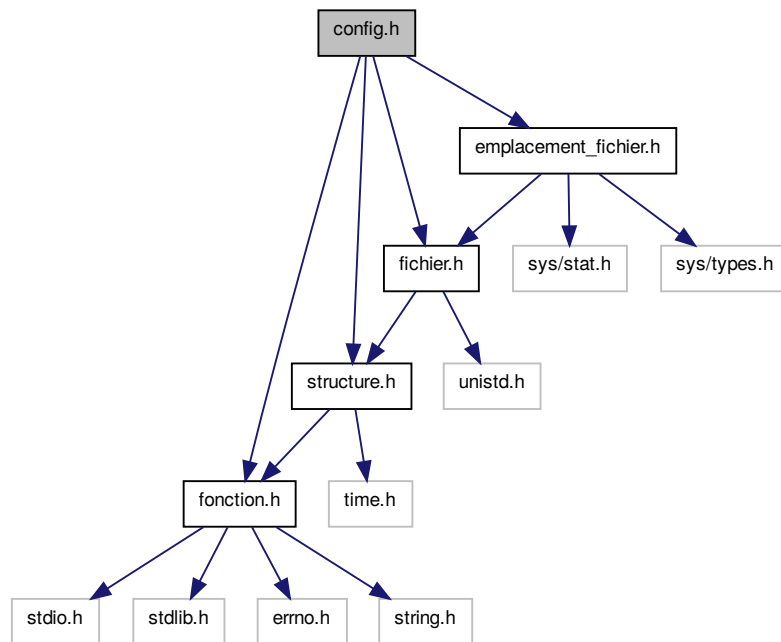


4.6 config.h File Reference

Prototypes des fonction des fichiers lies a la config.

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

Include dependency graph for config.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)

4.6.1 Detailed Description

Prototypes des fonction des fichiers lies a la config.

Author

Remi BERTHO

Date

27/03/14

Version

2.2.0

4.6.2 Macro Definition Documentation

4.6.2.1 #define NOM_DOSSIER_CONFIG "config"

Definit NOM_DOSSIER_CONFIG a "config"

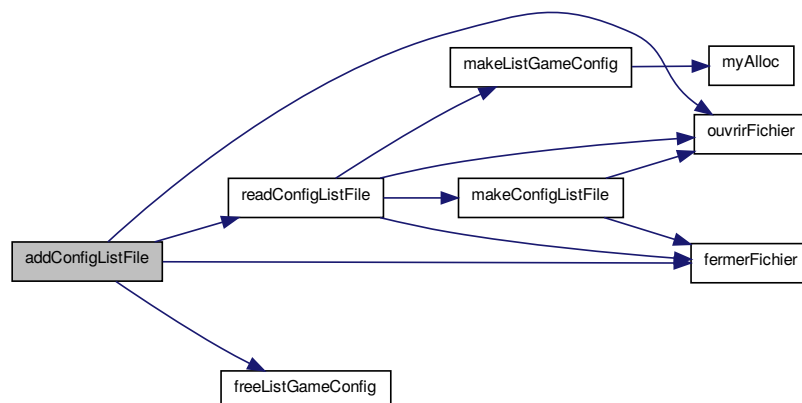
4.6.2.2 #define NOM_FICHER_CONFIG "configuration"

Definit NOM_FICHER_CONFIG a "configuratio"

4.6.3 Function Documentation

4.6.3.1 int addConfigListFile (char * new_config_name)

Here is the call graph for this function:

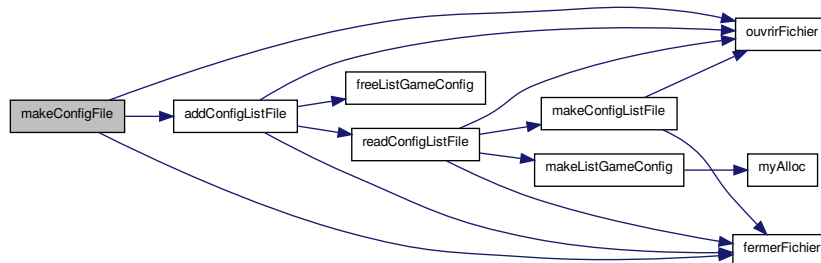


4.6.3.2 void freeListGameConfig (list_game_config * ptr_list_config)

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

4.6.3.3 int makeConfigFile (game_config config)

Here is the call graph for this function:



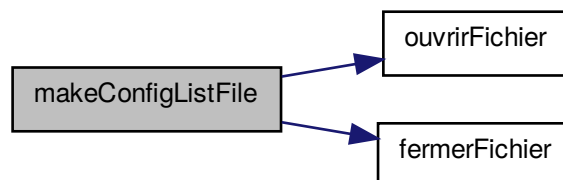
4.6.3.4 int makeConfigListFile ()

Crée les dossiers et le fichier qui va contenir les configs

Returns

VRAI si tout s'est bien passé, FAUX sinon

Here is the call graph for this function:



4.6.3.5 list_game_config* makeListGameConfig (int nb_config)

Crée 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:



4.6.3.6 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.

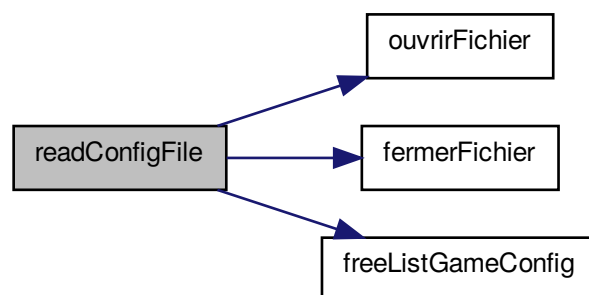
Parameters

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:



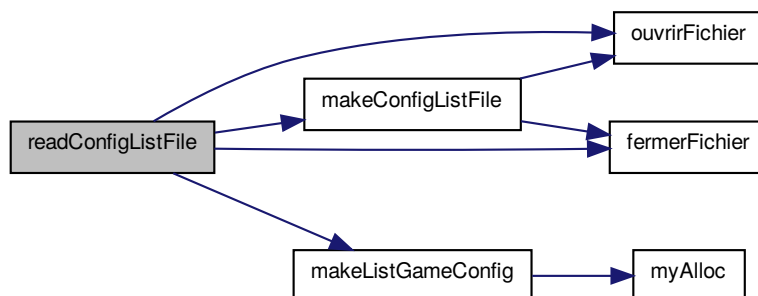
4.6.3.7 list_game_config* readConfigListFile ()

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:



4.6.3.8 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

4.6.3.9 int removeConfigListFile (int num_suppr, list_game_config * ptr_list_config)

Enleve une configuration du fichier qui les liste

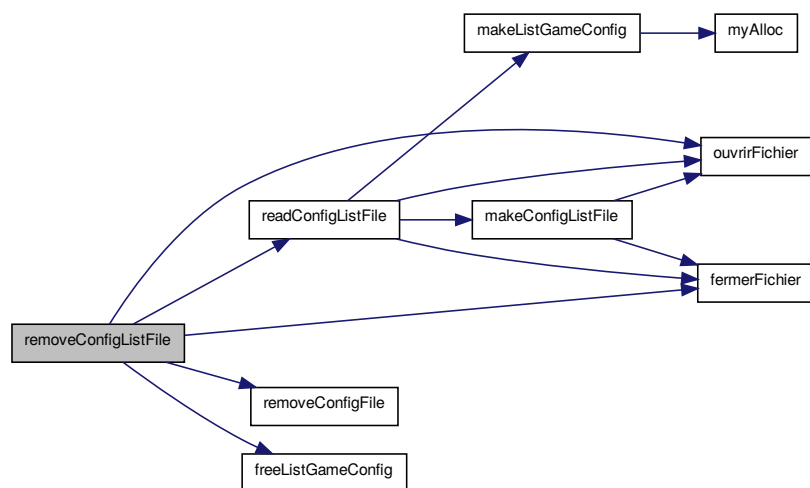
Parameters

in	<i>num_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:

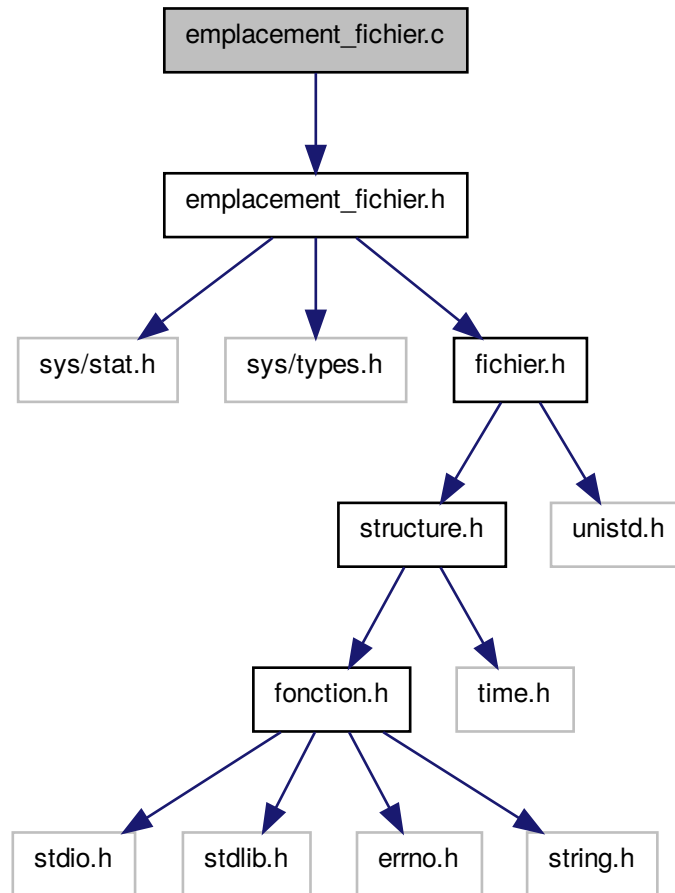


4.7 emplacement_fichier.c File Reference

Fonctions qui l'emplacement des fichiers sauvegardes.

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

Include dependency graph for emplacement_fichier.c:



Functions

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

4.7.1 Detailed Description

Fonctions qui l'emplacement des fichiers sauvegardes.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.7.2 Function Documentation

4.7.2.1 int changerCheminFichier (char * *nouveauChemin*)

Changer le chemin de sauvegarde des fichier par nouveauChemin

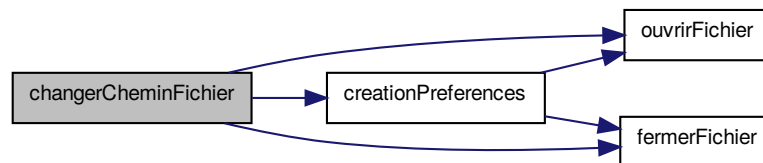
Parameters

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

Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



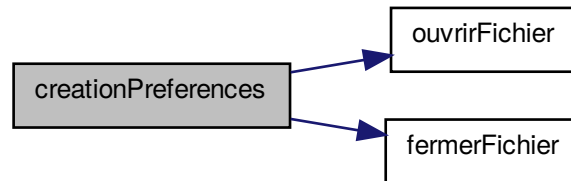
4.7.2.2 void creationPreferences ()

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:

**4.7.2.3 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

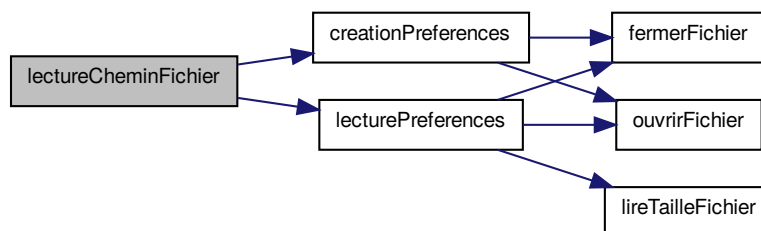
Parameters

<code>in, out</code>	<code>*<i>nom_fichier</i></code>	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:

**4.7.2.4 int lecturePreferences (char * *nom_fichier*)**

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

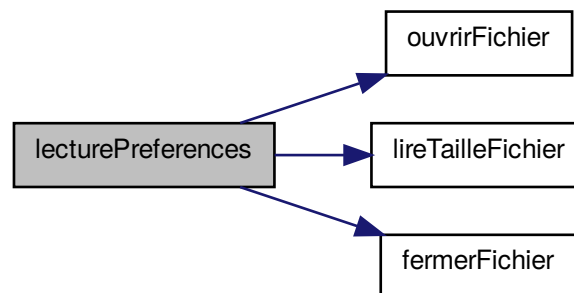
Parameters

<code>in, out</code>	<code>*nom_fichier</code>	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:

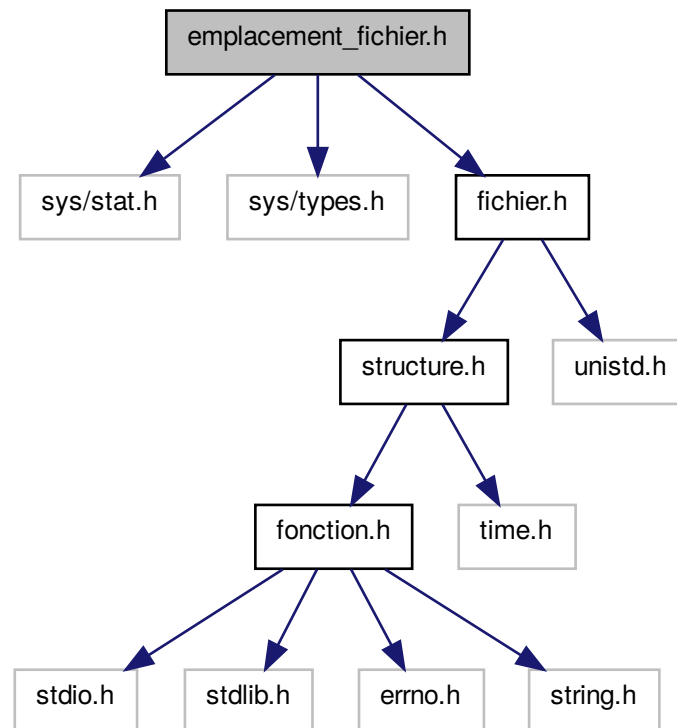


4.8 emplacement_fichier.h File Reference

Prototypes des fonctions qui l'emplacement des fichiers sauvegardes.

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


Include dependency graph for emplacement_fichier.h:



Macros

- `#define NOM_FICHER_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)`

4.8.1 Detailed Description

Prototypes des fonctions qui l'emplacement des fichiers sauvegardes.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.8.2 Macro Definition Documentation

4.8.2.1 #define NOM_DOSSIER_CSUPER ".csuper"

Definit NOM_DOSSIER_CSUPER

4.8.2.2 #define NOM_FICHIER_PREFERENCES "preferences.txt"

Definit NOM_FICHIER_PREFERENCES a "preferences.txt"

4.8.3 Function Documentation

4.8.3.1 int changerCheminFichier (char * *nouveauChemin*)

Changer le chemin de sauvegarde des fichier par nouveauChemin

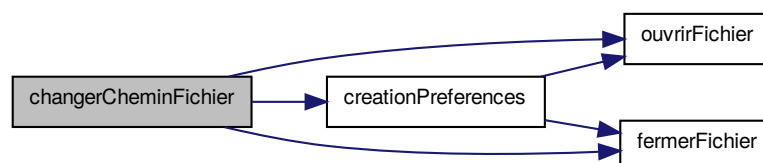
Parameters

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

Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



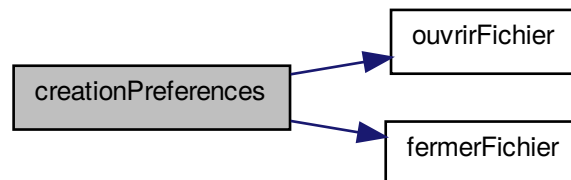
4.8.3.2 int creationPreferences ()

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:



4.8.3.3 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

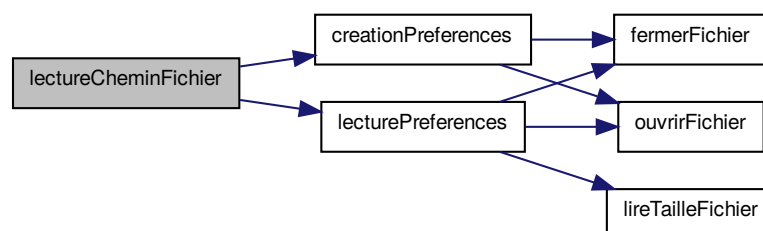
Parameters

<code>in, out</code>	<code>*nom_fichier</code>	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:



4.8.3.4 int lecturePreferences (char * nom_fichier)

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

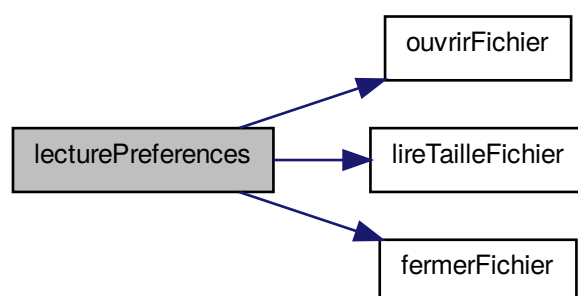
Parameters

<code>in, out</code>	<code>*nom_fichier</code>	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:

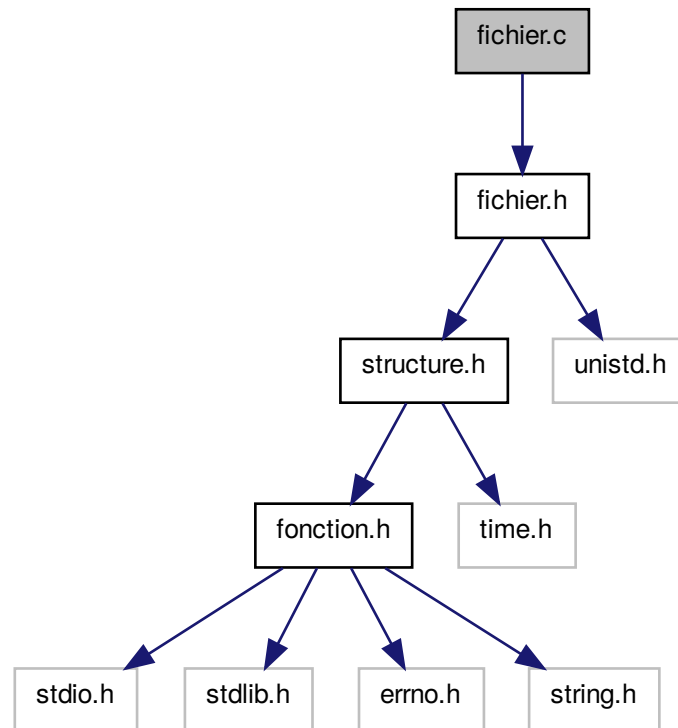


4.9 fichier.c File Reference

Fonction de gestion des fichiers.

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

Include dependency graph for fichier.c:



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)

4.9.1 Detailed Description

Fonction de gestion des fichiers.

Author

Remi BERTHO

Date

09/03/14

Version

2.1.0

4.9.2 Function Documentation

4.9.2.1 `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

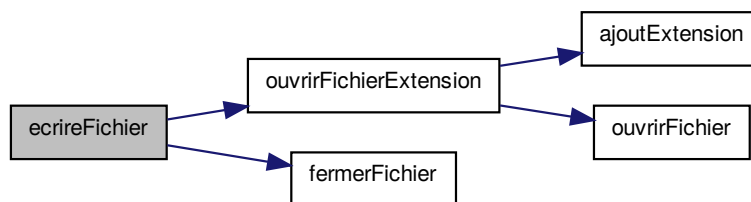
Parameters

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:

4.9.2.2 `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

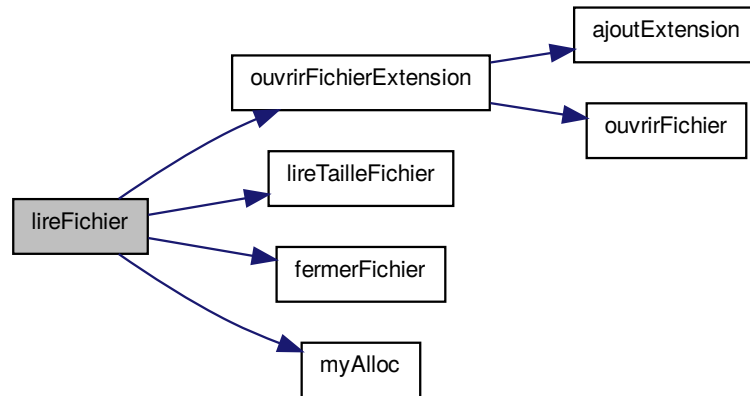
Parameters

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:



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

Mets a jour le fichier avec les nouveaux scores

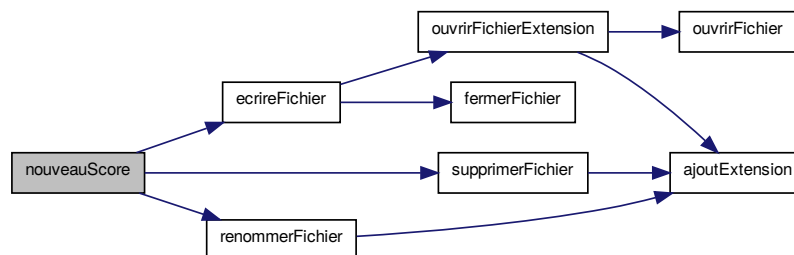
Parameters

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:



4.9.2.4 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

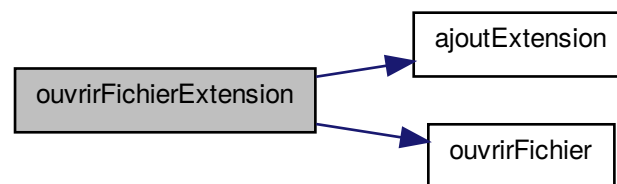
Parameters

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:



4.9.2.5 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:



4.9.2.6 int supprimerFichier (char * *nom*)

Supprime le fichier dont le nom est en parametre

Parameters

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

Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:

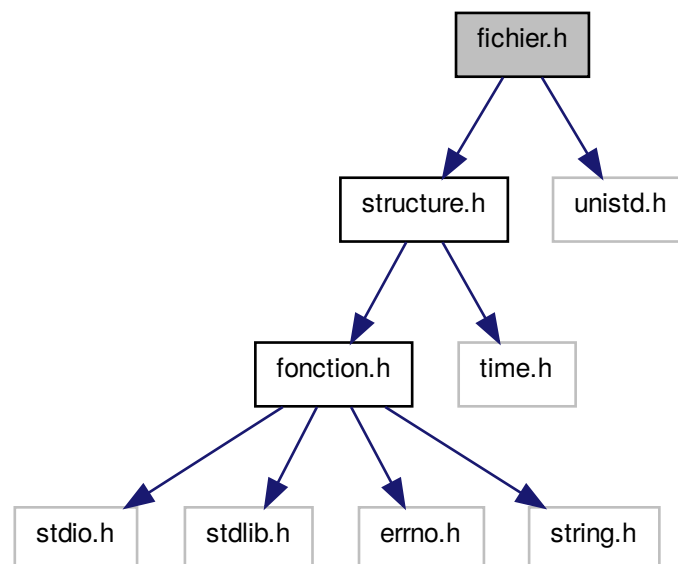


4.10 fichier.h File Reference

Prototypes des fonction de gestion des fichiers.

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

Include dependency graph for fichier.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)`

4.10.1 Detailed Description

Prototypes des fonction de gestion des fichiers.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.10.2 Macro Definition Documentation

4.10.2.1 `#define EXTENSION_FICHIER "csu"`

Definit l'extension du fichier a "csu"

4.10.2.2 `#define TAILLE_MAX_NOM_FICHIER 250`

Definit la taille max d'un nom a 250

4.10.2.3 `#define TYPE_FICHIER "CompteurScoreUniversel"`

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

4.10.3 Function Documentation

4.10.3.1 `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

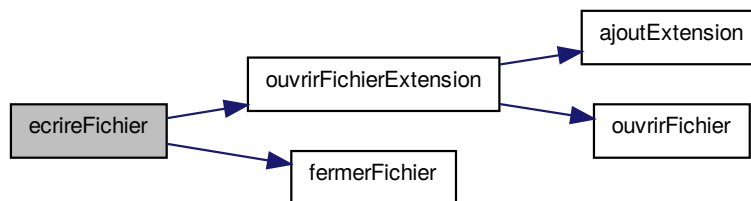
Parameters

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:



4.10.3.2 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

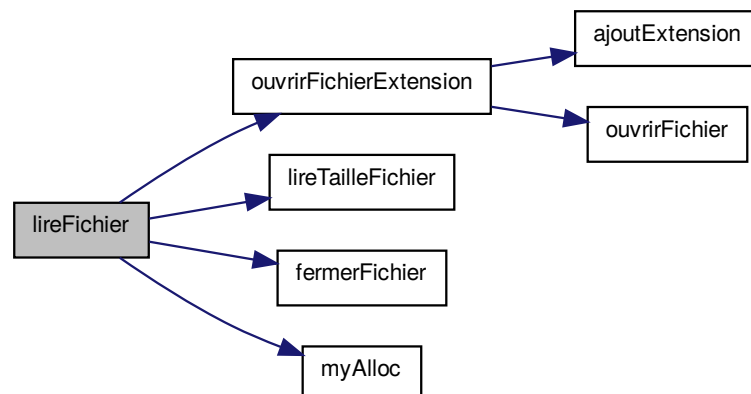
Parameters

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:



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

Mets a jour le fichier avec les nouveaux scores

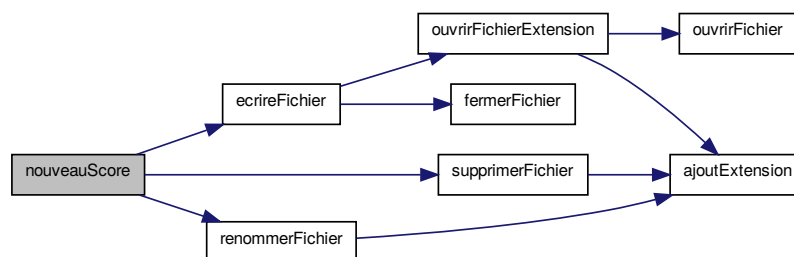
Parameters

in	*nom	le nom du fichier
in	*ptr_struct_fichier	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:



4.10.3.4 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

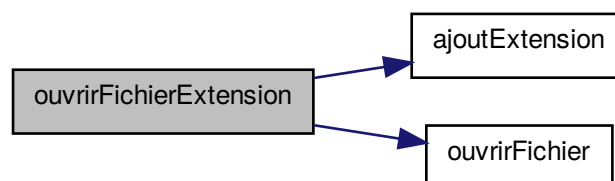
Parameters

in	nom[]	le nom du fichier
in	mode[]	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:



4.10.3.5 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:



4.10.3.6 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

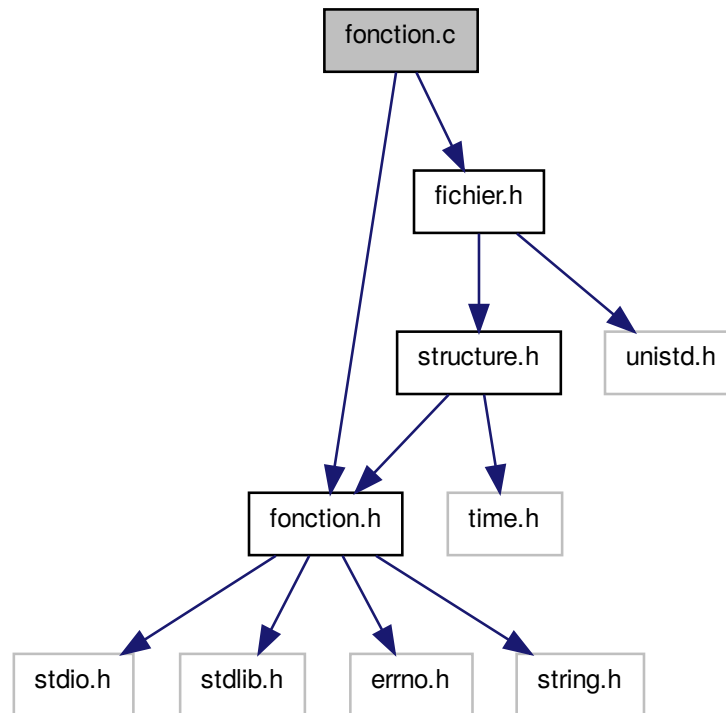
Here is the call graph for this function:



4.11 fonction.c File Reference

Fonctions essentielles au fonctionnement du programme.

```
#include "fonction.h"
#include "fichier.h"
Include dependency graph for fonction.c:
```



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)

4.11.1 Detailed Description

Fonctions essentielles au fonctionnement du programme.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.11.2 Function Documentation

4.11.2.1 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
----	--------------------	-------------------

4.11.2.2 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

4.11.2.3 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

4.11.2.4 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

4.11.2.5 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

4.11.2.6 void mauvais_choix ()

Affiche un message d'erreur.

4.11.2.7 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

4.11.2.8 void myRealloc (void ** *ptr*, int *taille_alloue*)4.11.2.9 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

4.11.2.10 void systemEfface ()

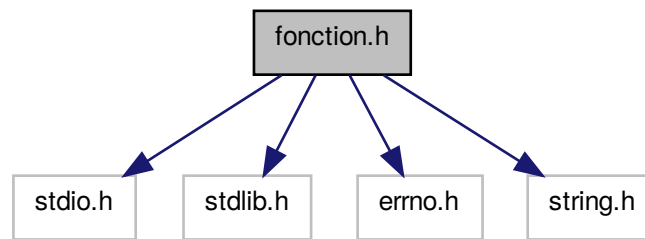
Efface la console de l'utilisateur.

4.12 fonction.h File Reference

Prototypes des fonctions essentielles au fonctionnement du programme.

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

Include dependency graph for fonction.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)

4.12.1 Detailed Description

Prototypes des fonctions essentielles au fonctionnement du programme.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.12.2 Macro Definition Documentation

4.12.2.1 #define FAUX 0

Definit FAUX a 0

4.12.2.2 #define VRAI 1

Definit VRAI a 1

4.12.3 Function Documentation

4.12.3.1 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
----	--------------------	-------------------

4.12.3.2 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

4.12.3.3 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

4.12.3.4 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

4.12.3.5 int lireTailleFichier (FILE * *ptr_fichier*)

Lis la taille du fichier

Parameters

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

Returns

entier ayant la taille du fichier

4.12.3.6 void mauvais_choix ()

Affiche un message d'erreur.

4.12.3.7 void* myAlloc (int *taille_alloue*)

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

Parameters

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

Returns

un pointeur sur la structure alloué

4.12.3.8 void myRealloc (void ** *ptr*, int *taille_alloue*)**4.12.3.9 FILE* ouvrirFichier (char *nom*[], char *mode*[])**

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

Parameters

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

Returns

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

4.12.3.10 void systemEfface ()

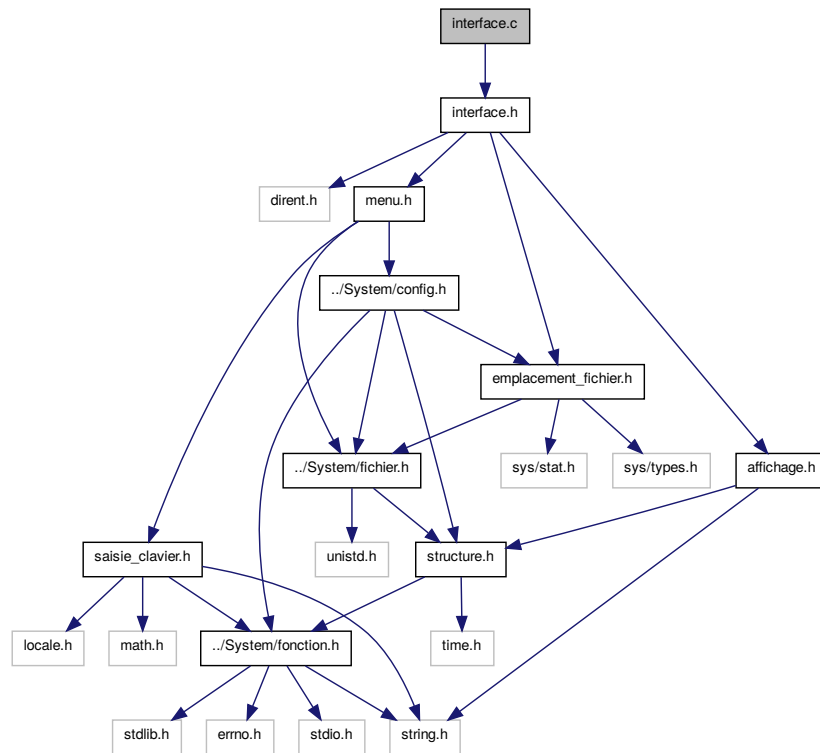
Efface la console de l'utilisateur.

4.13 interface.c File Reference

Fonctions qui gerent l'interface graphique du logiciel.

```
#include "interface.h"
```

Include dependency graph for interface.c:



Functions

- void [afficheFichier](#) ()
- void [supprimerFichierNom](#) ()
- void [listerFichier](#) ()
- void [jouer](#) (Fichier_Jeu *ptr_struct_fichier, char *nom_fichier)
- void [nouvellePartie](#) ()
- void [chargerPartie](#) ()
- void [menuPrincipal](#) ()
- void [menuPreferences](#) ()
- void [nouveauCheminFichier](#) ()
- void [lireCheminFichier](#) ()
- void [chargerPartieLocale](#) (char *nom_fichier)
- void [afficheFichierLocale](#) (char *nom_fichier)
- void [newGameConfig](#) ()
- void [removeGameConfig](#) ()
- void [printListGameConfig](#) ()
- void [printGameConfigFile](#) ()

4.13.1 Detailed Description

Fonctions qui gerent l'interface graphique du logiciel.

Author

Remi BERTHO

Date

09/03/14

Version

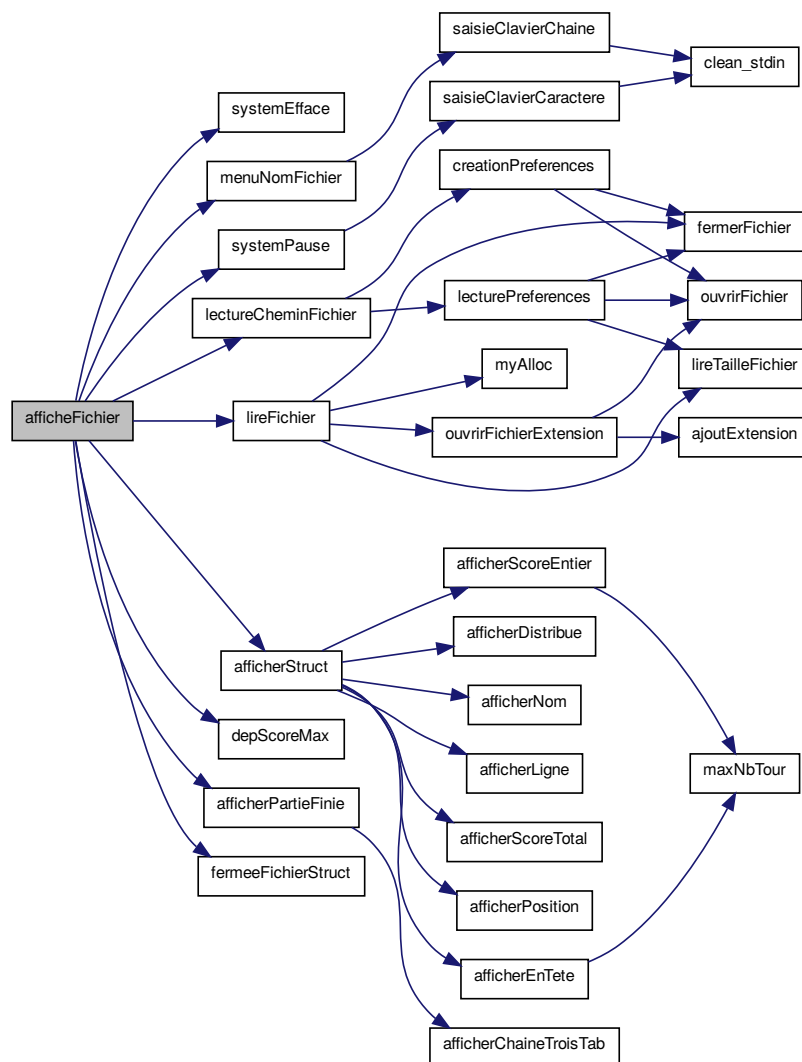
2.1.0

4.13.2 Function Documentation

4.13.2.1 void afficheFichier ()

Demande le nom d'un fichier et l'affiche

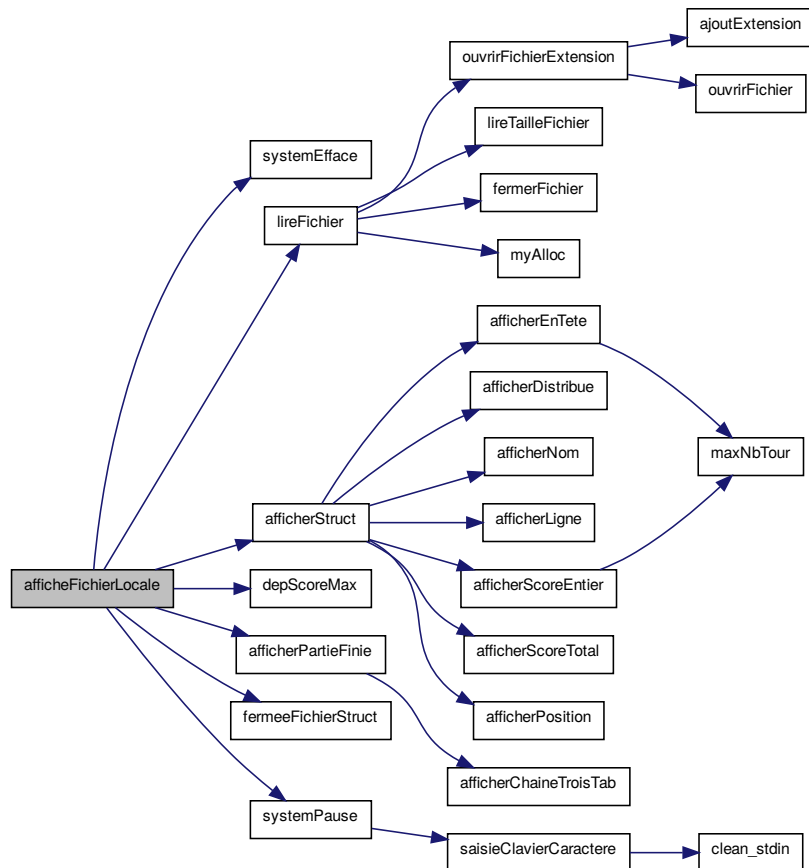
Here is the call graph for this function:



4.13.2.2 void afficheFichierLocale (char * nom_fichier)

Affiche le fichier dont le nom a ete donne en parametre

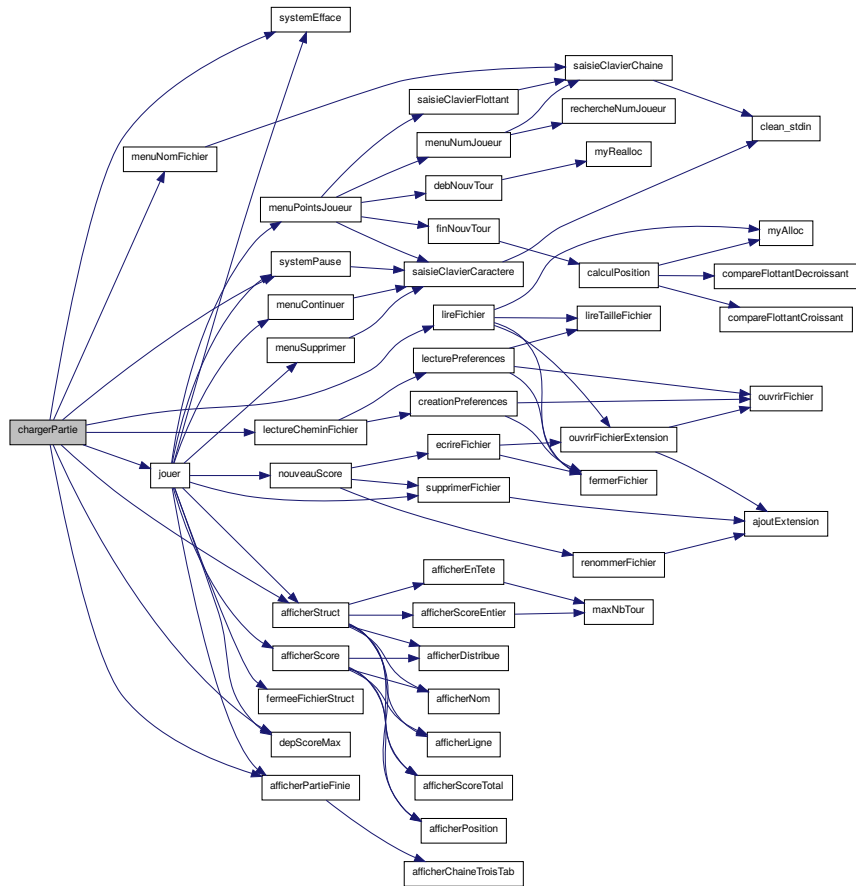
Here is the call graph for this function:



4.13.2.3 void chargerPartie ()

Charge une partie a partir d'un fichier et lance la fonction de comptage des points

Here is the call graph for this function:



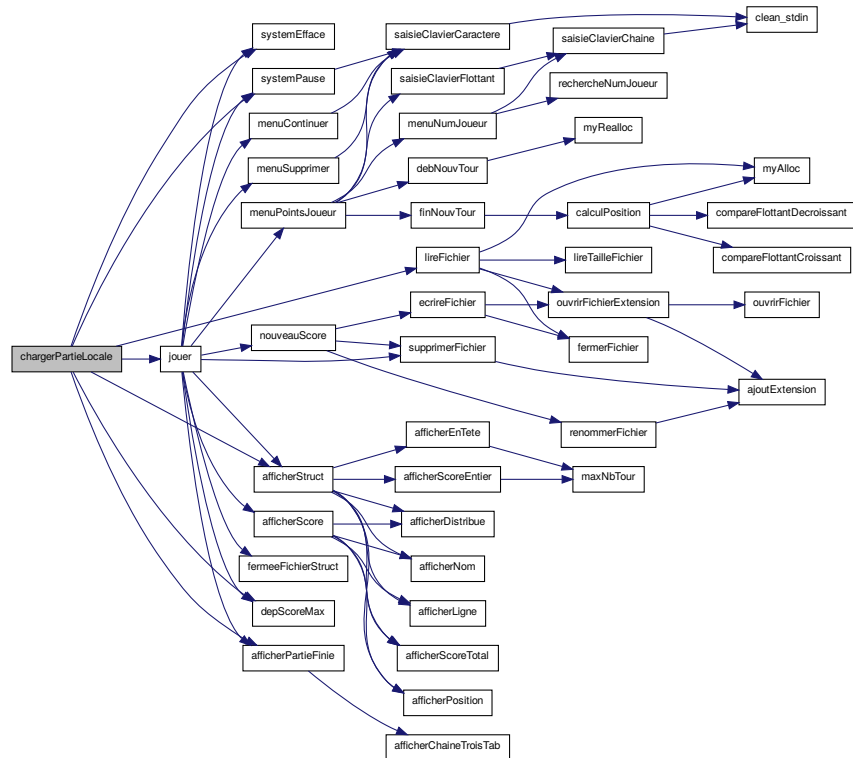
4.13.2.4 void chargerPartieLocale (char * nom_fichier)

Charge une partie a partir d'un fichier dont le nom a ete donne et lance la fonction de comptage des points

Parameters

in	nom_fichier,le	nom du fichier
----	----------------	----------------

Here is the call graph for this function:



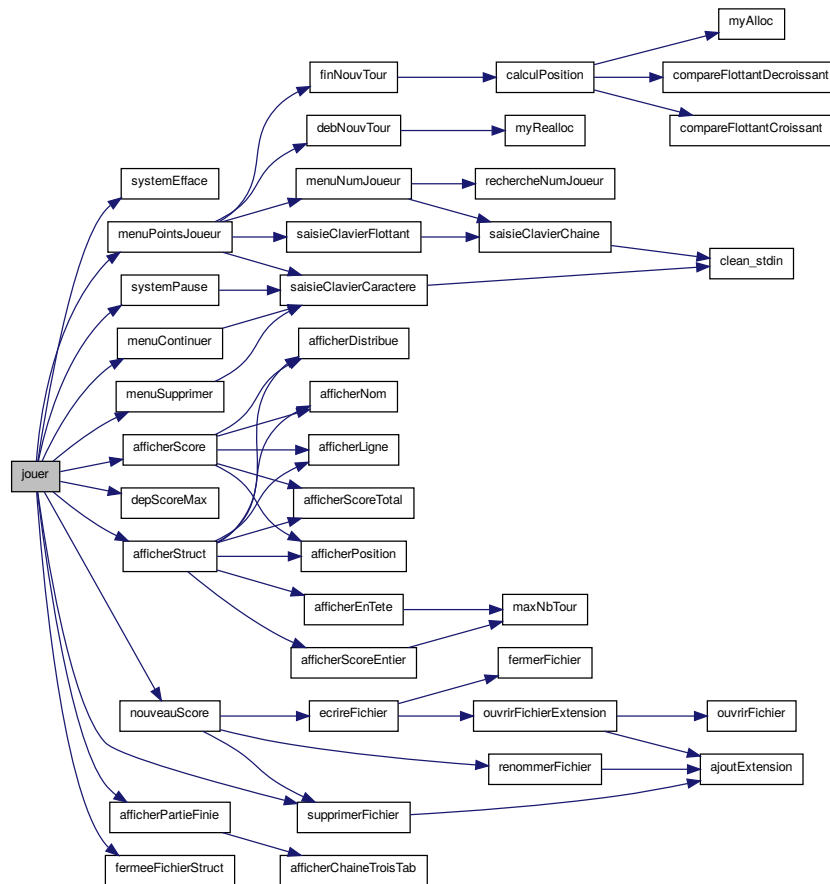
4.13.2.5 void jouer (Fichier_Jeu * ptr_struct_fichier, char * nom_fichier)

Lance la partie de comptage jusqu'a la fin

Parameters

in, out	*ptr_struct_fichier	la structure Fichier_Jeu
in	*nom_fichier	le nom du fichier dans lequel on stocke les informations

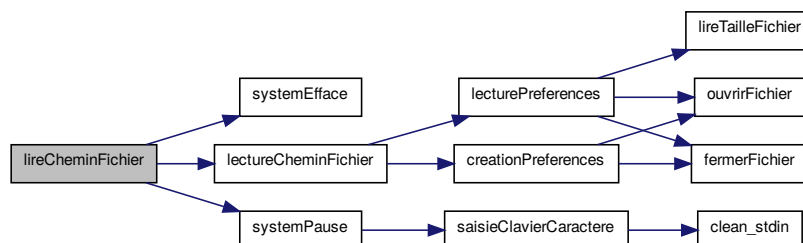
Here is the call graph for this function:



4.13.2.6 void lireCheminFichier ()

Lis le chemin de fichier et l'affiche

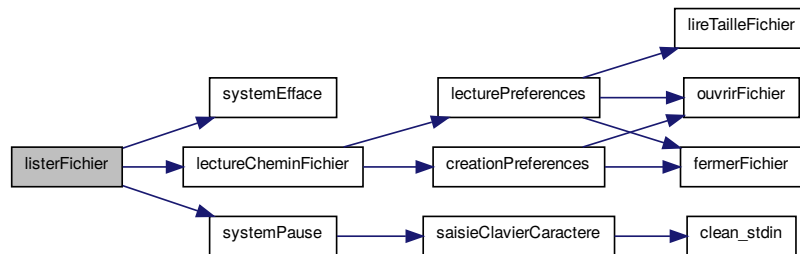
Here is the call graph for this function:



4.13.2.7 void listerFichier ()

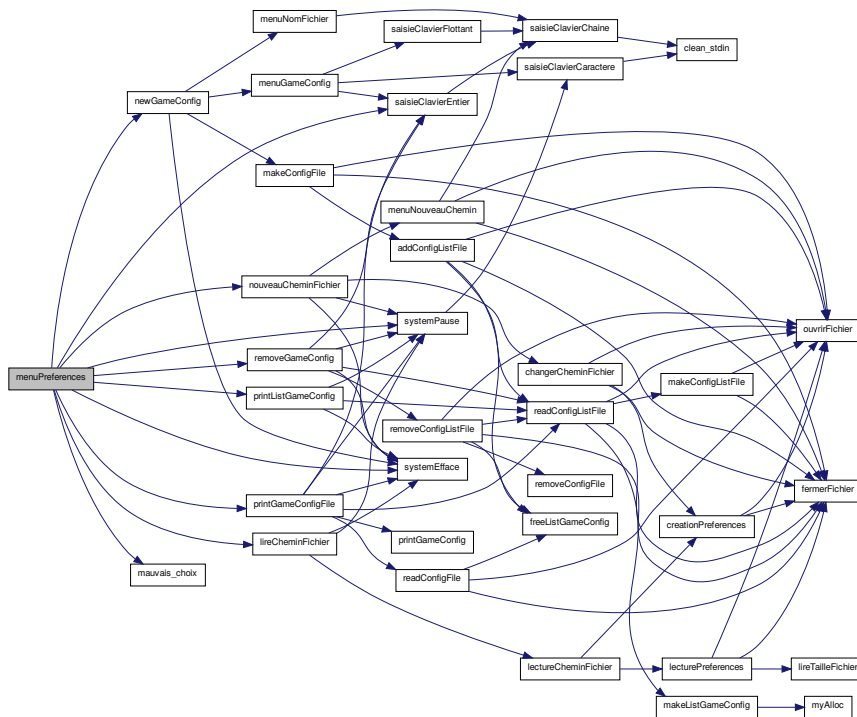
Liste tout les fichier du dossier courant contenant l'extension .jeu

Here is the call graph for this function:



4.13.2.8 void menuPreferences ()

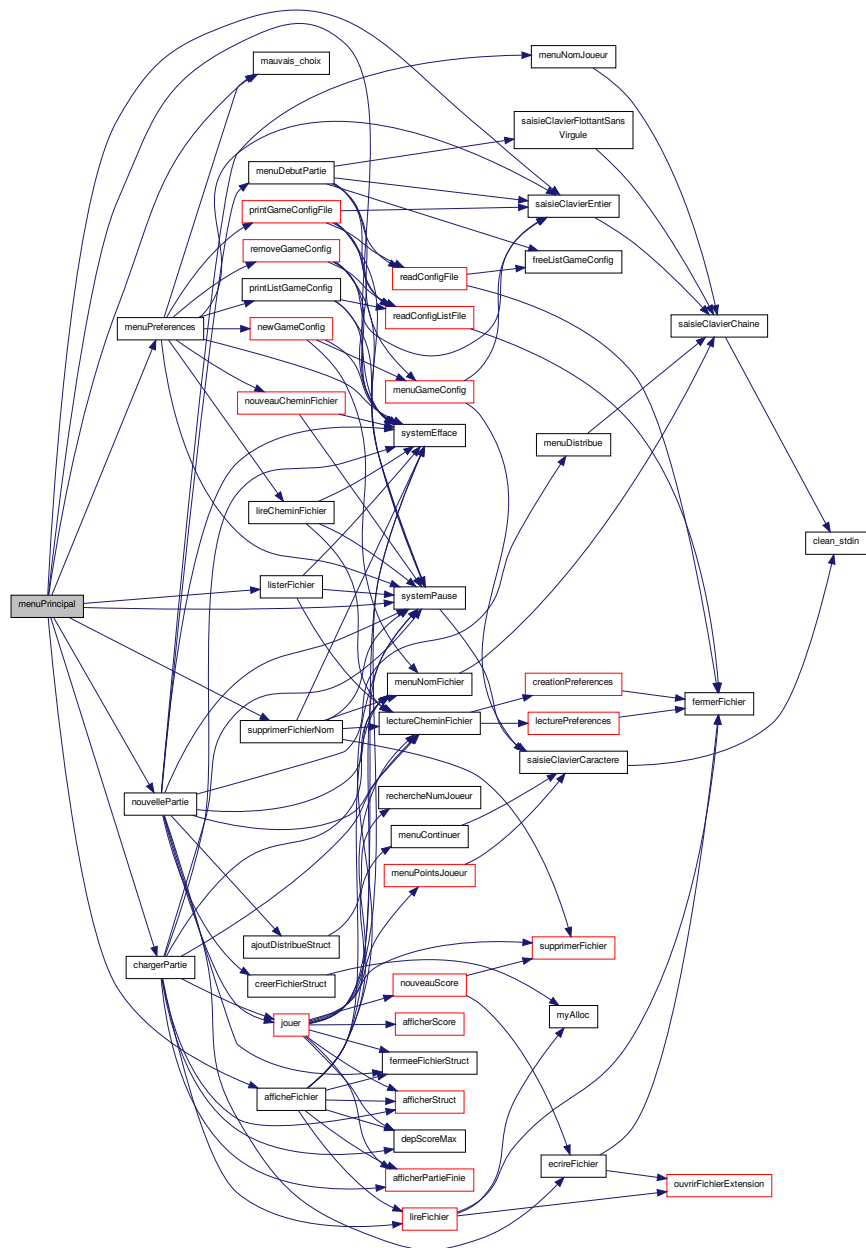
Here is the call graph for this function:



4.13.2.9 void menuPrincipal ()

Lance un menu que redirige vers l'action que l'on veut effectuer

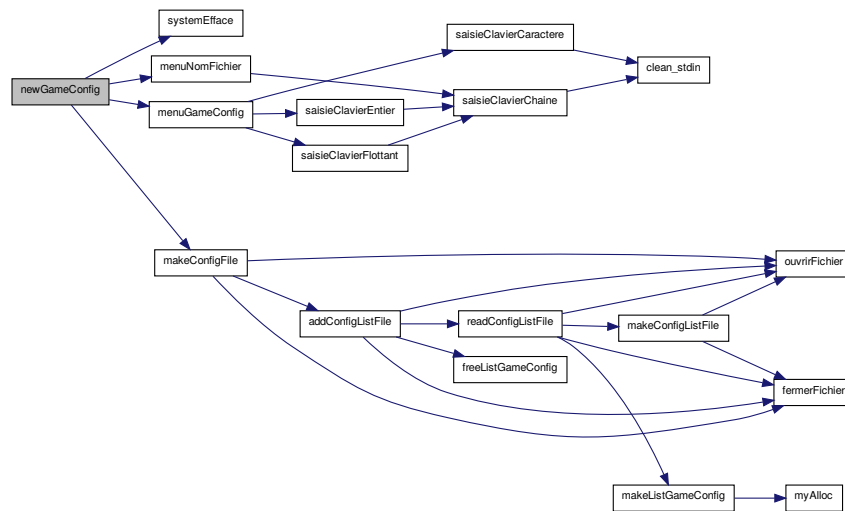
Here is the call graph for this function:



4.13.2.10 void newGameConfig ()

Add a new game configuration

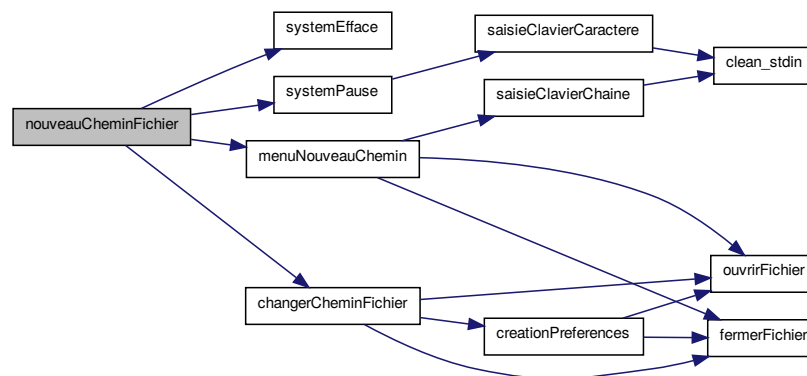
Here is the call graph for this function:



4.13.2.11 void nouveauCheminFichier ()

Charge un nouveau chemin que l'on demande a l'utilisateur

Here is the call graph for this function:

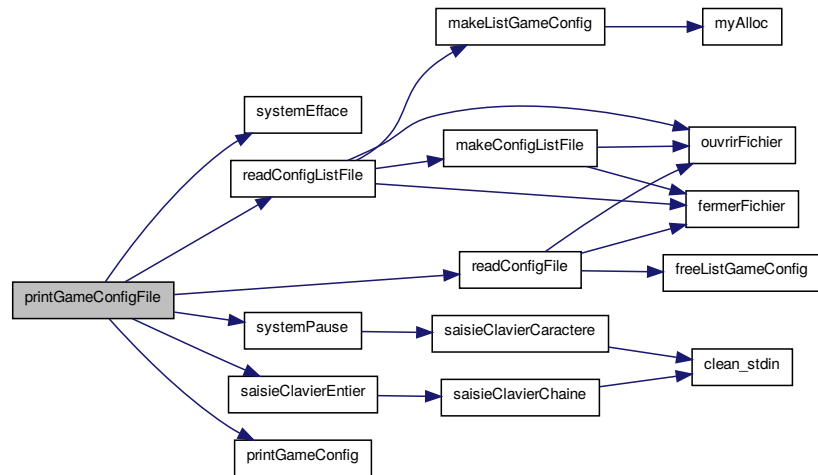


4.13.2.12 void nouvellePartie ()

Initialise une nouvelle partie et lance la fonction de comptage des points

4.13.2.13 void printGameConfigFile ()

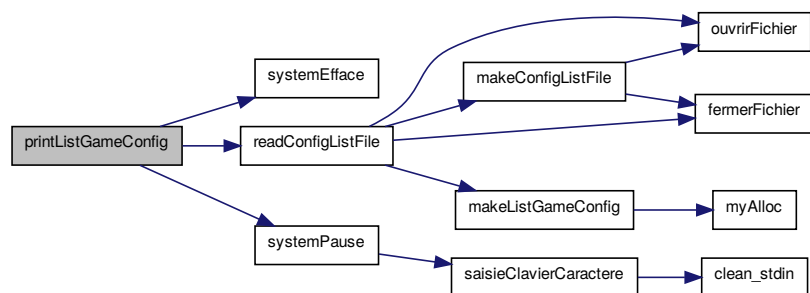
Here is the call graph for this function:



4.13.2.14 void printListGameConfig ()

Print the list of game configuration

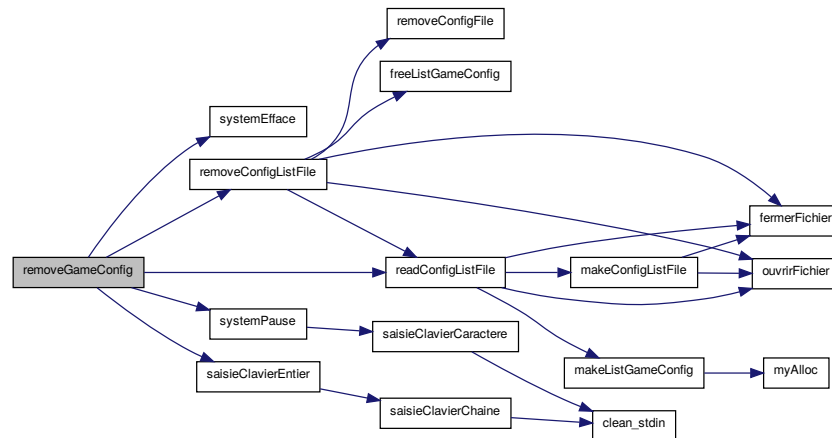
Here is the call graph for this function:



4.13.2.15 void removeGameConfig ()

Ask and remove a game configuration

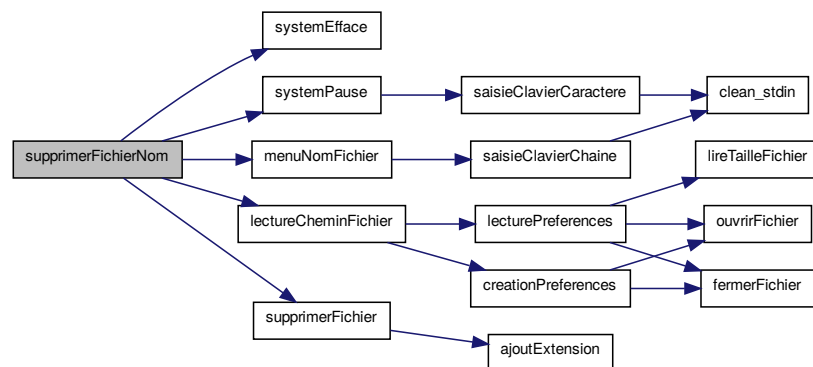
Here is the call graph for this function:



4.13.2.16 void supprimerFichierNom ()

Demande le nom d'un fichier et le supprime

Here is the call graph for this function:



4.14 interface.h File Reference

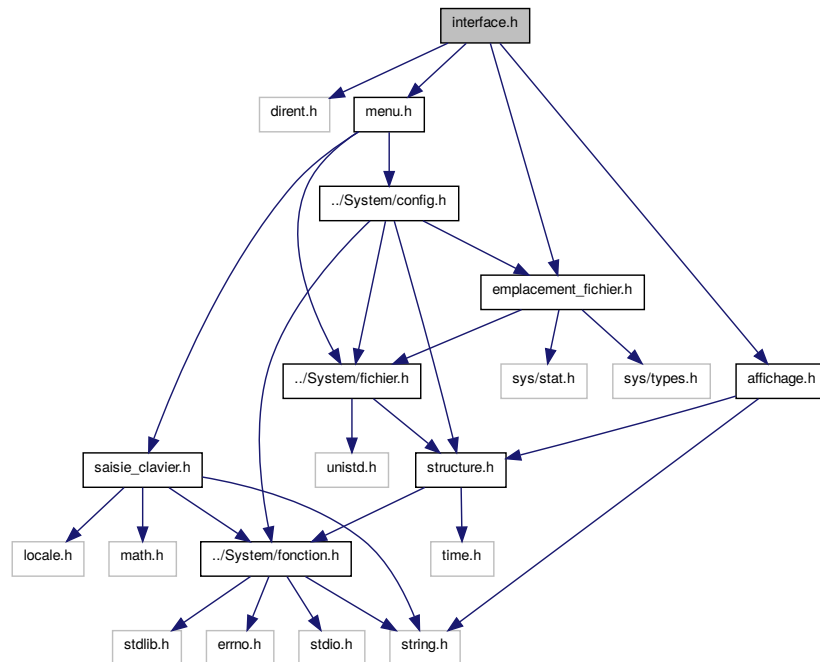
Prototypes des fonctions qui gerent l'interface graphique du logiciel.

```

#include <dirent.h>
#include "menu.h"
#include "affichage.h"
#include "../System/emplacement_fichier.h"

```


Include dependency graph for interface.h:



Enumerations

- enum `MenuPrincipal` {
`nouvPart =1`, `charPart =2`, `affFich =3`, `supprFich =4`,
`listFich =5`, `pref =6`, `quit =7`, `easterEggs = 42` }
- enum `MenuPreferences` {
`nouvChem =1`, `lireChem =2`, `newGameConf =3`, `removeGameConf =4`,
`printListGameConf =5`, `printGameConf =6`, `menuPrinc =7`, `easterEggs2 = 42` }

Functions

- void `afficheFichier` ()
- void `supprimerFichierNom` ()
- void `listerFichier` ()
- void `jouer` (`Fichier_Jeu *ptr_struct_fichier`, `char *nom_fichier`)
- void `nouvellePartie` ()
- void `chargerPartie` ()
- void `menuPrincipal` ()
- void `menuPreferences` ()
- void `nouveauCheminFichier` ()
- void `lireCheminFichier` ()
- void `chargerPartieLocale` (`char *nom_fichier`)
- void `afficheFichierLocale` (`char *nom_fichier`)
- void `newGameConfig` ()
- void `removeGameConfig` ()
- void `printListGameConfig` ()
- void `printGameConfigFile` ()

4.14.1 Detailed Description

Prototypes des fonctions qui gerent l'interface graphique du logiciel.

Author

Remi BERTHO

Date

09/03/14

Version

2.1.0

4.14.2 Enumeration Type Documentation

4.14.2.1 enum MenuPreferences

Enumerator

nouvChem

lireChem

newGameConf

removeGameConf

printListGameConf

printGameConf

menuPrinc

easterEggs2

4.14.2.2 enum MenuPrincipal

Enumerator

nouvPart

charPart

affFich

supprFich

listFich

pref

quit

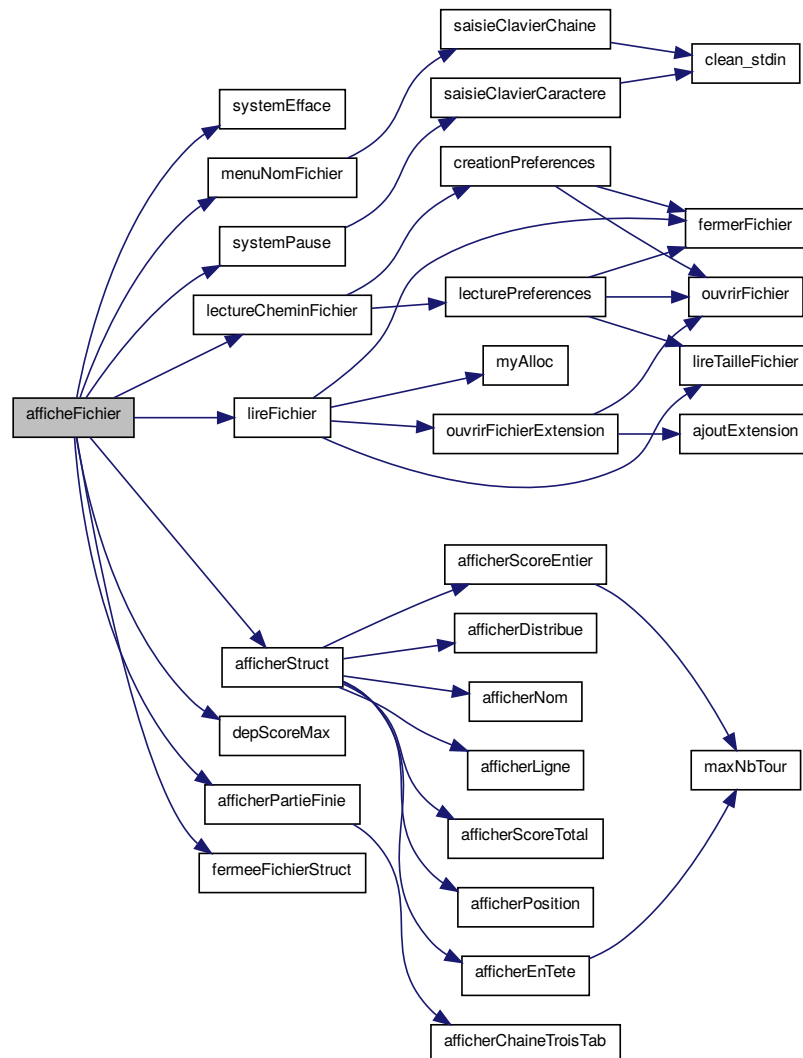
easterEggs

4.14.3 Function Documentation

4.14.3.1 void afficheFichier ()

Demande le nom d'un fichier et l'affiche

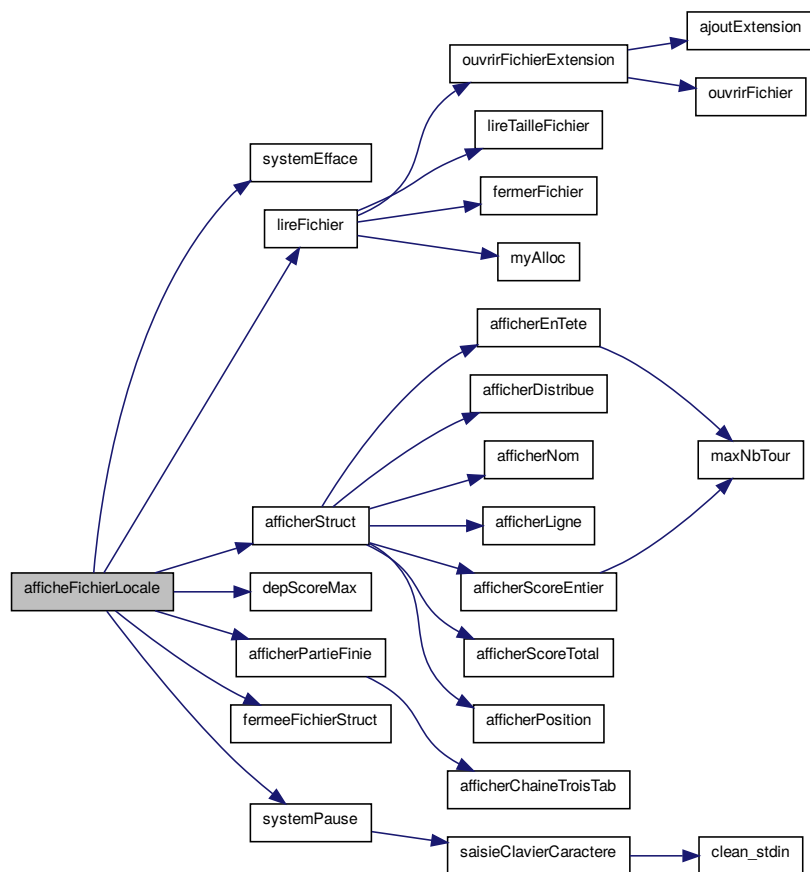
Here is the call graph for this function:



4.14.3.2 void afficheFichierLocale (char * nom_fichier)

Affiche le fichier dont le nom a ete donne en parametre

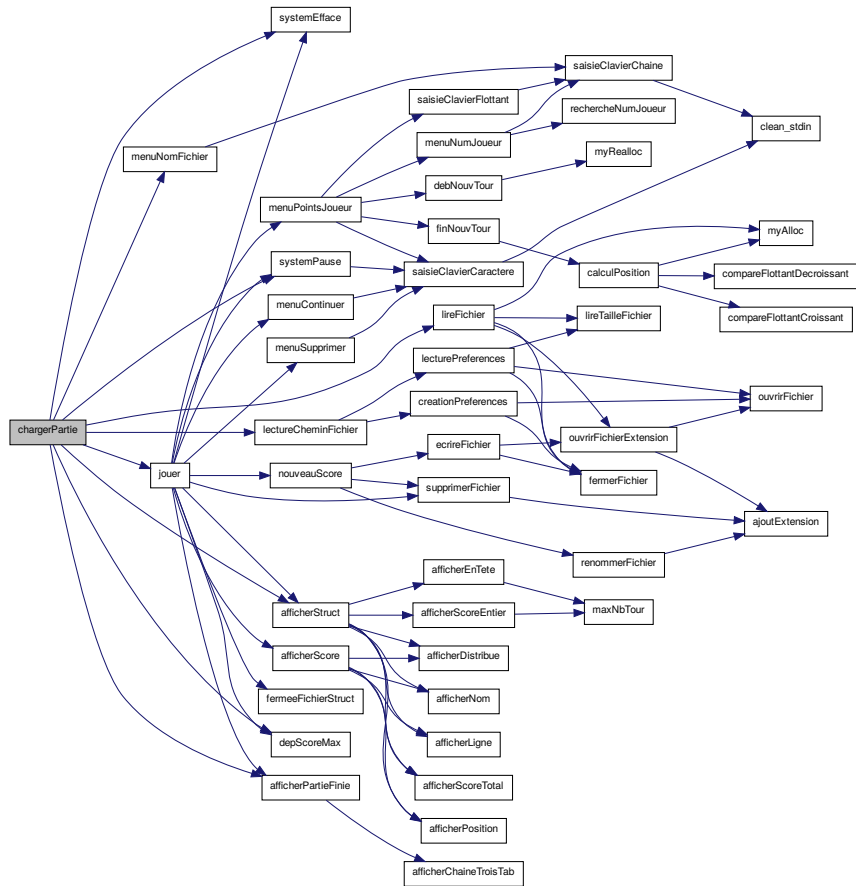
Here is the call graph for this function:



4.14.3.3 void chargerPartie ()

Charge une partie a partir d'un fichier et lance la fonction de comptage des points

Here is the call graph for this function:



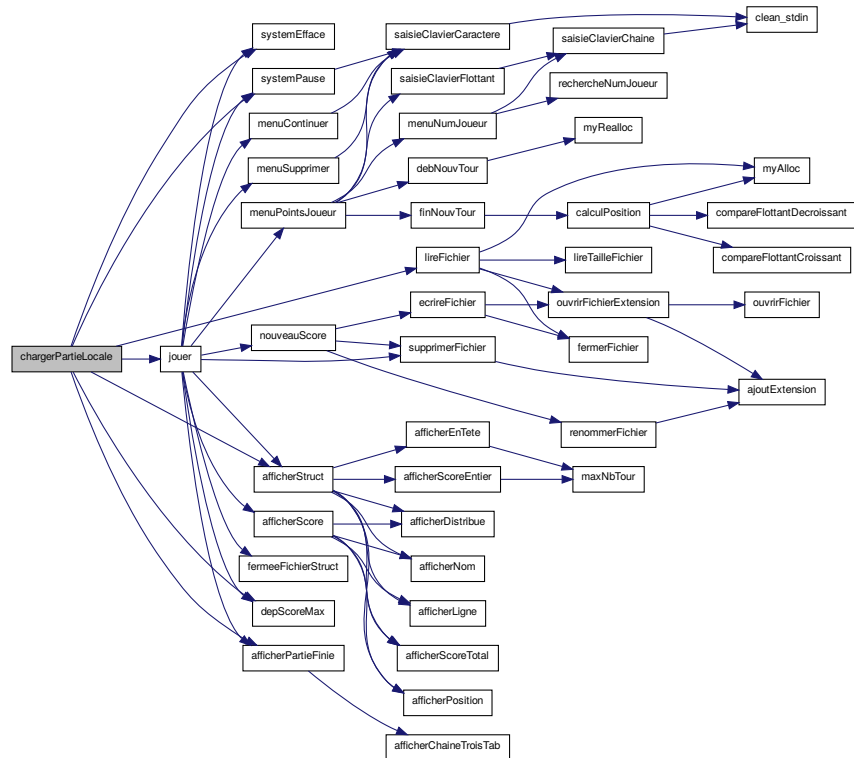
4.14.3.4 void chargerPartieLocale (char * nom_fichier)

Charge une partie a partir d'un fichier dont le nom a ete donne et lance la fonction de comptage des points

Parameters

in	nom_fichier,le	nom du fichier
----	----------------	----------------

Here is the call graph for this function:



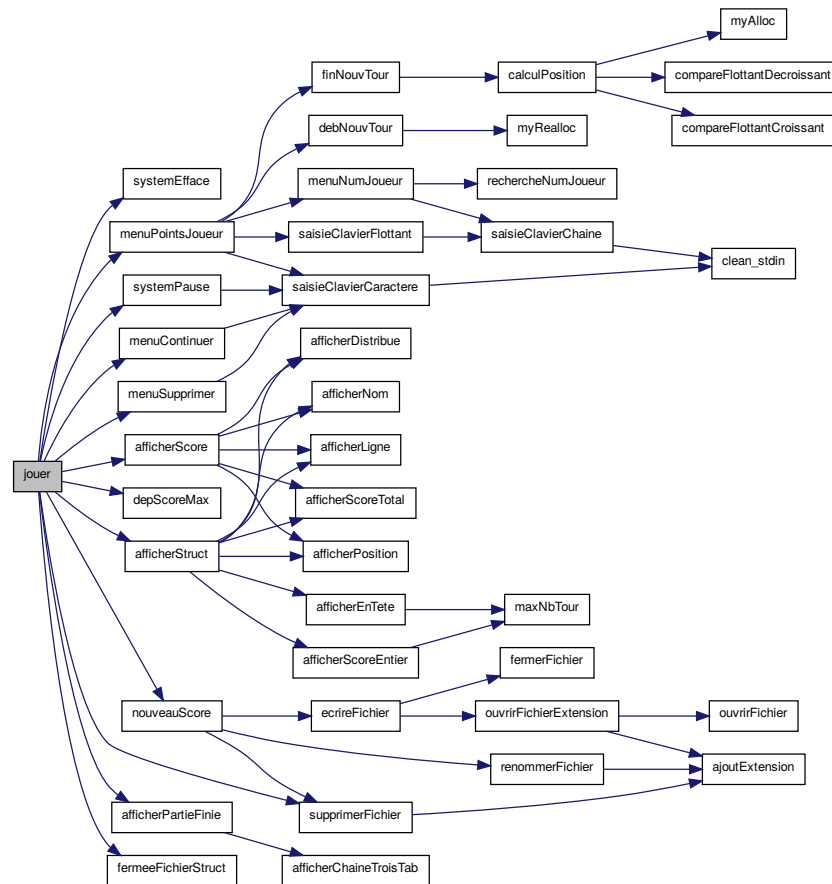
4.14.3.5 void jouer (Fichier_Jeu * ptr_struct_fichier, char * nom_fichier)

Lance la partie de comptage jusqu'a la fin

Parameters

in, out	*ptr_struct_fichier	la structure Fichier_Jeu
in	*nom_fichier	le nom du fichier dans lequel on stocke les informations

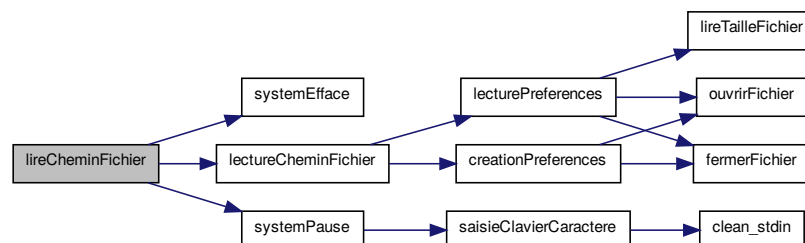
Here is the call graph for this function:



4.14.3.6 void lireCheminFichier ()

Lis le chemin de fichier et l'affiche

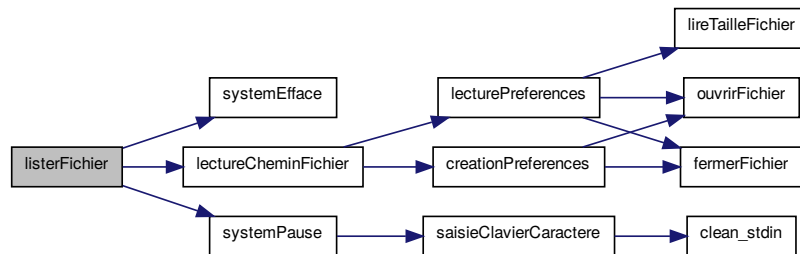
Here is the call graph for this function:



4.14.3.7 void listerFichier ()

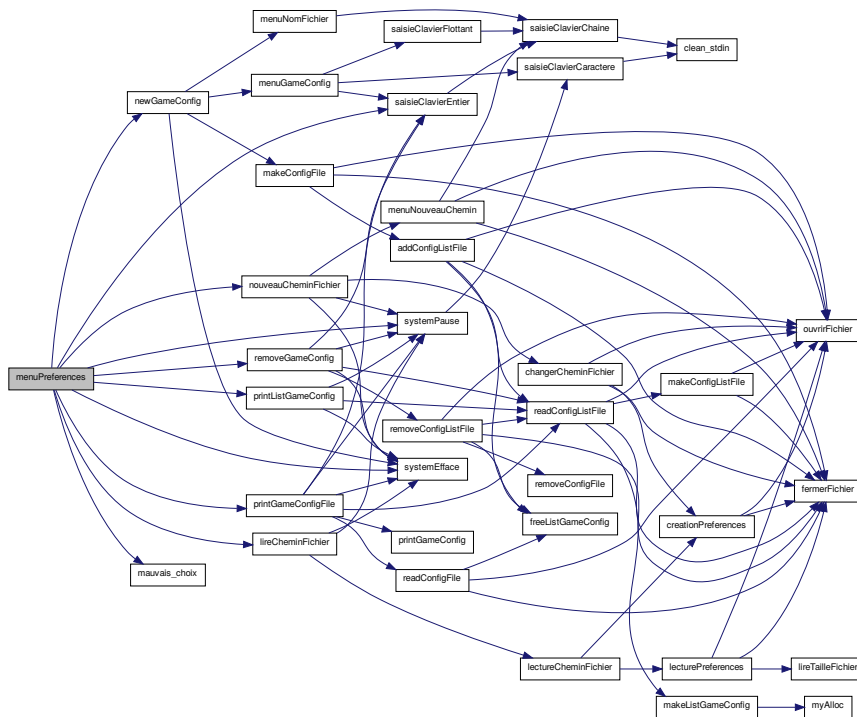
Liste tout les fichier du dossier courant contenant l'extension .jeu

Here is the call graph for this function:



4.14.3.8 void menuPreferences ()

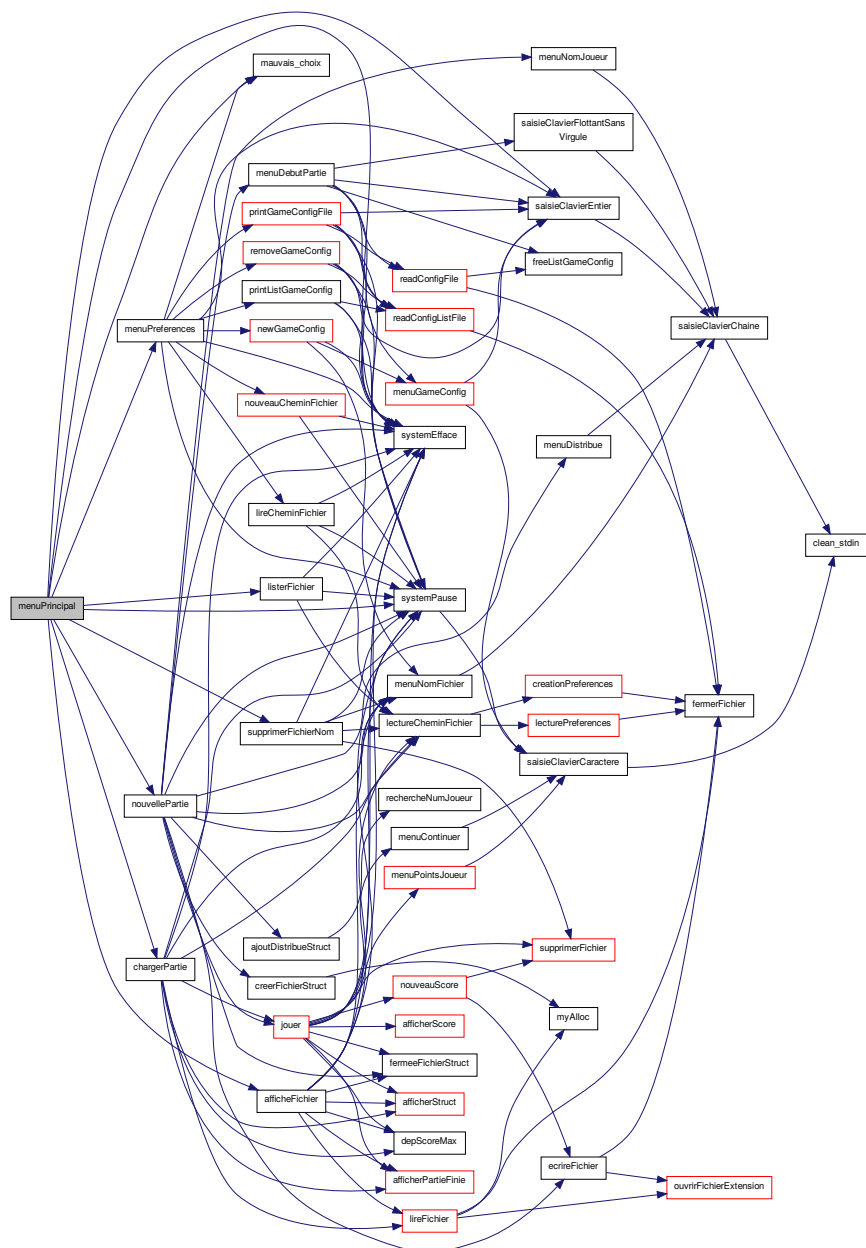
Here is the call graph for this function:



4.14.3.9 void menuPrincipal ()

Lance un menu que redirige vers l'action que l'on veut effectuer

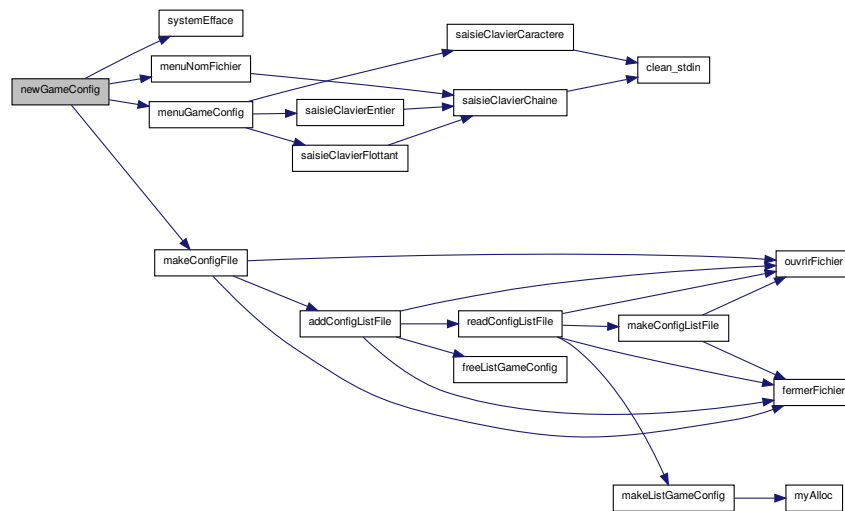
Here is the call graph for this function:



4.14.3.10 void newGameConfig ()

Add a new game configuration

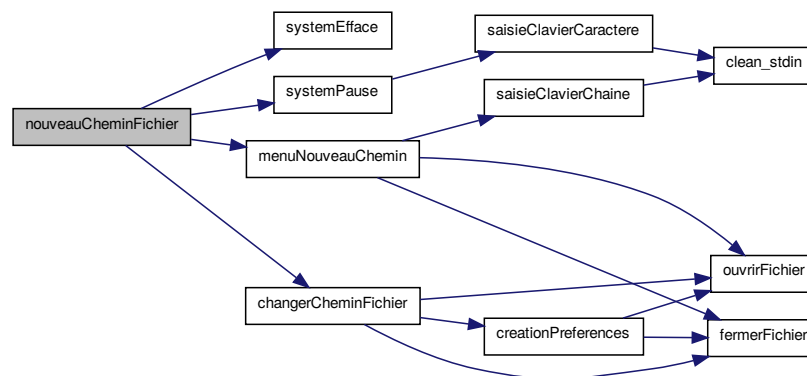
Here is the call graph for this function:



4.14.3.11 void nouveauCheminFichier ()

Charge un nouveau chemin que l'on demande a l'utilisateur

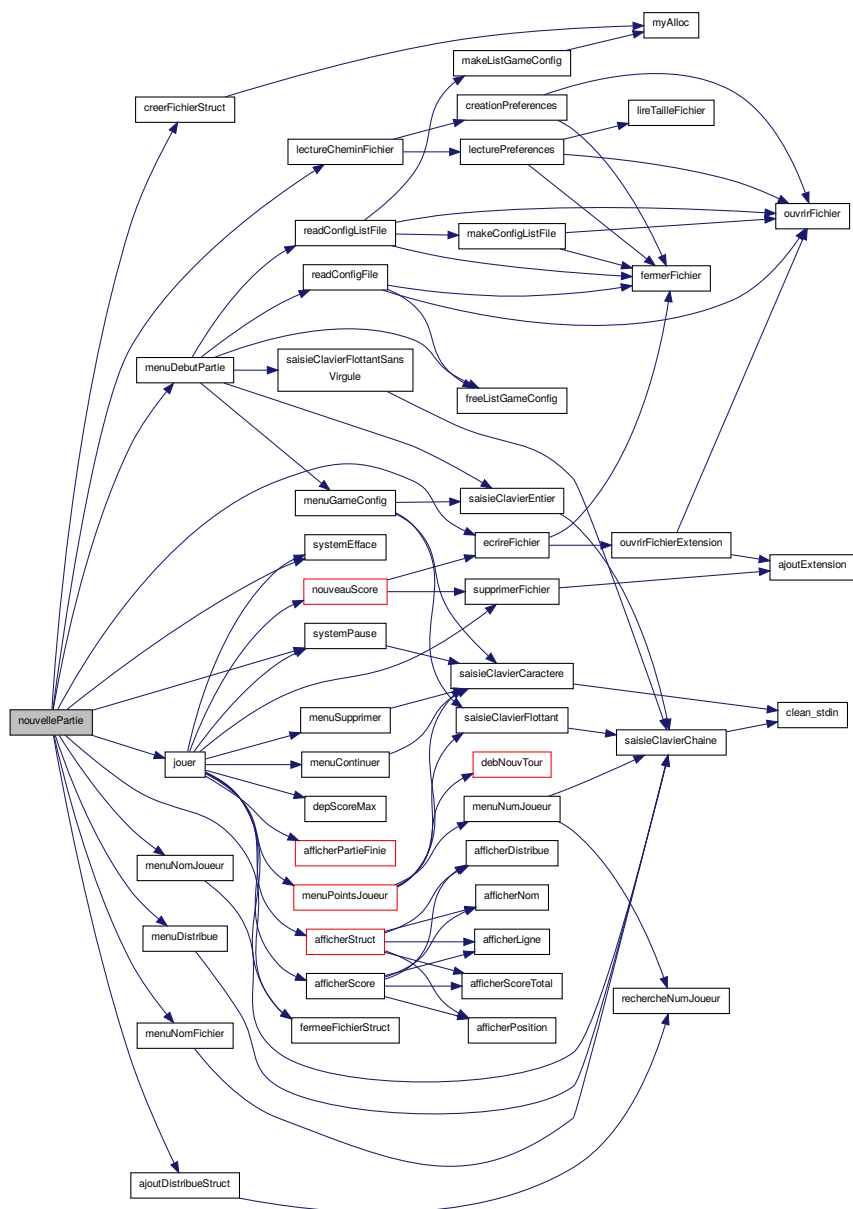
Here is the call graph for this function:



4.14.3.12 void nouvellePartie ()

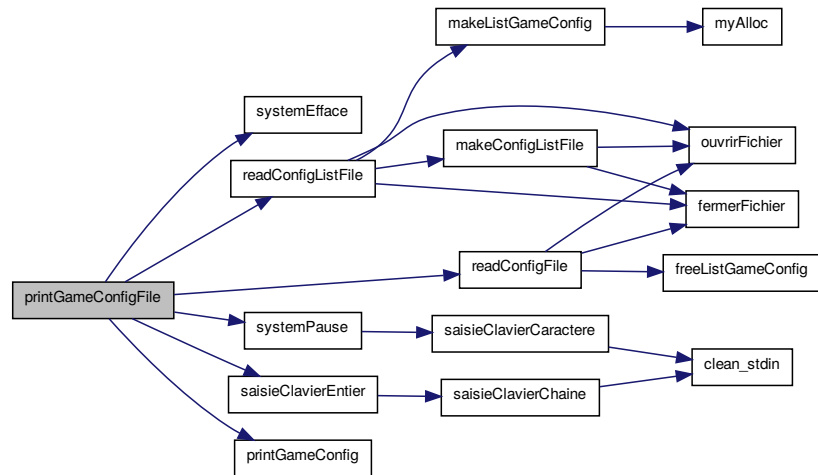
Initialise une nouvelle partie et lance la fonction de comptage des points

Here is the call graph for this function:



4.14.3.13 void printGameConfigFile ()

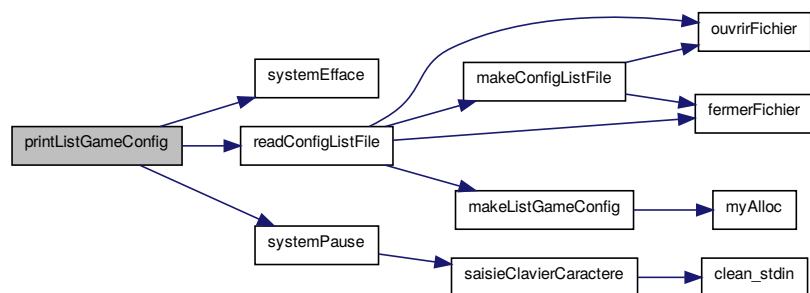
Here is the call graph for this function:



4.14.3.14 void printListGameConfig ()

Print the list of game configuration

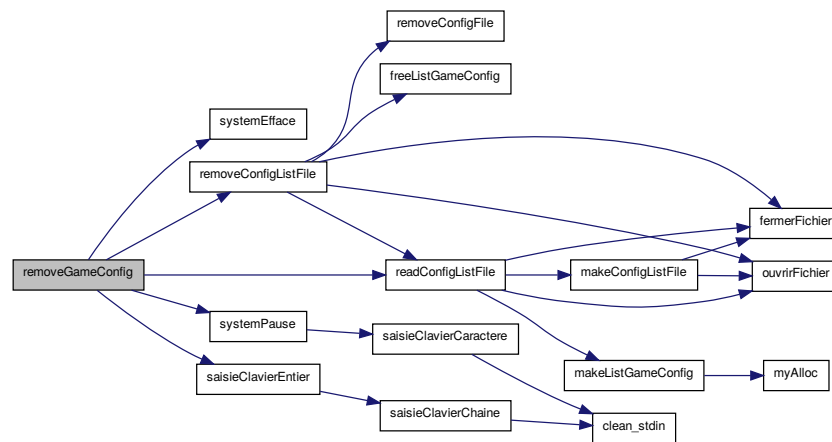
Here is the call graph for this function:



4.14.3.15 void removeGameConfig ()

Ask and remove a game configuration

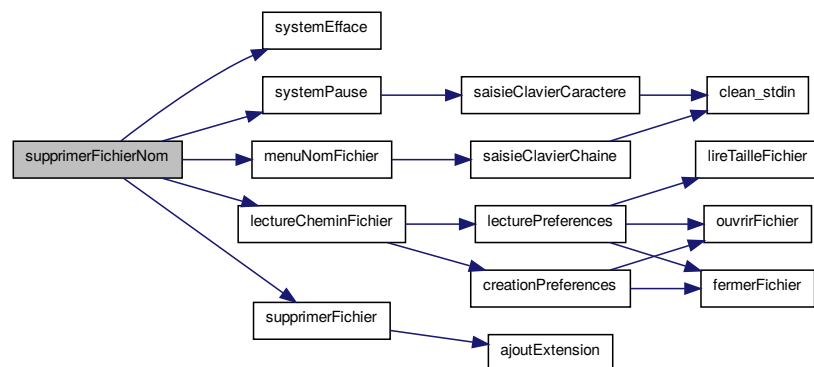
Here is the call graph for this function:



4.14.3.16 void supprimerFichierNom ()

Demande le nom d'un fichier et le supprime

Here is the call graph for this function:

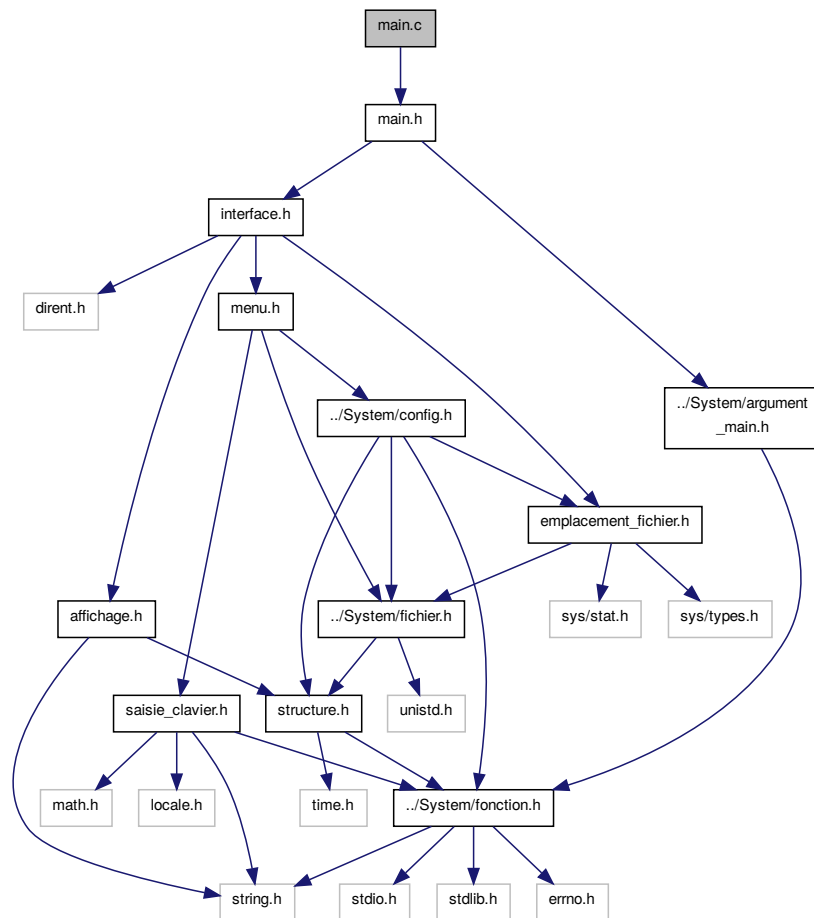


4.15 main.c File Reference

Lancement du programme.

```
#include "main.h"
```

Include dependency graph for main.c:



Functions

- int [main](#) (int argc, char *argv[])

4.15.1 Detailed Description

Lancement du programme.

Author

Remi BERTHO

Date

09/03/14

Version

2.1.0

4.15.2 Function Documentation

4.15.2.1 int main (int *argc*, char * *argv*[])

Lance le programme

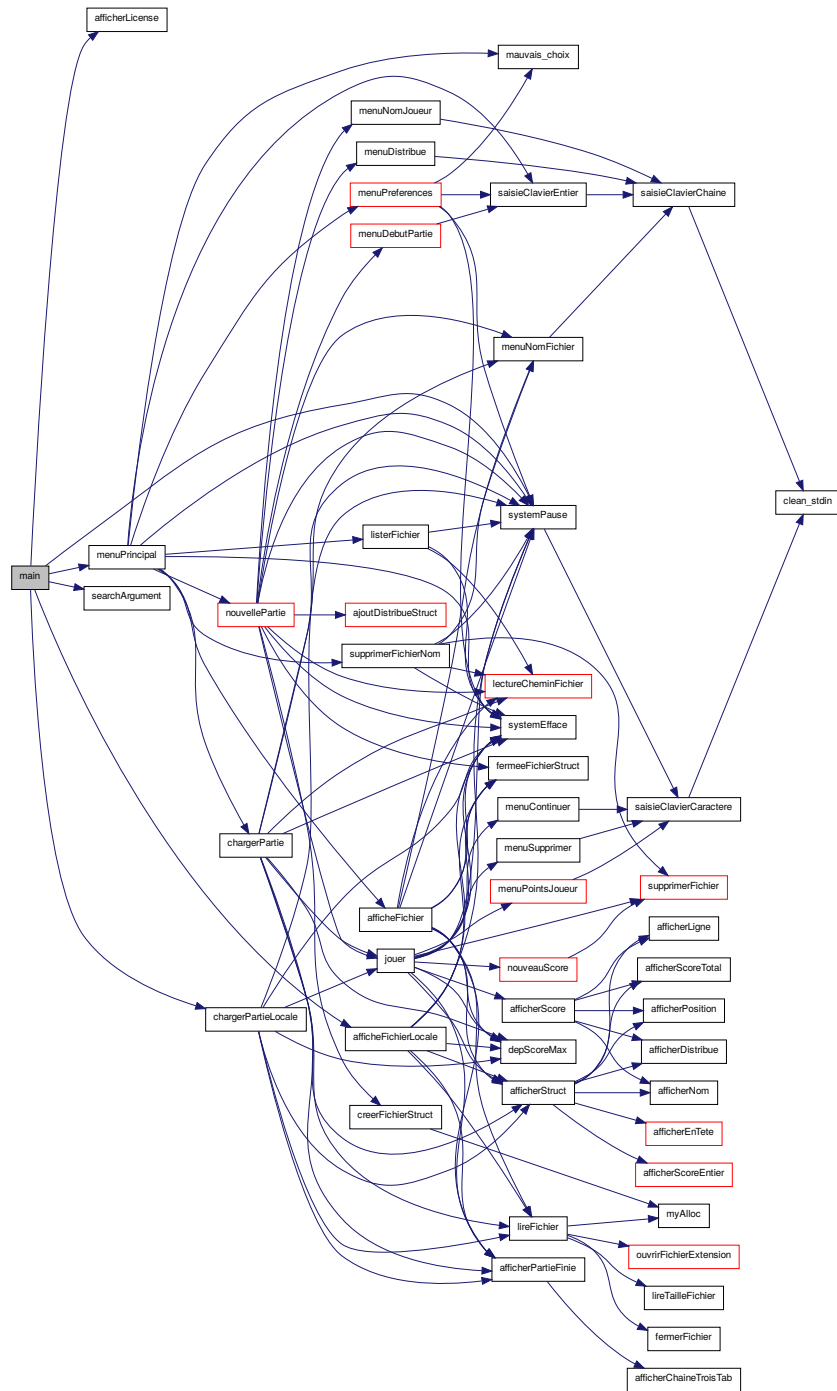
Parameters

in	<i>argc</i>	le nombre d 'argument
in	<i>argv</i>	le tableau des arguments

Returns

0 si tout s'est bien passe

Here is the call graph for this function:

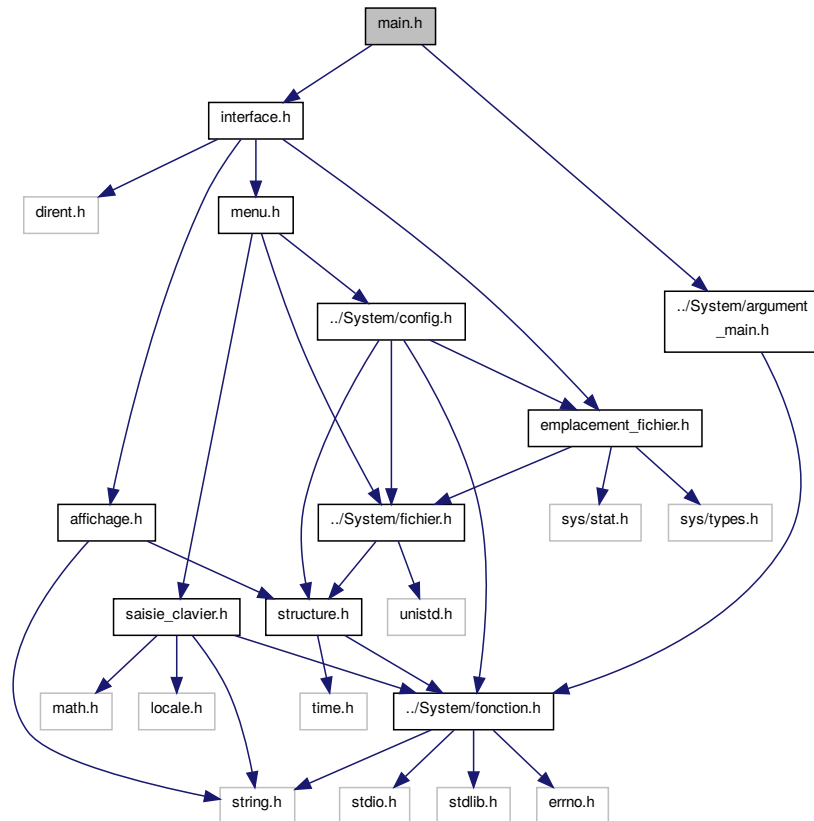


4.16 main.h File Reference

Lancement du programme.

```
#include "interface.h"
#include "../System/argument_main.h"
```


Include dependency graph for main.h:



Functions

- int [main](#) (int argc, char *argv[])

4.16.1 Detailed Description

Lancement du programme.

Author

Remi BERTHO

Date

09/03/14

Version

2.1.0

4.16.2 Function Documentation

4.16.2.1 `int main (int argc, char * argv[])`

Lance le programme

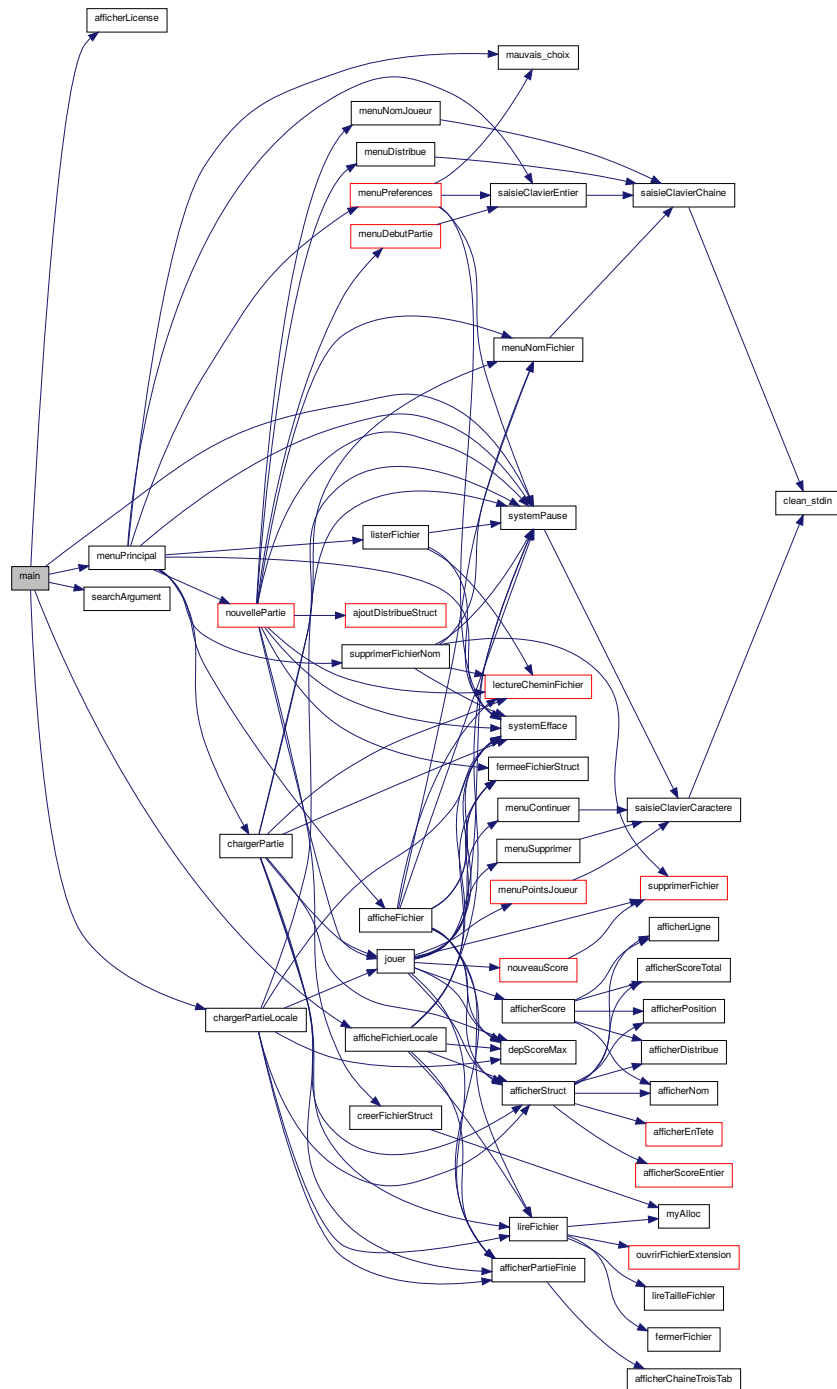
Parameters

in	<i>argc</i>	le nombre d 'argument
in	<i>argv</i>	le tableau des arguments

Returns

0 si tout s'est bien passe

Here is the call graph for this function:

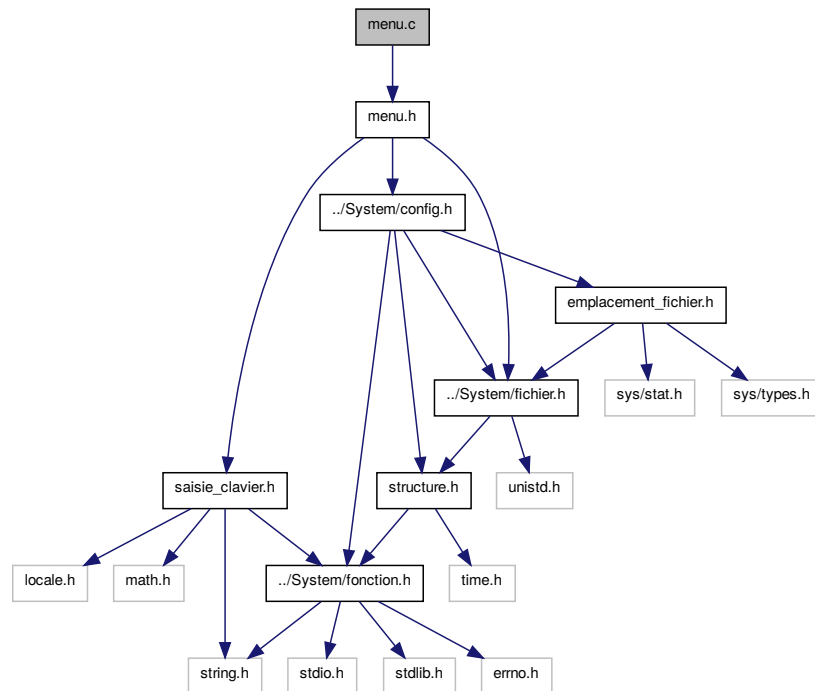


4.17 menu.c File Reference

Fonctions qui gerent les menus demandant de rentrer des valeurs du logiciel.

```
#include "menu.h"
```

Include dependency graph for menu.c:



Functions

- char * [menuNomFichier](#) (char nom_fichier[TAILLE_MAX_NOM_FICHER])
- void [menuDebutPartie](#) (float *ptr_nb_joueur, [game_config](#) *ptr_config)
- void [menuGameConfig](#) ([game_config](#) *ptr_config)
- void [menuDistribue](#) (char *nom_distribue)
- void [menuNomJoueur](#) ([Fichier_Jeu](#) *ptr_struct_fichier)
- void [menuPointsJoueur](#) ([Fichier_Jeu](#) *ptr_struct_fichier)
- int [menuNumJoueur](#) ([Fichier_Jeu](#) *ptr_struct_fichier)
- int [menuContinuer](#) ()
- int [menuSupprimer](#) ()
- void [menuNouveauChemin](#) (char *nouveauChemin)

4.17.1 Detailed Description

Fonctions qui gerent les menus demandant de rentrer des valeurs du logiciel.

Author

Remi BERTHO

Date

09/03/14

Version

2.1.0

4.17.2 Function Documentation

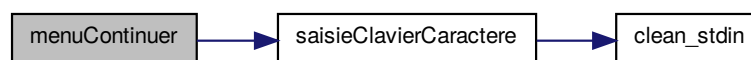
4.17.2.1 int menuContinuer ()

Demande si l'on veut continuer ou pas et l'enregistre dans la variable `arret`

Returns

VRAI si l'on veut continuer, FAUX sinon

Here is the call graph for this function:



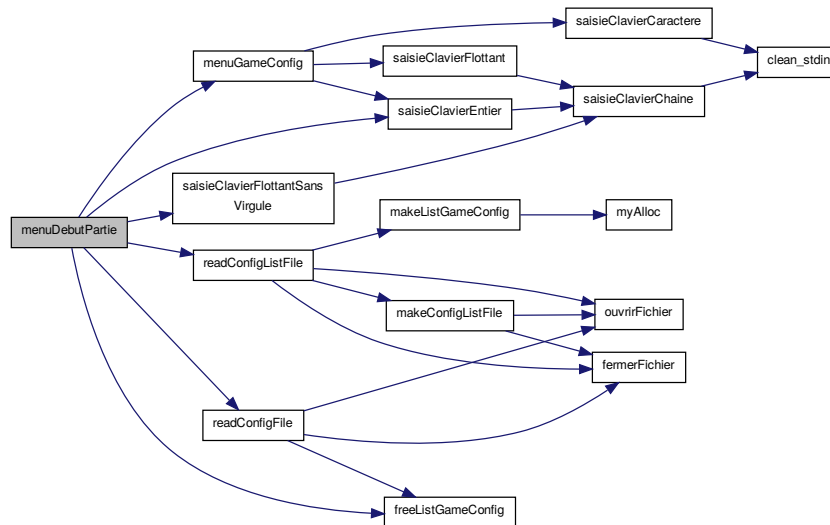
4.17.2.2 void menuDebutPartie (float * *ptr_nb_joueur*, game_config * *ptr_config*)

Demande et enregistre le nombre de joueur, le nombre maximum et le nom de la personne qui comme a distribuer

Parameters

<i>in, out</i>	<i>ptr_nb_joueur</i>	le nombre de joueur
<i>in, out</i>	<i>ptr_config</i>	la configuration de la partie

Here is the call graph for this function:



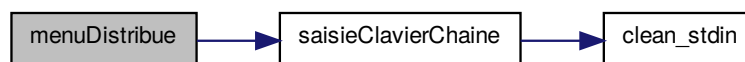
4.17.2.3 void menuDistribue (char * *nom_distribue*)

Demande et enregistre le nom de la personne qui commence a distribuer

Parameters

<i>in, out</i>	<i>*nom_distribue</i>	le nom de la personne qui commence a distribuer
----------------	-----------------------	---

Here is the call graph for this function:



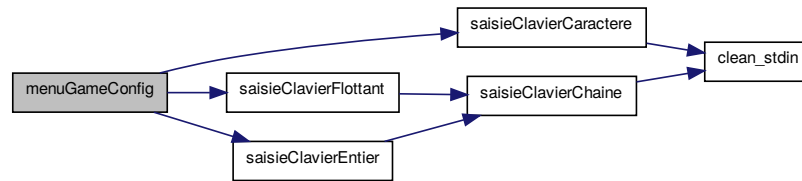
4.17.2.4 void menuGameConfig (game_config * *ptr_config*)

Demande et enregistre la configuration de jeu

Parameters

<i>in, out</i>	<i>ptr_config</i>	la configuration de la partie
----------------	-------------------	-------------------------------

Here is the call graph for this function:



4.17.2.5 char * menuNomFichier (char nom_fichier[TAILLE_MAX_NOM_FICHIER])

Demande et enregistre le nom du fichier

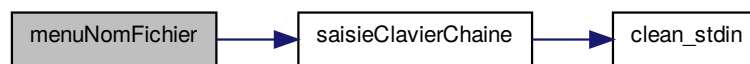
Parameters

in, out	nom_fichier	la chaine de caractere contenant le nom du fichier
---------	-------------	--

Returns

la chaine de caractere contenant le nom du fichier

Here is the call graph for this function:



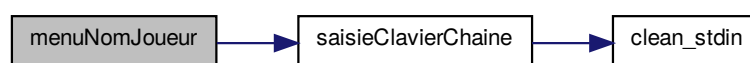
4.17.2.6 void menuNomJoueur (Fichier_Jeu * ptr_struct_fichier)

Demande et enregistre le nom des joueurs

Parameters

in, out	*ptr_struct_fichier	la structure Fichier_Jeu ou l'on veut enregistrer le nom des joueurs
---------	---------------------	--

Here is the call graph for this function:



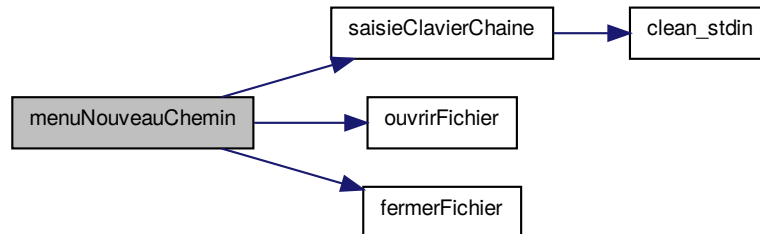
4.17.2.7 void menuNouveauChemin (char * *nouveauChemin*)

Demande et enregistre le nouveau chemin

Parameters

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

Here is the call graph for this function:

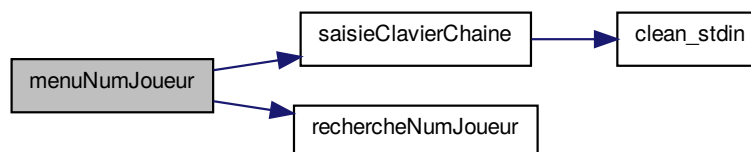
4.17.2.8 `int menuNumJoueur (Fichier_Jeu * ptr_struct_fichier)`

Demande et enregistre le numero du joueur

Parameters

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

Here is the call graph for this function:

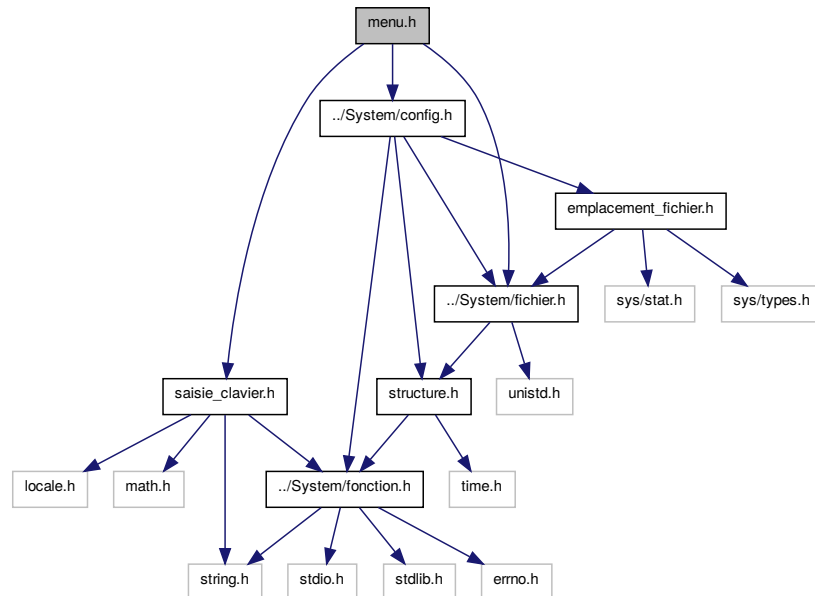
4.17.2.9 `void menuPointsJoueur (Fichier_Jeu * ptr_struct_fichier)`

Debut un nouveau tour, demande et enregistre les points et fini le tour

Parameters

in, out	<i>*ptr_struct_fichier</i>	la structure Fichier_Jeu ou l'on veut faire un nouveau tour
---------	----------------------------	---

Include dependency graph for menu.h:



Functions

- char * [menuNomFichier](#) (char nom_fichier[TAILLE_MAX_NOM_FICHIER])
- void [menuDebutPartie](#) (float *ptr_nb_joueur, [game_config](#) *ptr_config)
- void [menuGameConfig](#) ([game_config](#) *ptr_config)
- void [menuDistribue](#) (char *nom_distribue)
- void [menuNomJoueur](#) ([Fichier_Jeu](#) *ptr_struct_fichier)
- void [menuPointsJoueur](#) ([Fichier_Jeu](#) *ptr_struct_fichier)
- int [menuNumJoueur](#) ([Fichier_Jeu](#) *ptr_struct_fichier)
- int [menuContinuer](#) ()
- int [menuSupprimer](#) ()
- void [menuNouveauChemin](#) (char *nouveauChemin)

4.18.1 Detailed Description

Prototypes des fonctions qui gerent les menus demandant de rentrer des valeurs du logiciel.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.18.2 Function Documentation

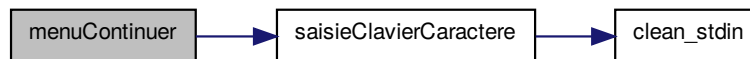
4.18.2.1 int menuContinuer ()

Demande si l'on veut continuer ou pas et l'enregistre dans la variable arret

Returns

VRAI si l'on veut continuer, FAUX sinon

Here is the call graph for this function:



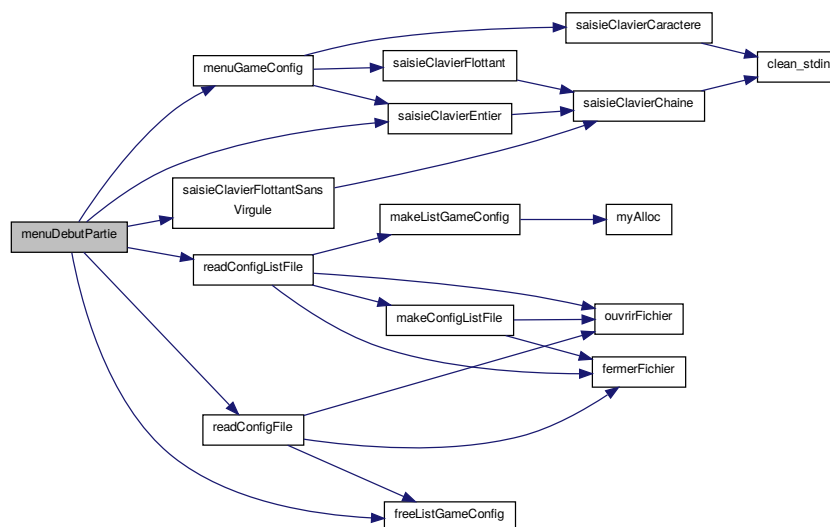
4.18.2.2 void menuDebutPartie (float * ptr_nb_joueur, game_config * ptr_config)

Demande et enregistre le nombre de joueur, le nombre maximum et le nom de la personne qui comme a distribuer

Parameters

in, out	<i>ptr_nb_joueur</i>	le nombre de joueur
in, out	<i>ptr_config</i>	la configuration de la partie

Here is the call graph for this function:



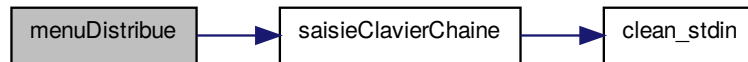
4.18.2.3 void menuDistribue (char * nom_distribue)

Demande et enregistre le nom de la personne qui comme a distribuer

Parameters

<i>in, out</i>	<i>*nom_distribue</i>	le nom de la personne qui commence a distribuer
----------------	-----------------------	---

Here is the call graph for this function:



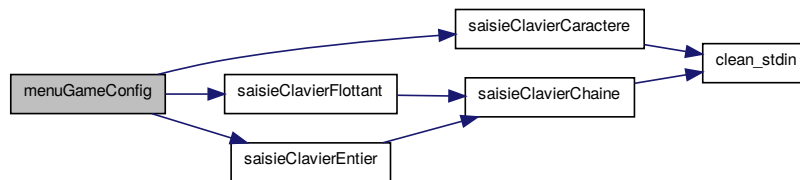
4.18.2.4 void menuGameConfig (game_config * ptr_config)

Demande et enregistre la configuration de jeu

Parameters

<i>in, out</i>	<i>ptr_config</i>	la configuration de la partie
----------------	-------------------	-------------------------------

Here is the call graph for this function:



4.18.2.5 char* menuNomFichier (char nom_fichier[TAILLE_MAX_NOM_FICHIER])

Demande et enregistre le nom du fichier

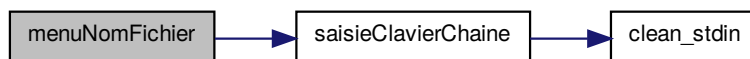
Parameters

<i>in, out</i>	<i>nom_fichier</i>	la chaine de caractere contenant le nom du fichier
----------------	--------------------	--

Returns

la chaine de caractere contenant le nom du fichier

Here is the call graph for this function:

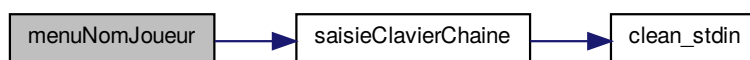
**4.18.2.6 void menuNomJoueur (Fichier_Jeu * ptr_struct_fichier)**

Demande et enregistre le nom des joueurs

Parameters

in, out	<i>*ptr_struct_fichier</i>	la structure Fichier_Jeu ou l'on veut enregistrer le nom des joueurs
---------	----------------------------	--

Here is the call graph for this function:

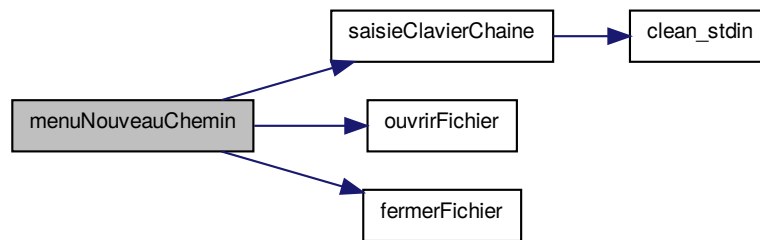
**4.18.2.7 void menuNouveauChemin (char * nouveauChemin)**

Demande et enregistre le nouveau chemin

Parameters

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

Here is the call graph for this function:



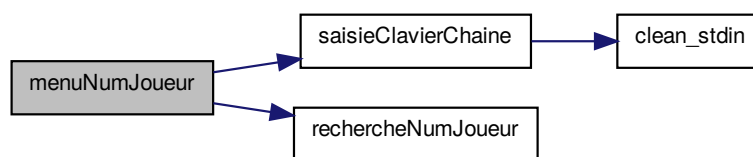
4.18.2.8 int menuNumJoueur (Fichier_Jeu * ptr_struct_fichier)

Demande et enregistre le numero du joueur

Parameters

in, out	*ptr_struct_fichier	la structure Fichier_Jeu
---------	---------------------	--

Here is the call graph for this function:



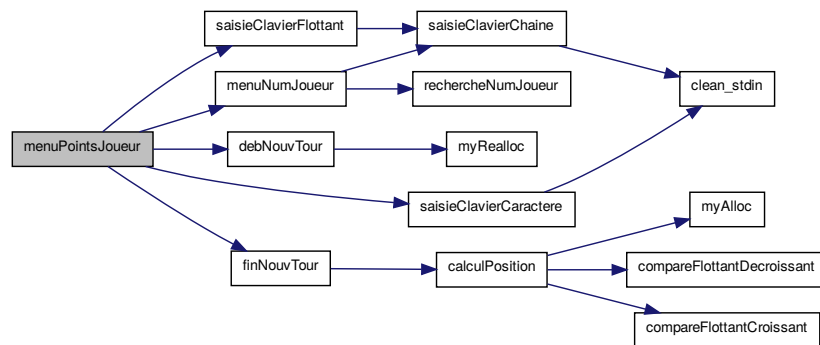
4.18.2.9 void menuPointsJoueur (Fichier_Jeu * ptr_struct_fichier)

Debut un nouveau tour, demande et enregistre les points et fini le tour

Parameters

in, out	*ptr_struct_fichier	la structure Fichier_Jeu ou l'on veut faire un nouveau tour
---------	---------------------	---

Here is the call graph for this function:



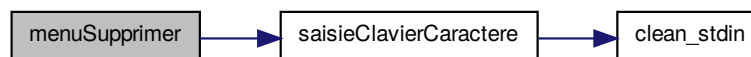
4.18.2.10 int menuSupprimer ()

Demande si l'on veut supprimer le fichier ou pas et l'enregistre dans la variable `suppr`

Returns

VRAI si l'on veut supprimer le fichier, FAUX sinon

Here is the call graph for this function:

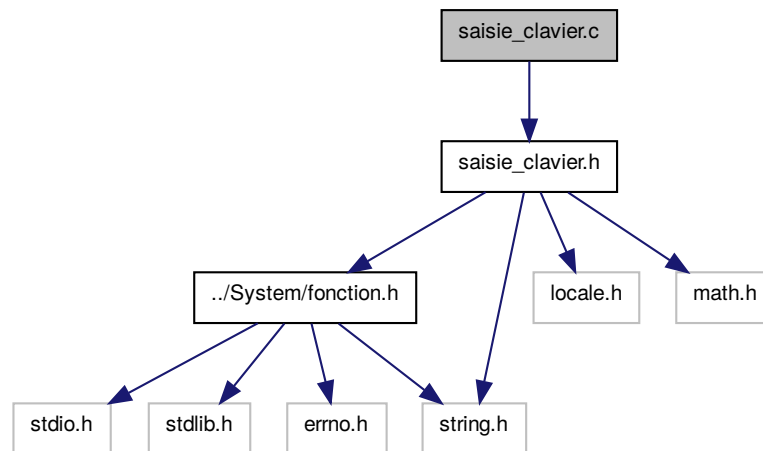


4.19 saisie_clavier.c File Reference

Fonctions de saisie clavier.


```
#include "saisie_clavier.h"
```

Include dependency graph for saisie_clavier.c:



Functions

- void `clean_stdin` (void)
- char * `saisieClavierChaine` (char *chaine, int nb_caract_plus_un)
- void `saisieClavierEntier` (int *nb)
- void `saisieClavierFlottant` (float *nb)
- void `saisieClavierFlottantSansVirgule` (float *nb)
- void `saisieClavierDouble` (double *nb)
- char * `saisieClavierCaractere` (char *c)
- void `systemPause` ()

4.19.1 Detailed Description

Fonctions de saisie clavier.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.19.2 Function Documentation

4.19.2.1 void `clean_stdin` (void)

Vide la cache de stdin.

4.19.2.2 char * saisieClavierCaractere (char * c)

Fait une saisie clavier d'un caractere.

Parameters

in, out	*c	le caractere que l'on veut saisir
---------	----	-----------------------------------

Here is the call graph for this function:



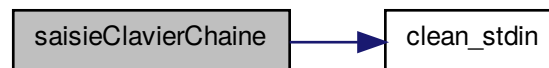
4.19.2.3 char * saisieClavierChaine (char * chaine, int nb_caract_plus_un)

Fait une saisie clavier de chaine de caractere au clavier dans chaine sur nb-caract_plus_un moins 1 caractere

Parameters

in, out	*chaine	une chaine de caractere
in	nb_caract_plus_un	le nombre de caractere moins un que la fonction va lire

Here is the call graph for this function:



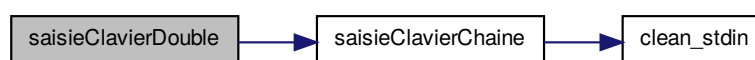
4.19.2.4 void saisieClavierDouble (double * nb)

Fait une saisie clavier d'un double, met 0 si l'entree n'est pas un chiffre.

Parameters

in, out	*nb	le nombre que l'on veut saisir.
---------	-----	---------------------------------

Here is the call graph for this function:



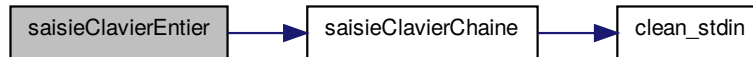
4.19.2.5 void * saisieClavierEntier (int * nb)

Fait une saisie clavier d'un entier, met 0 si l'entree n'est pas un chiffre.

Parameters

in, out	*nb	le nombre que l'on veut saisir.
---------	-----	---------------------------------

Here is the call graph for this function:



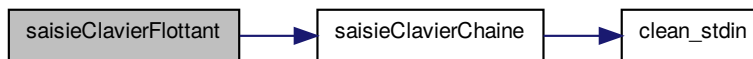
4.19.2.6 void saisieClavierFlottant (float * nb)

Fait une saisie clavier d'un flottant, met 0 si l'entree n'est pas un chiffre.

Parameters

in, out	*nb	le nombre que l'on veut saisir.
---------	-----	---------------------------------

Here is the call graph for this function:



4.19.2.7 void saisieClavierFlottantSansVirgule (float * nb)

Fait une saisie clavier d'un flottant sans virgule, met 0 si l'entree n'est pas un chiffre.

Parameters

in, out	*nb	le nombre que l'on veut saisir.
---------	-----	---------------------------------

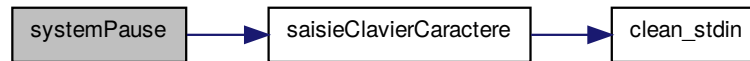
Here is the call graph for this function:



4.19.2.8 void systemPause ()

Demande a l'utilisateur d'appuyer sur entrer pour continuer le programme.

Here is the call graph for this function:

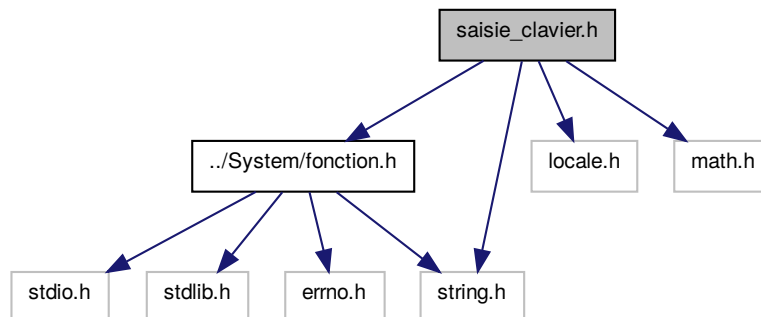


4.20 saisie_clavier.h File Reference

Prototypes des fonctions de saisie clavier.

```
#include "../System/fonction.h"
#include <locale.h>
#include <string.h>
#include <math.h>
```

Include dependency graph for `saisie_clavier.h`:



Macros

- `#define NB_CARACT_INT 12`
- `#define NB_CARACT_FLOT 39`
- `#define NB_CARACT_DOUB 309`

Functions

- void `clean_stdin` (void)
- char * `saisieClavierChaine` (char *chaine, int nb_caract_plus_un)
- void `saisieClavierEntier` (int *nb)
- void `saisieClavierFlottant` (float *nb)
- void `saisieClavierFlottantSansVirgule` (float *nb)

- void [saisieClavierDouble](#) (double *nb)
- char * [saisieClavierCaractere](#) (char *c)
- void [systemPause](#) ()

4.20.1 Detailed Description

Prototypes des fonctions de saisie clavier.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.20.2 Macro Definition Documentation

4.20.2.1 #define NB_CHARACTER DOUB 309

Definit NB_CHARACTER DOUB a 309

4.20.2.2 #define NB_CHARACTER FLOT 39

Definit NB_CHARACTER FLOT a 39

4.20.2.3 #define NB_CHARACTER INT 12

Definit NB_CHARACTER INT a 12

4.20.3 Function Documentation

4.20.3.1 void clean_stdin (void)

Vide la cache de stdin.

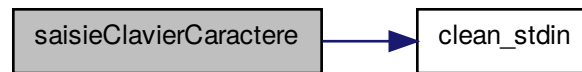
4.20.3.2 char* saisieClavierCaractere (char * c)

Fait une saisie clavier d'un caractere.

Parameters

in, out	*c	le caractere que l'on veut saisir
---------	----	-----------------------------------

Here is the call graph for this function:



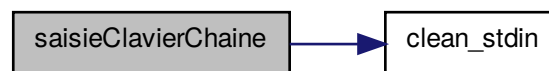
4.20.3.3 char* saisieClavierChaine (char * chaine, int nb_caract_plus_un)

Fait une saisie clavier de chaine de caractere au clavier dans chaine sur nb-caract_plus_un moins 1 caractere

Parameters

in, out	*chaine	une chaine de caractere
in	nb_caract_plus_un	le nombre de caractere moins un que la fonction va lire

Here is the call graph for this function:



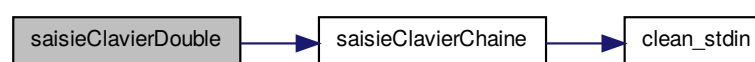
4.20.3.4 void saisieClavierDouble (double * nb)

Fait une saisie clavier d'un double, met 0 si l'entree n'est pas un chiffre.

Parameters

in, out	*nb	le nombre que l'on veut saisir.
---------	-----	---------------------------------

Here is the call graph for this function:



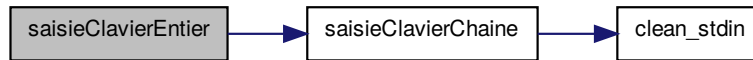
4.20.3.5 void saisieClavierEntier (int * nb)

Fait une saisie clavier d'un entier, met 0 si l'entree n'est pas un chiffre.

Parameters

in, out	*nb	le nombre que l'on veut saisir.
---------	-----	---------------------------------

Here is the call graph for this function:



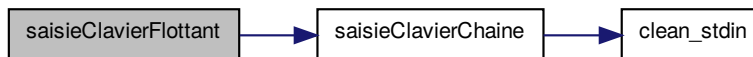
4.20.3.6 void saisieClavierFlottant (float * nb)

Fait une saisie clavier d'un flottant, met 0 si l'entree n'est pas un chiffre.

Parameters

in, out	*nb	le nombre que l'on veut saisir.
---------	-----	---------------------------------

Here is the call graph for this function:



4.20.3.7 void saisieClavierFlottantSansVirgule (float * nb)

Fait une saisie clavier d'un flottant sans virgule, met 0 si l'entree n'est pas un chiffre.

Parameters

in, out	*nb	le nombre que l'on veut saisir.
---------	-----	---------------------------------

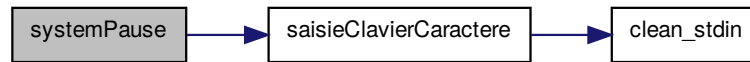
Here is the call graph for this function:



4.20.3.8 void systemPause ()

Demande a l'utilisateur d'appuyer sur entrer pour continuer le programme.

Here is the call graph for this function:

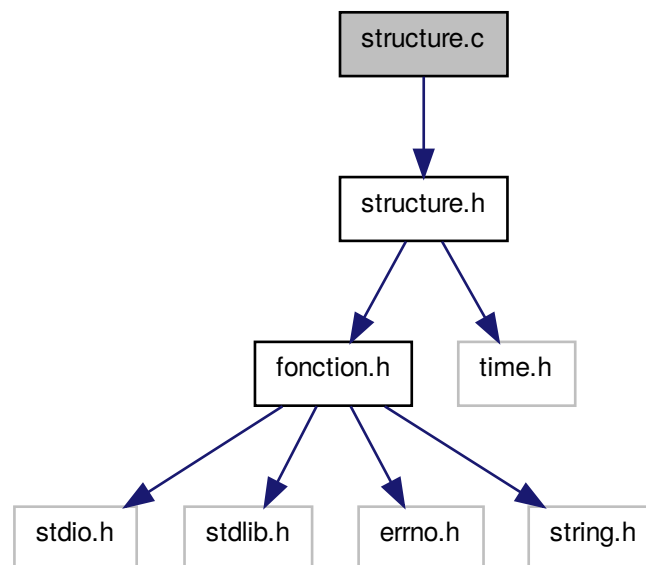


4.21 structure.c File Reference

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

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

Include dependency graph for structure.c:



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)

4.21.1 Detailed Description

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

Author

Remi BERTHO

Date

09/03/14

Version

2.1.0

4.21.2 Function Documentation

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

Ajoute la personne qui distribue dans la structure.

Parameters

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

Here is the call graph for this function:



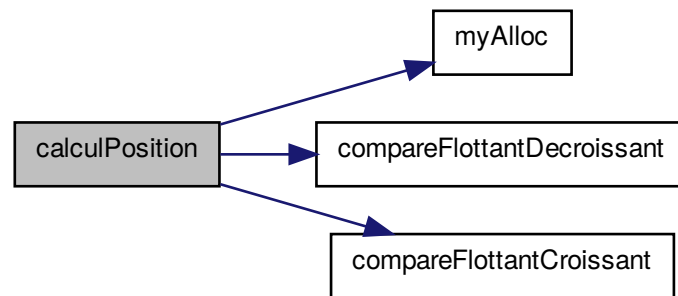
4.21.2.2 void calculPosition (Fichier_Jeu * ptr_struct_fichier)

Calcule les positions des joueurs

Parameters

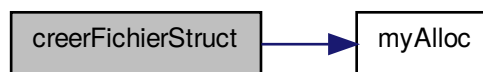
in, out	*ptr_struct_fichier	un pointeur sur la structure Fichier_Jeu
---------	---------------------	--

Here is the call graph for this function:



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

Here is the call graph for this function:



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

Here is the call graph for this function:



4.21.2.5 `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

4.21.2.6 void fermeeFichierStruct (Fichier_Jeu * ptr_struct_fichier)

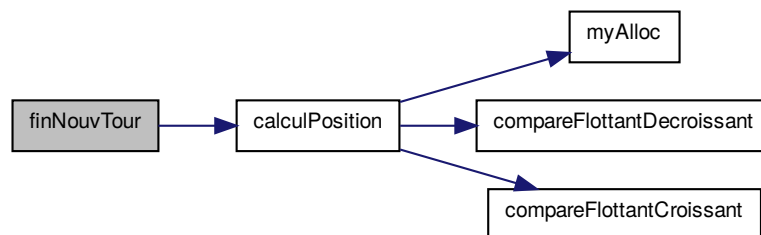
Desalloue la memoire attribuee a la structure [Fichier_Jeu](#) mis en parametre

Parameters

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

4.21.2.7 void finNouvTour (Fichier_Jeu * ptr_struct_fichier, int num_joueur)

Here is the call graph for this function:



4.21.2.8 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

4.21.2.9 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

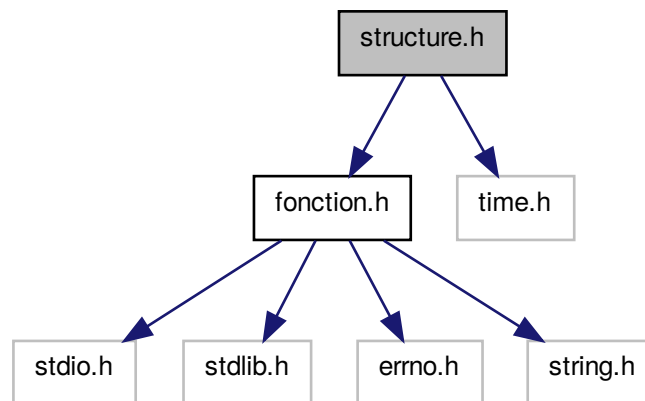
4.22 structure.h File Reference

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

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

```
#include <time.h>
```

Include dependency graph for structure.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)

4.22.1 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

4.22.2 Macro Definition Documentation

4.22.2.1 #define TAILLE_MAX_NOM 30

Definit la taille max d'un nom a 30

4.22.2.2 #define VERSION 1.4

Definit la version a 1.4

4.22.3 Function Documentation

4.22.3.1 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:



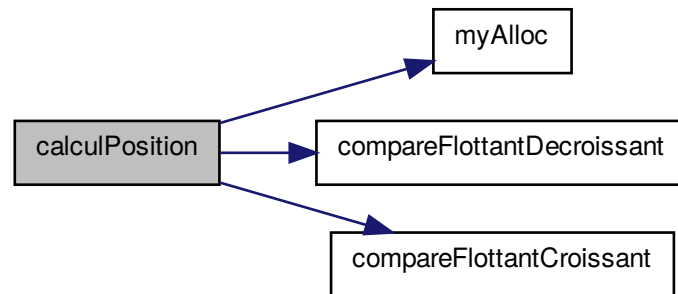
4.22.3.2 void calculPosition (Fichier_Jeu * *ptr_struct_fichier*)

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:



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

Here is the call graph for this function:



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

Here is the call graph for this function:



4.22.3.5 `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 dépassement, FAUX sinon

4.22.3.6 void fermeeFichierStruct (Fichier_Jeu * ptr_struct_fichier)

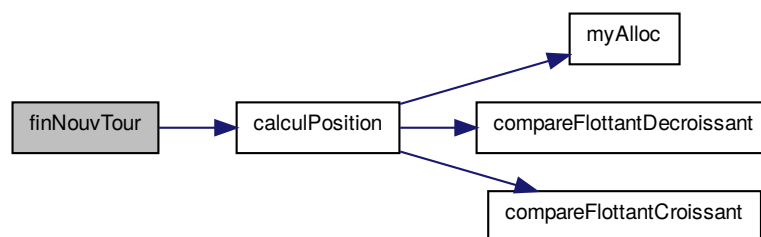
Desalloue la memoire attribuee a la structure [Fichier_Jeu](#) mis en parametre

Parameters

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

4.22.3.7 void finNouvTour (Fichier_Jeu * ptr_struct_fichier, int num_joueur)

Here is the call graph for this function:



4.22.3.8 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

4.22.3.9 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, 24
 - config.h, 30
- affFich
 - interface.h, 72
- affichage.c, 9
 - afficherChaineTroisTab, 10
 - afficherDistribue, 10
 - afficherEnTete, 10
 - afficherLicense, 11
 - afficherLigne, 11
 - afficherNom, 11
 - afficherPartieFinie, 11
 - afficherPosition, 12
 - afficherScore, 12
 - afficherScoreEntier, 13
 - afficherScoreTotal, 13
 - afficherStruct, 13
 - printGameConfig, 14
- affichage.h, 14
 - afficherChaineTroisTab, 16
 - afficherDistribue, 16
 - afficherEnTete, 16
 - afficherLicense, 16
 - afficherLigne, 16
 - afficherNom, 17
 - afficherPartieFinie, 17
 - afficherPosition, 17
 - afficherScore, 17
 - afficherScoreEntier, 18
 - afficherScoreTotal, 18
 - afficherStruct, 19
 - printGameConfig, 19
- afficheFichier
 - interface.c, 60
 - interface.h, 72
- afficheFichierLocale
 - interface.c, 60
 - interface.h, 73
- afficherChaineTroisTab
 - affichage.c, 10
 - affichage.h, 16
- afficherDistribue
 - affichage.c, 10
 - affichage.h, 16
- afficherEnTete
 - affichage.c, 10
 - affichage.h, 16
- afficherLicense
 - affichage.c, 11
 - affichage.h, 16
- afficherLigne
 - affichage.c, 11
 - affichage.h, 16
- afficherNom
 - affichage.c, 11
 - affichage.h, 17
- afficherPartieFinie
 - affichage.c, 11
 - affichage.h, 17
- afficherPosition
 - affichage.c, 12
 - affichage.h, 17
- afficherScore
 - affichage.c, 12
 - affichage.h, 17
- afficherScoreEntier
 - affichage.c, 13
 - affichage.h, 18
- afficherScoreTotal
 - affichage.c, 13
 - affichage.h, 18
- afficherStruct
 - affichage.c, 13
 - affichage.h, 19
- ajoutDistribueStruct
 - structure.c, 113
 - structure.h, 117
- ajoutExtension
 - fonction.c, 54
 - fonction.h, 57
- annee
 - Fichier_Jeu, 6
- argument_main.c, 19
 - searchArgument, 20
- argument_main.h, 21
 - LECTURE_FICHIER, 22
 - OUVERTURE_FICHIER, 22
 - searchArgument, 22
- calculPosition
 - structure.c, 113
 - structure.h, 117
- changerCheminFichier
 - emplacement_fichier.c, 36
 - emplacement_fichier.h, 40
- charPart
 - interface.h, 72
- chargerPartie

- interface.c, 61
- interface.h, 74
- chargerPartieLocale
 - interface.c, 62
 - interface.h, 75
- clean_stdin
 - saisie_clavier.c, 103
 - saisie_clavier.h, 108
- compareFlottantCroissant
 - fonction.c, 54
 - fonction.h, 57
- compareFlottantDecroissant
 - fonction.c, 54
 - fonction.h, 57
- config
 - Fichier_Jeu, 6
- config.c, 23
 - addConfigListFile, 24
 - freeListGameConfig, 24
 - makeConfigFile, 24
 - makeConfigListFile, 25
 - makeListGameConfig, 25
 - readConfigFile, 26
 - readConfigListFile, 26
 - removeConfigFile, 27
 - removeConfigListFile, 27
- config.h, 28
 - addConfigListFile, 30
 - freeListGameConfig, 30
 - makeConfigFile, 30
 - makeConfigListFile, 31
 - makeListGameConfig, 31
 - NOM_DOSSIER_CONFIG, 30
 - NOM_FICHER_CONFIG, 30
 - readConfigFile, 32
 - readConfigListFile, 32
 - removeConfigFile, 33
 - removeConfigListFile, 33
- creationPreferences
 - emplacement_fichier.c, 36
 - emplacement_fichier.h, 40
- creerFichierStruct
 - structure.c, 114
 - structure.h, 119
- debNouvTour
 - structure.c, 114
 - structure.h, 119
- depScoreMax
 - structure.c, 114
 - structure.h, 119
- distribue
 - Fichier_Jeu, 6
- EXTENSION_FICHER
 - fichier.h, 49
- easterEggs
 - interface.h, 72
- easterEggs2
 - interface.h, 72
- ecrireFichier
 - fichier.c, 44
 - fichier.h, 49
- emplacement_fichier.c, 34
 - changerCheminFichier, 36
 - creationPreferences, 36
 - lectureCheminFichier, 37
 - lecturePreferences, 37
- emplacement_fichier.h, 38
 - changerCheminFichier, 40
 - creationPreferences, 40
 - lectureCheminFichier, 41
 - lecturePreferences, 41
- FAUX
 - fonction.h, 57
- fermeeFichierStruct
 - structure.c, 115
 - structure.h, 121
- fermerFichier
 - fonction.c, 54
 - fonction.h, 57
- fichier.c, 42
 - ecrireFichier, 44
 - lireFichier, 44
 - nouveauScore, 45
 - ouvrirFichierExtension, 45
 - renommerFichier, 47
 - supprimerFichier, 47
- fichier.h, 48
 - EXTENSION_FICHER, 49
 - ecrireFichier, 49
 - lireFichier, 50
 - nouveauScore, 51
 - ouvrirFichierExtension, 51
 - renommerFichier, 51
 - supprimerFichier, 52
 - TYPE_FICHER, 49
- Fichier_Jeu, 5
 - annee, 6
 - config, 6
 - distribue, 6
 - jour, 6
 - mois, 6
 - nb_joueur, 6
 - nb_tour, 6
 - nom_joueur, 6
 - point, 6
 - point_tot, 6
 - position, 6
 - taille_max_nom, 6
 - version, 7
- finNouvTour
 - structure.c, 115
 - structure.h, 121
- fonction.c, 52
 - ajoutExtension, 54
 - compareFlottantCroissant, 54

- compareFlottantDecroissant, 54
- fermerFichier, 54
- lireTailleFichier, 54
- mauvais_choix, 55
- myAlloc, 55
- myRealloc, 55
- ouvrirFichier, 55
- systemEfface, 55
- fonction.h, 55
 - ajoutExtension, 57
 - compareFlottantCroissant, 57
 - compareFlottantDecroissant, 57
 - FAUX, 57
 - fermerFichier, 57
 - lireTailleFichier, 57
 - mauvais_choix, 58
 - myAlloc, 58
 - myRealloc, 58
 - ouvrirFichier, 58
 - systemEfface, 58
 - VRAI, 57
- freeListGameConfig
 - config.c, 24
 - config.h, 30
- game_config, 7
 - name, 7
 - nb_max, 7
 - number_after_comma, 7
 - sens_premier, 7
 - tour_par_tour, 7
 - use_distributor, 7
- interface.h
 - affFich, 72
 - charPart, 72
 - easterEggs, 72
 - easterEggs2, 72
 - lireChem, 72
 - listFich, 72
 - menuPrinc, 72
 - newGameConf, 72
 - nouvChem, 72
 - nouvPart, 72
 - pref, 72
 - printGameConf, 72
 - printListGameConf, 72
 - quit, 72
 - removeGameConf, 72
 - supprFich, 72
- interface.c, 58
 - afficheFichier, 60
 - afficheFichierLocale, 60
 - chargerPartie, 61
 - chargerPartieLocale, 62
 - jouer, 63
 - lireCheminFichier, 64
 - listerFichier, 64
 - menuPreferences, 65
 - menuPrincipal, 65
 - newGameConfig, 66
 - nouveauCheminFichier, 67
 - nouvellePartie, 67
 - printGameConfigFile, 68
 - printListGameConfig, 69
 - removeGameConfig, 69
 - supprimerFichierNom, 70
- interface.h, 70
 - afficheFichier, 72
 - afficheFichierLocale, 73
 - chargerPartie, 74
 - chargerPartieLocale, 75
 - jouer, 76
 - lireCheminFichier, 77
 - listerFichier, 77
 - MenuPreferences, 72
 - menuPreferences, 78
 - MenuPrincipal, 72
 - menuPrincipal, 78
 - newGameConfig, 79
 - nouveauCheminFichier, 80
 - nouvellePartie, 80
 - printGameConfigFile, 81
 - printListGameConfig, 82
 - removeGameConfig, 82
 - supprimerFichierNom, 83
- jouer
 - interface.c, 63
 - interface.h, 76
- jour
 - Fichier_Jeu, 6
- LECTURE_FICHIER
 - argument_main.h, 22
- lectureCheminFichier
 - emplacement_fichier.c, 37
 - emplacement_fichier.h, 41
- lecturePreferences
 - emplacement_fichier.c, 37
 - emplacement_fichier.h, 41
- lireChem
 - interface.h, 72
- lireCheminFichier
 - interface.c, 64
 - interface.h, 77
- lireFichier
 - fichier.c, 44
 - fichier.h, 50
- lireTailleFichier
 - fonction.c, 54
 - fonction.h, 57
- listFich
 - interface.h, 72
- list_game_config, 8
 - name_game_config, 8
 - nb_config, 8
- listerFichier

- interface.c, 64
- interface.h, 77
- main
 - main.c, 85
 - main.h, 87
- main.c, 83
 - main, 85
- main.h, 86
 - main, 87
- makeConfigFile
 - config.c, 24
 - config.h, 30
- makeConfigListFile
 - config.c, 25
 - config.h, 31
- makeListGameConfig
 - config.c, 25
 - config.h, 31
- mauvais_choix
 - fonction.c, 55
 - fonction.h, 58
- maxNbTour
 - structure.c, 115
 - structure.h, 121
- menu.c, 90
 - menuContinuer, 91
 - menuDebutPartie, 91
 - menuDistribue, 92
 - menuGameConfig, 92
 - menuNomFichier, 93
 - menuNomJoueur, 93
 - menuNouveauChemin, 93
 - menuNumJoueur, 95
 - menuPointsJoueur, 95
 - menuSupprimer, 96
- menu.h, 96
 - menuContinuer, 98
 - menuDebutPartie, 98
 - menuDistribue, 98
 - menuGameConfig, 99
 - menuNomFichier, 99
 - menuNomJoueur, 100
 - menuNouveauChemin, 100
 - menuNumJoueur, 101
 - menuPointsJoueur, 101
 - menuSupprimer, 102
- menuPrinc
 - interface.h, 72
- menuContinuer
 - menu.c, 91
 - menu.h, 98
- menuDebutPartie
 - menu.c, 91
 - menu.h, 98
- menuDistribue
 - menu.c, 92
 - menu.h, 98
- menuGameConfig
 - menu.c, 92
 - menu.h, 99
- menuNomFichier
 - menu.c, 93
 - menu.h, 99
- menuNomJoueur
 - menu.c, 93
 - menu.h, 100
- menuNouveauChemin
 - menu.c, 93
 - menu.h, 100
- menuNumJoueur
 - menu.c, 95
 - menu.h, 101
- menuPointsJoueur
 - menu.c, 95
 - menu.h, 101
- MenuPreferences
 - interface.h, 72
- menuPreferences
 - interface.c, 65
 - interface.h, 78
- MenuPrincipal
 - interface.h, 72
- menuPrincipal
 - interface.c, 65
 - interface.h, 78
- menuSupprimer
 - menu.c, 96
 - menu.h, 102
- mois
 - Fichier_Jeu, 6
- myAlloc
 - fonction.c, 55
 - fonction.h, 58
- myRealloc
 - fonction.c, 55
 - fonction.h, 58
- NB_CARACT_DOUB
 - saisie_clavier.h, 108
- NB_CARACT_FLOT
 - saisie_clavier.h, 108
- NB_CARACT_INT
 - saisie_clavier.h, 108
- NOM_DOSSIER_CONFIG
 - config.h, 30
- NOM_DOSSIER_CSUPER
 - emplacement_fichier.h, 40
- NOM_FICHIER_CONFIG
 - config.h, 30
- 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
- newGameConf
 - interface.h, 72
- newGameConfig
 - interface.c, 66
 - interface.h, 79
- nom_joueur
 - Fichier_Jeu, 6
- nouvChem
 - interface.h, 72
- nouvPart
 - interface.h, 72
- nouveauCheminFichier
 - interface.c, 67
 - interface.h, 80
- nouveauScore
 - fichier.c, 45
 - fichier.h, 51
- nouvellePartie
 - interface.c, 67
 - interface.h, 80
- number_after_comma
 - game_config, 7
- OUVERTURE_FICHIER
 - argument_main.h, 22
- ouvrirFichier
 - fonction.c, 55
 - fonction.h, 58
- ouvrirFichierExtension
 - fichier.c, 45
 - fichier.h, 51
- point
 - Fichier_Jeu, 6
- point_tot
 - Fichier_Jeu, 6
- position
 - Fichier_Jeu, 6
- pref
 - interface.h, 72
- printGameConf
 - interface.h, 72
- printListGameConf
 - interface.h, 72
- printGameConfig
 - affichage.c, 14
 - affichage.h, 19
- printGameConfigFile
 - interface.c, 68
 - interface.h, 81
- printListGameConfig
 - interface.c, 69
 - interface.h, 82
- quit
 - interface.h, 72
- readConfigFile
 - config.c, 26
 - config.h, 32
- readConfigListFile
 - config.c, 26
 - config.h, 32
- rechercheNumJoueur
 - structure.c, 115
 - structure.h, 121
- removeGameConf
 - interface.h, 72
- removeConfigFile
 - config.c, 27
 - config.h, 33
- removeConfigListFile
 - config.c, 27
 - config.h, 33
- removeGameConfig
 - interface.c, 69
 - interface.h, 82
- renommerFichier
 - fichier.c, 47
 - fichier.h, 51
- saisie_clavier.c, 102
 - clean_stdin, 103
 - saisieClavierCaractere, 103
 - saisieClavierChaine, 105
 - saisieClavierDouble, 105
 - saisieClavierEntier, 105
 - saisieClavierFlottant, 106
 - saisieClavierFlottantSansVirgule, 106
 - systemPause, 106
- saisie_clavier.h, 107
 - clean_stdin, 108
 - NB_CHARACTER_DOUB, 108
 - NB_CHARACTER_FLOT, 108
 - NB_CHARACTER_INT, 108
 - saisieClavierCaractere, 108
 - saisieClavierChaine, 109
 - saisieClavierDouble, 109
 - saisieClavierEntier, 109
 - saisieClavierFlottant, 111
 - saisieClavierFlottantSansVirgule, 111
 - systemPause, 111
- saisieClavierCaractere
 - saisie_clavier.c, 103
 - saisie_clavier.h, 108
- saisieClavierChaine
 - saisie_clavier.c, 105
 - saisie_clavier.h, 109
- saisieClavierDouble
 - saisie_clavier.c, 105
 - saisie_clavier.h, 109
- saisieClavierEntier
 - saisie_clavier.c, 105
 - saisie_clavier.h, 109

saisieClavierFlottant
 saisie_clavier.c, 106
 saisie_clavier.h, 111
saisieClavierFlottantSansVirgule
 saisie_clavier.c, 106
 saisie_clavier.h, 111
searchArgument
 argument_main.c, 20
 argument_main.h, 22
sens_premier
 game_config, 7
structure.c, 112
 ajoutDistribueStruct, 113
 calculPosition, 113
 creerFichierStruct, 114
 debNouvTour, 114
 depScoreMax, 114
 fermeeFichierStruct, 115
 finNouvTour, 115
 maxNbTour, 115
 rechercheNumJoueur, 115
structure.h, 116
 ajoutDistribueStruct, 117
 calculPosition, 117
 creerFichierStruct, 119
 debNouvTour, 119
 depScoreMax, 119
 fermeeFichierStruct, 121
 finNouvTour, 121
 maxNbTour, 121
 rechercheNumJoueur, 121
 TAILLE_MAX_NOM, 117
 VERSION, 117
supprFich
 interface.h, 72
supprimerFichier
 fichier.c, 47
 fichier.h, 52
supprimerFichierNom
 interface.c, 70
 interface.h, 83
systemEfface
 fonction.c, 55
 fonction.h, 58
systemPause
 saisie_clavier.c, 106
 saisie_clavier.h, 111

TAILLE_MAX_NOM
 structure.h, 117
TYPE_FICHER
 fichier.h, 49
taille_max_nom
 Fichier_Jeu, 6
tour_par_tour
 game_config, 7

use_distributor
 game_config, 7

VERSION
 structure.h, 117
VRAI
 fonction.h, 57
version
 Fichier_Jeu, 7