

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

2.1.3

Generated by Doxygen 1.8.6

Sat Mar 22 2014 17:43:28

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	5
3.1.2	Field Documentation	5
3.1.2.1	annee	5
3.1.2.2	distribue	5
3.1.2.3	jour	6
3.1.2.4	mois	6
3.1.2.5	nb_joueur	6
3.1.2.6	nb_max	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	sens_premier	6
3.1.2.13	taille_max_nom	6
3.1.2.14	version	6
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	11

4.1.2.4	afficherLicense	11
4.1.2.5	afficherLigne	12
4.1.2.6	afficherNom	12
4.1.2.7	afficherPartieFinie	13
4.1.2.8	afficherPosition	13
4.1.2.9	afficherScore	14
4.1.2.10	afficherScoreEntier	15
4.1.2.11	afficherScoreTotal	15
4.1.2.12	afficherStruct	16
4.2	affichage.h File Reference	17
4.2.1	Detailed Description	18
4.2.2	Function Documentation	19
4.2.2.1	afficherChaineTroisTab	19
4.2.2.2	afficherDistribue	19
4.2.2.3	afficherEnTete	19
4.2.2.4	afficherLicense	20
4.2.2.5	afficherLigne	20
4.2.2.6	afficherNom	21
4.2.2.7	afficherPartieFinie	21
4.2.2.8	afficherPosition	22
4.2.2.9	afficherScore	22
4.2.2.10	afficherScoreEntier	23
4.2.2.11	afficherScoreTotal	24
4.2.2.12	afficherStruct	24
4.3	argument_main.c File Reference	25
4.3.1	Detailed Description	26
4.3.2	Function Documentation	26
4.3.2.1	searchArgument	26
4.4	argument_main.h File Reference	27
4.4.1	Detailed Description	28
4.4.2	Macro Definition Documentation	28
4.4.2.1	CHAINE_LECTURE_FICHIER	28
4.4.2.2	CHAINE_LECTURE_FICHIER_RED	29
4.4.2.3	CHAINE_OUVERTURE_FICHIER	29
4.4.2.4	CHAINE_OUVERTURE_FICHIER_RED	29
4.4.2.5	LECTURE_FICHIER	29
4.4.2.6	OUVERTURE_FICHIER	29
4.4.3	Function Documentation	29
4.4.3.1	searchArgument	29
4.5	emplacement_fichier.c File Reference	29

4.5.1	Detailed Description	30
4.5.2	Function Documentation	31
4.5.2.1	changerCheminFichier	31
4.5.2.2	creationPreferences	31
4.5.2.3	lectureCheminFichier	32
4.5.2.4	lecturePreferences	33
4.6	emplacement_fichier.h File Reference	34
4.6.1	Detailed Description	36
4.6.2	Macro Definition Documentation	37
4.6.2.1	NOM_DOSSIER	37
4.6.2.2	NOM_FICHIER	37
4.6.2.3	PORTABLE	37
4.6.3	Function Documentation	37
4.6.3.1	changerCheminFichier	37
4.6.3.2	creationPreferences	38
4.6.3.3	lectureCheminFichier	38
4.6.3.4	lecturePreferences	39
4.7	fichier.c File Reference	40
4.7.1	Detailed Description	41
4.7.2	Function Documentation	42
4.7.2.1	ecrireFichier	42
4.7.2.2	lireFichier	42
4.7.2.3	nouveauScore	43
4.7.2.4	ouvrirFichierExtension	44
4.7.2.5	renommerFichier	45
4.7.2.6	supprimerFichier	45
4.8	fichier.h File Reference	46
4.8.1	Detailed Description	47
4.8.2	Macro Definition Documentation	48
4.8.2.1	EXTENSION_FICHIER	48
4.8.2.2	TAILLE_MAX_NOM_FICHIER	48
4.8.2.3	TYPE_FICHIER	48
4.8.3	Function Documentation	48
4.8.3.1	ecrireFichier	48
4.8.3.2	lireFichier	49
4.8.3.3	nouveauScore	50
4.8.3.4	ouvrirFichierExtension	51
4.8.3.5	renommerFichier	51
4.8.3.6	supprimerFichier	52
4.9	fonction.c File Reference	52

4.9.1	Detailed Description	53
4.9.2	Function Documentation	54
4.9.2.1	compareFlottantCroissant	54
4.9.2.2	compareFlottantDecroissant	55
4.9.2.3	fermerFichier	55
4.9.2.4	lireTailleFichier	56
4.9.2.5	mauvais_choix	56
4.9.2.6	myAlloc	57
4.9.2.7	myRealloc	57
4.9.2.8	ouvrirFichier	57
4.9.2.9	systemEfface	58
4.9.2.10	systemPause	59
4.10	fonction.h File Reference	59
4.10.1	Detailed Description	61
4.10.2	Macro Definition Documentation	61
4.10.2.1	FAUX	61
4.10.2.2	VRAI	61
4.10.3	Function Documentation	61
4.10.3.1	compareFlottantCroissant	61
4.10.3.2	compareFlottantDecroissant	62
4.10.3.3	fermerFichier	63
4.10.3.4	lireTailleFichier	63
4.10.3.5	mauvais_choix	64
4.10.3.6	myAlloc	64
4.10.3.7	myRealloc	65
4.10.3.8	ouvrirFichier	65
4.10.3.9	systemEfface	66
4.10.3.10	systemPause	66
4.11	interface.c File Reference	67
4.11.1	Detailed Description	68
4.11.2	Function Documentation	69
4.11.2.1	afficheFichier	69
4.11.2.2	afficheFichierLocale	70
4.11.2.3	chargerPartie	71
4.11.2.4	chargerPartieLocale	72
4.11.2.5	jouer	73
4.11.2.6	lireCheminFichier	74
4.11.2.7	listerFichier	74
4.11.2.8	menuPrincipal	75
4.11.2.9	nouveauCheminFichier	76

4.11.2.10 nouvellePartie	77
4.11.2.11 supprimerFichierNom	78
4.12 interface.h File Reference	79
4.12.1 Detailed Description	81
4.12.2 Enumeration Type Documentation	81
4.12.2.1 Menu	81
4.12.3 Function Documentation	81
4.12.3.1 afficheFichier	81
4.12.3.2 afficheFichierLocale	82
4.12.3.3 chargerPartie	83
4.12.3.4 chargerPartieLocale	84
4.12.3.5 jouer	85
4.12.3.6 lireCheminFichier	86
4.12.3.7 listerFichier	87
4.12.3.8 menuPrincipal	88
4.12.3.9 nouveauCheminFichier	89
4.12.3.10 nouvellePartie	89
4.12.3.11 supprimerFichierNom	90
4.13 main.c File Reference	91
4.13.1 Detailed Description	92
4.13.2 Function Documentation	92
4.13.2.1 main	92
4.14 main.h File Reference	94
4.14.1 Detailed Description	95
4.14.2 Function Documentation	95
4.14.2.1 main	95
4.15 menu.c File Reference	96
4.15.1 Detailed Description	97
4.15.2 Function Documentation	98
4.15.2.1 menuContinuer	98
4.15.2.2 menuDebutPartie	98
4.15.2.3 menuDistribue	99
4.15.2.4 menuNomFichier	99
4.15.2.5 menuNomJoueur	100
4.15.2.6 menuNouveauChemin	101
4.15.2.7 menuPointsJoueur	101
4.15.2.8 menuSupprimer	102
4.16 menu.h File Reference	103
4.16.1 Detailed Description	104
4.16.2 Function Documentation	105

4.16.2.1	menuContinuer	105
4.16.2.2	menuDebutPartie	105
4.16.2.3	menuDistribue	106
4.16.2.4	menuNomFichier	106
4.16.2.5	menuNomJoueur	107
4.16.2.6	menuNouveauChemin	108
4.16.2.7	menuPointsJoueur	108
4.16.2.8	menuSupprimer	109
4.17	saisie_clavier.c File Reference	110
4.17.1	Detailed Description	111
4.17.2	Function Documentation	111
4.17.2.1	clean_stdin	111
4.17.2.2	saisieClavierCaractere	111
4.17.2.3	saisieClavierChaine	112
4.17.2.4	saisieClavierDouble	113
4.17.2.5	saisieClavierEntier	113
4.17.2.6	saisieClavierFlottant	114
4.17.2.7	saisieClavierFlottantSansVirgule	115
4.18	saisie_clavier.h File Reference	116
4.18.1	Detailed Description	117
4.18.2	Macro Definition Documentation	118
4.18.2.1	NB_CHARACTER_DOUB	118
4.18.2.2	NB_CHARACTER_FLOT	118
4.18.2.3	NB_CHARACTER_INT	118
4.18.3	Function Documentation	118
4.18.3.1	clean_stdin	118
4.18.3.2	saisieClavierCaractere	118
4.18.3.3	saisieClavierChaine	119
4.18.3.4	saisieClavierDouble	120
4.18.3.5	saisieClavierEntier	120
4.18.3.6	saisieClavierFlottant	121
4.18.3.7	saisieClavierFlottantSansVirgule	122
4.19	structure.c File Reference	123
4.19.1	Detailed Description	124
4.19.2	Function Documentation	125
4.19.2.1	ajoutDistribueStruct	125
4.19.2.2	calculPosition	125
4.19.2.3	creerFichierStruct	126
4.19.2.4	debNouvTour	127
4.19.2.5	depScoreMax	127

4.19.2.6	fermeeFichierStruct	128
4.19.2.7	finNouvTour	129
4.19.2.8	maxNbTour	129
4.20	structure.h File Reference	130
4.20.1	Detailed Description	132
4.20.2	Macro Definition Documentation	132
4.20.2.1	TAILLE_MAX_NOM	132
4.20.2.2	VERSION	132
4.20.3	Function Documentation	132
4.20.3.1	ajoutDistribueStruct	132
4.20.3.2	calculPosition	132
4.20.3.3	creerFichierStruct	133
4.20.3.4	debNouvTour	134
4.20.3.5	depScoreMax	135
4.20.3.6	fermeeFichierStruct	136
4.20.3.7	finNouvTour	137
4.20.3.8	maxNbTour	137
Index		138

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Fichier_Jeu	5
---------------------------------------	---

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	17
argument_main.c	Lancement du programme	25
argument_main.h	Lancement du programme	27
emplacement_fichier.c	Fonctions qui l'emplacement des fichiers sauvegardes	29
emplacement_fichier.h	Prototypes des fonctions qui l'emplacement des fichiers sauvegardes	34
fichier.c	Fonction de gestion des fichiers	40
fichier.h	Prototypes des fonction de gestion des fichiers	46
fonction.c	Fonctions essentielles au fonctionnement du programme	52
fonction.h	Prototypes des fonctions essentielles au fonctionnement du programme	59
interface.c	Fonctions qui gerent l'interface graphique du logiciel	67
interface.h	Prototypes des fonctions qui gerent l'interface graphique du logiciel	79
main.c	Lancement du programme	91
main.h	Lancement du programme	94
menu.c	Fonctions qui gerent les menus demandant de rentrer des valeurs du logiciel	96
menu.h	Prototypes des fonctions qui gerent les menus demandant de rentrer des valeurs du logiciel	103
saisie_clavier.c	Fonctions de saisie clavier	110
saisie_clavier.h	Prototypes des fonctions de saisie clavier	116
structure.c	Fonction de gestion des fichiers de la struction contenant les informations	123

[structure.h](#)Prototypes des fonction de gestion des fichiers de la struction contenant les informations . . . [130](#)

Chapter 3

Data Structure Documentation

3.1 Fichier_Jeu Struct Reference

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

Data Fields

- float [version](#)
- float [taille_max_nom](#)
- float [jour](#)
- float [mois](#)
- float [annee](#)
- float [nb_joueur](#)
- float [nb_max](#)
- char [sens_premier](#)
- 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 float distribue

Numero de la personne qui doit distribuer.

3.1.2.3 float jour

Jour de creation de la structure.

3.1.2.4 float mois

Mois de creation de la structure.

3.1.2.5 float nb_joueur

Nombre de joueurs.

3.1.2.6 float nb_max

Nombre maximum que peut prendre un joueur.

3.1.2.7 float* nb_tour

Nombre de tour dans le jeu par joueur.

3.1.2.8 char** nom_joueur

Tableau contenant tout les noms de joueurs.

3.1.2.9 float** point

Tableau contenant les points de chaque joueur a chaque tour.

3.1.2.10 float* point_tot

Tableau contenant tout les points totaux des joueurs.

3.1.2.11 float* position

Tableau contenant la position des joueurs.

3.1.2.12 char sens_premier

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

3.1.2.13 float taille_max_nom

Taille maximum que peut prendre un nom de joueur.

3.1.2.14 float version

Version de la structure.

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

- [structure.h](#)

Chapter 4

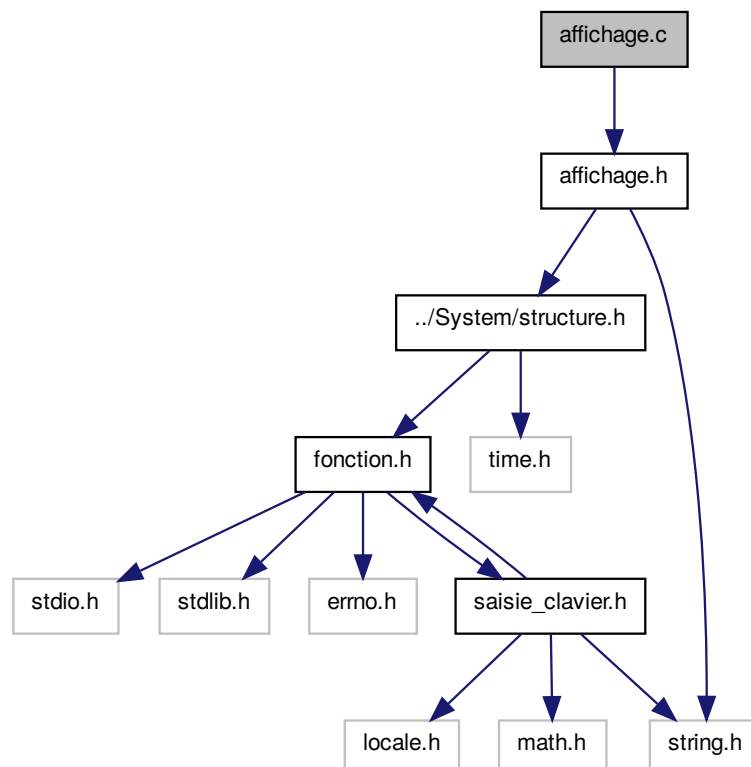
File Documentation

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](#) ()

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

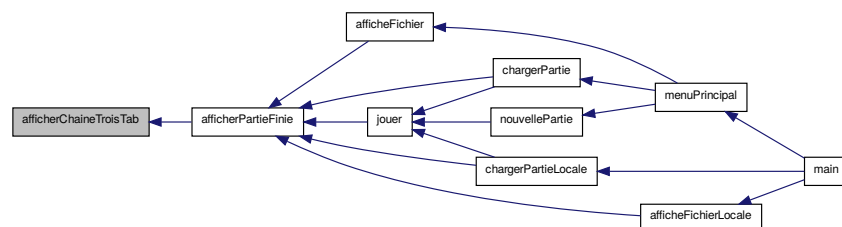
4.1.2.1 void [afficherChaineTroisTab](#) (char * *chaine*)

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

Parameters

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

Here is the caller graph for this function:



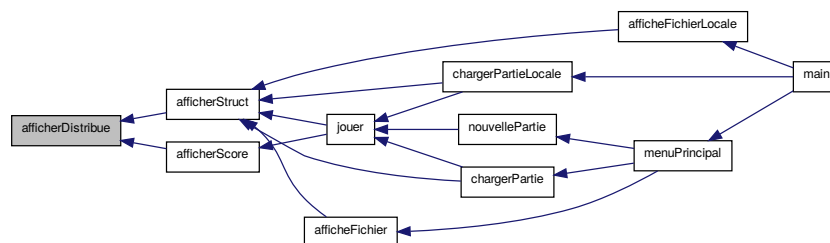
4.1.2.2 void [afficherDistribue](#) ([Fichier_Jeu](#) * *ptr_struct_fichier*)

Affiche la personne devant distribuer

Parameters

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

Here is the caller graph for this function:



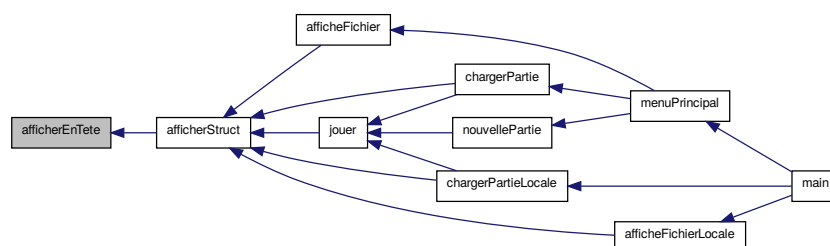
4.1.2.3 void afficherEnTete (Fichier_Jeu * ptr_struct_fichier)

Affiche l'en tete de la structure

Parameters

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

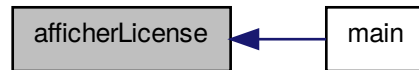
Here is the caller graph for this function:



4.1.2.4 void afficherLicense ()

Affiche la license

Here is the caller graph for this function:



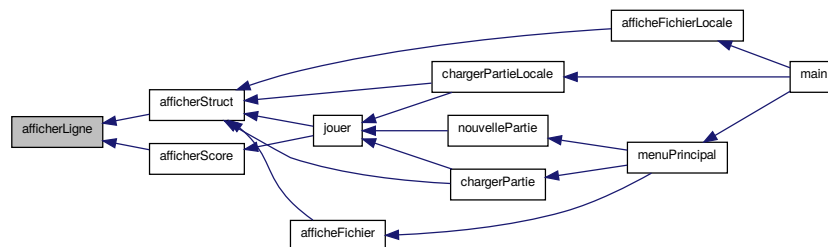
4.1.2.5 void afficherLigne (int *taille_ligne*)

Affiche *taille_ligne* - apres une tabulation

Parameters

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

Here is the caller graph for this function:



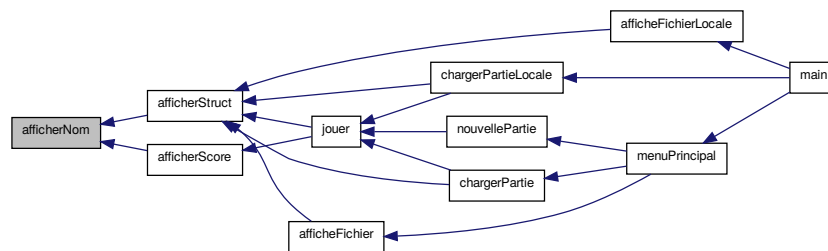
4.1.2.6 void afficherNom (Fichier_Jeu * *ptr_struct_fichier*, int * *ptr_taille_ligne*)

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

Here is the caller graph for this function:



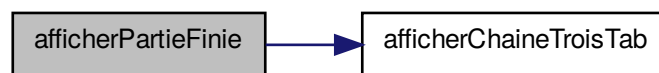
4.1.2.7 void afficherPartieFinie (Fichier_Jeu * ptr_struct_fichier)

Affiche un podium des resultats

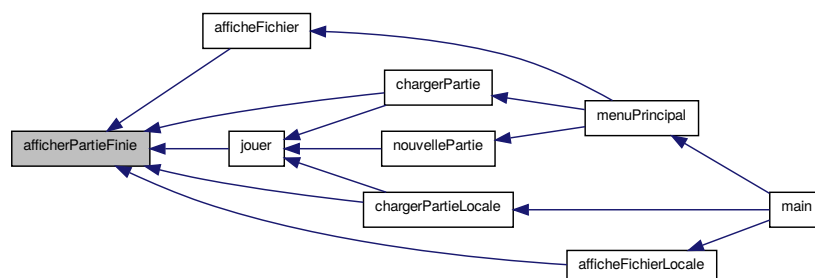
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



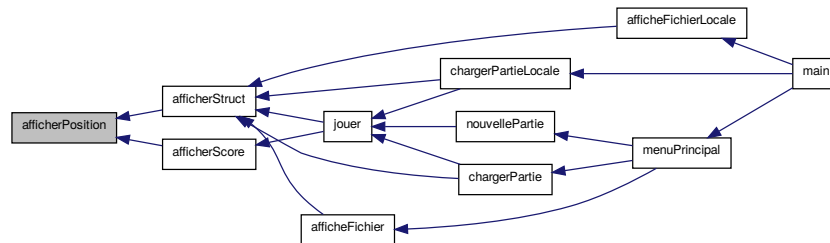
4.1.2.8 void afficherPosition (Fichier_Jeu * ptr_struct_fichier)

Affiche les positions des joueurs

Parameters

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

Here is the caller graph for this function:



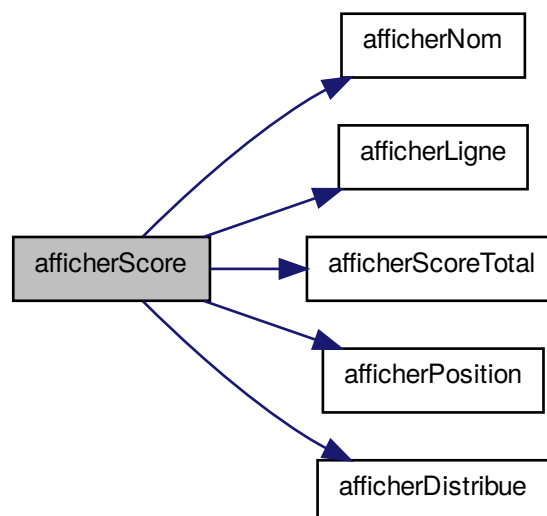
4.1.2.9 void afficherScore (Fichier_Jeu * ptr_struct_fichier)

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

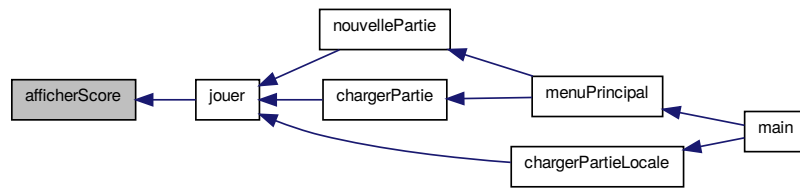
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



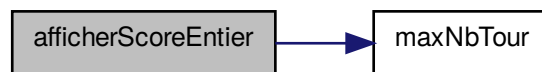
4.1.2.10 void afficherScoreEntier (Fichier_Jeu * ptr_struct_fichier)

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

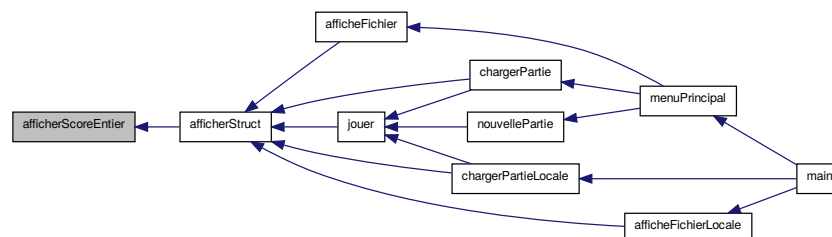
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



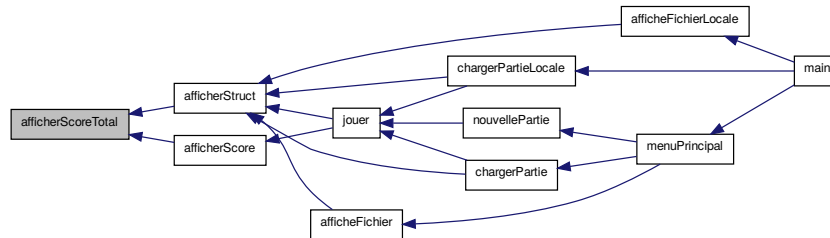
4.1.2.11 void afficherScoreTotal (Fichier_Jeu * ptr_struct_fichier)

Affiche le score total des joueurs

Parameters

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

Here is the caller graph for this function:



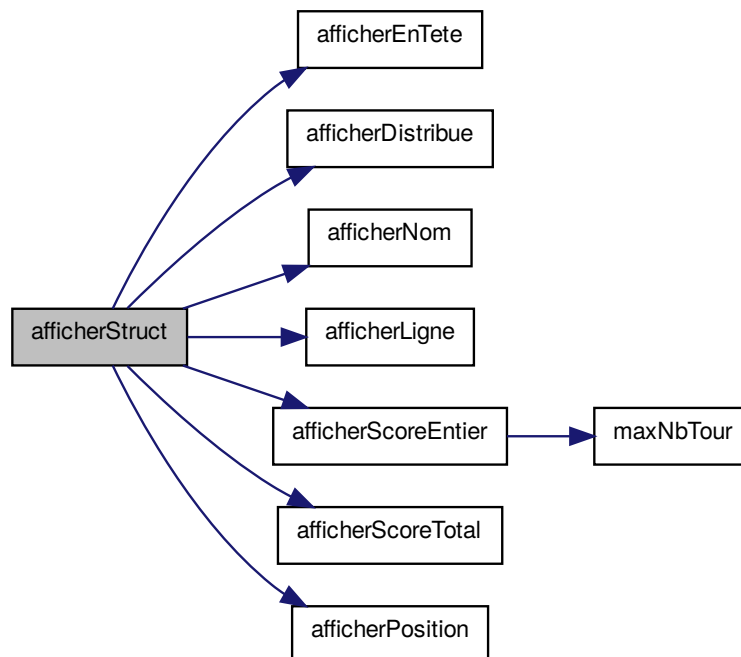
4.1.2.12 void afficherStruct (Fichier_Jeu * ptr_struct_fichier)

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

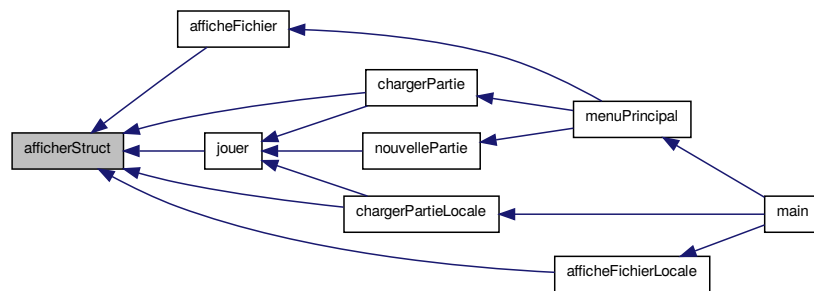
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



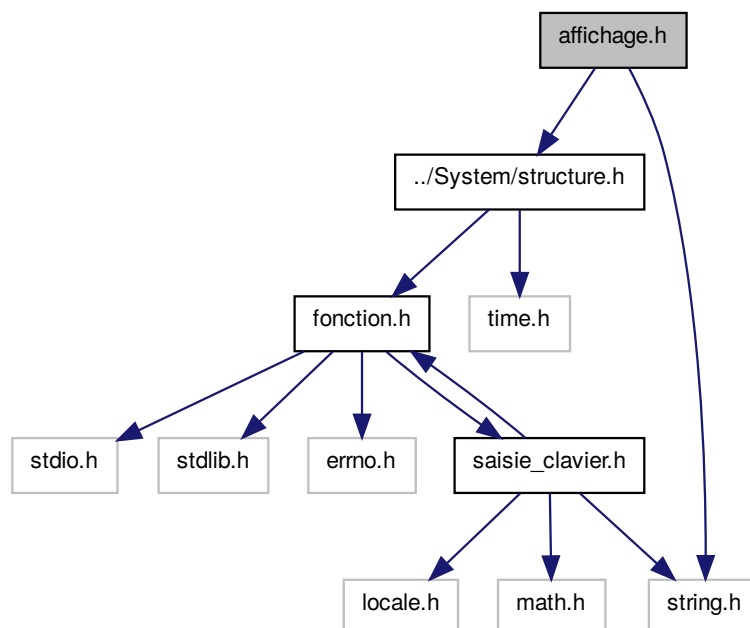
4.2 affichage.h File Reference

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

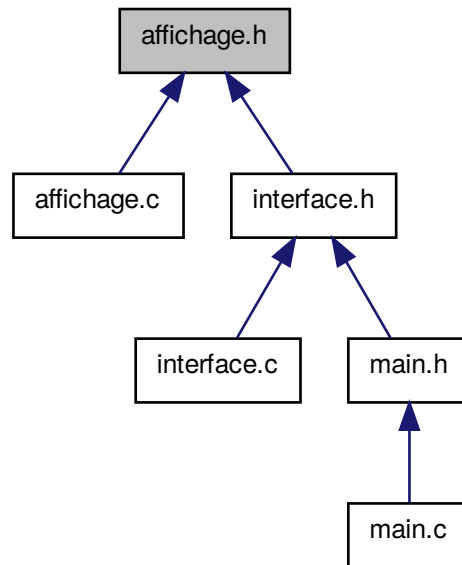
```
#include "../System/structure.h"
```

```
#include <string.h>
```

Include dependency graph for `affichage.h`:



This graph shows which files directly or indirectly include this file:



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](#) ()

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

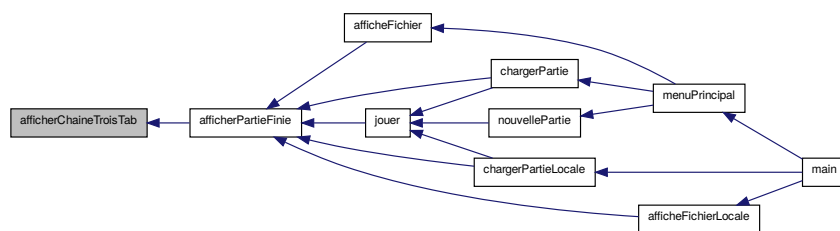
4.2.2.1 void afficherChaineTroisTab (char * *chaine*)

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

Parameters

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

Here is the caller graph for this function:



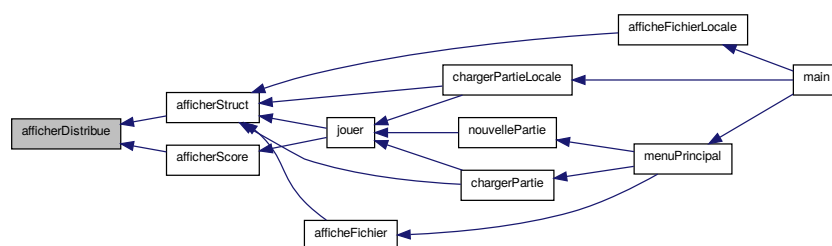
4.2.2.2 void afficherDistribue (Fichier_Jeu * *ptr_struct_fichier*)

Affiche la personne devant distribuer

Parameters

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

Here is the caller graph for this function:



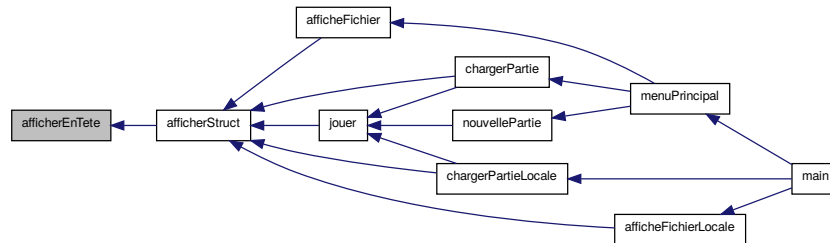
4.2.2.3 void afficherEnTete (Fichier_Jeu * *ptr_struct_fichier*)

Affiche l'en tete de la structure

Parameters

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

Here is the caller graph for this function:



4.2.2.4 void afficherLicense ()

Affiche la license

Here is the caller graph for this function:



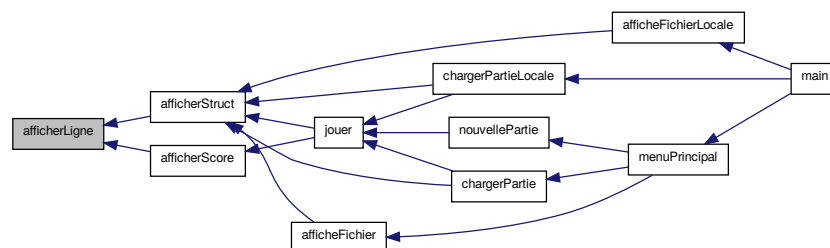
4.2.2.5 void afficherLigne (int taille_ligne)

Affiche taille_ligne - apres une tabulation

Parameters

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

Here is the caller graph for this function:



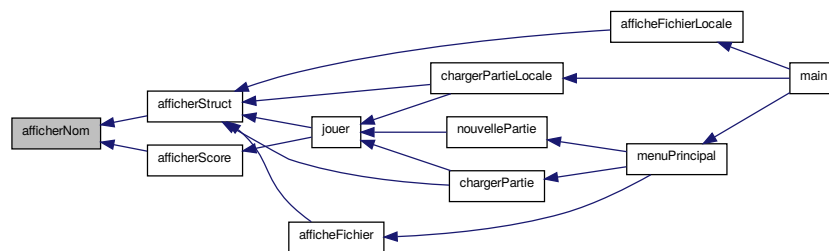
4.2.2.6 void afficherNom (Fichier_Jeu * ptr_struct_fichier, int * ptr_taille_ligne)

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

Parameters

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

Here is the caller graph for this function:



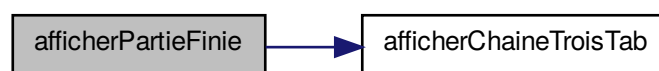
4.2.2.7 void afficherPartieFinie (Fichier_Jeu * ptr_struct_fichier)

Affiche un podium des resultats

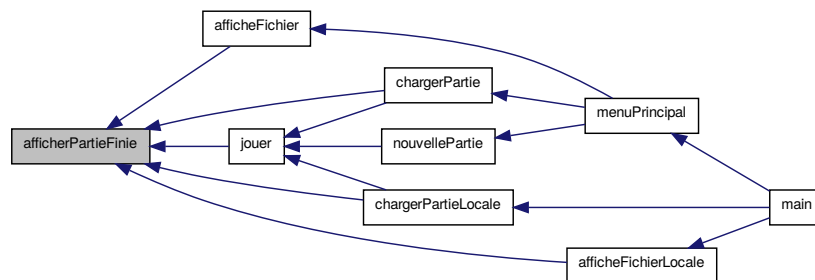
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



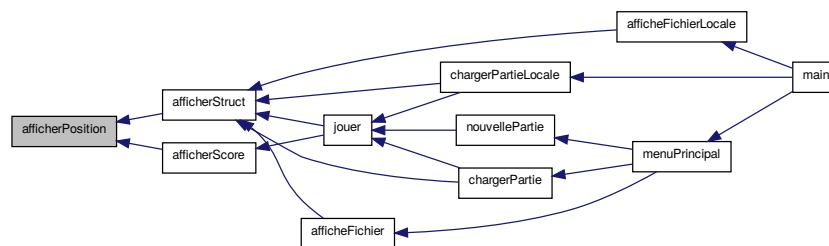
4.2.2.8 void afficherPosition (Fichier_Jeu * ptr_struct_fichier)

Affiche les positions des joueurs

Parameters

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

Here is the caller graph for this function:



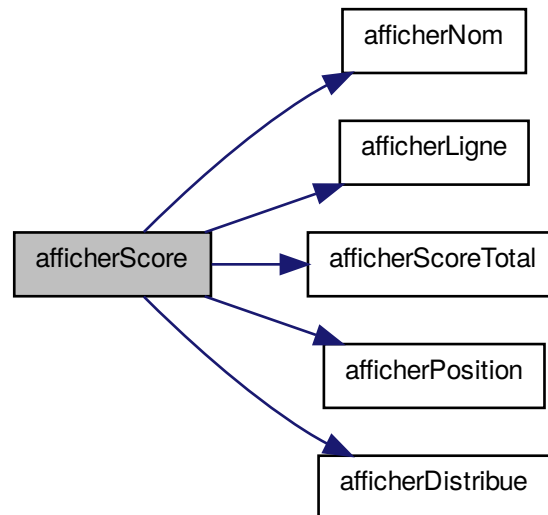
4.2.2.9 void afficherScore (Fichier_Jeu * ptr_struct_fichier)

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

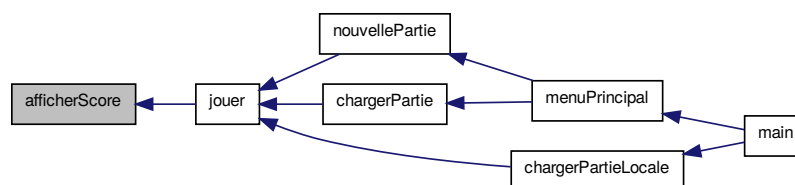
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



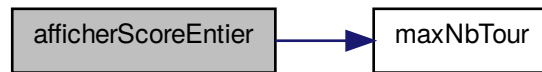
4.2.2.10 void afficherScoreEntier (Fichier_Jeu * ptr_struct_fichier)

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

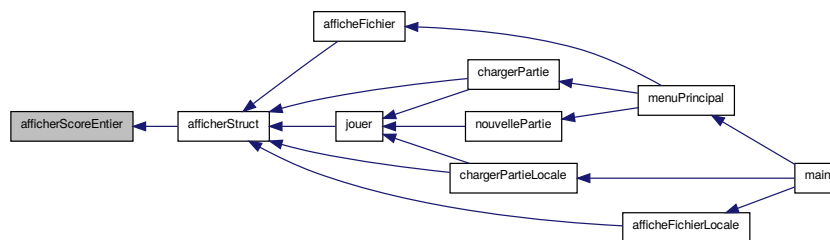
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



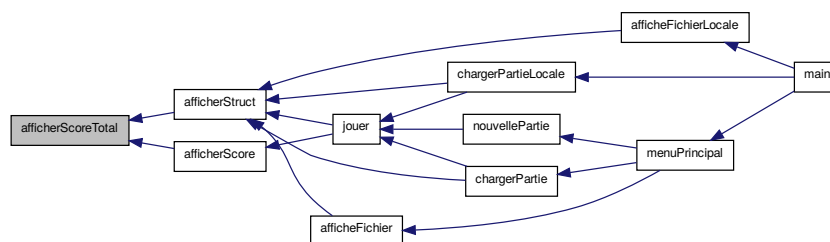
4.2.2.11 void afficherScoreTotal (Fichier_Jeu * ptr_struct_fichier)

Affiche le score total des joueurs

Parameters

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

Here is the caller graph for this function:



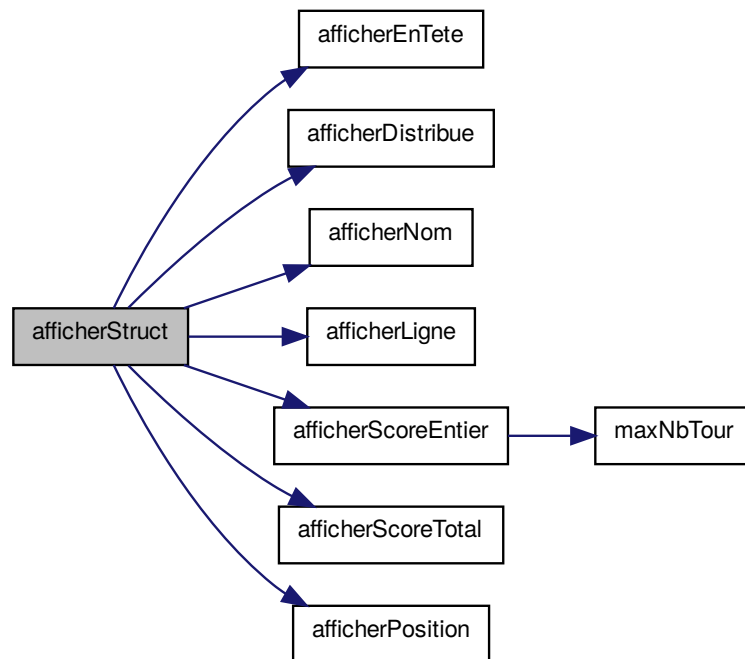
4.2.2.12 void afficherStruct (Fichier_Jeu * ptr_struct_fichier)

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

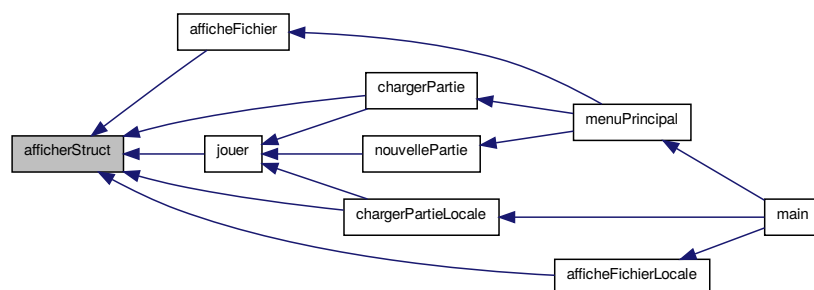
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:

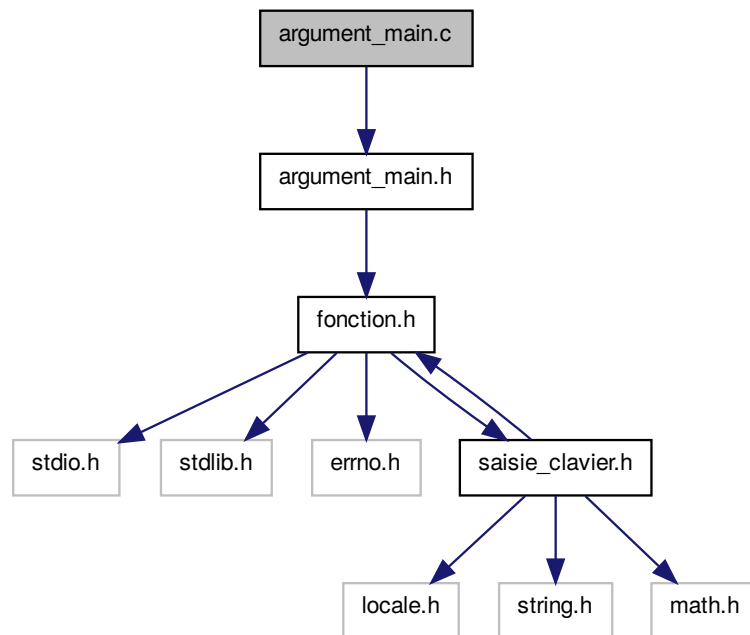


4.3 argument_main.c File Reference

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

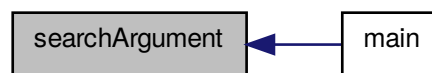
Parameters

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

Here is the caller graph for this function:

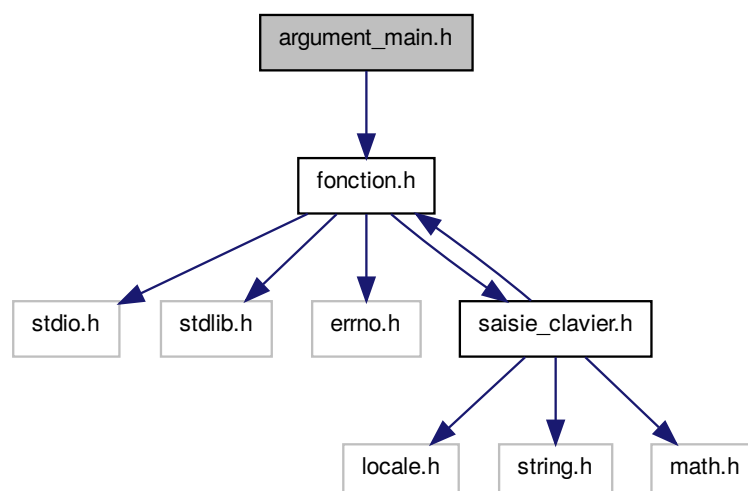


4.4 argument_main.h File Reference

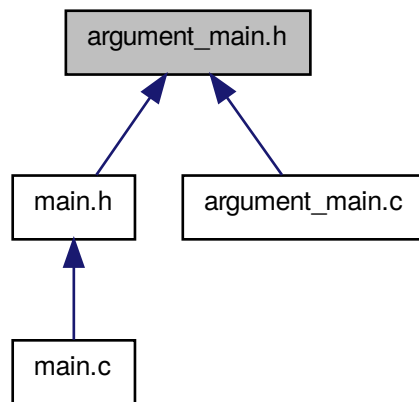
Lancement du programme.

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

Include dependency graph for argument_main.h:



This graph shows which files directly or indirectly include this file:



Macros

- `#define CHAINE_Lecture_FICHIER "--read"`
- `#define CHAINE_Lecture_FICHIER_RED "-r"`
- `#define LECTURE_FICHIER 0`
- `#define CHAINE_OUVERTURE_FICHIER "--open"`
- `#define CHAINE_OUVERTURE_FICHIER_RED "-o"`
- `#define OUVERTURE_FICHIER 1`

Functions

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

4.4.1 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"`

Définit l'appel à la lecture du fichier à "--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

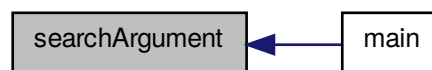
Parameters

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

Here is the caller graph for this function:

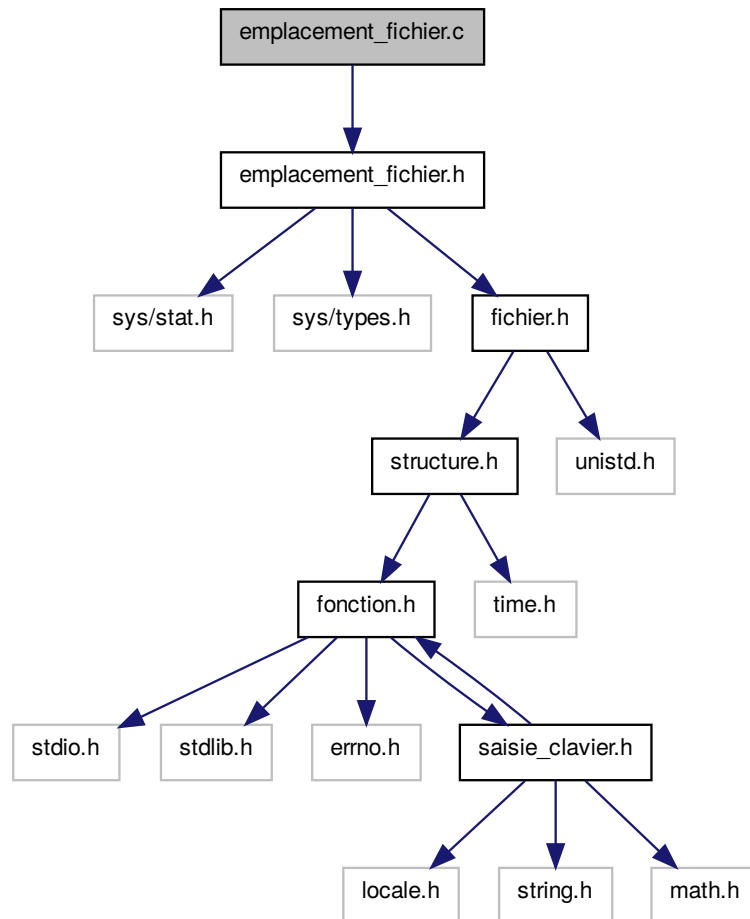


4.5 emplacement_fichier.c File Reference

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)

4.5.1 Detailed Description

Fonctions qui l'emplacement des fichiers sauvegardes.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.5.2 Function Documentation

4.5.2.1 int changerCheminFichier (char * *nouveauChemin*)

Changer le chemin de sauvegarde des fichier par nouveauChemin

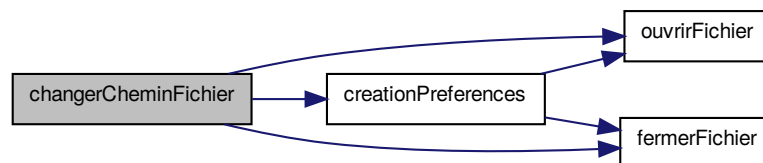
Parameters

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

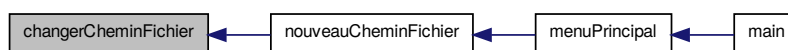
Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



Here is the caller graph for this function:



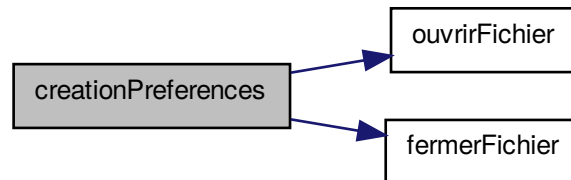
4.5.2.2 void creationPreferences ()

Cree le dossier et le fichier qui va contenir les preferences

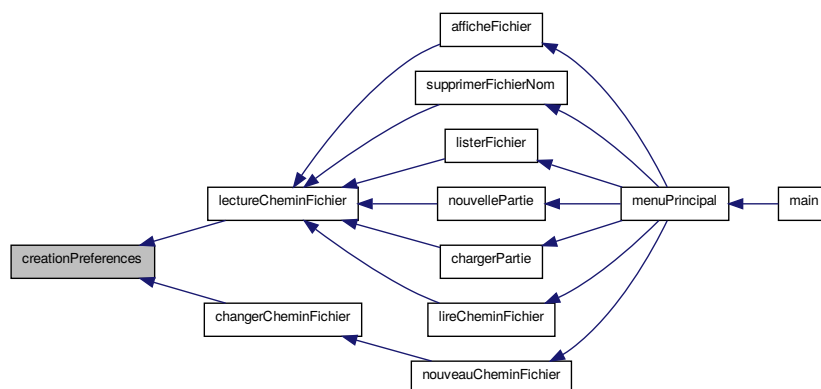
Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



Here is the caller graph for this function:



4.5.2.3 int lectureCheminFichier (char * nom_fichier)

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

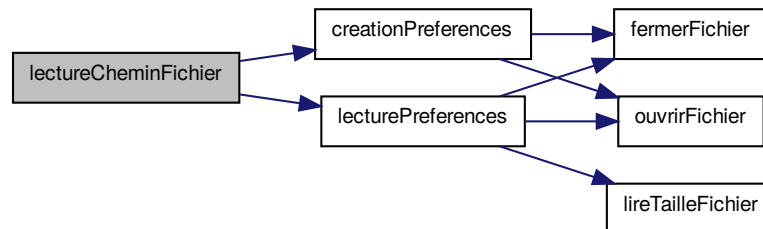
Parameters

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

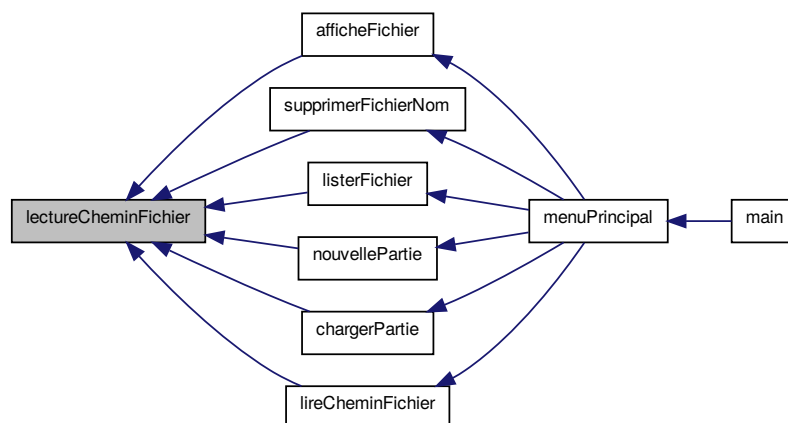
Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



Here is the caller graph for this function:



4.5.2.4 int lecturePreferences (char * nom_fichier)

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

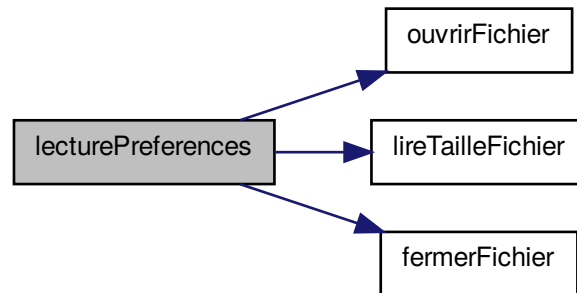
Parameters

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

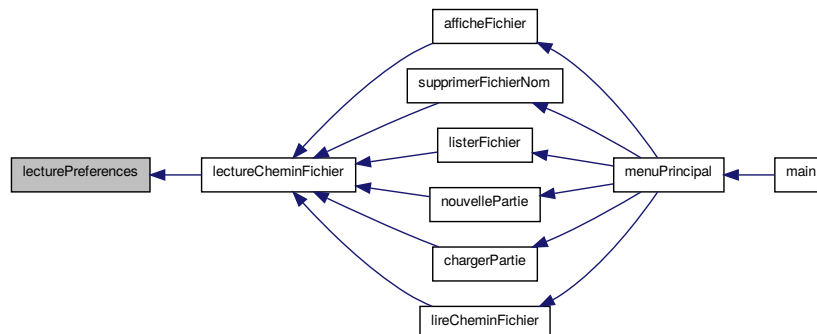
Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



Here is the caller graph for this function:



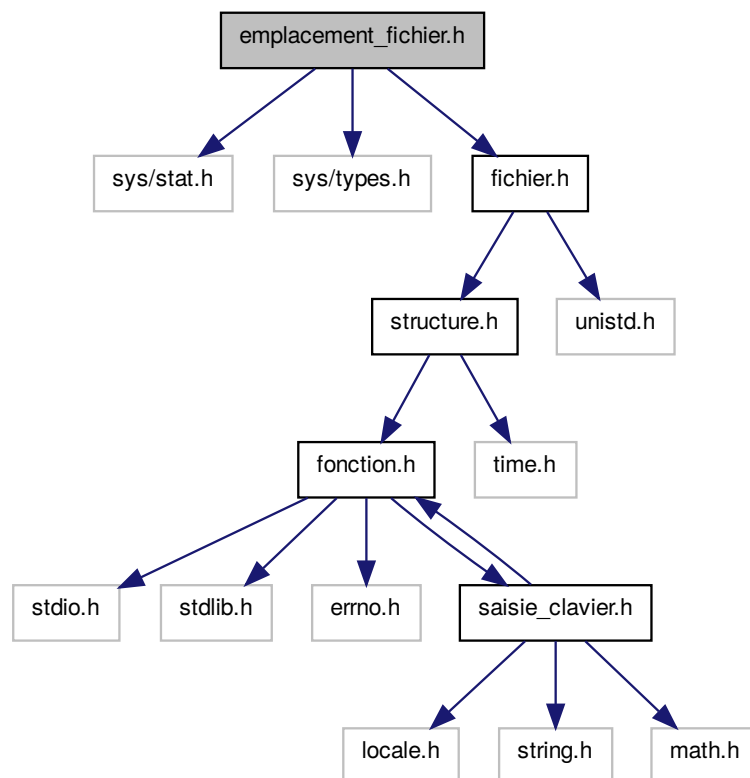
4.6 emplacement_fichier.h File Reference

Prototypes des fonctions qui l'emplacement des fichiers sauvegardes.

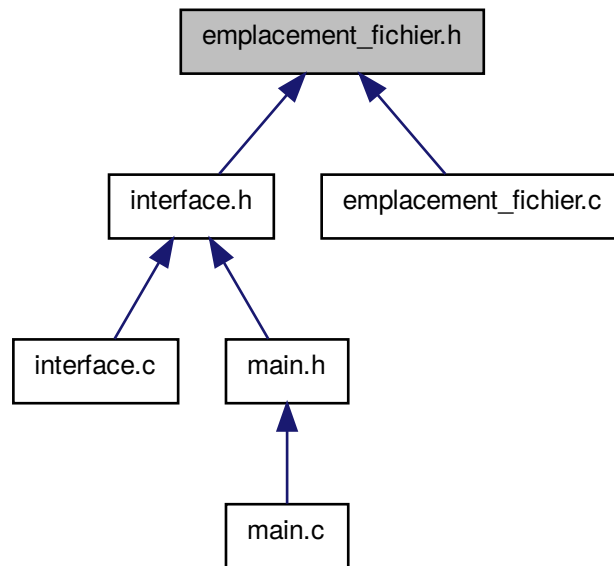
```

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

Include dependency graph for emplacement_fichier.h:



This graph shows which files directly or indirectly include this file:



Macros

- `#define` [PORTABLE](#)
- `#define` [NOM_FICHIER](#) "preferences.txt"
- `#define` [NOM_DOSSIER](#) ".csuper"

Functions

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

4.6.1 Detailed Description

Prototypes des fonctions qui l'emplacement des fichiers sauvegardes.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.6.2 Macro Definition Documentation

4.6.2.1 #define NOM_DOSSIER ".csuper"

Definit NOM_DOSSIER a ".csuper"

4.6.2.2 #define NOM_FICHIER "preferences.txt"

Definit NOM_FICHIER a "preferences.txt"

4.6.2.3 #define PORTABLE

Definit PORTABLE

4.6.3 Function Documentation

4.6.3.1 int changerCheminFichier (char * *nouveauChemin*)

Changer le chemin de sauvegarde des fichier par nouveauChemin

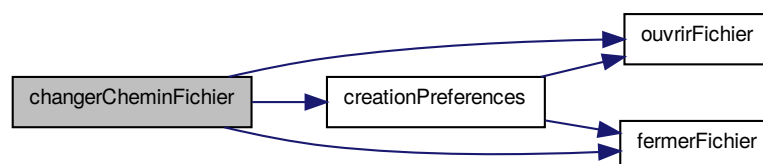
Parameters

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

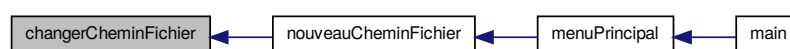
Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



Here is the caller graph for this function:



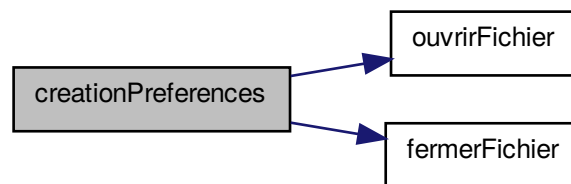
4.6.3.2 int creationPreferences ()

Cree le dossier et le fichier qui va contenir les preferences

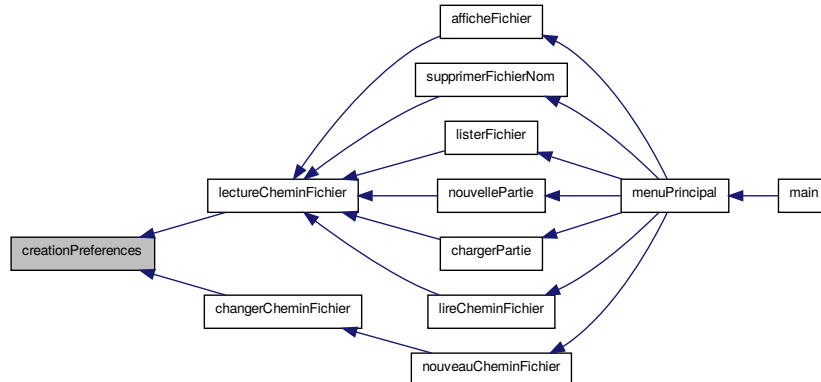
Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



Here is the caller graph for this function:



4.6.3.3 int lectureCheminFichier (char * nom_fichier)

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

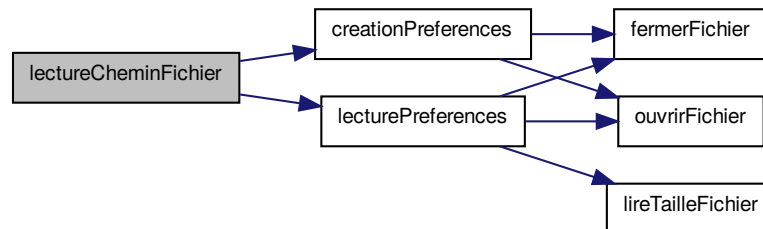
Parameters

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

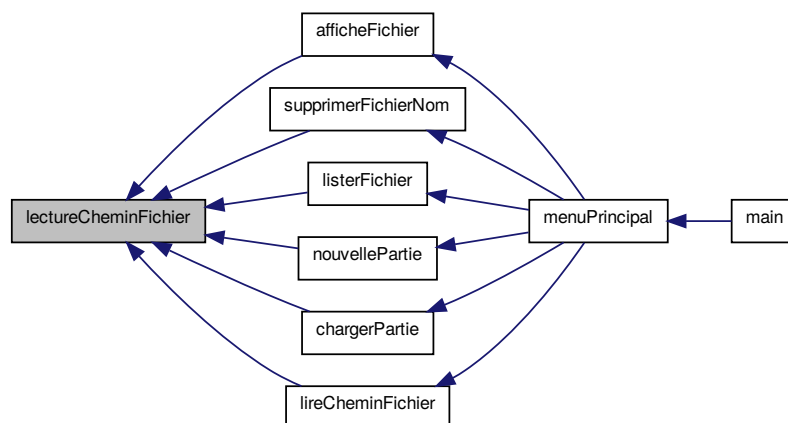
Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



Here is the caller graph for this function:



4.6.3.4 int lecturePreferences (char * nom_fichier)

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

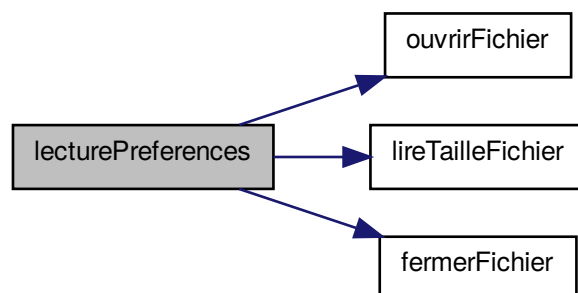
Parameters

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

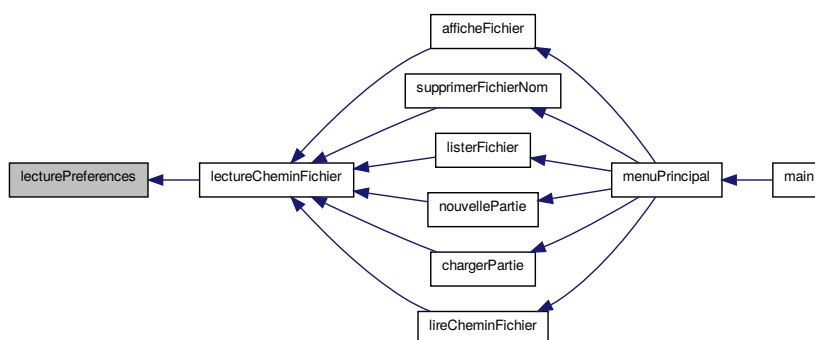
Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



Here is the caller graph for this function:

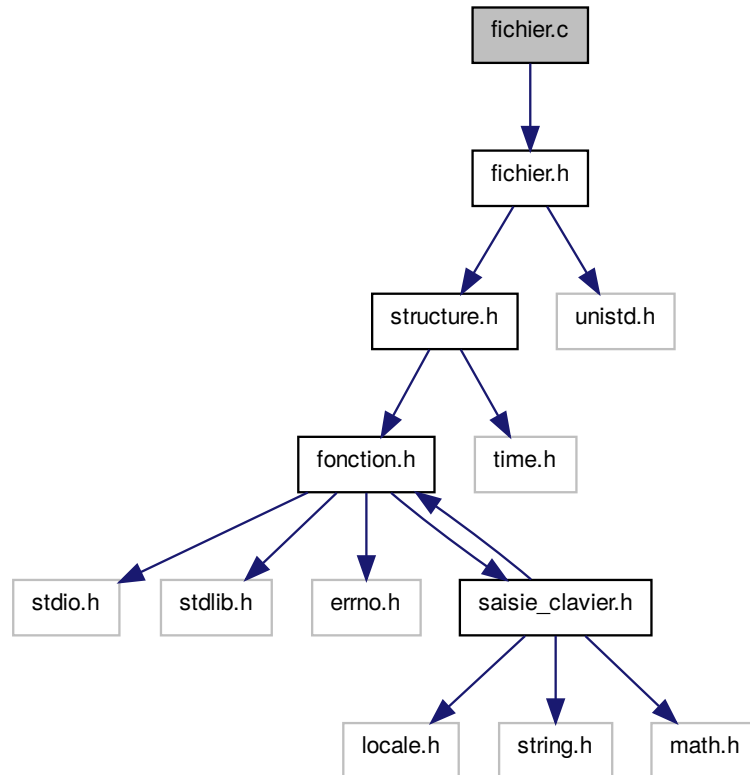


4.7 fichier.c File Reference

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)`

4.7.1 Detailed Description

Fonction de gestion des fichiers.

Author

Remi BERTHO

Date

09/03/14

Version

2.1.0

4.7.2 Function Documentation

4.7.2.1 int ecrireFichier (char * nom, Fichier_Jeu * ptr_struct_fichier)

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

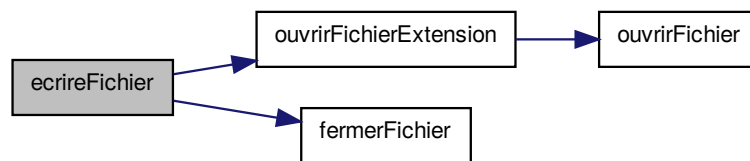
Parameters

in	*nom	le nom du fichier
in	*ptr_struct_fichier	la structure du fichier avec lequel on veut créer le fichier

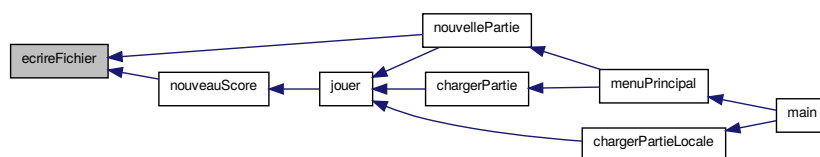
Returns

VRAI si tout s'est bien passé, FAUX sinon

Here is the call graph for this function:



Here is the caller graph for this function:



4.7.2.2 Fichier_Jeu * lireFichier (char * nom)

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

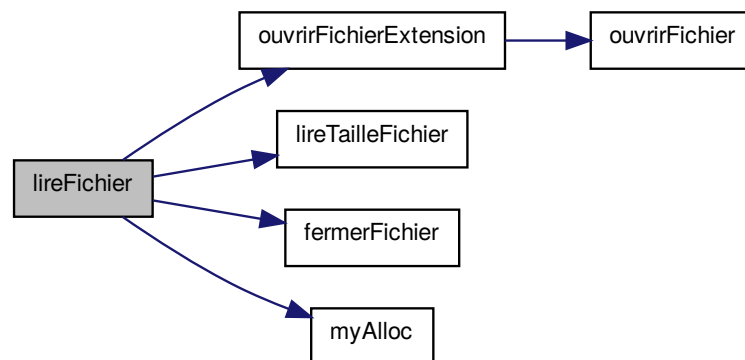
Parameters

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

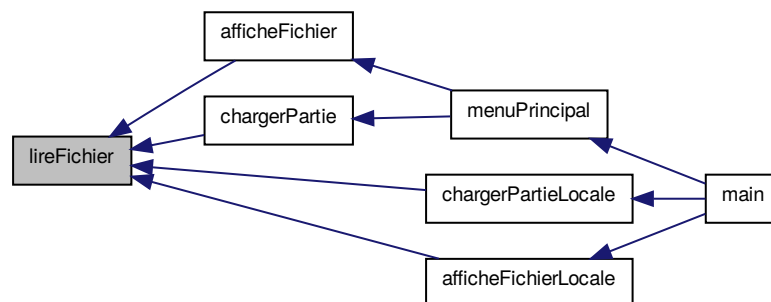
Returns

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

Here is the call graph for this function:



Here is the caller graph for this function:



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

Mets a jour le fichier avec les nouveaux scores

Parameters

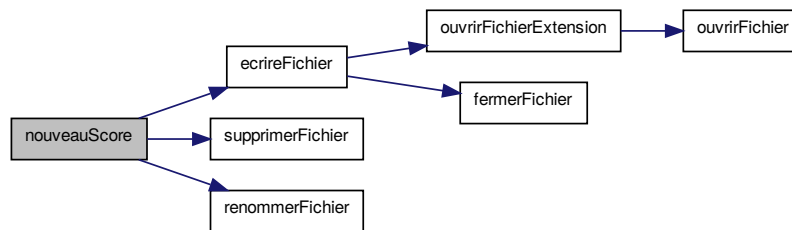
<code>in</code>	<code>*nom</code>	le nom du fichier
-----------------	-------------------	-------------------

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

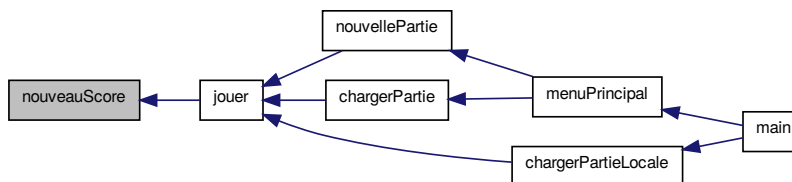
Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



Here is the caller graph for this function:



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

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

Parameters

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

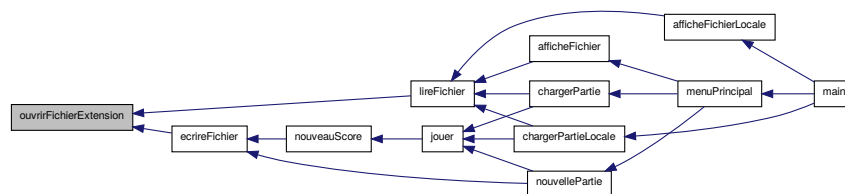
Returns

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

Here is the call graph for this function:



Here is the caller graph for this function:



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

Renomme le fichier dont le nom est en parametre

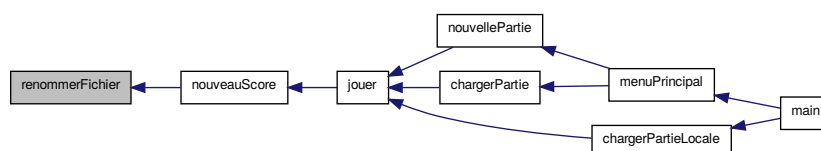
Parameters

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

Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the caller graph for this function:



4.7.2.6 int supprimerFichier (char * *nom*)

Supprime le fichier dont le nom est en parametre

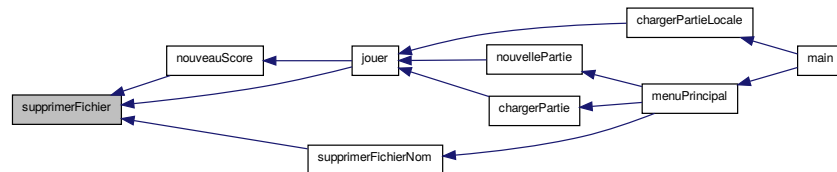
Parameters

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

Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the caller graph for this function:



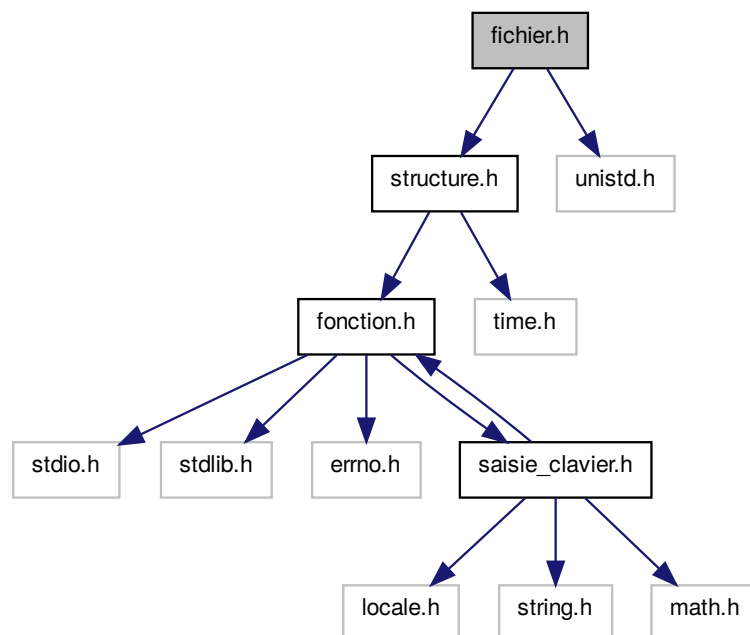
4.8 fichier.h File Reference

Prototypes des fonction de gestion des fichiers.

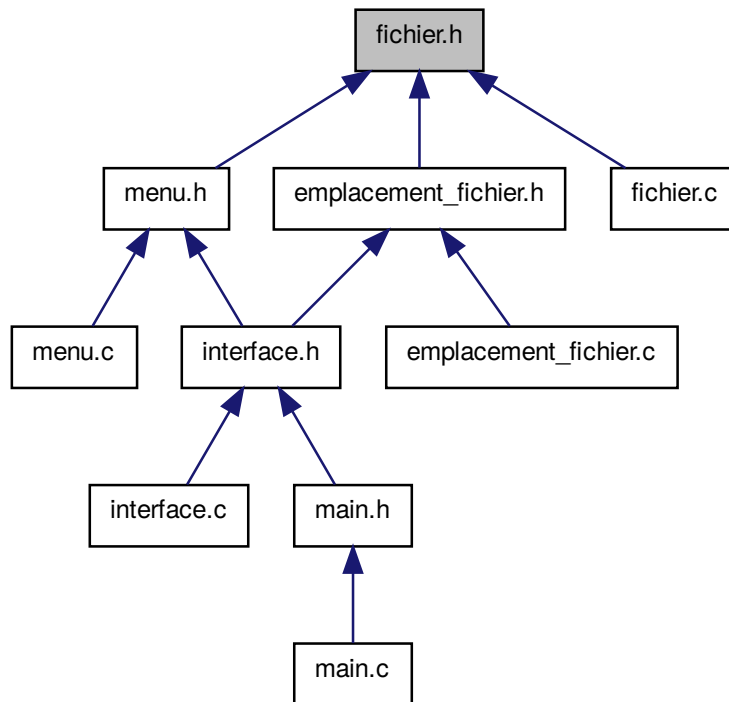
```
#include "structure.h"
```

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

Include dependency graph for fichier.h:



This graph shows which files directly or indirectly include this file:



Macros

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

Functions

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

4.8.1 Detailed Description

Prototypes des fonction de gestion des fichiers.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.8.2 Macro Definition Documentation

4.8.2.1 #define EXTENSION_FICHIER "csu"

Definit l'extension du fichier a "csu"

4.8.2.2 #define TAILLE_MAX_NOM_FICHIER 250

Definit la taille max d'un nom a 250

4.8.2.3 #define TYPE_FICHIER "CompteurScoreUniversel"

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

4.8.3 Function Documentation

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

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

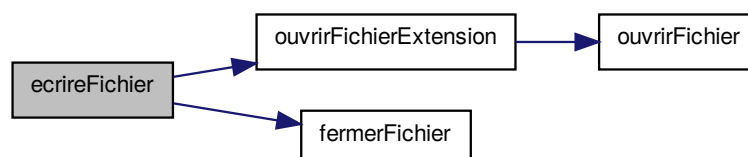
Parameters

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

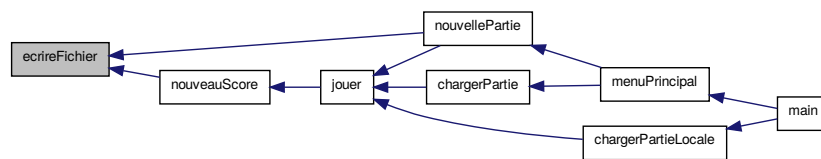
Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



Here is the caller graph for this function:



4.8.3.2 Fichier_Jeu* lireFichier (char * nom)

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

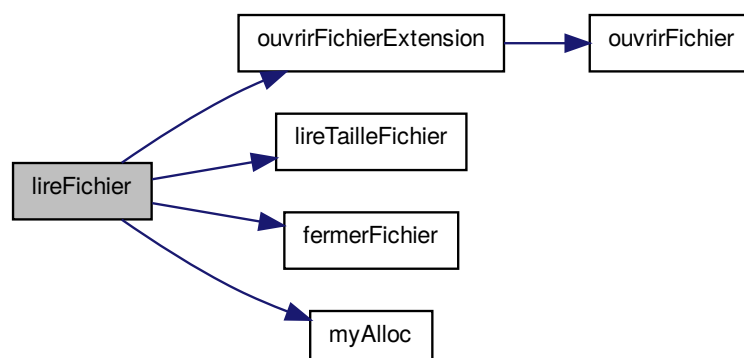
Parameters

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

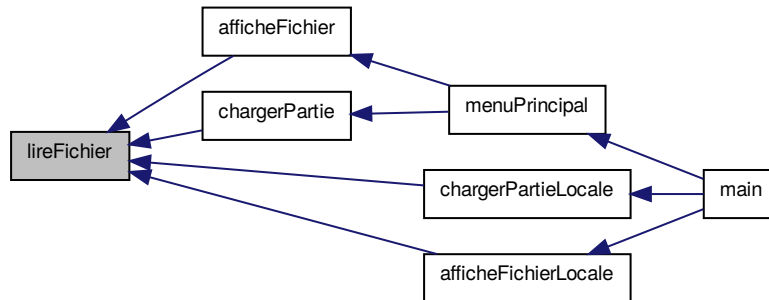
Returns

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

Here is the call graph for this function:



Here is the caller graph for this function:



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

Mets a jour le fichier avec les nouveaux scores

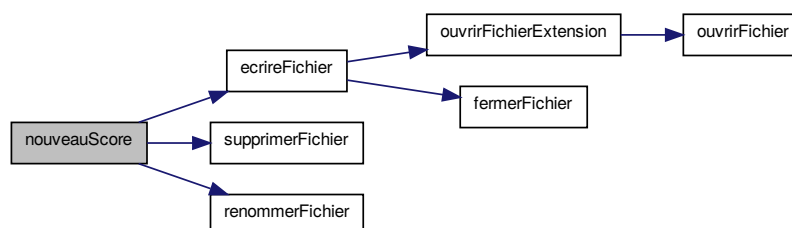
Parameters

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

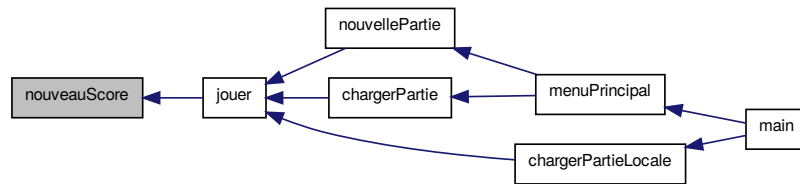
Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



Here is the caller graph for this function:



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

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

Parameters

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

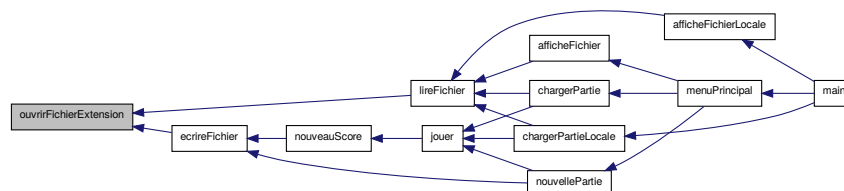
Returns

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

Here is the call graph for this function:



Here is the caller graph for this function:



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

Renomme le fichier dont le nom est en parametre

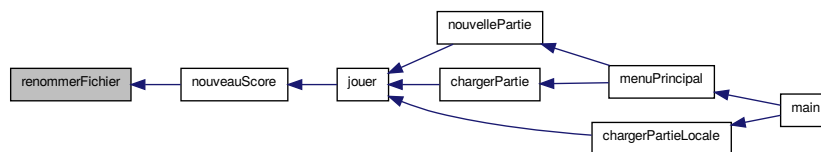
Parameters

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

Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the caller graph for this function:

**4.8.3.6 int supprimerFichier (char * nom)**

Supprime le fichier dont le nom est en parametre

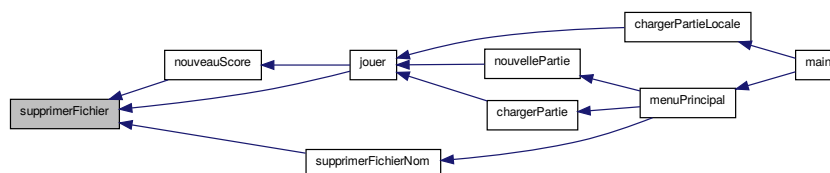
Parameters

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

Returns

VRAI si tout s'est bien passe, FAUX sinon

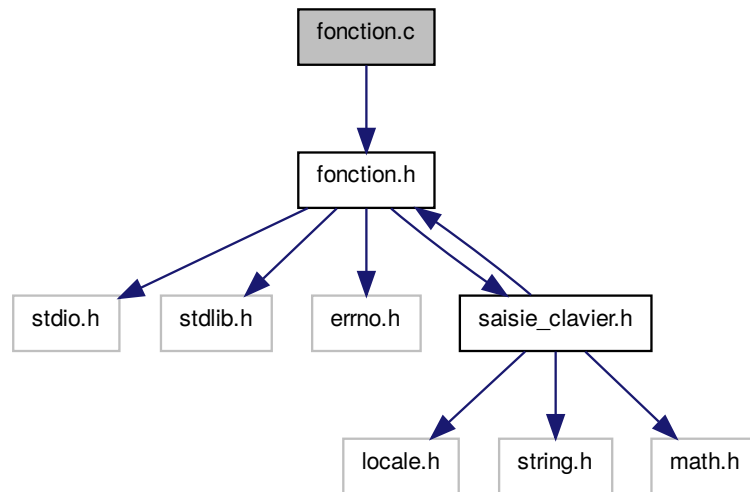
Here is the caller graph for this function:

**4.9 fonction.c File Reference**

Fonctions essentielles au fonctionnement du programme.

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

Include dependency graph for fonction.c:



Functions

- void `mauvais_choix` ()
- void `systemPause` ()
- 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)

4.9.1 Detailed Description

Fonctions essentielles au fonctionnement du programme.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.9.2 Function Documentation

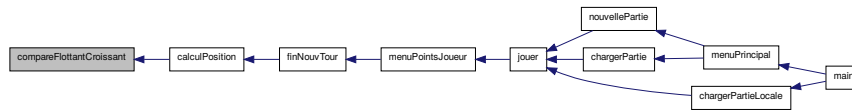
4.9.2.1 `int compareFlottantCroissant (void const * a, void const * b)`

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

Here is the caller graph for this function:

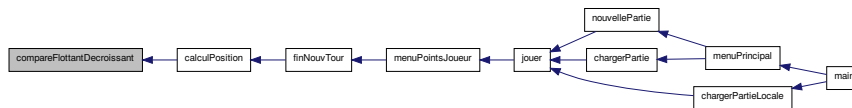
4.9.2.2 int int compareFlottantDecroissant (void const * *a*, void const * *b*)

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

Here is the caller graph for this function:

4.9.2.3 int fermerFichier (FILE * *ptr_fichier*)

Ferme le fichier

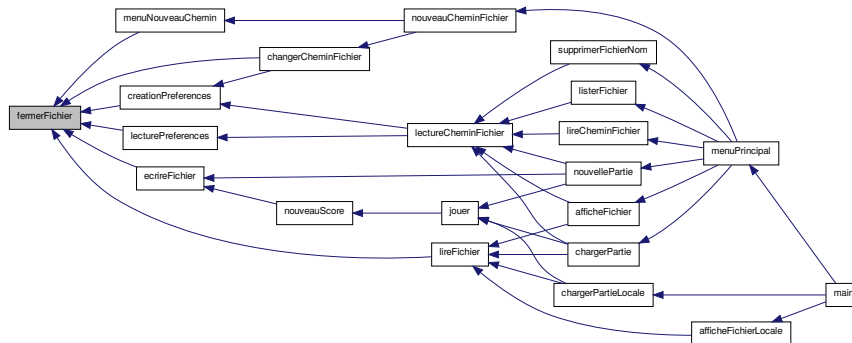
Parameters

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

Returns

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

Here is the caller graph for this function:



4.9.2.4 int lireTailleFichier (FILE * ptr_fichier)

Lis la taille du fichier

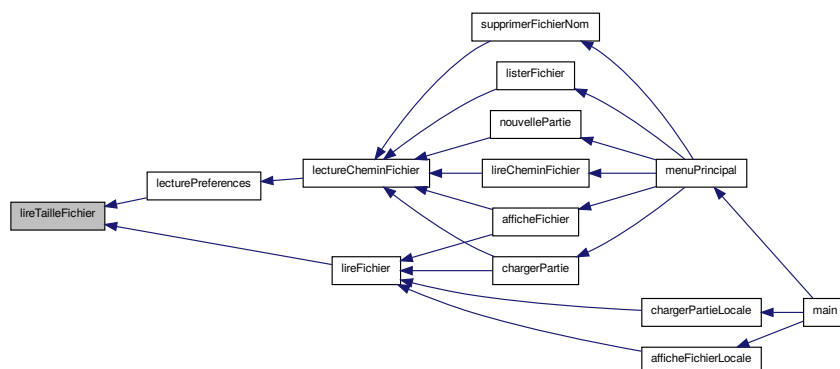
Parameters

in	*ptr_fichier	le fichier
----	--------------	------------

Returns

entier ayant la taille du fichier

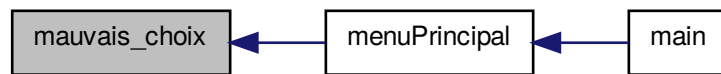
Here is the caller graph for this function:



4.9.2.5 void mauvais_choix ()

Affiche un message d'erreur.

Here is the caller graph for this function:



4.9.2.6 void * myAlloc (int *taille_alloue*)

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

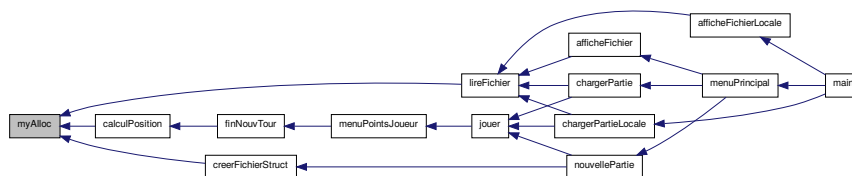
Parameters

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

Returns

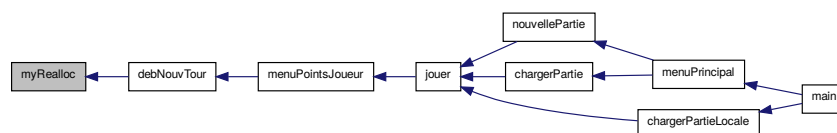
un pointeur sur la structure alloué

Here is the caller graph for this function:



4.9.2.7 void myRealloc (void ** *ptr*, int *taille_alloue*)

Here is the caller graph for this function:



4.9.2.8 FILE * ouvrirFichier (char *nom*[], char *mode*[])

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

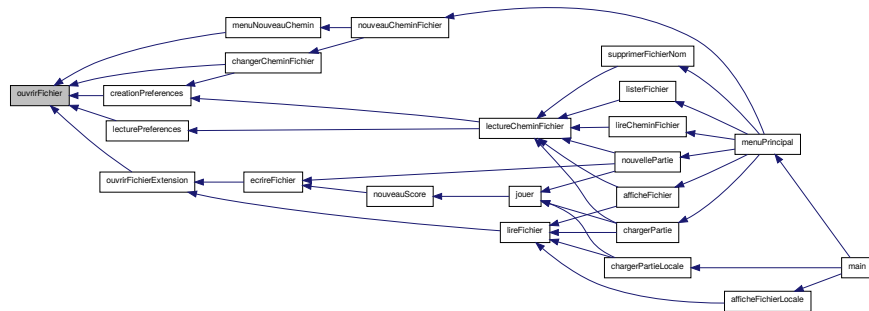
Parameters

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

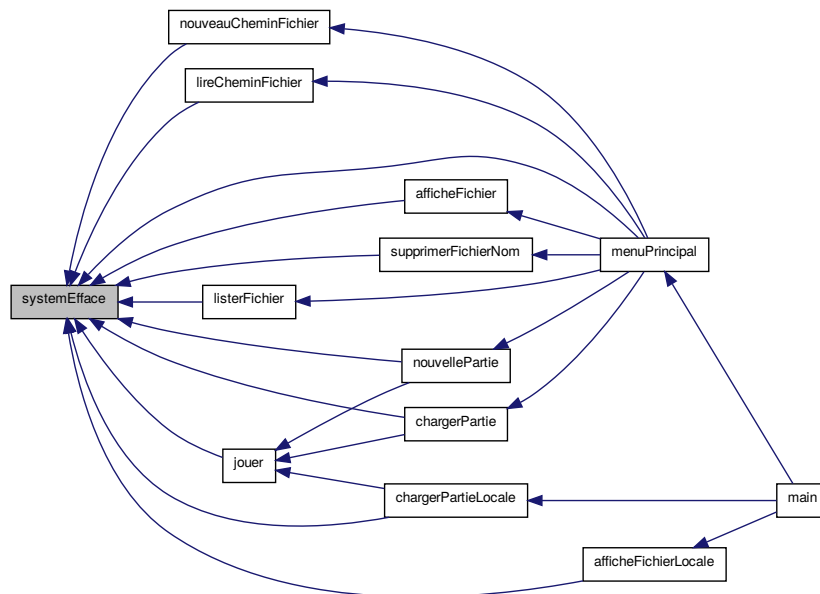
Here is the caller graph for this function:



4.9.2.9 void systemEfface ()

Efface la console de l'utilisateur.

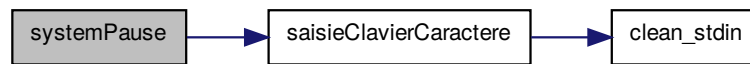
Here is the caller graph for this function:



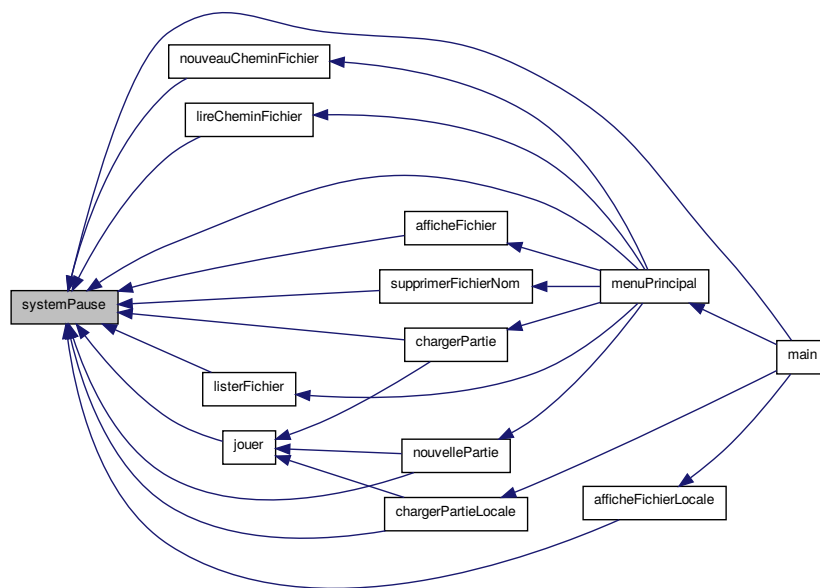
4.9.2.10 void systemPause ()

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

Here is the call graph for this function:



Here is the caller graph for this function:

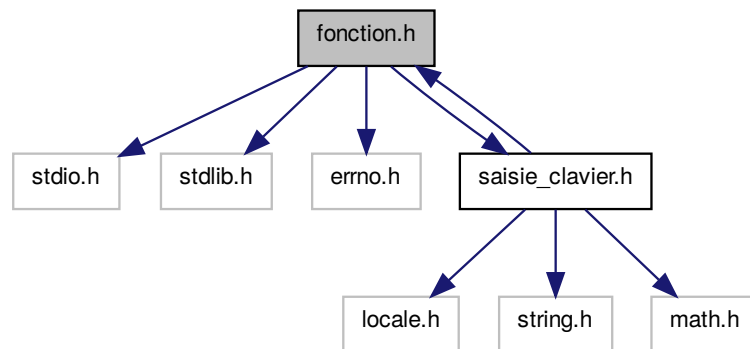


4.10 fonction.h File Reference

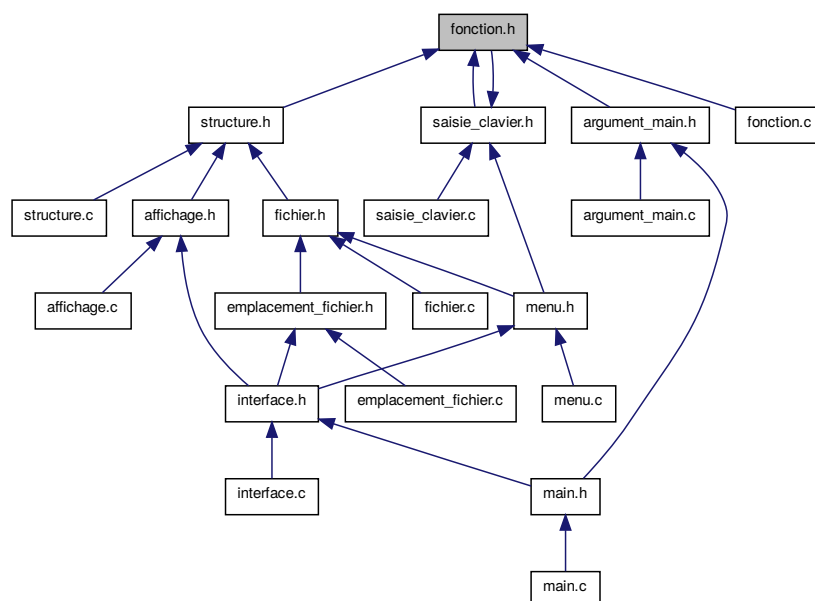
Prototypes des fonctions essentielles au fonctionnement du programme.

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include "saisie_clavier.h"
```

Include dependency graph for fonction.h:



This graph shows which files directly or indirectly include this file:



Macros

- `#define` [VRAI](#) 1
- `#define` [FAUX](#) 0

Functions

- void [mauvais_choix](#) ()
- void [systemPause](#) ()

- 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)

4.10.1 Detailed Description

Prototypes des fonctions essentielles au fonctionnement du programme.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.10.2 Macro Definition Documentation

4.10.2.1 #define FAUX 0

Definit FAUX a 0

4.10.2.2 #define VRAI 1

Definit VRAI a 1

4.10.3 Function Documentation

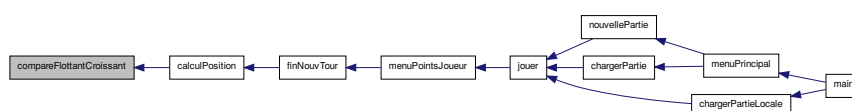
4.10.3.1 int compareFlottantCroissant (void const * a, void const * b)

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

Here is the caller graph for this function:



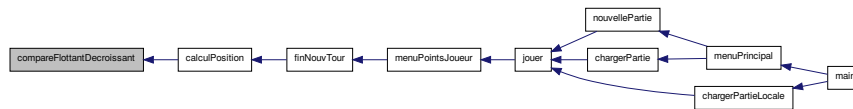
4.10.3.2 `int compareFlottantDecroissant (void const * a, void const * b)`

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

Here is the caller graph for this function:

4.10.3.3 int fermerFichier (FILE * *ptr_fichier*)

Ferme le fichier

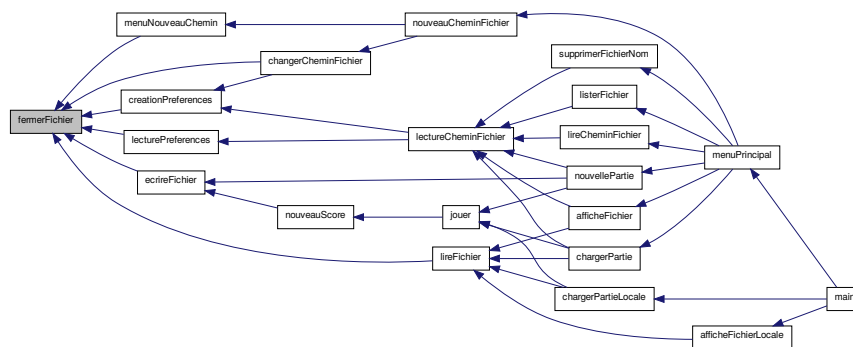
Parameters

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

Returns

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

Here is the caller graph for this function:

4.10.3.4 int lireTailleFichier (FILE * *ptr_fichier*)

Lis la taille du fichier

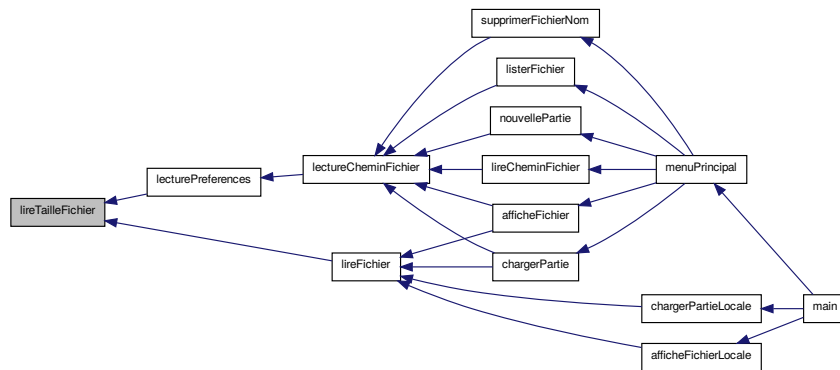
Parameters

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

Returns

entier ayant la taille du fichier

Here is the caller graph for this function:



4.10.3.5 void mauvais_choix ()

Affiche un message d'erreur.

Here is the caller graph for this function:



4.10.3.6 void* myAlloc (int *taille_alloue*)

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

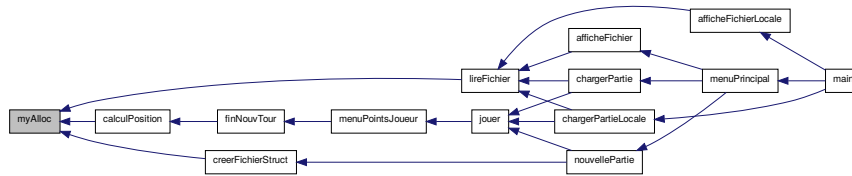
Parameters

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

Returns

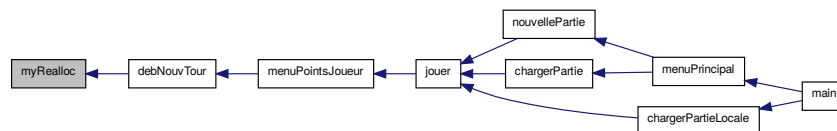
un pointeur sur la structure alloué

Here is the caller graph for this function:



4.10.3.7 void myRealloc (void ** ptr, int taille_alloue)

Here is the caller graph for this function:



4.10.3.8 FILE* ouvrirFichier (char *nom*[], char *mode*[])

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

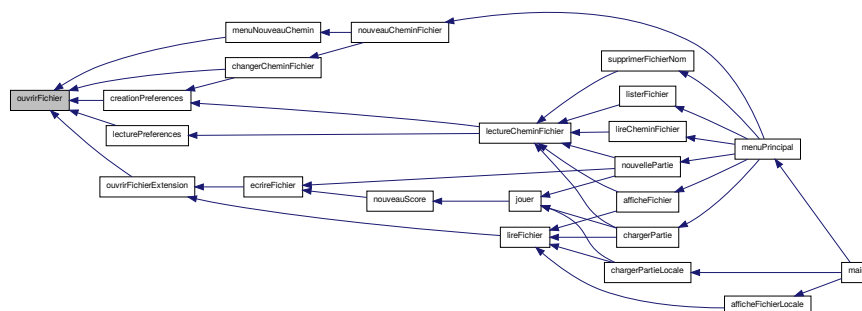
Parameters

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

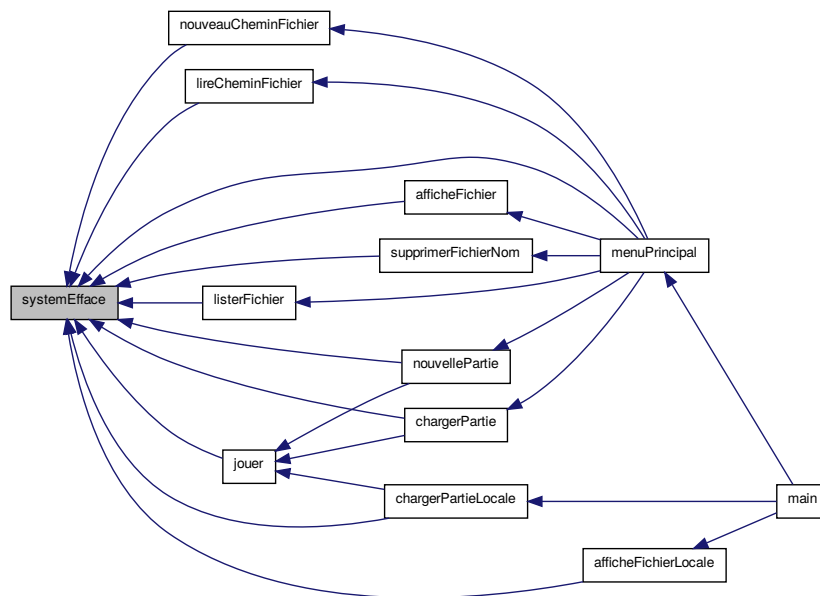
Here is the caller graph for this function:



4.10.3.9 void systemEfface ()

Efface la console de l'utilisateur.

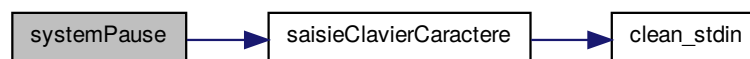
Here is the caller graph for this function:



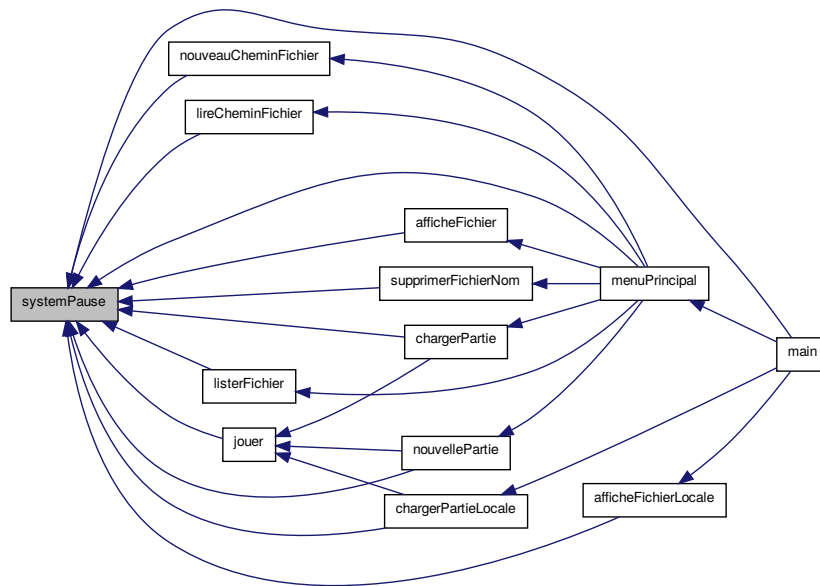
4.10.3.10 void systemPause ()

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

Here is the call graph for this function:



Here is the caller graph for this function:

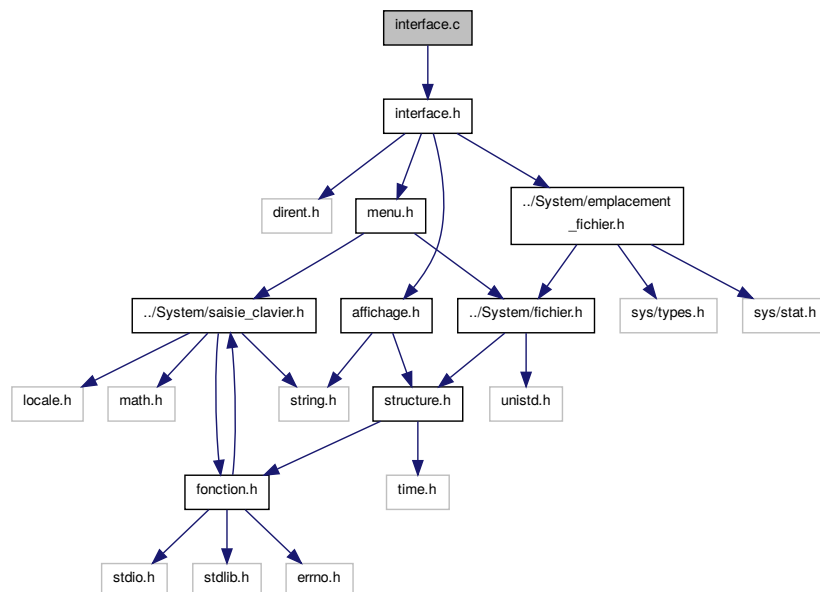


4.11 interface.c File Reference

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 `nouveauCheminFichier` ()
- void `lireCheminFichier` ()
- void `chargerPartieLocale` (char *nom_fichier)
- void `afficheFichierLocale` (char *nom_fichier)

4.11.1 Detailed Description

Fonctions qui gerent l'interface graphique du logiciel.

Author

Remi BERTHO

Date

09/03/14

Version

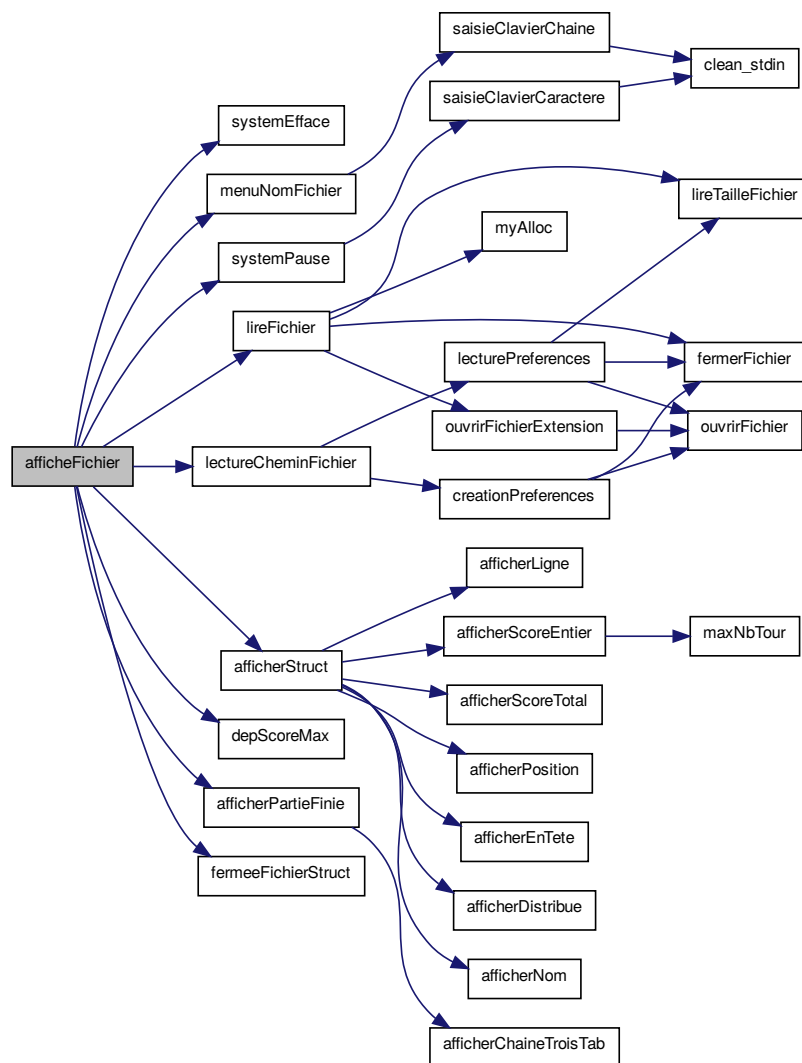
2.1.0

4.11.2 Function Documentation

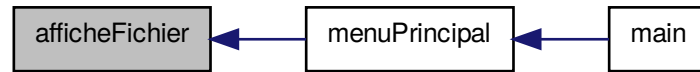
4.11.2.1 void afficheFichier ()

Demande le nom d'un fichier et l'affiche

Here is the call graph for this function:



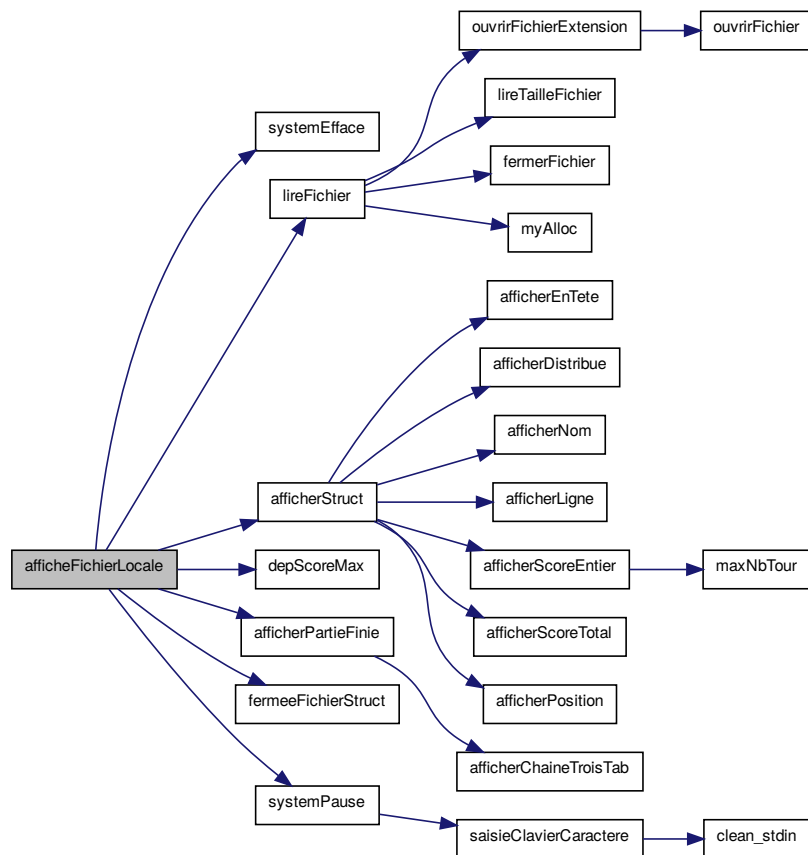
Here is the caller graph for this function:



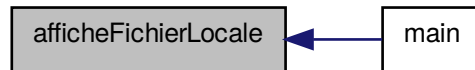
4.11.2.2 void afficheFichierLocale (char * nom_fichier)

Affiche le fichier dont le nom a ete donne en parametre

Here is the call graph for this function:



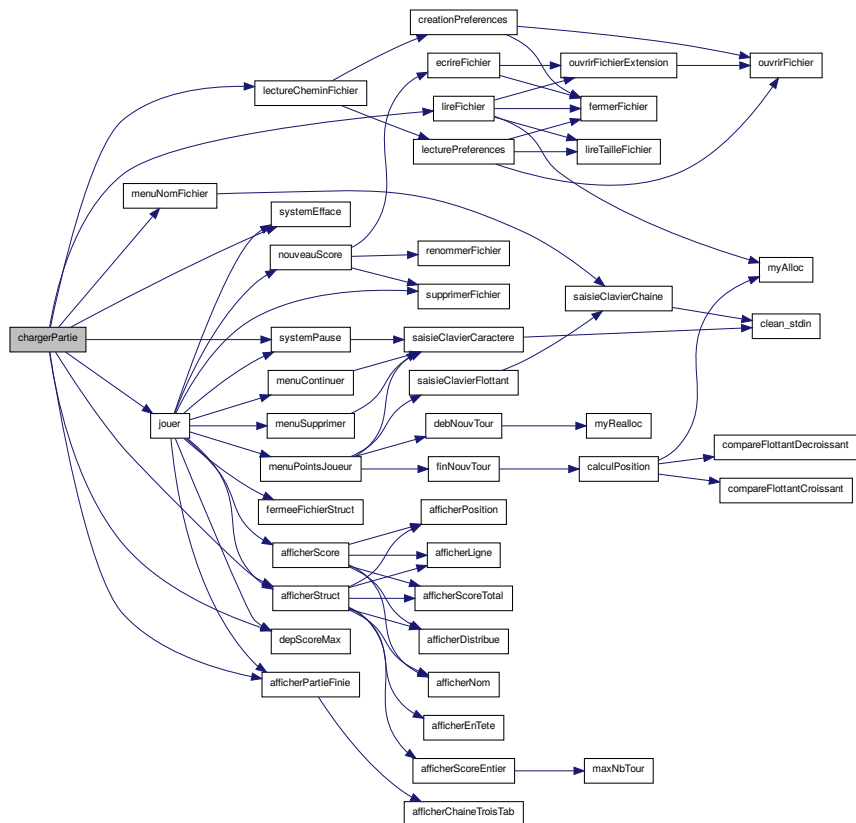
Here is the caller graph for this function:



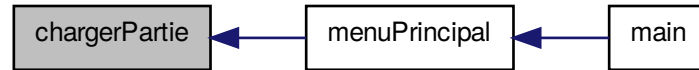
4.11.2.3 void chargerPartie ()

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

Here is the call graph for this function:



Here is the caller graph for this function:



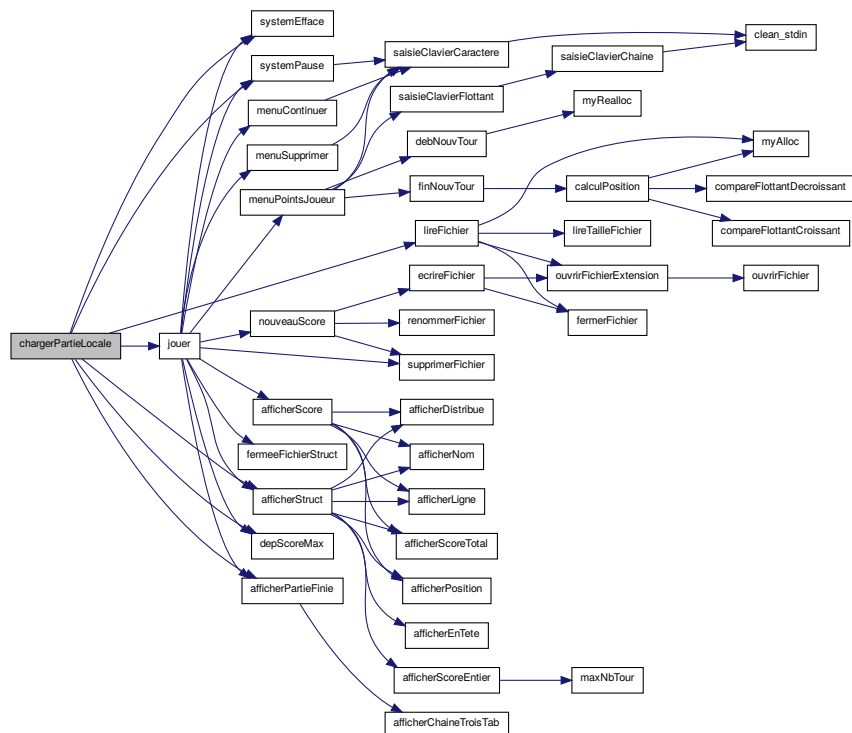
4.11.2.4 void chargerPartieLocale (char * nom_fichier)

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

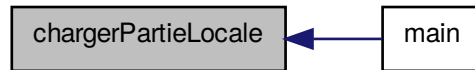
Parameters

in	<i>nom_fichier,le</i>	nom du fichier
----	-----------------------	----------------

Here is the call graph for this function:



Here is the caller graph for this function:



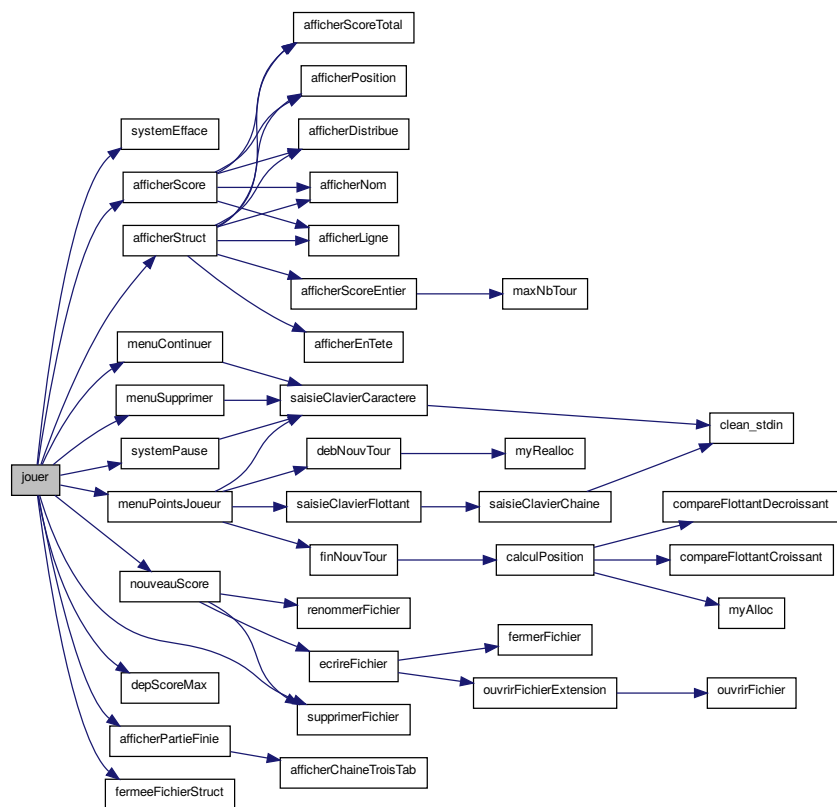
4.11.2.5 void jouer (Fichier_Jeu * ptr_struct_fichier, char * nom_fichier)

Lance la partie de comptage jusqu'a la fin

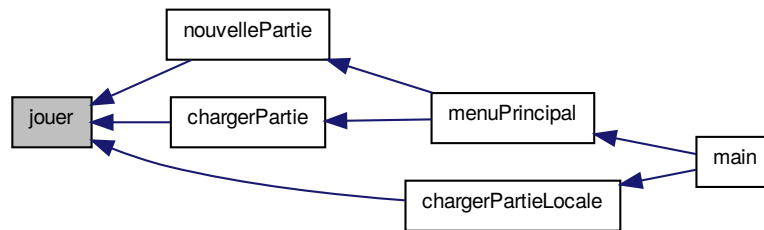
Parameters

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:



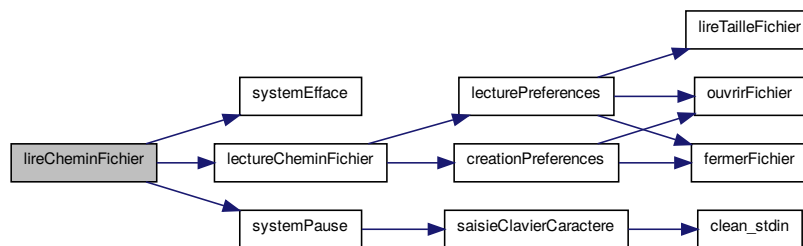
Here is the caller graph for this function:



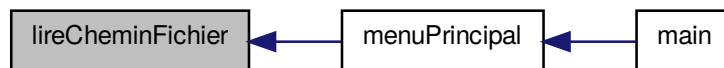
4.11.2.6 void lireCheminFichier ()

Lis le chemin de fichier et l'affiche

Here is the call graph for this function:



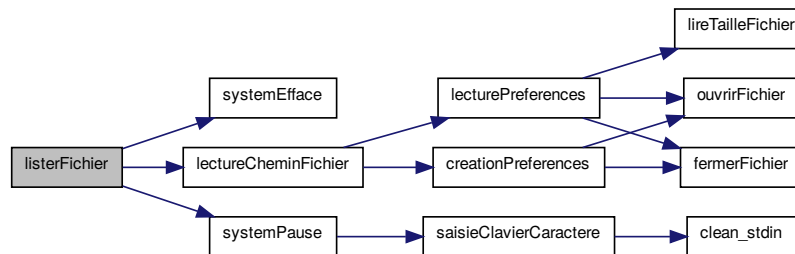
Here is the caller graph for this function:



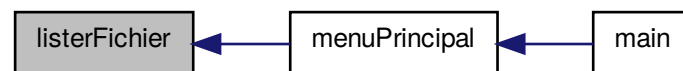
4.11.2.7 void listerFichier ()

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

Here is the call graph for this function:



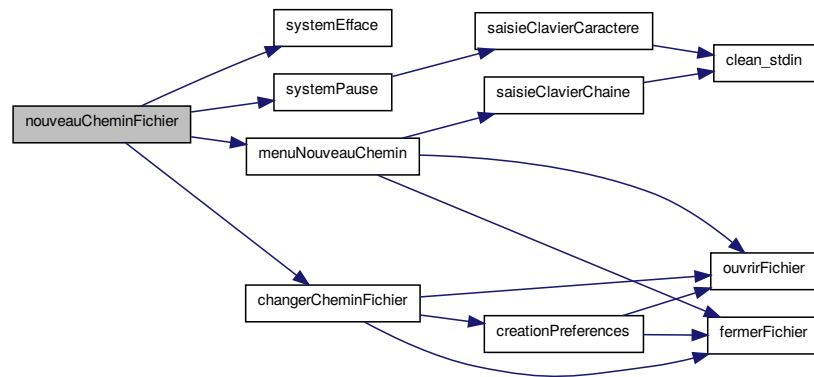
Here is the caller graph for this function:



4.11.2.8 void menuPrincipal ()

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

Here is the call graph for this function:



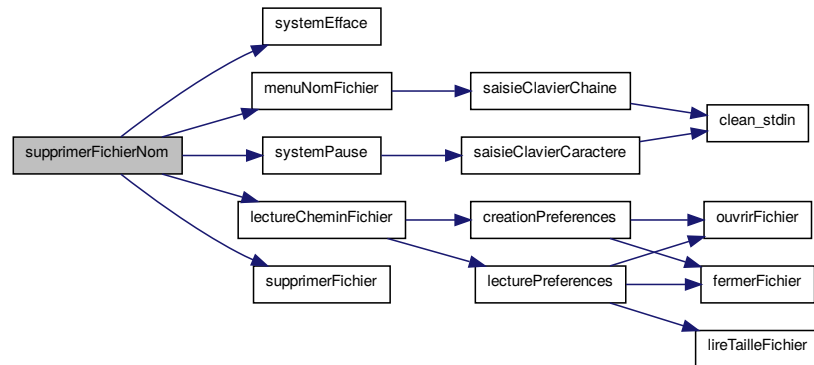
Here is the caller graph for this function:



4.11.2.10 void nouvellePartie ()

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

Here is the call graph for this function:



Here is the caller graph for this function:



4.12 interface.h File Reference

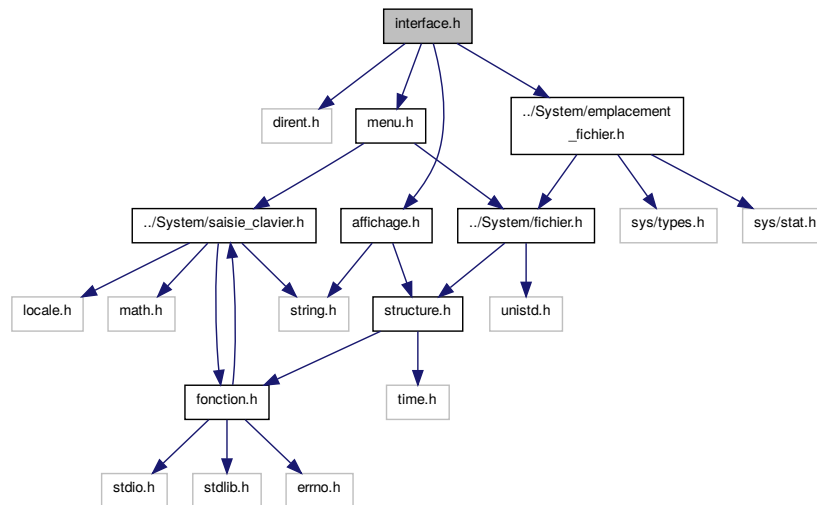
Prototypes des fonctions qui gerent l'interface graphique du logiciel.

```

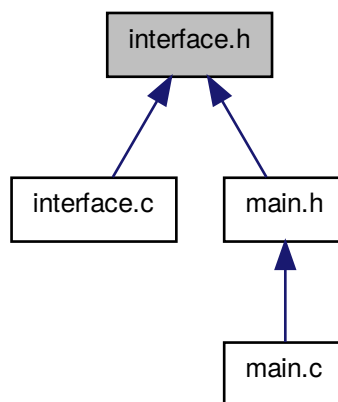
#include <dirent.h>
#include "menu.h"
#include "affichage.h"
#include "../System/emplacement_fichier.h"

```

Include dependency graph for interface.h:



This graph shows which files directly or indirectly include this file:



Enumerations

- enum `Menu` {
`nouvPart` =1, `charPart` =2, `afficheFich` =3, `supprFich` =4,
`listFich` =5, `quit` =6, `easterEggs` = 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 [nouveauCheminFichier](#) ()
- void [lireCheminFichier](#) ()
- void [chargerPartieLocale](#) (char *nom_fichier)
- void [afficheFichierLocale](#) (char *nom_fichier)

4.12.1 Detailed Description

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

Author

Remi BERTHO

Date

09/03/14

Version

2.1.0

4.12.2 Enumeration Type Documentation

4.12.2.1 enum Menu

Enumerator

nouvPart

charPart

affFich

supprFich

listFich

quit

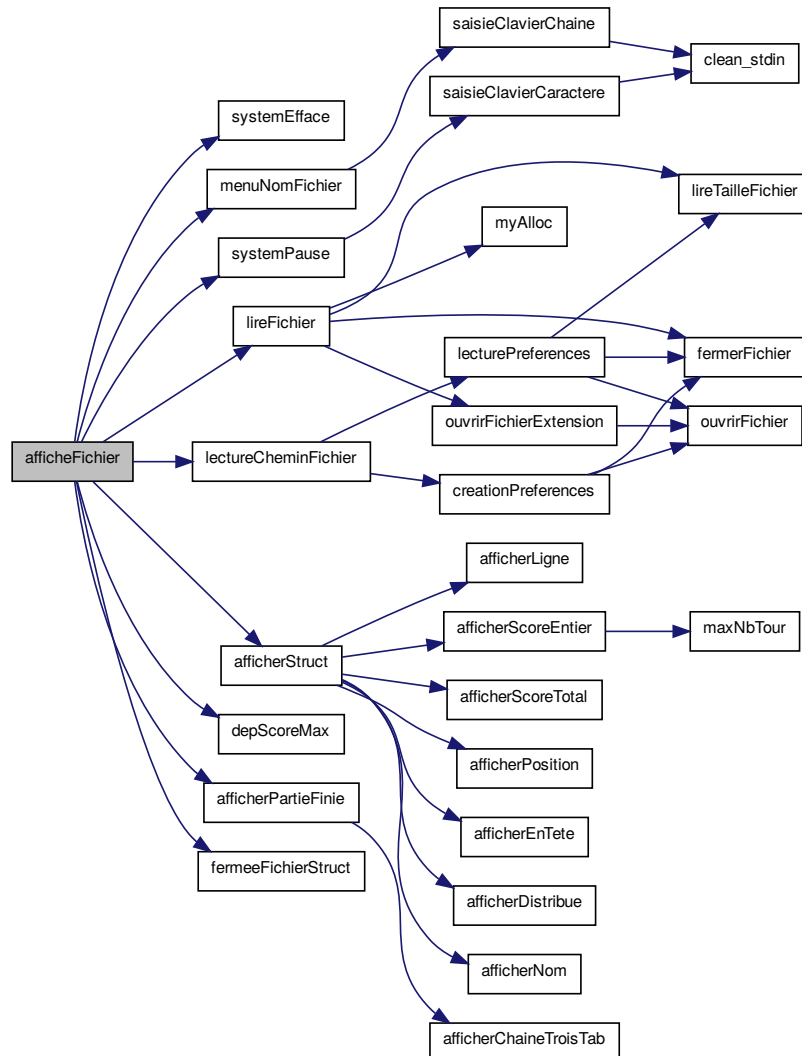
easterEggs

4.12.3 Function Documentation

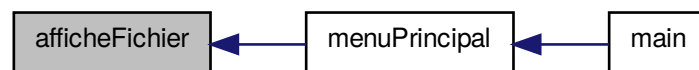
4.12.3.1 void afficheFichier ()

Demande le nom d'un fichier et l'affiche

Here is the call graph for this function:



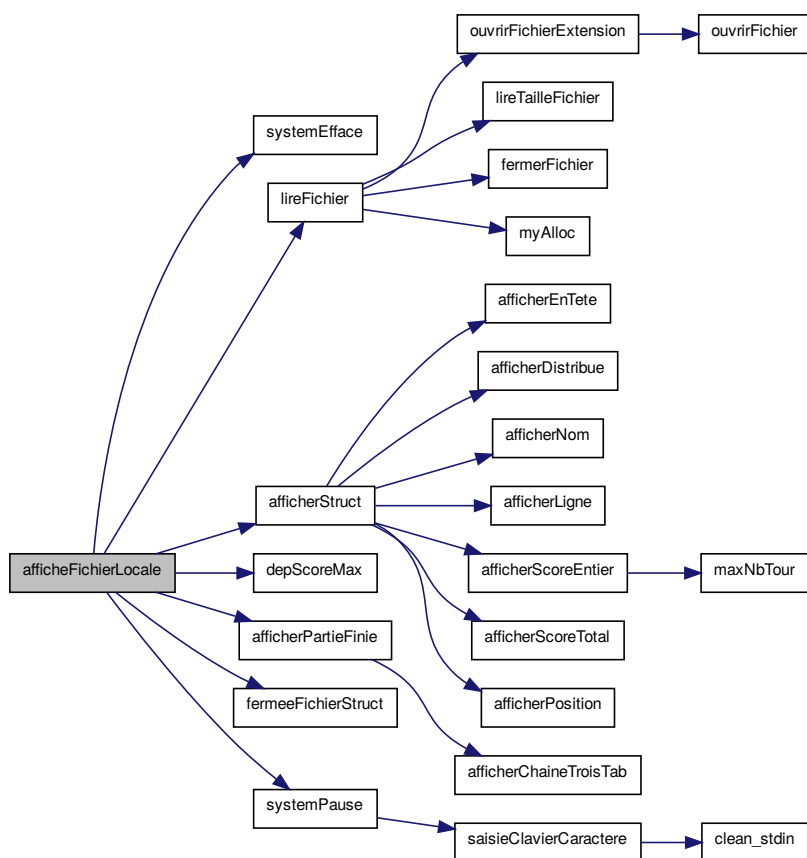
Here is the caller graph for this function:



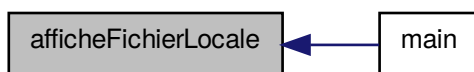
4.12.3.2 void afficheFichierLocale (char * nom_fichier)

Affiche le fichier dont le nom a ete donne en parametre

Here is the call graph for this function:



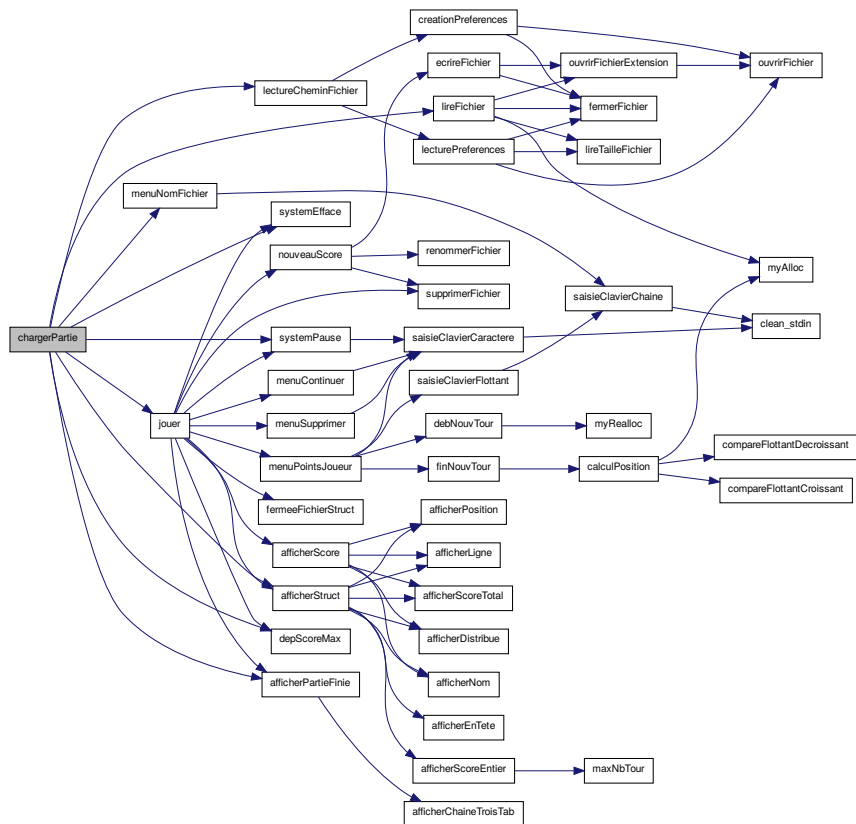
Here is the caller graph for this function:



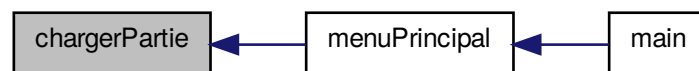
4.12.3.3 void chargerPartie ()

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

Here is the call graph for this function:



Here is the caller graph for this function:



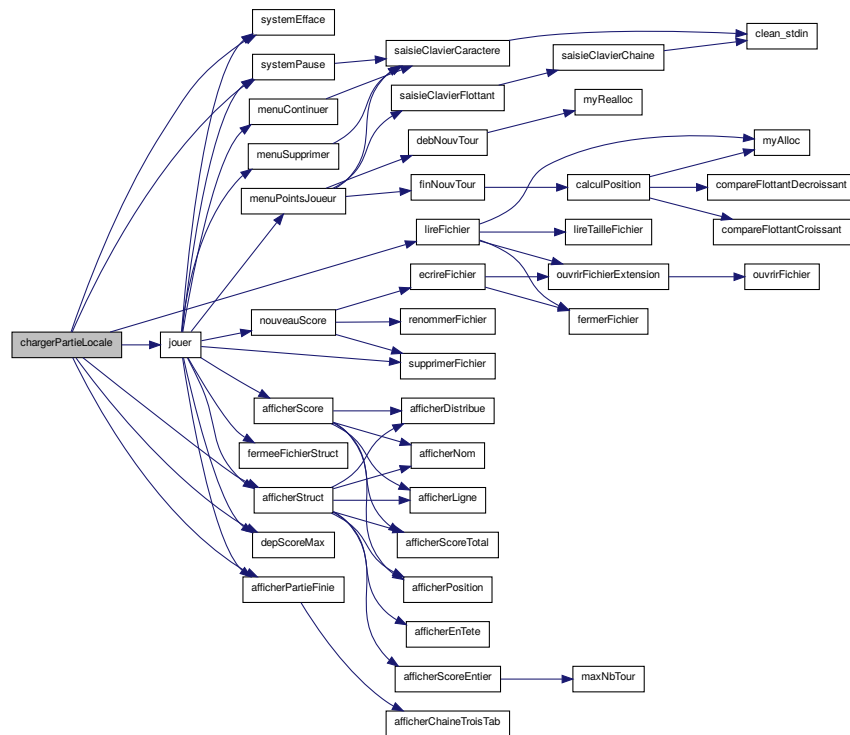
4.12.3.4 void chargerPartieLocale (char * nom_fichier)

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

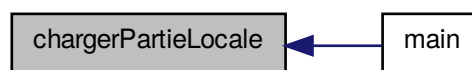
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



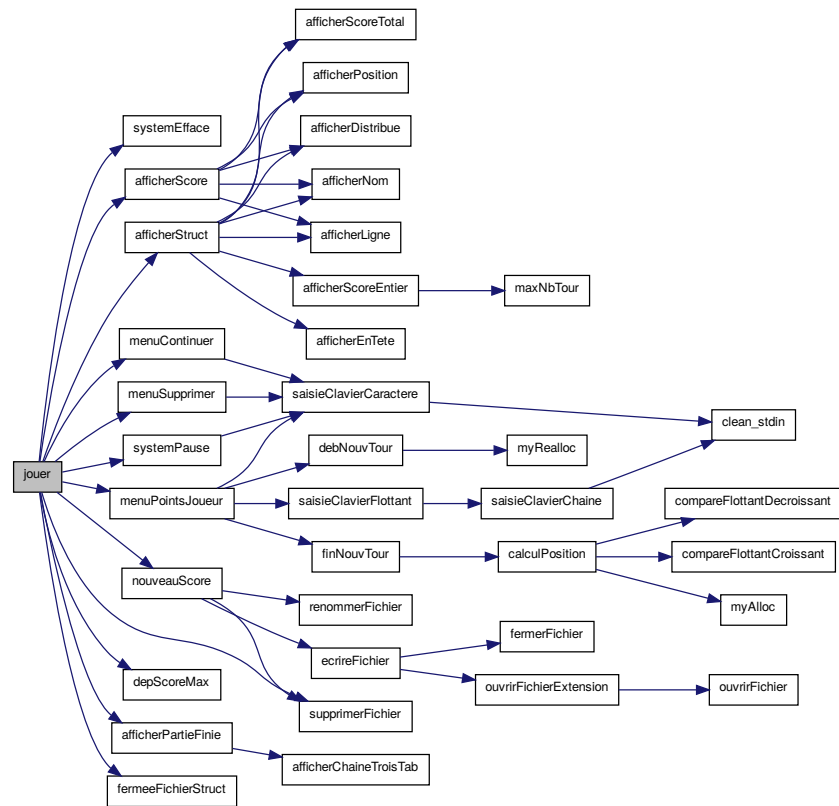
4.12.3.5 void jouer (Fichier_Jeu * ptr_struct_fichier, char * nom_fichier)

Lance la partie de comptage jusqu'a la fin

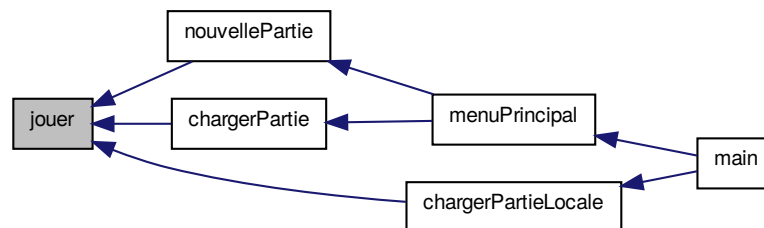
Parameters

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:



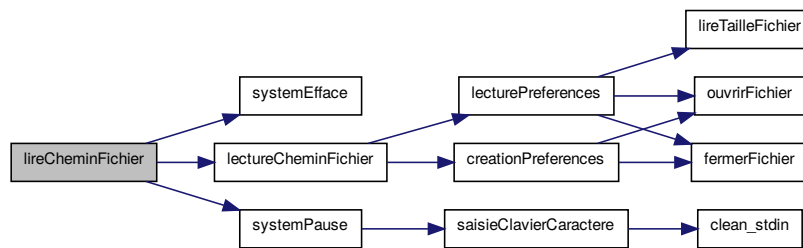
Here is the caller graph for this function:



4.12.3.6 void lireCheminFichier ()

Lis le chemin de fichier et l'affiche

Here is the call graph for this function:



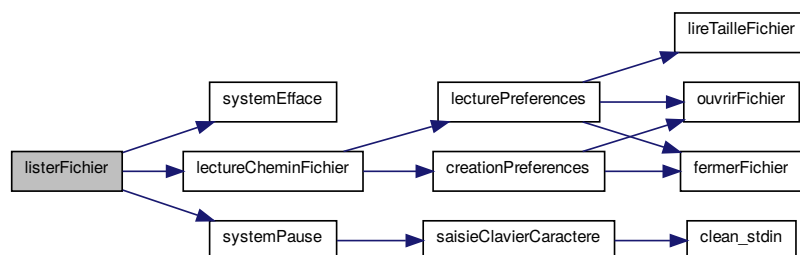
Here is the caller graph for this function:



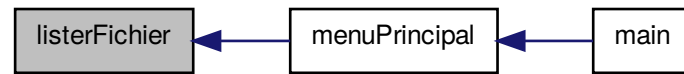
4.12.3.7 void listerFichier ()

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

Here is the call graph for this function:



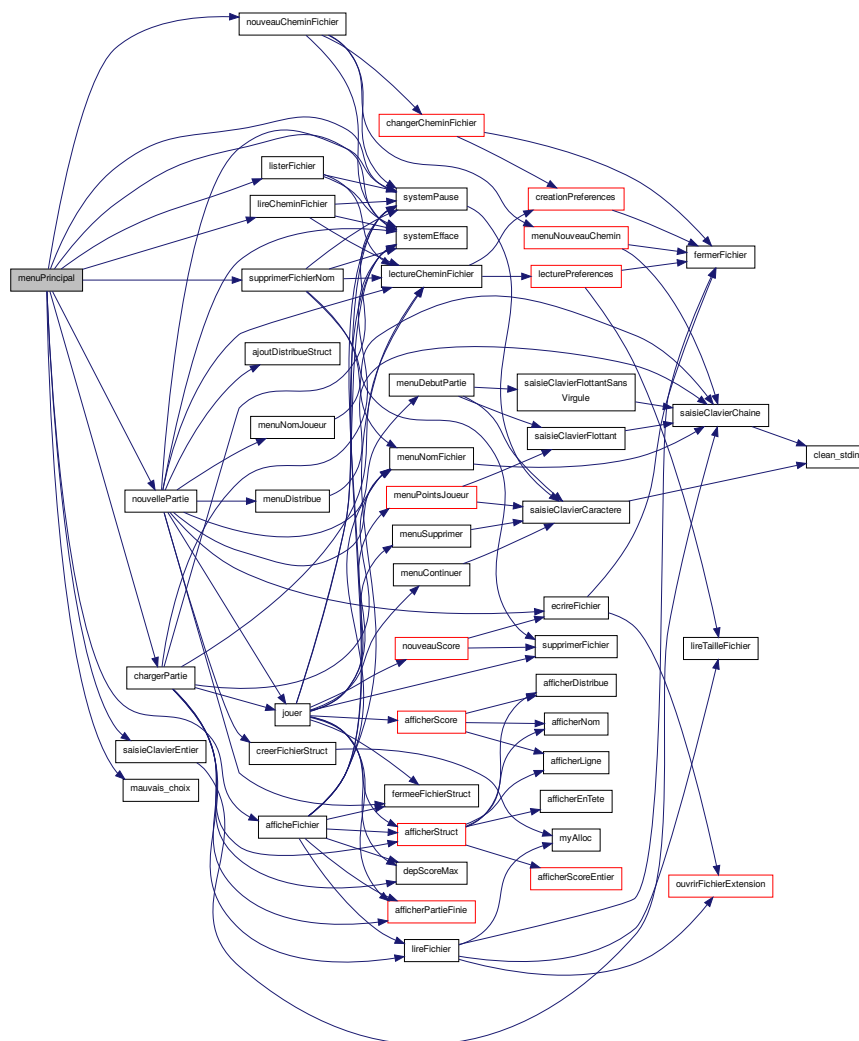
Here is the caller graph for this function:



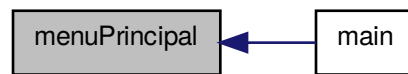
4.12.3.8 void menuPrincipal ()

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

Here is the call graph for this function:



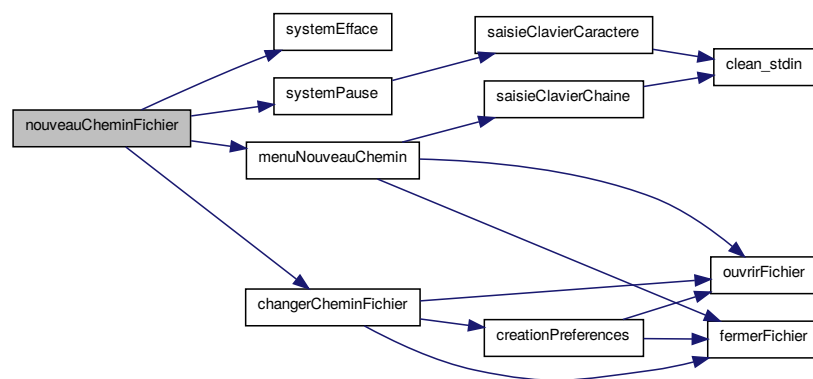
Here is the caller graph for this function:



4.12.3.9 void nouveauCheminFichier ()

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

Here is the call graph for this function:



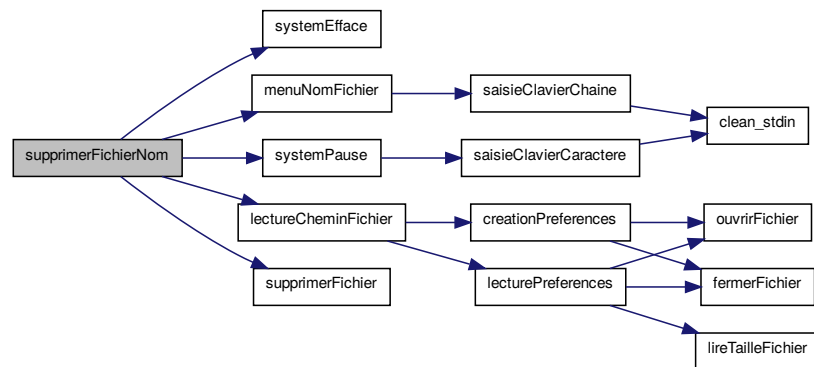
Here is the caller graph for this function:



4.12.3.10 void nouvellePartie ()

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

Here is the call graph for this function:



Here is the caller graph for this function:

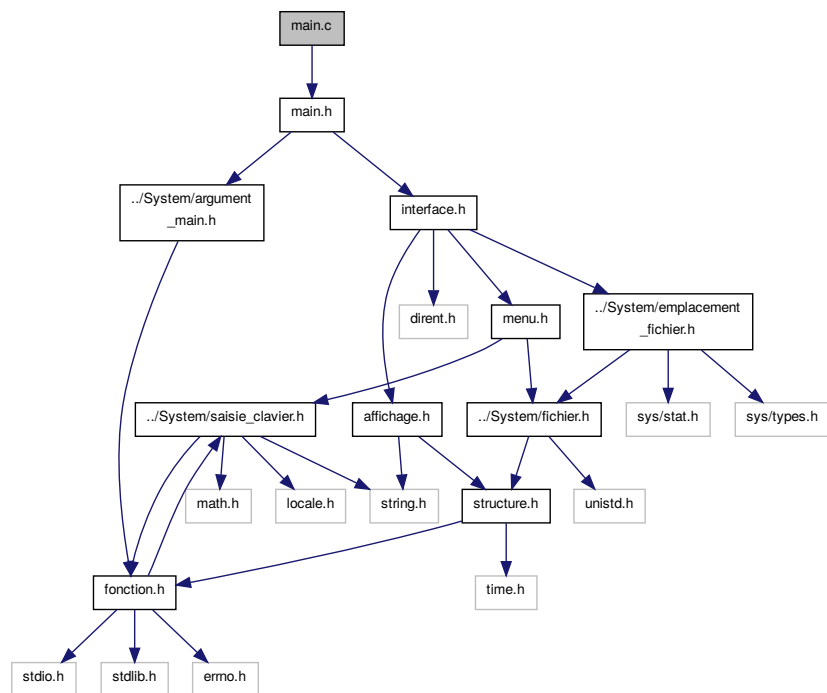


4.13 main.c File Reference

Lancement du programme.

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

Include dependency graph for main.c:



Functions

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

4.13.1 Detailed Description

Lancement du programme.

Author

Remi BERTHO

Date

09/03/14

Version

2.1.0

4.13.2 Function Documentation

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

Lance le programme

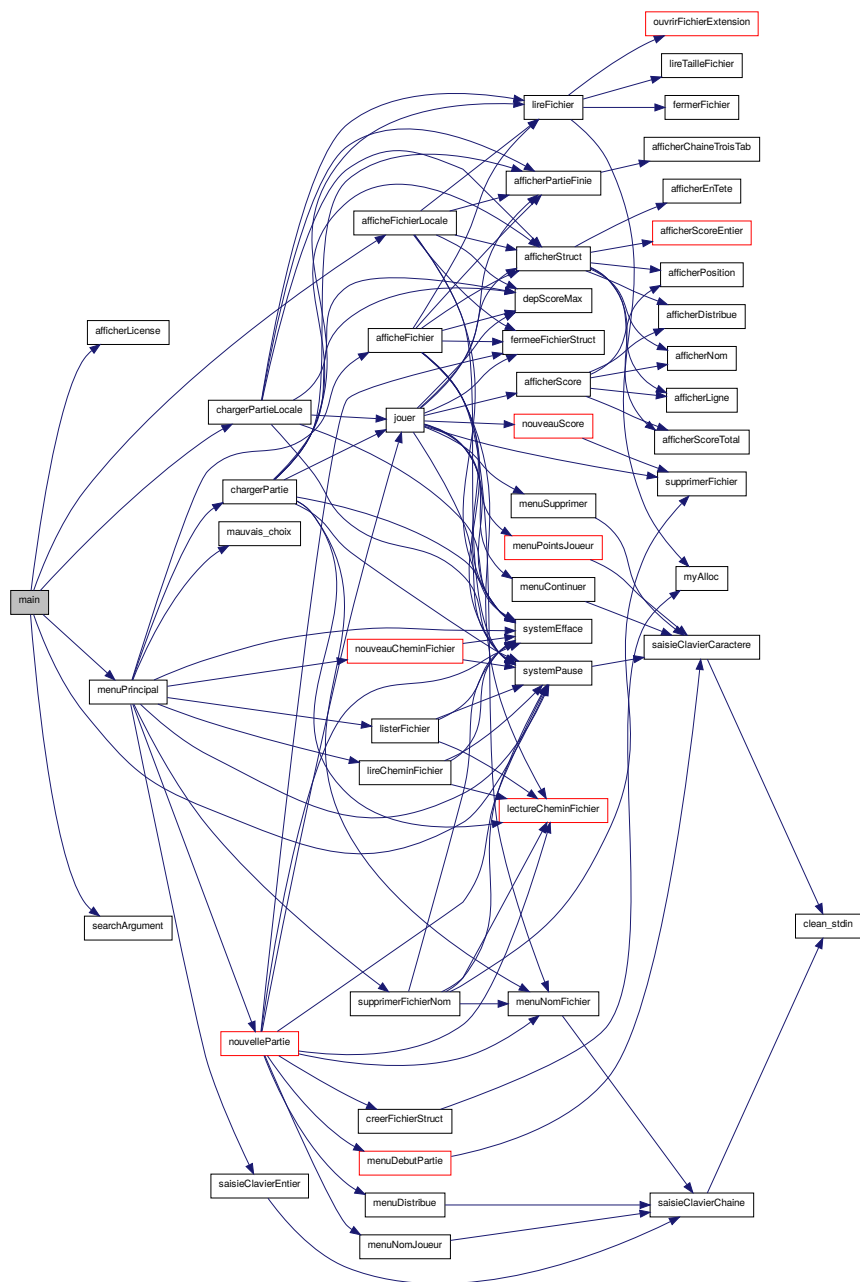
Parameters

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

Returns

0 si tout s'est bien passé

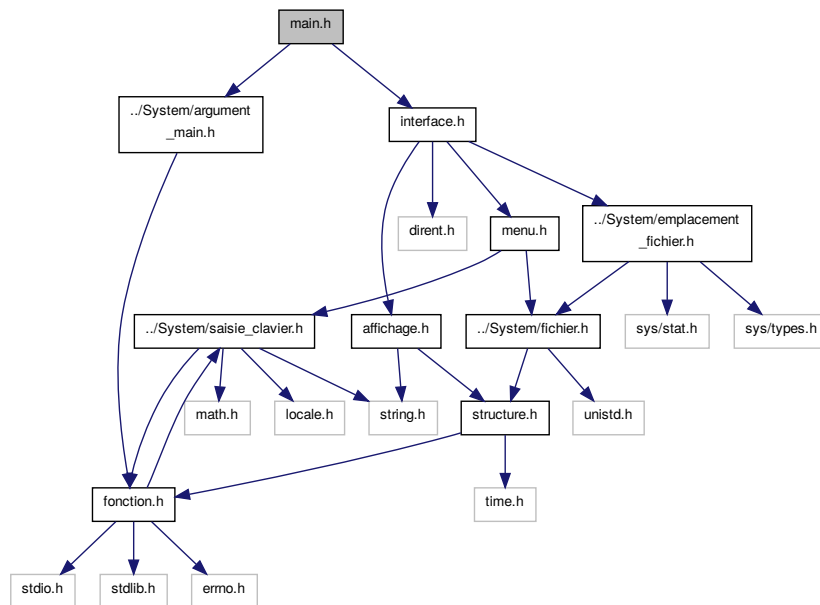
Here is the call graph for this function:



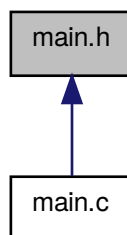
4.14 main.h File Reference

Lancement du programme.

```
#include "interface.h"
#include "../System/argument_main.h"
Include dependency graph for main.h:
```



This graph shows which files directly or indirectly include this file:



Functions

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

4.14.1 Detailed Description

Lancement du programme.

Author

Remi BERTHO

Date

09/03/14

Version

2.1.0

4.14.2 Function Documentation

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

Lance le programme

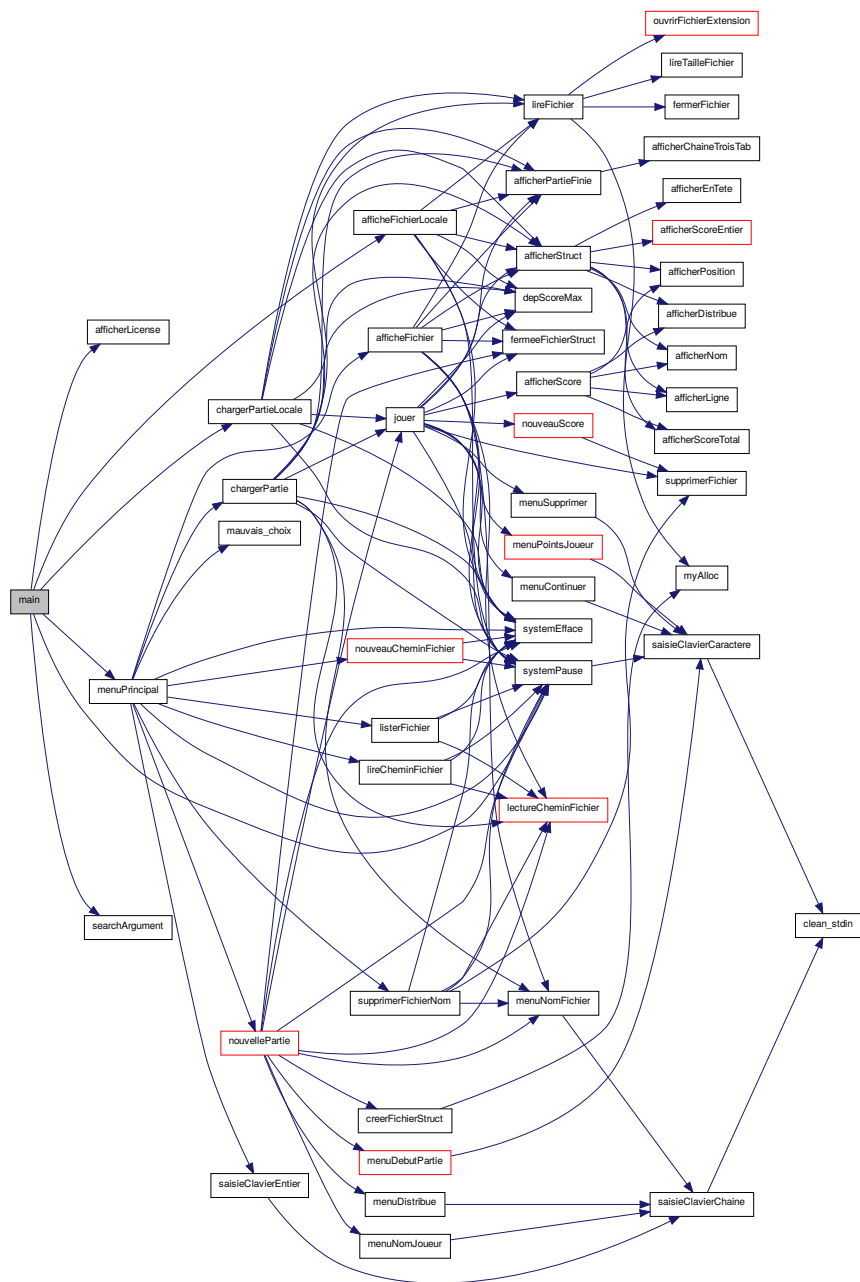
Parameters

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

Returns

0 si tout s'est bien passe

Here is the call graph for this function:

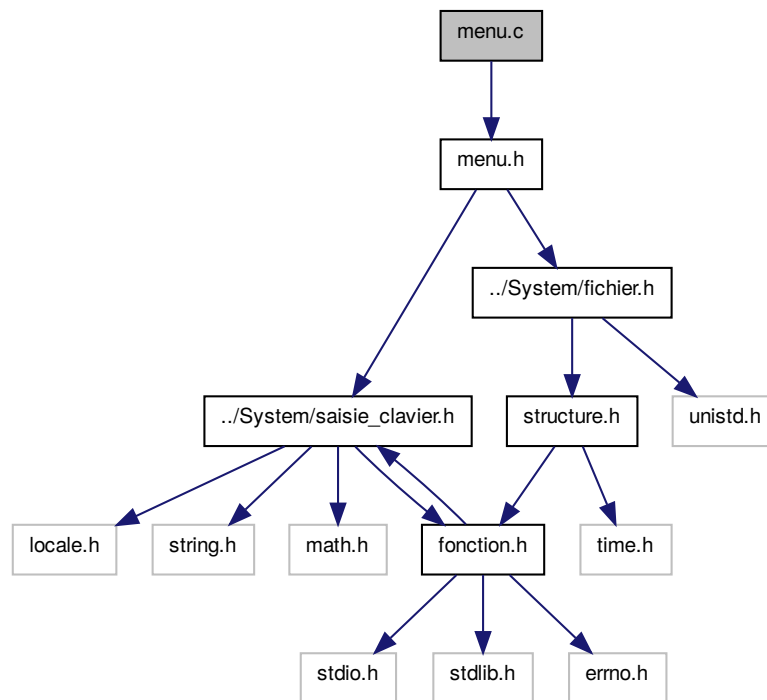


4.15 menu.c File Reference

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_FICHIER])
- void [menuDebutPartie](#) (float *ptr_nb_joueur, float *ptr_nb_max, char *ptr_sens_premier)
- void [menuDistribue](#) (char *nom_distribue)
- void [menuNomJoueur](#) (Fichier_Jeu *ptr_struct_fichier)
- void [menuPointsJoueur](#) (Fichier_Jeu *ptr_struct_fichier)
- int [menuContinuer](#) ()
- int [menuSupprimer](#) ()
- void [menuNouveauChemin](#) (char *nouveauChemin)

4.15.1 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

4.15.2 Function Documentation

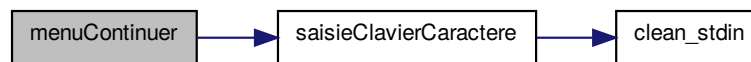
4.15.2.1 int menuContinuer ()

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

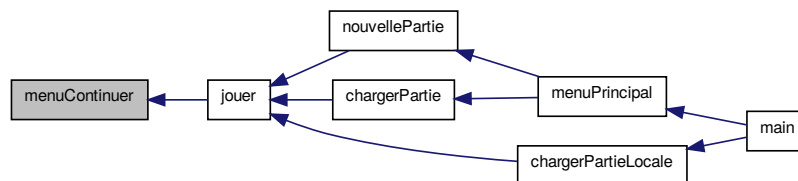
Returns

VRAI si l'on veut continuer, FAUX sinon

Here is the call graph for this function:



Here is the caller graph for this function:



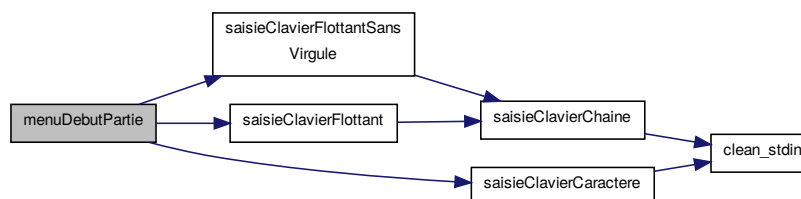
4.15.2.2 void menuDebutPartie (float * ptr_nb_joueur, float * ptr_nb_max, char * ptr_sens_premier)

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_nb_max</i>	le nombre maximum
in, out	<i>*ptr_sens_premier</i>	definit le sens du premier

Here is the call graph for this function:



Here is the caller graph for this function:



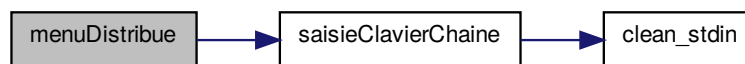
4.15.2.3 void menuDistribue (char * *nom_distribue*)

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:



Here is the caller graph for this function:



4.15.2.4 char * menuNomFichier (char *nom_fichier*[TAILLE_MAX_NOM_FICHIER])

Demande et enregistre le nom du fichier

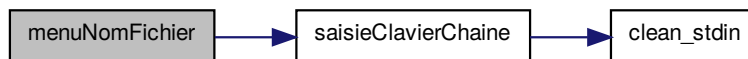
Parameters

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

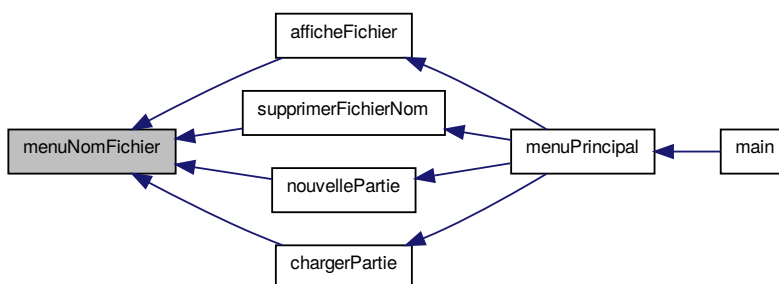
Returns

la chaine de caractere contenant le nom du fichier

Here is the call graph for this function:



Here is the caller graph for this function:



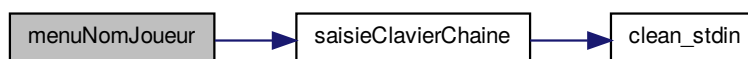
4.15.2.5 void menuNomJoueur (Fichier_Jeu * ptr_struct_fichier)

Demande et enregistre le nom des joueurs

Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



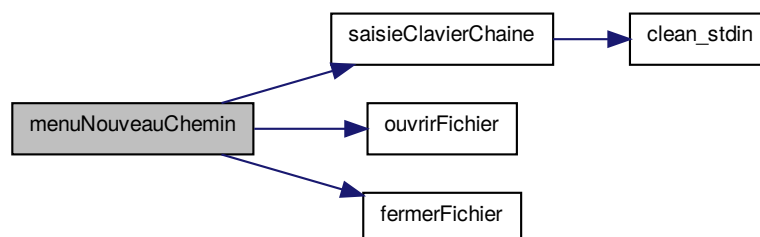
4.15.2.6 void menuNouveauChemin (char * *nouveauChemin*)

Demande et enregistre le nouveau chemin

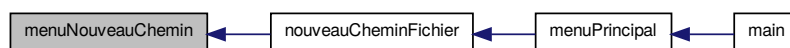
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



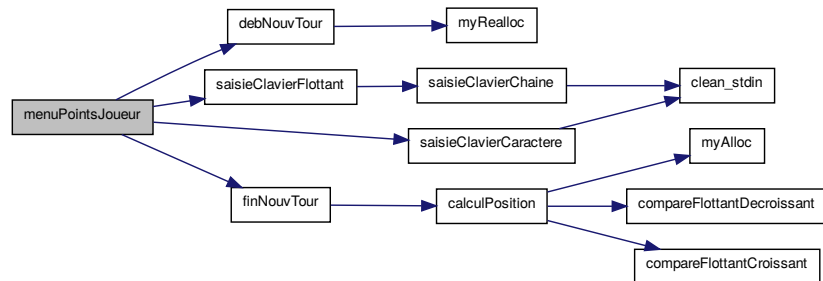
4.15.2.7 void menuPointsJoueur (Fichier_Jeu * *ptr_struct_fichier*)

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

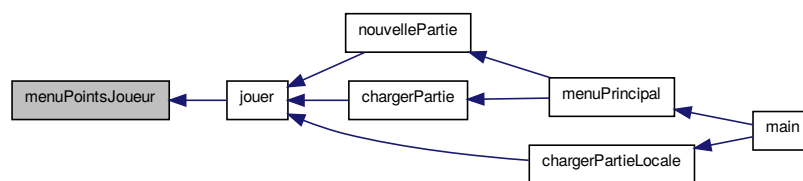
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



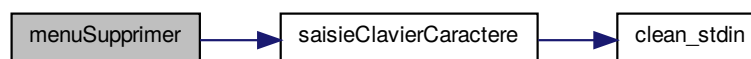
4.15.2.8 int menuSupprimer ()

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

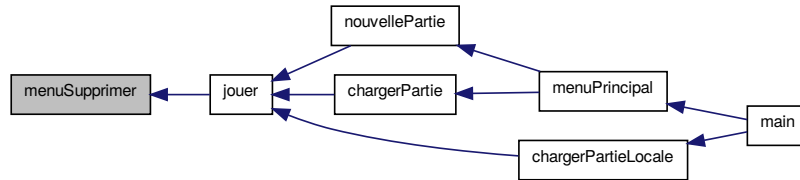
Returns

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

Here is the call graph for this function:



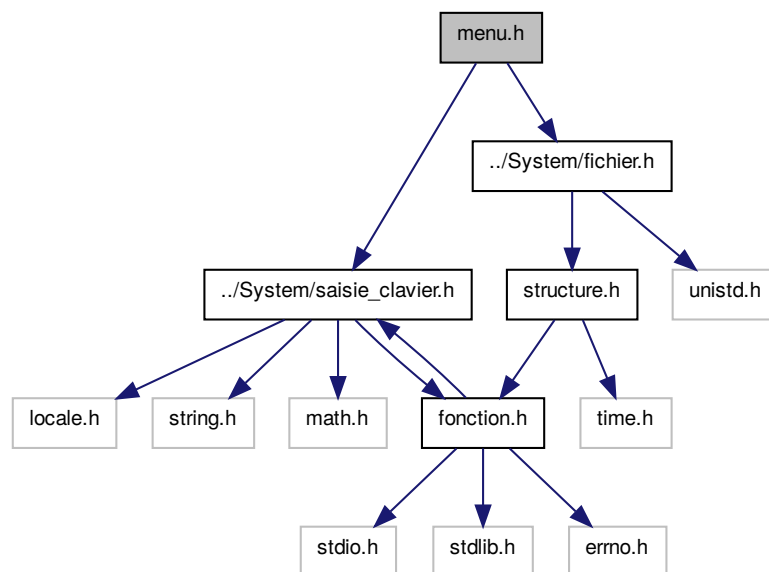
Here is the caller graph for this function:



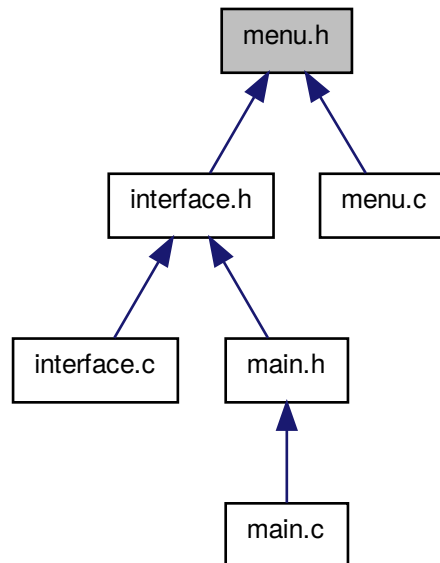
4.16 menu.h File Reference

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

```
#include "../System/saisie_clavier.h"
#include "../System/fichier.h"
Include dependency graph for menu.h:
```



This graph shows which files directly or indirectly include this file:



Functions

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

4.16.1 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

4.16.2 Function Documentation

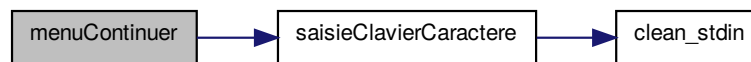
4.16.2.1 int menuContinuer ()

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

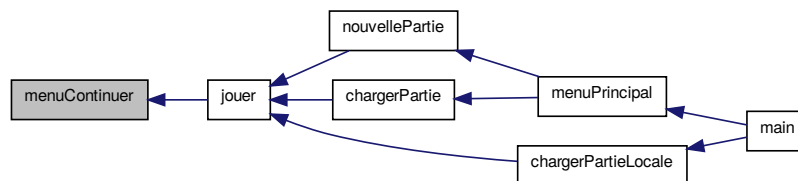
Returns

VRAI si l'on veut continuer, FAUX sinon

Here is the call graph for this function:



Here is the caller graph for this function:



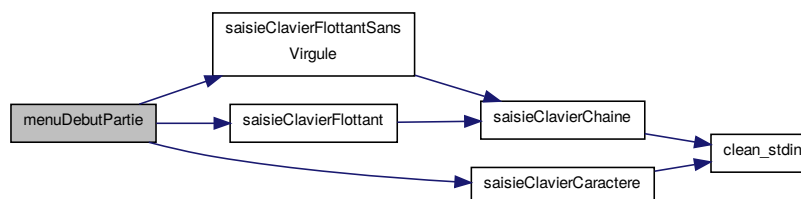
4.16.2.2 void menuDebutPartie (float * ptr_nb_joueur, float * ptr_nb_max, char * ptr_sens_premier)

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_nb_max</i>	le nombre maximum
in, out	<i>*ptr_sens_premier</i>	definit le sens du premier

Here is the call graph for this function:



Here is the caller graph for this function:



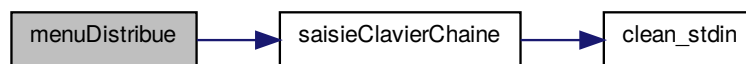
4.16.2.3 void menuDistribue (char * *nom_distribue*)

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:



Here is the caller graph for this function:



4.16.2.4 char* menuNomFichier (char *nom_fichier*[TAILLE_MAX_NOM_FICHIER])

Demande et enregistre le nom du fichier

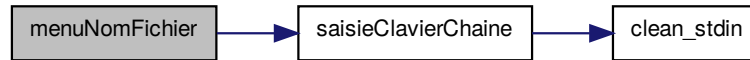
Parameters

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

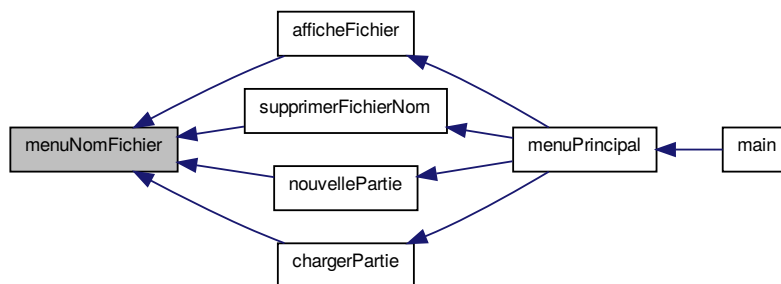
Returns

la chaine de caractere contenant le nom du fichier

Here is the call graph for this function:



Here is the caller graph for this function:



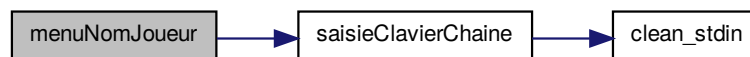
4.16.2.5 void menuNomJoueur (Fichier_Jeu * ptr_struct_fichier)

Demande et enregistre le nom des joueurs

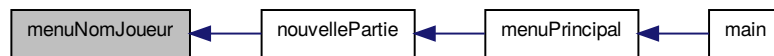
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



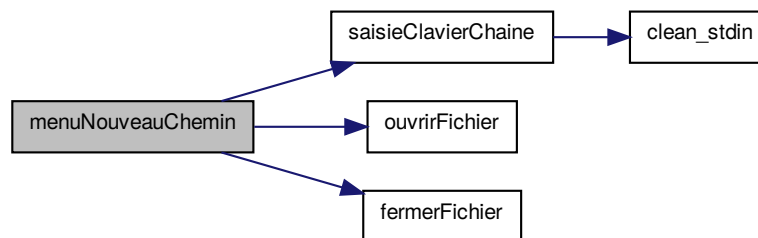
4.16.2.6 void menuNouveauChemin (char * *nouveauChemin*)

Demande et enregistre le nouveau chemin

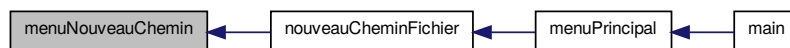
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



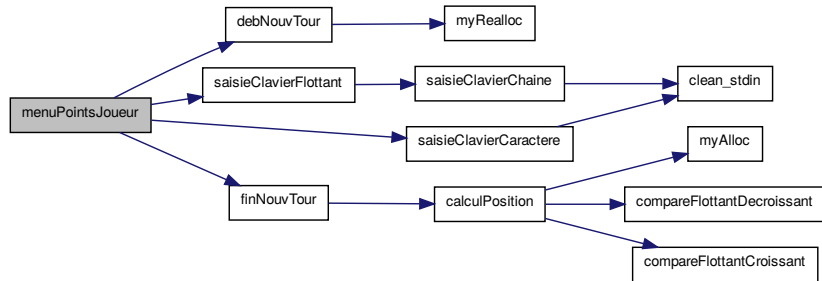
4.16.2.7 void menuPointsJoueur (Fichier_Jeu * *ptr_struct_fichier*)

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

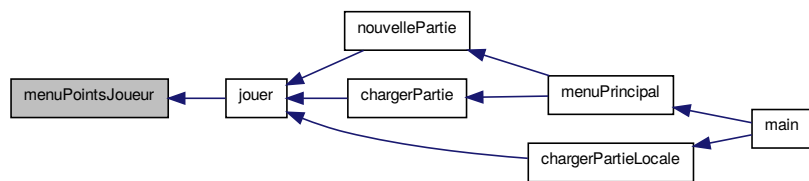
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



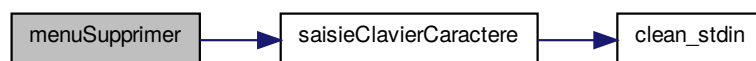
4.16.2.8 `int menuSupprimer ()`

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

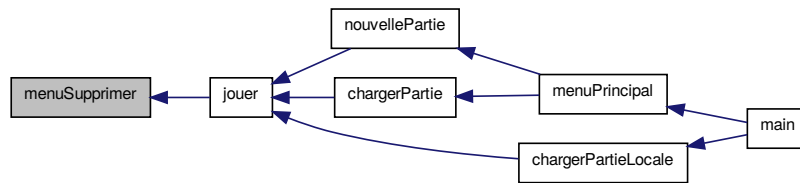
Returns

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

Here is the call graph for this function:



Here is the caller graph for this function:

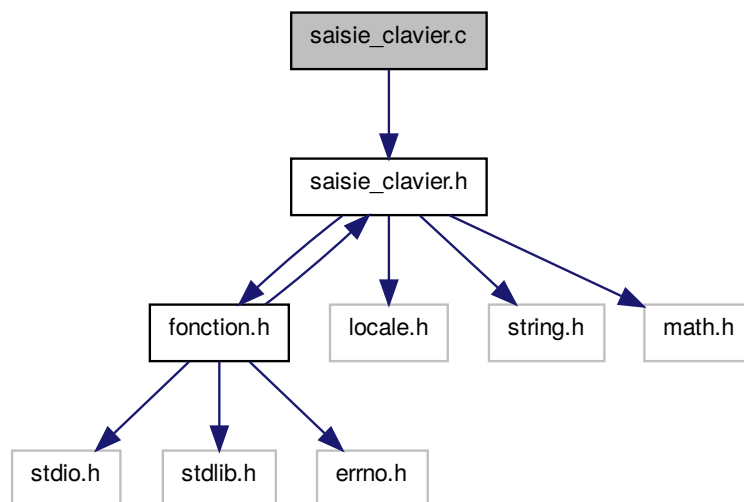


4.17 saisie_clavier.c File Reference

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)

4.17.1 Detailed Description

Fonctions de saisie clavier.

Author

Remi BERTHO

Date

13/02/14

Version

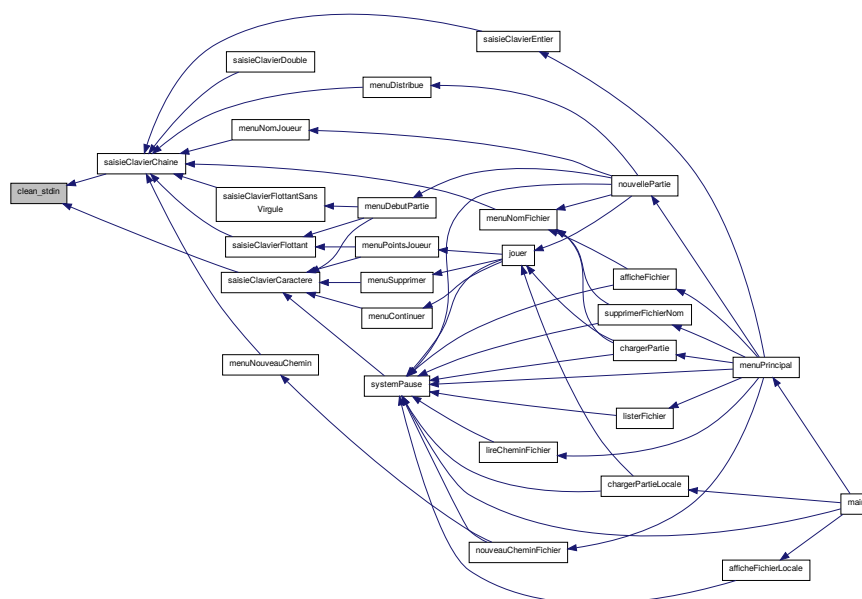
2.0

4.17.2 Function Documentation

4.17.2.1 void clean_stdin (void)

Vide la cache de stdin.

Here is the caller graph for this function:



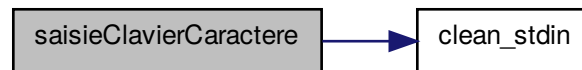
4.17.2.2 char * saisieClavierCaractere (char * c)

Fait une saisie clavier d'un caractere.

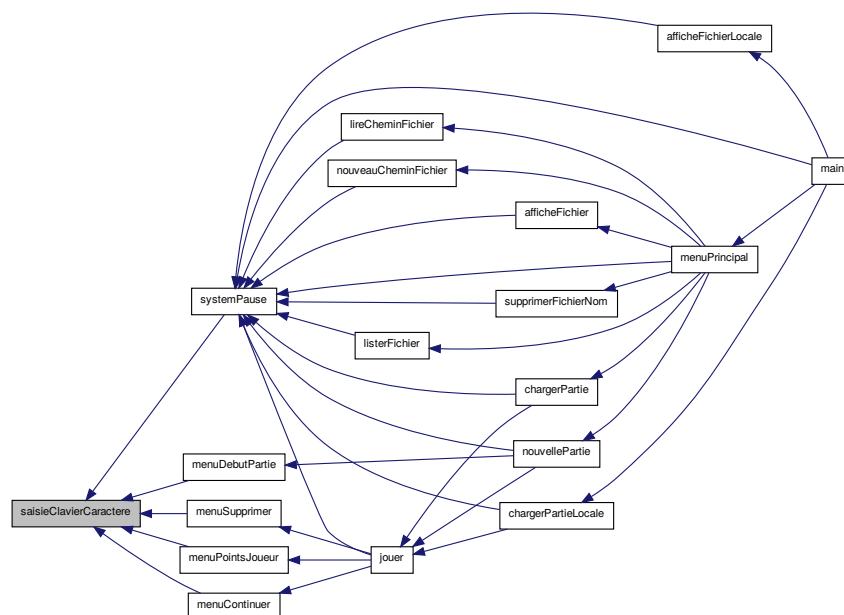
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



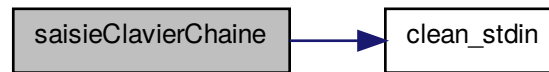
4.17.2.3 char * saisieClavierChaine (char * chaine, int nb_caract_plus_un)

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

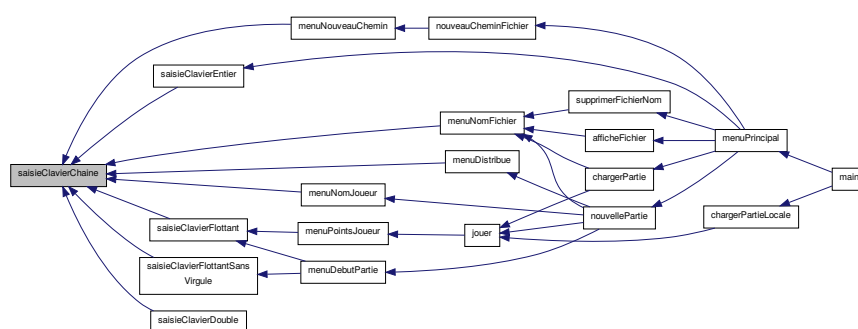
Parameters

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:



Here is the caller graph for this function:



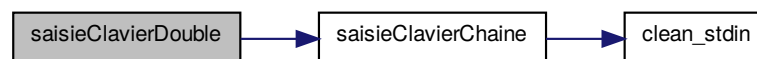
4.17.2.4 void saisieClavierDouble (double * nb)

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:



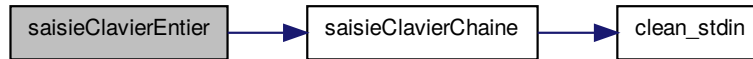
4.17.2.5 void * saisieClavierEntier (int * nb)

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

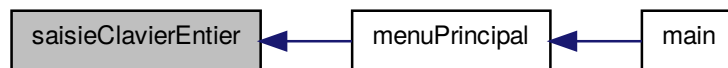
Parameters

<code>in, out</code>	<code>*nb</code>	le nombre que l'on veut saisir.
----------------------	------------------	---------------------------------

Here is the call graph for this function:



Here is the caller graph for this function:



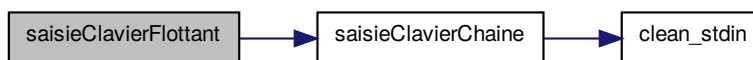
4.17.2.6 void saisieClavierFlottant (float * nb)

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

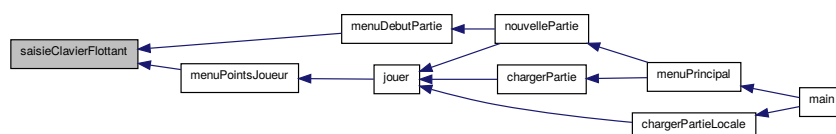
Parameters

<code>in, out</code>	<code>*nb</code>	le nombre que l'on veut saisir.
----------------------	------------------	---------------------------------

Here is the call graph for this function:



Here is the caller graph for this function:



4.17.2.7 void saisieClavierFlottantSansVirgule (float * nb)

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

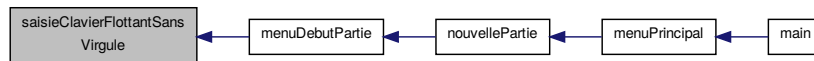
Parameters

<code>in, out</code>	<code>*nb</code>	le nombre que l'on veut saisir.
----------------------	------------------	---------------------------------

Here is the call graph for this function:



Here is the caller graph for this function:

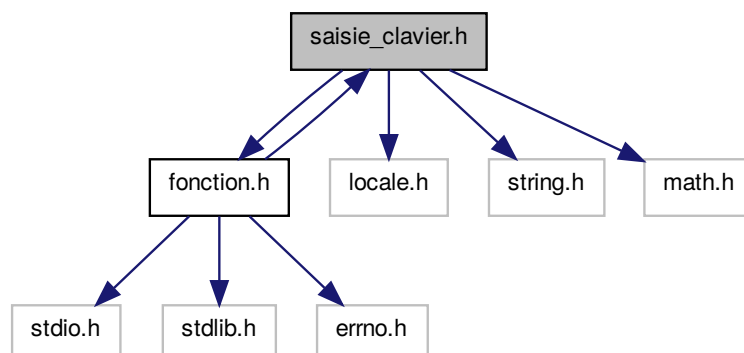


4.18 saisie_clavier.h File Reference

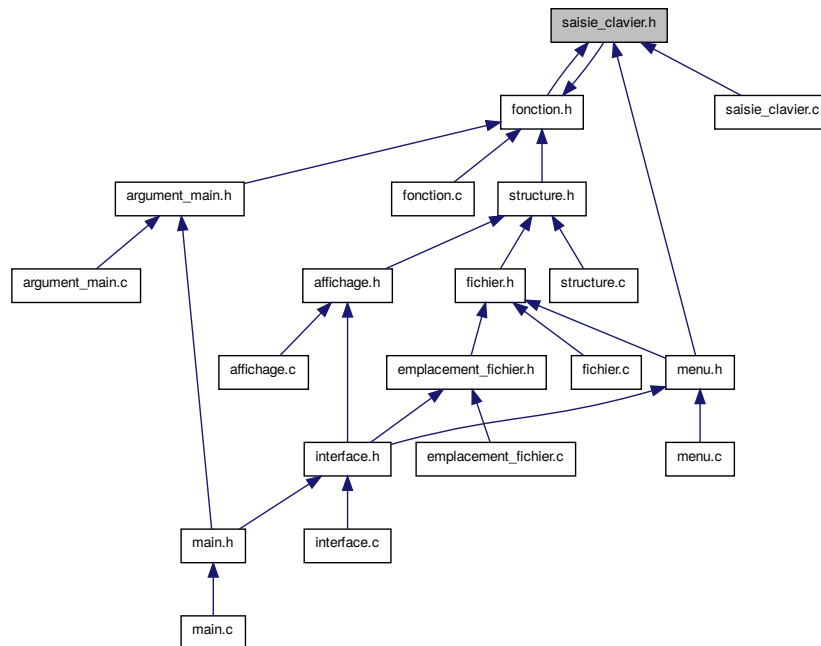
Prototypes des fonctions de saisie clavier.

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

Include dependency graph for `saisie_clavier.h`:



This graph shows which files directly or indirectly include this file:



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)

4.18.1 Detailed Description

Prototypes des fonctions de saisie clavier.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.18.2 Macro Definition Documentation

4.18.2.1 #define NB_CARACT_DOUB 309

Definit NB_CARACT_DOUB a 309

4.18.2.2 #define NB_CARACT_FLOT 39

Definit NB_CARACT_FLOT a 39

4.18.2.3 #define NB_CARACT_INT 12

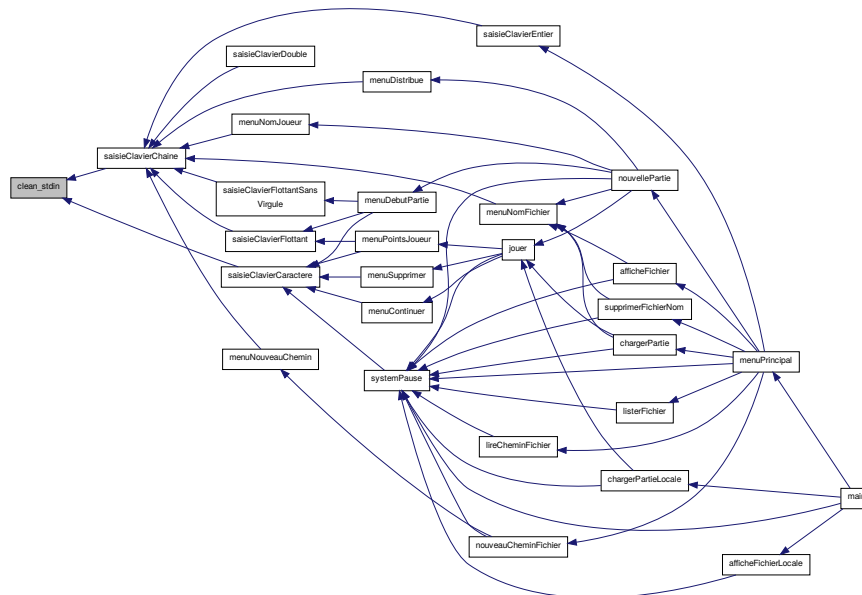
Definit NB_CARACT_INT a 12

4.18.3 Function Documentation

4.18.3.1 void clean_stdin (void)

Vide la cache de stdin.

Here is the caller graph for this function:



4.18.3.2 char* saisieClavierCaractere (char * c)

Fait une saisie clavier d'un caractere.

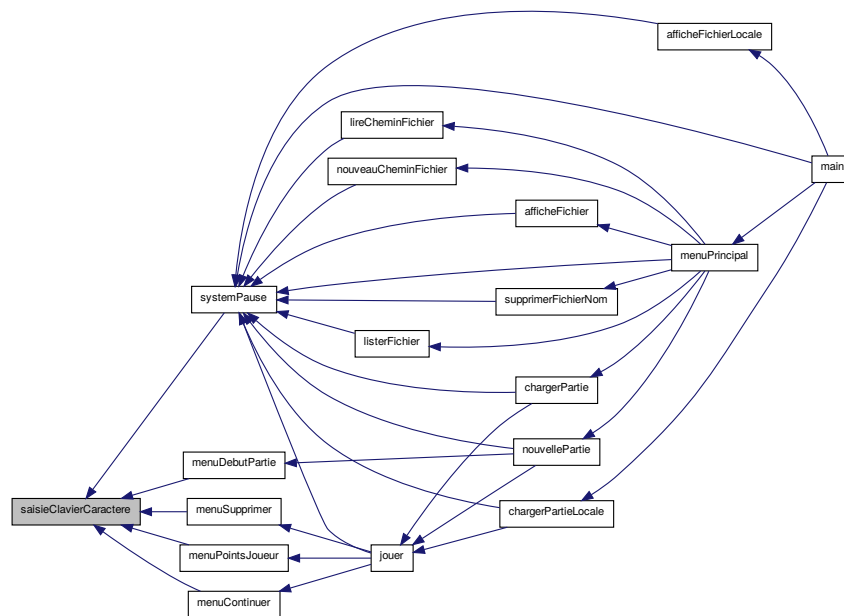
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



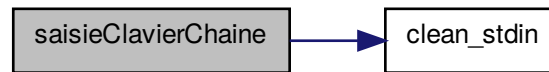
4.18.3.3 char* saisieClavierChaine (char * chaine, int nb_caract_plus_un)

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

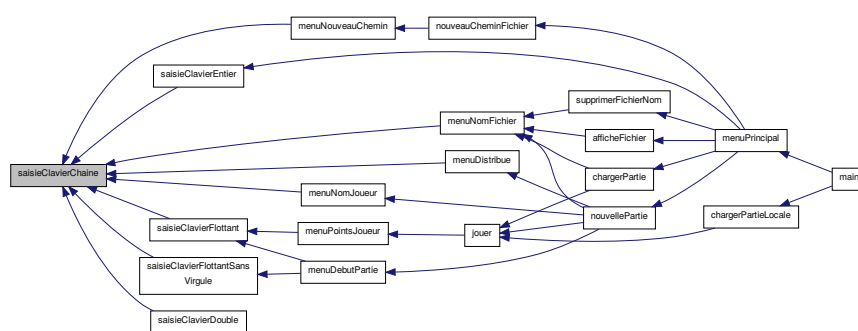
Parameters

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:



Here is the caller graph for this function:



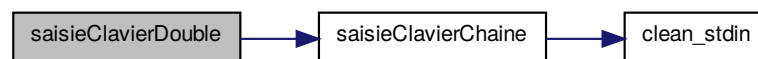
4.18.3.4 void saisieClavierDouble (double * nb)

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:



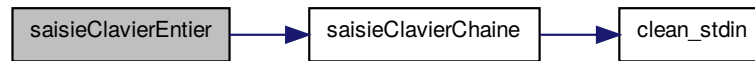
4.18.3.5 void saisieClavierEntier (int * nb)

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

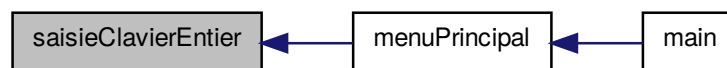
Parameters

<code>in, out</code>	<code>*nb</code>	le nombre que l'on veut saisir.
----------------------	------------------	---------------------------------

Here is the call graph for this function:



Here is the caller graph for this function:



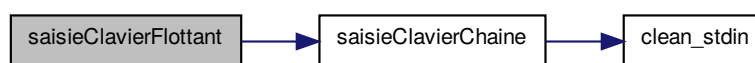
4.18.3.6 void saisieClavierFlottant (float * nb)

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

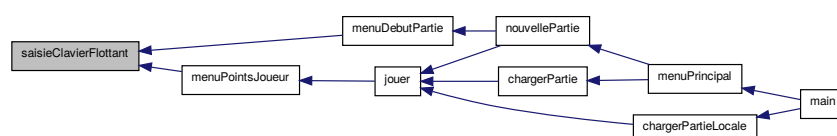
Parameters

<code>in, out</code>	<code>*nb</code>	le nombre que l'on veut saisir.
----------------------	------------------	---------------------------------

Here is the call graph for this function:



Here is the caller graph for this function:



4.18.3.7 void saisieClavierFlottantSansVirgule (float * nb)

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

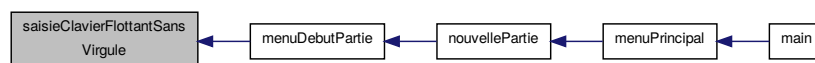
Parameters

<code>in, out</code>	<code>*nb</code>	le nombre que l'on veut saisir.
----------------------	------------------	---------------------------------

Here is the call graph for this function:



Here is the caller graph for this function:

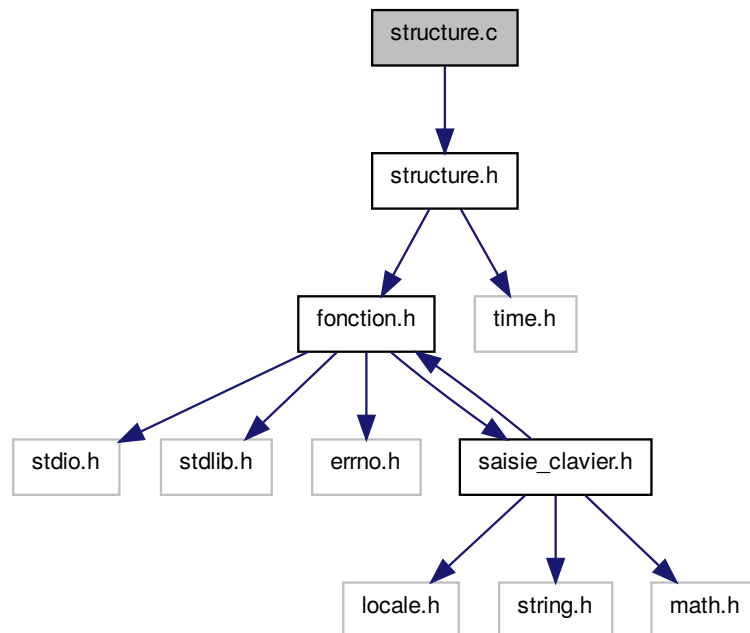


4.19 structure.c File Reference

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, float nb_max, char sens_premier)
- 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`)

4.19.1 Detailed Description

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

Author

Remi BERTHO

Date

09/03/14

Version

2.1.0

4.19.2 Function Documentation

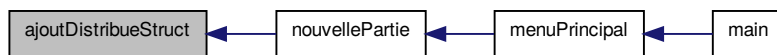
4.19.2.1 void ajoutDistribueStruct (Fichier_Jeu * ptr_struct_fichier, char * nom_distribue)

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 caller graph for this function:



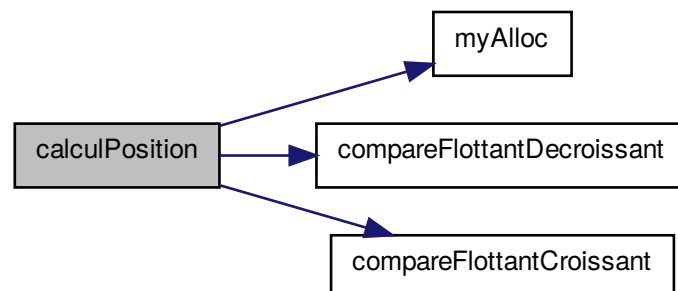
4.19.2.2 void calculPosition (Fichier_Jeu * ptr_struct_fichier)

Calcule les positions des joueurs

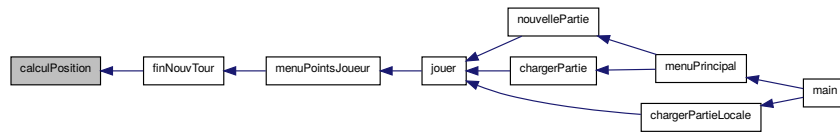
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



4.19.2.3 Fichier_Jeu * creerFichierStruct (float nb_joueur, float nb_max, char sens_premier)

Cree une structure Fichier_jeu a partir des donnees de la fonction.

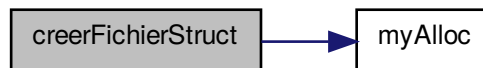
Parameters

in	<i>nb_joueur</i>	le nombre de joueur
in	<i>nb_max</i>	le nombre maximum de points
in	<i>sens_premier</i>	permet de savoir dans quelle sens est calculer le premier

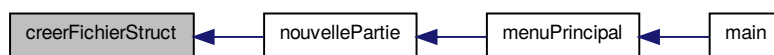
Returns

un pointeur sur le structure [Fichier_Jeu](#) cree

Here is the call graph for this function:

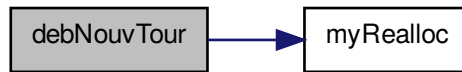


Here is the caller graph for this function:

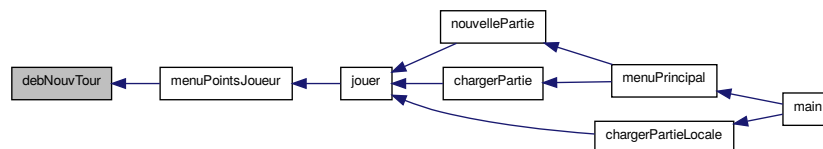


4.19.2.4 void debNouvTour (Fichier_Jeu * ptr_struct_fichier, int num_joueur)

Here is the call graph for this function:



Here is the caller graph for this function:



4.19.2.5 int depScoreMax (Fichier_Jeu * ptr_struct_fichier)

Verifie si quelqu'un a depasse le score maximum

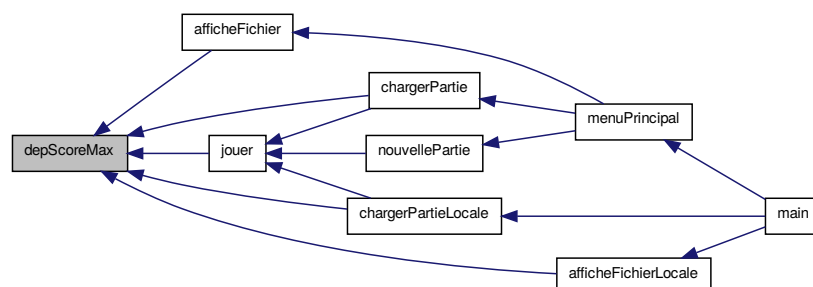
Parameters

in	*ptr_struct_fichier	la structure du fichier
----	---------------------	-------------------------

Returns

VRAI s'il y a un depassement, FAUX sinon

Here is the caller graph for this function:



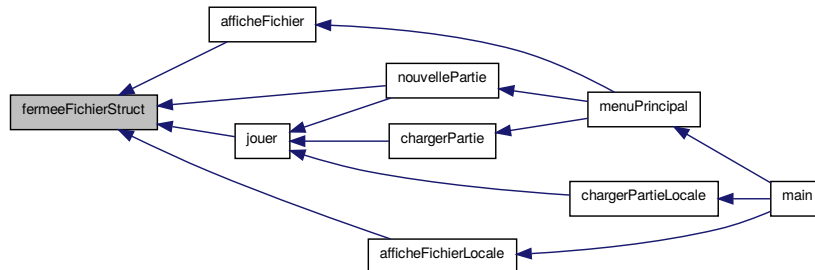
4.19.2.6 void fermeeFichierStruct (Fichier_Jeu * ptr_struct_fichier)

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

Parameters

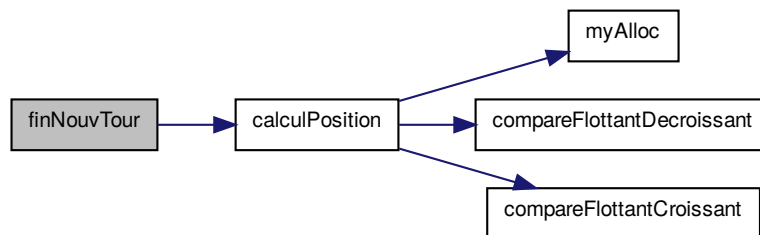
in, out	<i>*ptr_struct_fichier</i>	un pointeur sur la structure Fichier_Jeu a fermer
---------	----------------------------	---

Here is the caller graph for this function:

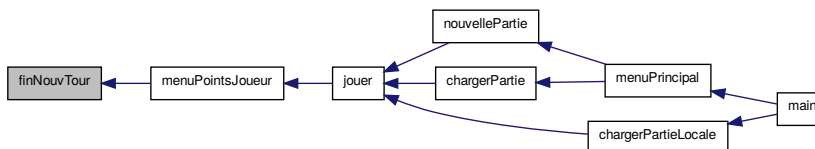


4.19.2.7 void finNouvTour (Fichier_Jeu * ptr_struct_fichier, int num_joueur)

Here is the call graph for this function:



Here is the caller graph for this function:



4.19.2.8 int maxNbTour (Fichier_Jeu * ptr_struct_fichier)

Cherche le maximum de nombre de tour

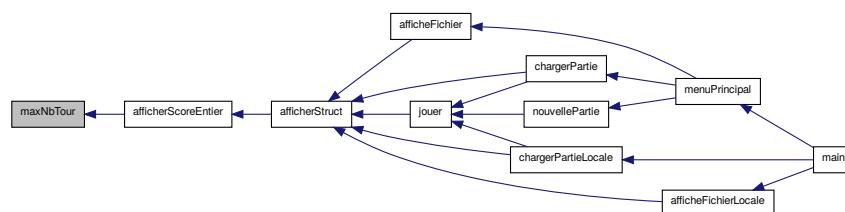
Parameters

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

Returns

le nombre maximum de nombre de tour

Here is the caller graph for this function:



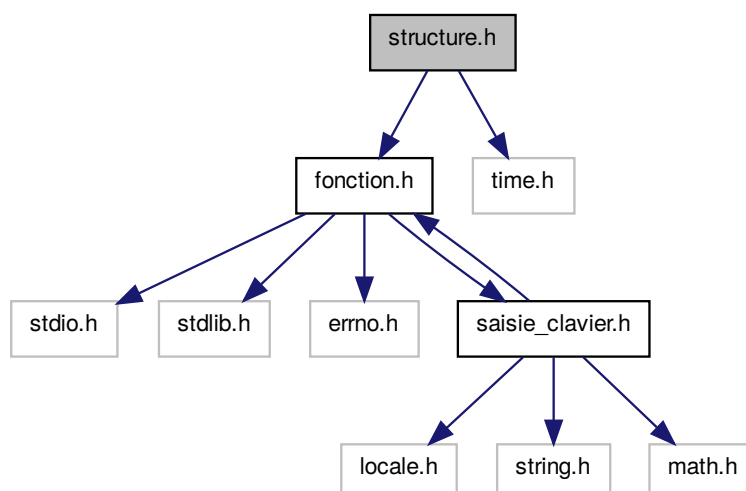
4.20 structure.h File Reference

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

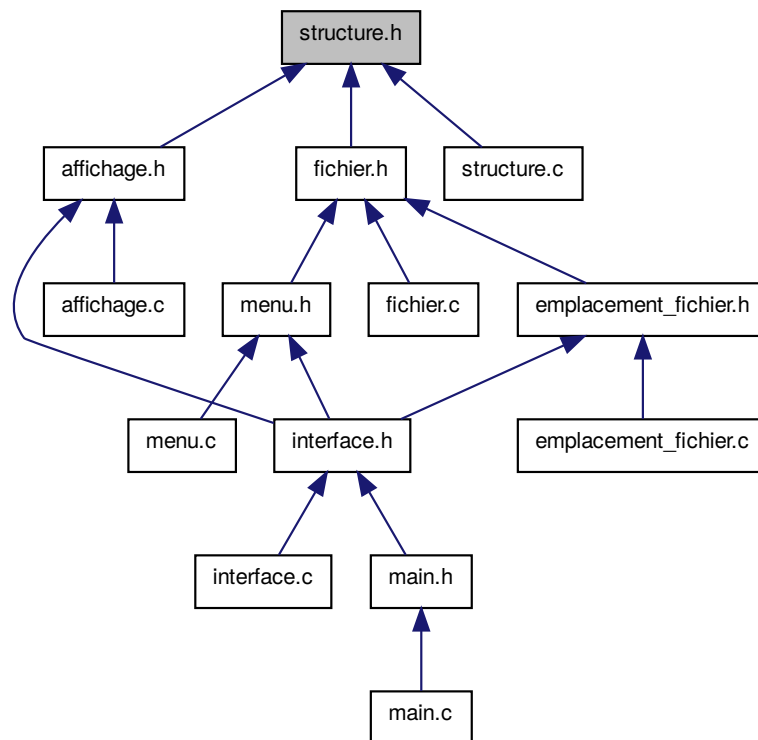
```
#include "fonction.h"
```

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

Include dependency graph for structure.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [Fichier_Jeu](#)

Macros

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

Functions

- [Fichier_Jeu * creerFichierStruct](#) (float nb_joueur, float nb_max, char sens_premier)
- 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)

4.20.1 Detailed Description

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

Author

Remi BERTHO

Date

09/03/14

Version

2.1.0

4.20.2 Macro Definition Documentation

4.20.2.1 #define TAILLE_MAX_NOM 30

Definit la taille max d'un nom a 30

4.20.2.2 #define VERSION 1.4

Definit la version a 1.4

4.20.3 Function Documentation

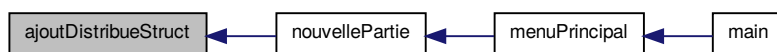
4.20.3.1 void ajoutDistribueStruct (Fichier_Jeu * ptr_struct_fichier, char * nom_distribue)

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 caller graph for this function:



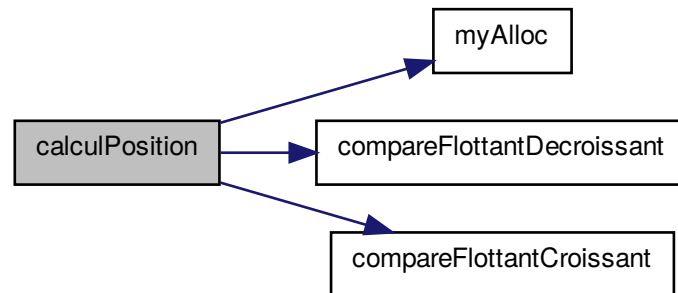
4.20.3.2 void calculPosition (Fichier_Jeu * ptr_struct_fichier)

Calcule les positions des joueurs

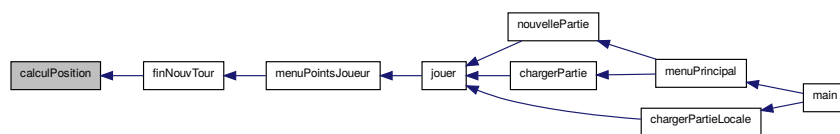
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



4.20.3.3 Fichier_Jeu* creerFichierStruct (float nb_joueur, float nb_max, char sens_premier)

Cree une structure Fichier_jeu a partir des donnees de la fonction.

Parameters

in	<i>nb_joueur</i>	le nombre de joueur
in	<i>nb_max</i>	le nombre maximum de points
in	<i>sens_premier</i>	permet de savoir dans quelle sens est calculer le premier

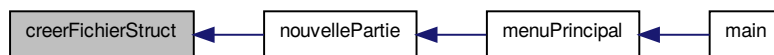
Returns

un pointeur sur le structure [Fichier_Jeu](#) cree

Here is the call graph for this function:



Here is the caller graph for this function:

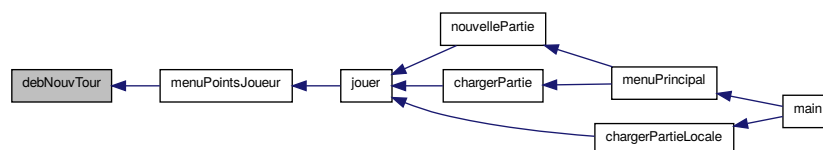


4.20.3.4 void debNouvTour (Fichier_Jeu * ptr_struct_fichier, int num_joueur)

Here is the call graph for this function:



Here is the caller graph for this function:



4.20.3.5 int depScoreMax (Fichier_Jeu * *ptr_struct_fichier*)

Verifie si quelqu'un a depasse le score maximum

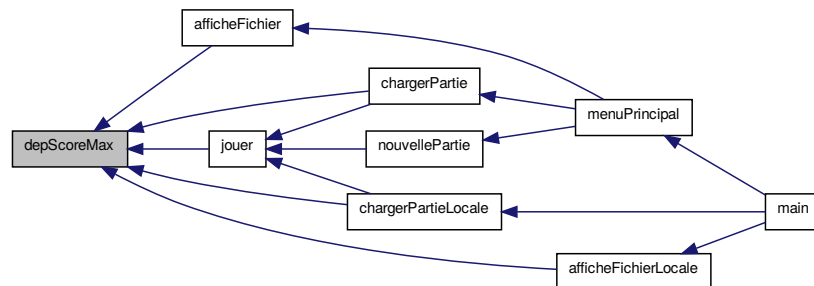
Parameters

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

Returns

VRAI s'il y a un depassement, FAUX sinon

Here is the caller graph for this function:



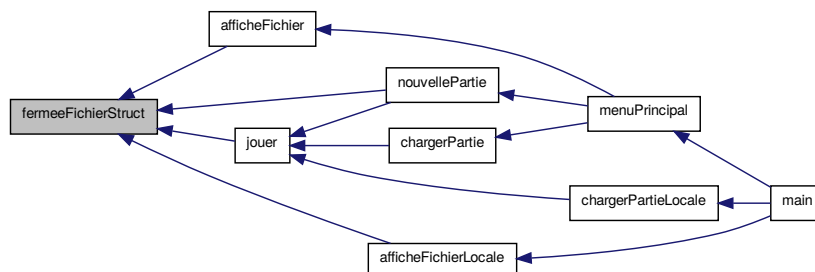
4.20.3.6 void fermeeFichierStruct (Fichier_Jeu * ptr_struct_fichier)

Desalloue la memoire attribuee a la structure `Fichier_Jeu` mis en parametre

Parameters

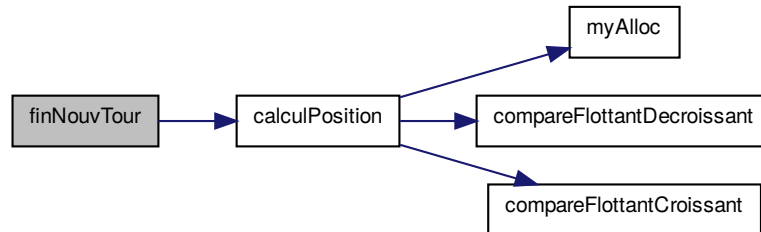
in, out	<i>*ptr_struct_fichier</i>	un pointeur sur la structure <code>Fichier_Jeu</code> a fermer
---------	----------------------------	--

Here is the caller graph for this function:

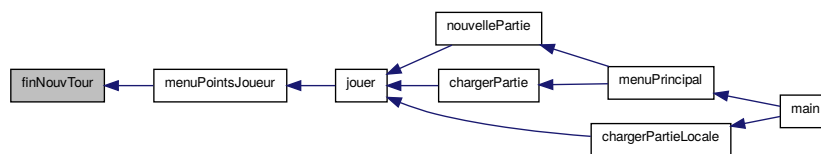


4.20.3.7 void finNouvTour (Fichier_Jeu * ptr_struct_fichier, int num_joueur)

Here is the call graph for this function:



Here is the caller graph for this function:



4.20.3.8 int maxNbTour (Fichier_Jeu * ptr_struct_fichier)

Cherche le maximum de nombre de tour

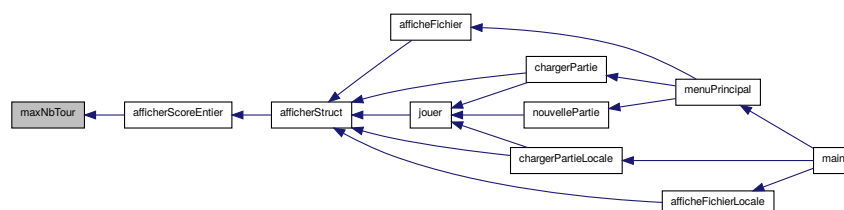
Parameters

in	*ptr_struct_fichier	la structure du fichier
----	---------------------	-------------------------

Returns

le nombre maximum de nombre de tour

Here is the caller graph for this function:



Index

affFich
 interface.h, 81

affichage.c, 9
 afficherChaineTroisTab, 10
 afficherDistribue, 10
 afficherEnTete, 11
 afficherLicense, 11
 afficherLigne, 12
 afficherNom, 12
 afficherPartieFinie, 13
 afficherPosition, 13
 afficherScore, 14
 afficherScoreEntier, 15
 afficherScoreTotal, 15
 afficherStruct, 16

affichage.h, 17
 afficherChaineTroisTab, 19
 afficherDistribue, 19
 afficherEnTete, 19
 afficherLicense, 20
 afficherLigne, 20
 afficherNom, 21
 afficherPartieFinie, 21
 afficherPosition, 22
 afficherScore, 22
 afficherScoreEntier, 23
 afficherScoreTotal, 24
 afficherStruct, 24

afficheFichier
 interface.c, 69
 interface.h, 81

afficheFichierLocale
 interface.c, 70
 interface.h, 82

afficherChaineTroisTab
 affichage.c, 10
 affichage.h, 19

afficherDistribue
 affichage.c, 10
 affichage.h, 19

afficherEnTete
 affichage.c, 11
 affichage.h, 19

afficherLicense
 affichage.c, 11
 affichage.h, 20

afficherLigne
 affichage.c, 12
 affichage.h, 20

afficherNom
 affichage.c, 12
 affichage.h, 21

afficherPartieFinie
 affichage.c, 13
 affichage.h, 21

afficherPosition
 affichage.c, 13
 affichage.h, 22

afficherScore
 affichage.c, 14
 affichage.h, 22

afficherScoreEntier
 affichage.c, 15
 affichage.h, 23

afficherScoreTotal
 affichage.c, 15
 affichage.h, 24

afficherStruct
 affichage.c, 16
 affichage.h, 24

ajoutDistribueStruct
 structure.c, 125
 structure.h, 132

annee
 Fichier_Jeu, 5

argument_main.c, 25
 searchArgument, 26

argument_main.h, 27
 LECTURE_FICHER, 29
 OUVERTURE_FICHER, 29
 searchArgument, 29

calculPosition
 structure.c, 125
 structure.h, 132

changerCheminFichier
 emplacement_fichier.c, 31
 emplacement_fichier.h, 37

charPart
 interface.h, 81

chargerPartie
 interface.c, 71
 interface.h, 83

chargerPartieLocale
 interface.c, 72
 interface.h, 84

clean_stdin
 saisie_clavier.c, 111
 saisie_clavier.h, 118

- compareFlottantCroissant
 - fonction.c, 54
 - fonction.h, 61
- compareFlottantDecroissant
 - fonction.c, 55
 - fonction.h, 61
- creationPreferences
 - emplacement_fichier.c, 31
 - emplacement_fichier.h, 37
- creerFichierStruct
 - structure.c, 126
 - structure.h, 133
- debNouvTour
 - structure.c, 126
 - structure.h, 134
- depScoreMax
 - structure.c, 127
 - structure.h, 134
- distribue
 - Fichier_Jeu, 5
- EXTENSION_FICHIER
 - fichier.h, 48
- easterEggs
 - interface.h, 81
- ecrireFichier
 - fichier.c, 42
 - fichier.h, 48
- emplacement_fichier.c, 29
 - changerCheminFichier, 31
 - creationPreferences, 31
 - lectureCheminFichier, 32
 - lecturePreferences, 33
- emplacement_fichier.h, 34
 - changerCheminFichier, 37
 - creationPreferences, 37
 - lectureCheminFichier, 38
 - lecturePreferences, 39
 - NOM_DOSSIER, 37
 - NOM_FICHIER, 37
 - PORTABLE, 37
- FAUX
 - fonction.h, 61
- fermeeFichierStruct
 - structure.c, 127
 - structure.h, 136
- fermerFichier
 - fonction.c, 55
 - fonction.h, 63
- fichier.c, 40
 - ecrireFichier, 42
 - lireFichier, 42
 - nouveauScore, 43
 - ouvrirFichierExtension, 44
 - renommerFichier, 45
 - supprimerFichier, 45
- fichier.h, 46
 - EXTENSION_FICHIER, 48
 - ecrireFichier, 48
 - lireFichier, 49
 - nouveauScore, 50
 - ouvrirFichierExtension, 51
 - renommerFichier, 51
 - supprimerFichier, 52
 - TYPE_FICHIER, 48
- Fichier_Jeu, 5
 - annee, 5
 - distribue, 5
 - jour, 5
 - mois, 6
 - nb_joueur, 6
 - nb_max, 6
 - nb_tour, 6
 - nom_joueur, 6
 - point, 6
 - point_tot, 6
 - position, 6
 - sens_premier, 6
 - taille_max_nom, 6
 - version, 6
- finNouvTour
 - structure.c, 129
 - structure.h, 136
- fonction.c, 52
 - compareFlottantCroissant, 54
 - compareFlottantDecroissant, 55
 - fermerFichier, 55
 - lireTailleFichier, 56
 - mauvais_choix, 56
 - myAlloc, 57
 - myRealloc, 57
 - ouvrirFichier, 57
 - systemEfface, 58
 - systemPause, 58
- fonction.h, 59
 - compareFlottantCroissant, 61
 - compareFlottantDecroissant, 61
 - FAUX, 61
 - fermerFichier, 63
 - lireTailleFichier, 63
 - mauvais_choix, 64
 - myAlloc, 64
 - myRealloc, 65
 - ouvrirFichier, 65
 - systemEfface, 65
 - systemPause, 66
 - VRAI, 61
- interface.h
 - affFich, 81
 - charPart, 81
 - easterEggs, 81
 - listFich, 81
 - nouvPart, 81
 - quit, 81
 - supprFich, 81

- interface.c, [67](#)
 - afficheFichier, [69](#)
 - afficheFichierLocale, [70](#)
 - chargerPartie, [71](#)
 - chargerPartieLocale, [72](#)
 - jouer, [73](#)
 - lireCheminFichier, [74](#)
 - listerFichier, [74](#)
 - menuPrincipal, [75](#)
 - nouveauCheminFichier, [76](#)
 - nouvellePartie, [77](#)
 - supprimerFichierNom, [78](#)
- interface.h, [79](#)
 - afficheFichier, [81](#)
 - afficheFichierLocale, [82](#)
 - chargerPartie, [83](#)
 - chargerPartieLocale, [84](#)
 - jouer, [85](#)
 - lireCheminFichier, [86](#)
 - listerFichier, [87](#)
 - Menu, [81](#)
 - menuPrincipal, [88](#)
 - nouveauCheminFichier, [89](#)
 - nouvellePartie, [89](#)
 - supprimerFichierNom, [90](#)
- jouer
 - interface.c, [73](#)
 - interface.h, [85](#)
- jour
 - Fichier_Jeu, [5](#)
- LECTURE_FICHER
 - argument_main.h, [29](#)
- lectureCheminFichier
 - emplacement_fichier.c, [32](#)
 - emplacement_fichier.h, [38](#)
- lecturePreferences
 - emplacement_fichier.c, [33](#)
 - emplacement_fichier.h, [39](#)
- lireCheminFichier
 - interface.c, [74](#)
 - interface.h, [86](#)
- lireFichier
 - fichier.c, [42](#)
 - fichier.h, [49](#)
- lireTailleFichier
 - fonction.c, [56](#)
 - fonction.h, [63](#)
- listFich
 - interface.h, [81](#)
- listerFichier
 - interface.c, [74](#)
 - interface.h, [87](#)
- main
 - main.c, [92](#)
 - main.h, [95](#)
- main.c, [91](#)
 - main, [92](#)
- main.h, [94](#)
 - main, [95](#)
- mauvais_choix
 - fonction.c, [56](#)
 - fonction.h, [64](#)
- maxNbTour
 - structure.c, [129](#)
 - structure.h, [137](#)
- Menu
 - interface.h, [81](#)
- menu.c, [96](#)
 - menuContinuer, [98](#)
 - menuDebutPartie, [98](#)
 - menuDistribue, [99](#)
 - menuNomFichier, [99](#)
 - menuNomJoueur, [100](#)
 - menuNouveauChemin, [101](#)
 - menuPointsJoueur, [101](#)
 - menuSupprimer, [102](#)
- menu.h, [103](#)
 - menuContinuer, [105](#)
 - menuDebutPartie, [105](#)
 - menuDistribue, [106](#)
 - menuNomFichier, [106](#)
 - menuNomJoueur, [107](#)
 - menuNouveauChemin, [108](#)
 - menuPointsJoueur, [108](#)
 - menuSupprimer, [109](#)
- menuContinuer
 - menu.c, [98](#)
 - menu.h, [105](#)
- menuDebutPartie
 - menu.c, [98](#)
 - menu.h, [105](#)
- menuDistribue
 - menu.c, [99](#)
 - menu.h, [106](#)
- menuNomFichier
 - menu.c, [99](#)
 - menu.h, [106](#)
- menuNomJoueur
 - menu.c, [100](#)
 - menu.h, [107](#)
- menuNouveauChemin
 - menu.c, [101](#)
 - menu.h, [108](#)
- menuPointsJoueur
 - menu.c, [101](#)
 - menu.h, [108](#)
- menuPrincipal
 - interface.c, [75](#)
 - interface.h, [88](#)
- menuSupprimer
 - menu.c, [102](#)
 - menu.h, [109](#)
- mois
 - Fichier_Jeu, [6](#)

myAlloc
 fonction.c, 57
 fonction.h, 64
myRealloc
 fonction.c, 57
 fonction.h, 65
NB_CARACT_DOUB
 saisie_clavier.h, 118
NB_CARACT_FLOT
 saisie_clavier.h, 118
NB_CARACT_INT
 saisie_clavier.h, 118
NOM_DOSSIER
 emplacement_fichier.h, 37
NOM_FICHER
 emplacement_fichier.h, 37
nb_joueur
 Fichier_Jeu, 6
nb_max
 Fichier_Jeu, 6
nb_tour
 Fichier_Jeu, 6
nom_joueur
 Fichier_Jeu, 6
nouvPart
 interface.h, 81
nouveauCheminFichier
 interface.c, 76
 interface.h, 89
nouveauScore
 fichier.c, 43
 fichier.h, 50
nouvellePartie
 interface.c, 77
 interface.h, 89
OUVERTURE_FICHER
 argument_main.h, 29
ouvrirFichier
 fonction.c, 57
 fonction.h, 65
ouvrirFichierExtension
 fichier.c, 44
 fichier.h, 51
PORTABLE
 emplacement_fichier.h, 37
point
 Fichier_Jeu, 6
point_tot
 Fichier_Jeu, 6
position
 Fichier_Jeu, 6
quit
 interface.h, 81
renommerFichier
 fichier.c, 45
 fichier.h, 51
saisie_clavier.c, 110
 clean_stdin, 111
 saisieClavierCaractere, 111
 saisieClavierChaine, 112
 saisieClavierDouble, 113
 saisieClavierEntier, 113
 saisieClavierFlottant, 114
 saisieClavierFlottantSansVirgule, 114
saisie_clavier.h, 116
 clean_stdin, 118
 NB_CARACT_DOUB, 118
 NB_CARACT_FLOT, 118
 NB_CARACT_INT, 118
 saisieClavierCaractere, 118
 saisieClavierChaine, 119
 saisieClavierDouble, 120
 saisieClavierEntier, 120
 saisieClavierFlottant, 121
 saisieClavierFlottantSansVirgule, 121
saisieClavierCaractere
 saisie_clavier.c, 111
 saisie_clavier.h, 118
saisieClavierChaine
 saisie_clavier.c, 112
 saisie_clavier.h, 119
saisieClavierDouble
 saisie_clavier.c, 113
 saisie_clavier.h, 120
saisieClavierEntier
 saisie_clavier.c, 113
 saisie_clavier.h, 120
saisieClavierFlottant
 saisie_clavier.c, 114
 saisie_clavier.h, 121
saisieClavierFlottantSansVirgule
 saisie_clavier.c, 114
 saisie_clavier.h, 121
searchArgument
 argument_main.c, 26
 argument_main.h, 29
sens_premier
 Fichier_Jeu, 6
structure.c, 123
 ajoutDistribueStruct, 125
 calculPosition, 125
 creerFichierStruct, 126
 debNouvTour, 126
 depScoreMax, 127
 fermeeFichierStruct, 127
 finNouvTour, 129
 maxNbTour, 129
structure.h, 130
 ajoutDistribueStruct, 132
 calculPosition, 132
 creerFichierStruct, 133
 debNouvTour, 134

- depScoreMax, [134](#)
- fermeeFichierStruct, [136](#)
- finNouvTour, [136](#)
- maxNbTour, [137](#)
- TAILLE_MAX_NOM, [132](#)
- VERSION, [132](#)
- supprFich
 - interface.h, [81](#)
- supprimerFichier
 - fichier.c, [45](#)
 - fichier.h, [52](#)
- supprimerFichierNom
 - interface.c, [78](#)
 - interface.h, [90](#)
- systemEfface
 - fonction.c, [58](#)
 - fonction.h, [65](#)
- systemPause
 - fonction.c, [58](#)
 - fonction.h, [66](#)
- TAILLE_MAX_NOM
 - structure.h, [132](#)
- TYPE_FICHIER
 - fichier.h, [48](#)
- taille_max_nom
 - Fichier_Jeu, [6](#)
- VERSION
 - structure.h, [132](#)
- VRAI
 - fonction.h, [61](#)
- version
 - Fichier_Jeu, [6](#)