

Csuper - Compteur de Score Universel Permettant l'Exemption de Reflexion
2.0

Generated by Doxygen 1.8.6

Sat Feb 15 2014 19:37:05

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	13
4.1.2.7	afficherPartieFinie	13
4.1.2.8	afficherPosition	14
4.1.2.9	afficherScore	14
4.1.2.10	afficherScoreEntier	15
4.1.2.11	afficherScoreTotal	16
4.1.2.12	afficherStruct	16
4.2	affichage.h File Reference	17
4.2.1	Detailed Description	19
4.2.2	Function Documentation	19
4.2.2.1	afficherChaineTroisTab	19
4.2.2.2	afficherDistribue	19
4.2.2.3	afficherEnTete	20
4.2.2.4	afficherLicense	20
4.2.2.5	afficherLigne	21
4.2.2.6	afficherNom	22
4.2.2.7	afficherPartieFinie	22
4.2.2.8	afficherPosition	23
4.2.2.9	afficherScore	23
4.2.2.10	afficherScoreEntier	24
4.2.2.11	afficherScoreTotal	25
4.2.2.12	afficherStruct	25
4.3	emplacement_fichier.c File Reference	26
4.3.1	Detailed Description	27
4.3.2	Function Documentation	28
4.3.2.1	changerCheminFichier	28
4.3.2.2	creationPreferences	28
4.3.2.3	lectureCheminFichier	29
4.3.2.4	lecturePreferences	30
4.4	emplacement_fichier.h File Reference	31
4.4.1	Detailed Description	33
4.4.2	Macro Definition Documentation	33
4.4.2.1	NOM_DOSSIER	33
4.4.2.2	NOM_FICHER	33
4.4.3	Function Documentation	33
4.4.3.1	changerCheminFichier	33
4.4.3.2	creationPreferences	34
4.4.3.3	lectureCheminFichier	35

4.4.3.4	lecturePreferences	36
4.5	fichier.c File Reference	37
4.5.1	Detailed Description	38
4.5.2	Function Documentation	38
4.5.2.1	ecrireFichier	38
4.5.2.2	lireFichier	39
4.5.2.3	nouveauScore	40
4.5.2.4	ouvrirFichierExtension	41
4.5.2.5	renommerFichier	42
4.5.2.6	supprimerFichier	42
4.6	fichier.h File Reference	43
4.6.1	Detailed Description	44
4.6.2	Macro Definition Documentation	45
4.6.2.1	EXTENSION_FICHIER	45
4.6.2.2	TAILLE_MAX_NOM_FICHIER	45
4.6.2.3	TYPE_FICHIER	45
4.6.3	Function Documentation	45
4.6.3.1	ecrireFichier	45
4.6.3.2	lireFichier	46
4.6.3.3	nouveauScore	47
4.6.3.4	ouvrirFichierExtension	48
4.6.3.5	renommerFichier	49
4.6.3.6	supprimerFichier	49
4.7	fonction.c File Reference	50
4.7.1	Detailed Description	51
4.7.2	Function Documentation	51
4.7.2.1	compareFlottantCroissant	51
4.7.2.2	compareFlottantDecroissant	51
4.7.2.3	fermerFichier	52
4.7.2.4	lireTailleFichier	53
4.7.2.5	mauvais_choix	53
4.7.2.6	ouvrirFichier	54
4.7.2.7	systemEfface	54
4.7.2.8	systemPause	55
4.8	fonction.h File Reference	56
4.8.1	Detailed Description	57
4.8.2	Macro Definition Documentation	58
4.8.2.1	FAUX	58
4.8.2.2	VRAI	58
4.8.3	Function Documentation	58

4.8.3.1	compareFlottantCroissant	58
4.8.3.2	compareFlottantDecroissant	58
4.8.3.3	fermerFichier	58
4.8.3.4	lireTailleFichier	59
4.8.3.5	mauvais_choix	59
4.8.3.6	ouvrirFichier	60
4.8.3.7	systemEfface	60
4.8.3.8	systemPause	61
4.9	interface.c File Reference	62
4.9.1	Detailed Description	63
4.9.2	Function Documentation	63
4.9.2.1	afficheFichier	63
4.9.2.2	ChargerPartie	64
4.9.2.3	jouer	65
4.9.2.4	lireCheminFichier	66
4.9.2.5	listerFichier	67
4.9.2.6	menuPrincipal	68
4.9.2.7	nouveauCheminFichier	69
4.9.2.8	nouvellePartie	69
4.9.2.9	supprimerFichierNom	70
4.10	interface.h File Reference	71
4.10.1	Detailed Description	73
4.10.2	Enumeration Type Documentation	73
4.10.2.1	Menu	73
4.10.3	Function Documentation	73
4.10.3.1	afficheFichier	73
4.10.3.2	ChargerPartie	74
4.10.3.3	jouer	75
4.10.3.4	lireCheminFichier	76
4.10.3.5	listerFichier	77
4.10.3.6	menuPrincipal	78
4.10.3.7	nouveauCheminFichier	79
4.10.3.8	nouvellePartie	79
4.10.3.9	supprimerFichierNom	80
4.11	main.c File Reference	81
4.11.1	Detailed Description	82
4.11.2	Function Documentation	82
4.11.2.1	main	82
4.12	menu.c File Reference	83
4.12.1	Detailed Description	84

4.12.2	Function Documentation	85
4.12.2.1	menuContinuer	85
4.12.2.2	menuDebutPartie	85
4.12.2.3	menuDistribue	86
4.12.2.4	menuNomFichier	87
4.12.2.5	menuNomJoueur	88
4.12.2.6	menuNouveauChemin	89
4.12.2.7	menuPointsJoueur	89
4.12.2.8	menuSupprimer	90
4.13	menu.h File Reference	91
4.13.1	Detailed Description	92
4.13.2	Function Documentation	92
4.13.2.1	menuContinuer	92
4.13.2.2	menuDebutPartie	93
4.13.2.3	menuDistribue	94
4.13.2.4	menuNomFichier	94
4.13.2.5	menuNomJoueur	95
4.13.2.6	menuNouveauChemin	96
4.13.2.7	menuPointsJoueur	96
4.13.2.8	menuSupprimer	97
4.14	saisie_clavier.c File Reference	98
4.14.1	Detailed Description	98
4.14.2	Function Documentation	99
4.14.2.1	clean_stdin	99
4.14.2.2	saisieClavierCaractere	99
4.14.2.3	saisieClavierChaine	100
4.14.2.4	saisieClavierDouble	101
4.14.2.5	saisieClavierEntier	101
4.14.2.6	saisieClavierFlottant	102
4.14.2.7	saisieClavierFlottantSansVirgule	102
4.15	saisie_clavier.h File Reference	103
4.15.1	Detailed Description	104
4.15.2	Macro Definition Documentation	105
4.15.2.1	NB_CARACT_DOUB	105
4.15.2.2	NB_CARACT_FLOT	105
4.15.2.3	NB_CARACT_INT	105
4.15.3	Function Documentation	105
4.15.3.1	clean_stdin	105
4.15.3.2	saisieClavierCaractere	105
4.15.3.3	saisieClavierChaine	106

4.15.3.4	saisieClavierDouble	107
4.15.3.5	saisieClavierEntier	107
4.15.3.6	saisieClavierFlottant	108
4.15.3.7	saisieClavierFlottantSansVirgule	109
4.16	structure.c File Reference	110
4.16.1	Detailed Description	111
4.16.2	Function Documentation	112
4.16.2.1	ajoutDistribueStruct	112
4.16.2.2	calculPosition	113
4.16.2.3	creerFichierStruct	113
4.16.2.4	debNouvTour	114
4.16.2.5	depScoreMax	114
4.16.2.6	fermeeFichierStruct	115
4.16.2.7	finNouvTour	115
4.17	structure.h File Reference	116
4.17.1	Detailed Description	117
4.17.2	Macro Definition Documentation	118
4.17.2.1	TAILLE_MAX_NOM	118
4.17.2.2	VERSION	118
4.17.3	Function Documentation	118
4.17.3.1	ajoutDistribueStruct	118
4.17.3.2	calculPosition	118
4.17.3.3	creerFichierStruct	119
4.17.3.4	debNouvTour	119
4.17.3.5	depScoreMax	120
4.17.3.6	fermeeFichierStruct	120
4.17.3.7	finNouvTour	121
Index		122

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
emplacement_fichier.c	Fonctions qui l'emplacement des fichiers sauvegards	26
emplacement_fichier.h	Prototypes des fonctions qui l'emplacement des fichiers sauvegards	31
fichier.c	Fonction de gestion des fichiers	37
fichier.h	Prototypes des fonction de gestion des fichiers	43
fonction.c	Fonctions essentielles au fonctionnement du programme	50
fonction.h	Prototypes des fonctions essentielles au fonctionnement du programme	56
interface.c	Fonctions qui gerent l'interface graphique du logiciel	62
interface.h	Prototypes des fonctions qui gerent l'interface graphique du logiciel	71
main.c	Lancement du programme	81
menu.c	Fonctions qui gerent les menus demandant de rentrer des valeurs du logiciel	83
menu.h	Prototypes des fonctions qui gerent les menus demandant de rentrer des valeurs du logiciel	91
saisie_clavier.c	Fonctions de saisie clavier	98
saisie_clavier.h	Prototypes des fonctions de saisie clavier	103
structure.c	Fonction de gestion des fichiers de la struction contenant les informations	110
structure.h	Prototypes des fonction de gestion des fichiers de la struction contenant les informations	116

Chapter 3

Data Structure Documentation

3.1 Fichier_Jeu Struct Reference

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

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.

3.1.2.8 char** nom_joueur

Tableau contenant tout les noms de joueurs.

3.1.2.9 float* point

Tableau contenat les points de chaque joueur a chaque tour.

3.1.2.10 float* point_tot

Tableau contenant tout les points totaux des joueurs.

3.1.2.11 float* position

Tableau contenant la position des joueurs.

3.1.2.12 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

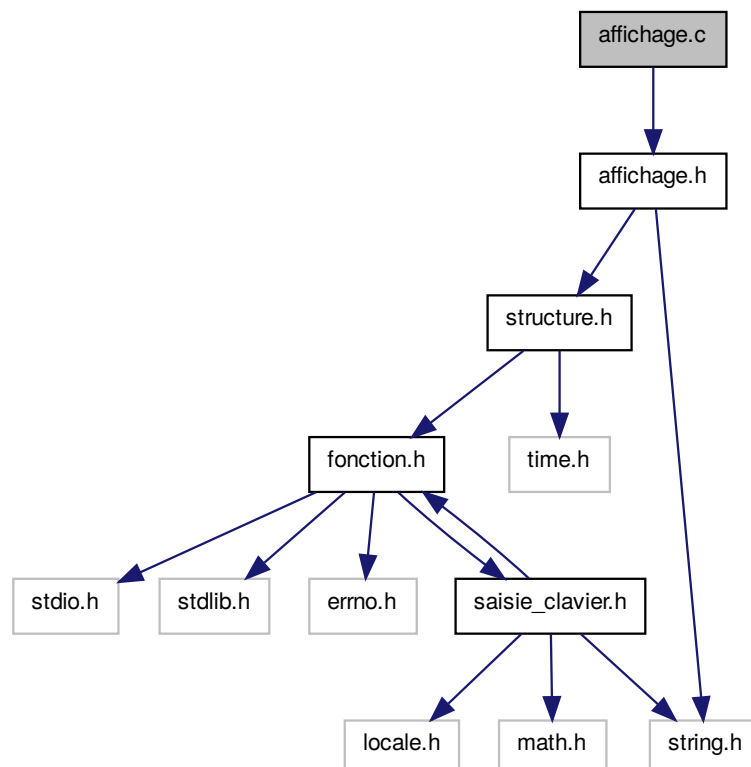
File Documentation

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

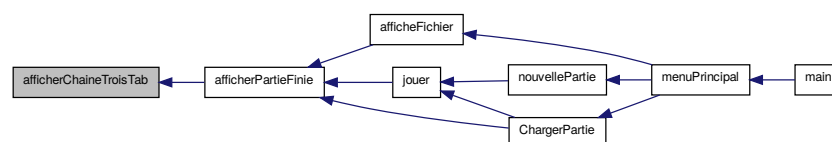
4.1.2.1 void [afficherChaineTroisTab](#) (char * *chaine*)

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:



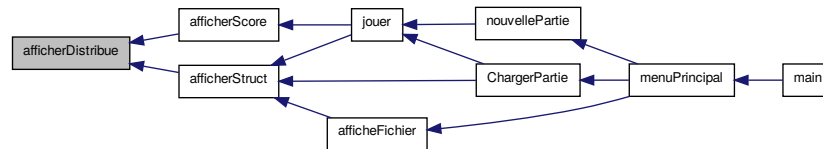
4.1.2.2 void [afficherDistribue](#) ([Fichier_Jeu](#) * *ptr_struct_fichier*)

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:



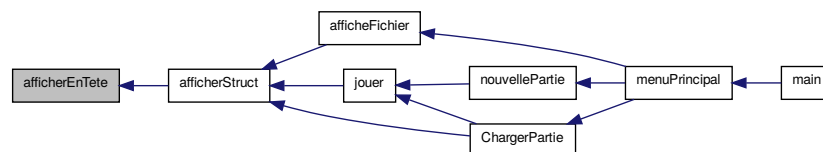
4.1.2.3 void afficherEnTete (Fichier_Jeu * ptr_struct_fichier)

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:



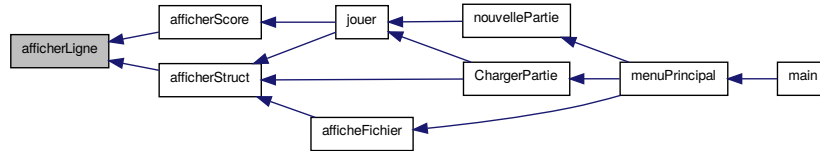
4.1.2.5 void afficherLigne (int *taille_ligne*)

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:



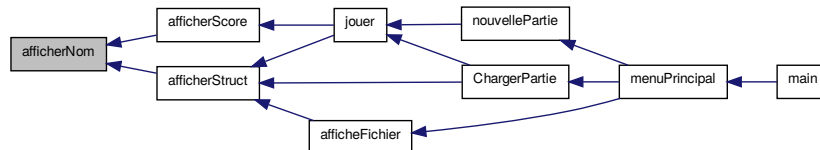
4.1.2.6 void afficherNom (Fichier_Jeu * ptr_struct_fichier, int * ptr_taille_ligne)

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:



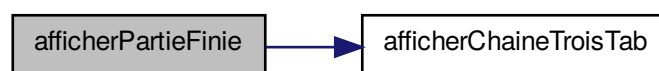
4.1.2.7 void afficherPartieFinie (Fichier_Jeu * ptr_struct_fichier)

Affiche un podium des resultats

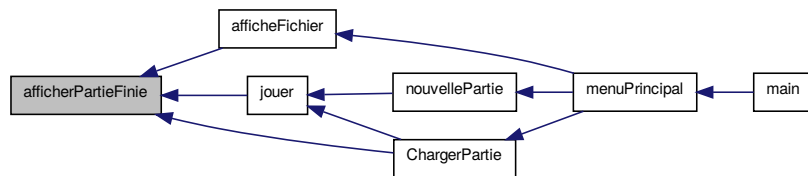
Parameters

in	<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:



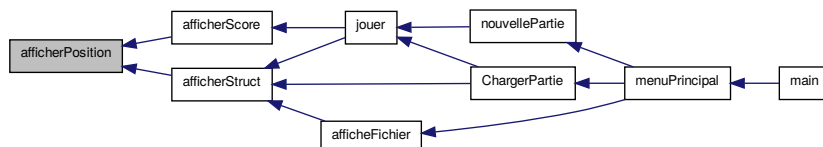
4.1.2.8 void afficherPosition (Fichier_Jeu * ptr_struct_fichier)

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:



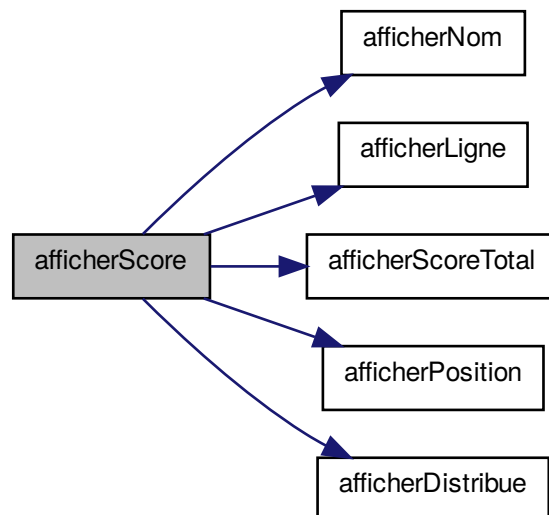
4.1.2.9 void afficherScore (Fichier_Jeu * ptr_struct_fichier)

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

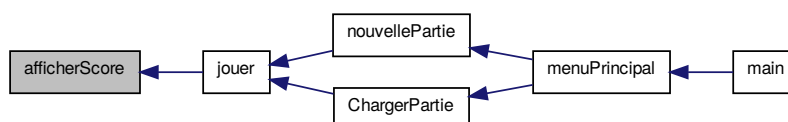
Parameters

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:



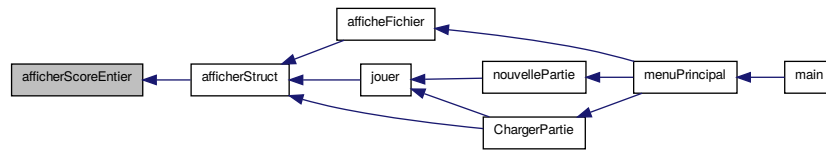
4.1.2.10 void afficherScoreEntier (Fichier_Jeu * ptr_struct_fichier)

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 caller graph for this function:



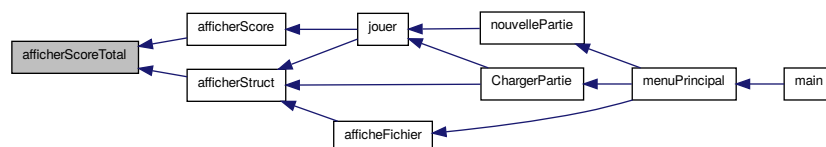
4.1.2.11 void afficherScoreTotal (Fichier_Jeu * ptr_struct_fichier)

Affiche le score total des joueurs

Parameters

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

Here is the caller graph for this function:



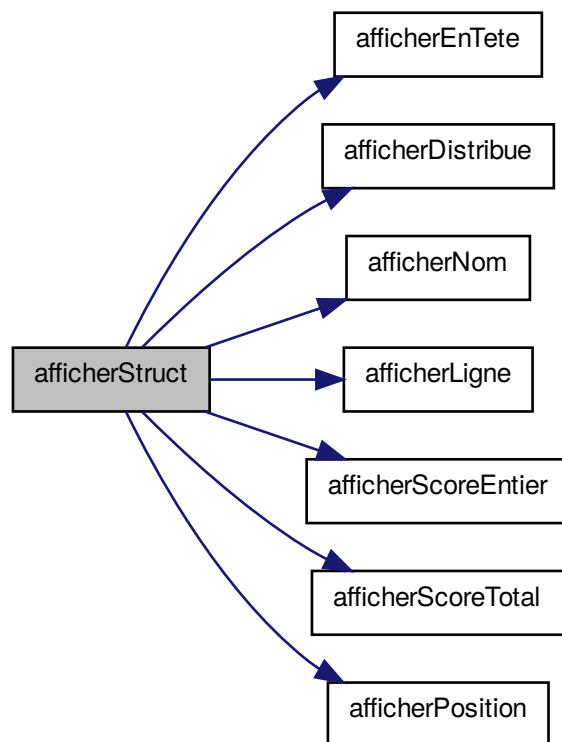
4.1.2.12 void afficherStruct (Fichier_Jeu * ptr_struct_fichier)

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

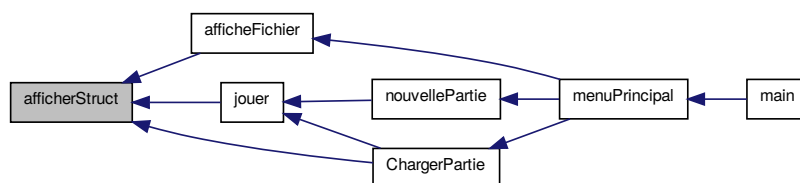
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:

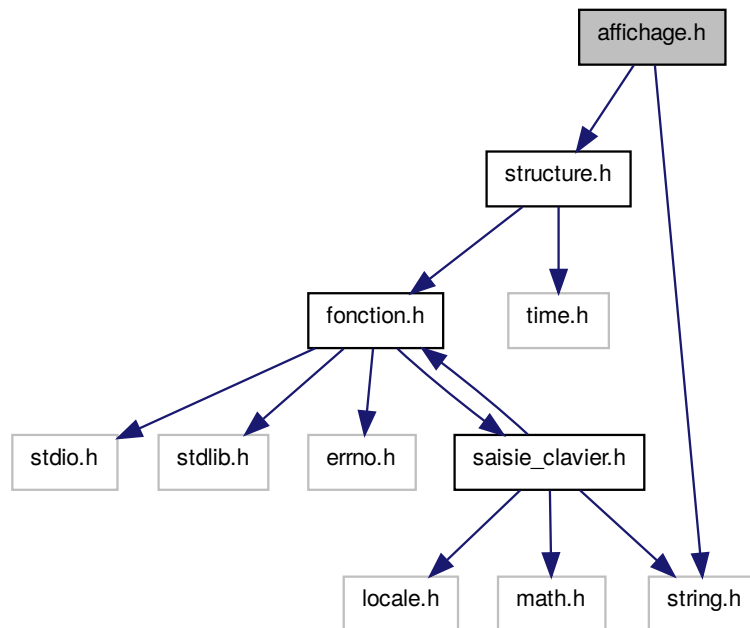


4.2 affichage.h File Reference

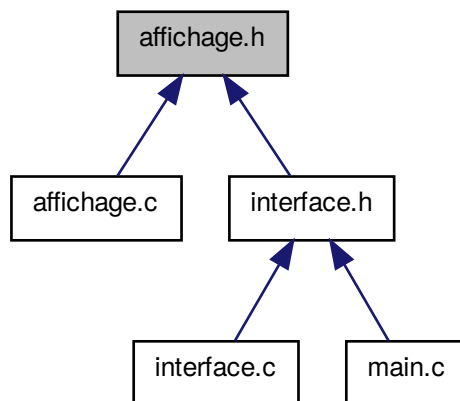
Prototypes des fonctions qui gerent l'affichage des scores du logiciel.

```
#include "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

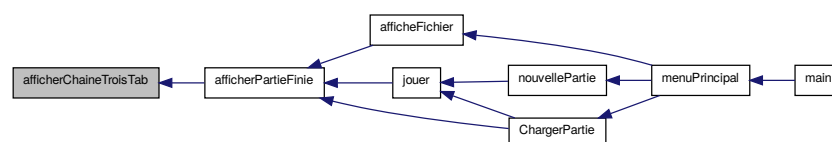
4.2.2.1 void [afficherChaineTroisTab](#) (char * *chaine*)

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:



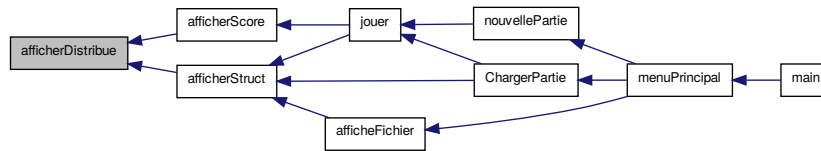
4.2.2.2 void [afficherDistribue](#) ([Fichier_Jeu](#) * *ptr_struct_fichier*)

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:



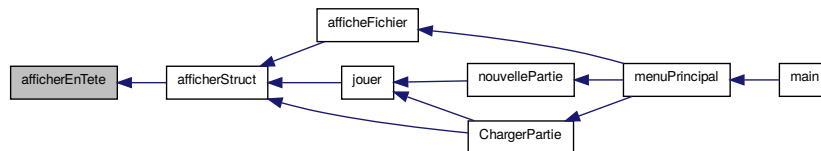
4.2.2.3 void afficherEnTete (Fichier_Jeu * ptr_struct_fichier)

Affiche l'en tete de la structure

Parameters

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

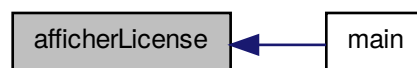
Here is the caller graph for this function:



4.2.2.4 void afficherLicense ()

Affiche la license

Here is the caller graph for this function:



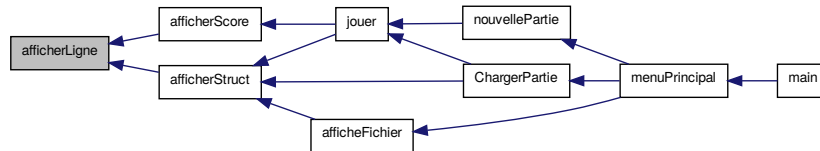
4.2.2.5 void afficherLigne (int *taille_ligne*)

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:



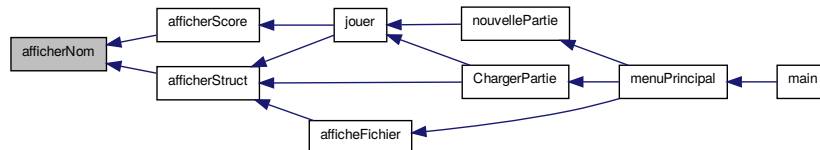
4.2.2.6 void afficherNom (Fichier_Jeu * ptr_struct_fichier, int * ptr_taille_ligne)

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

Parameters

in	<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:



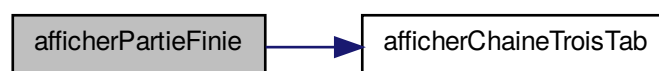
4.2.2.7 void afficherPartieFinie (Fichier_Jeu * ptr_struct_fichier)

Affiche un podium des resultats

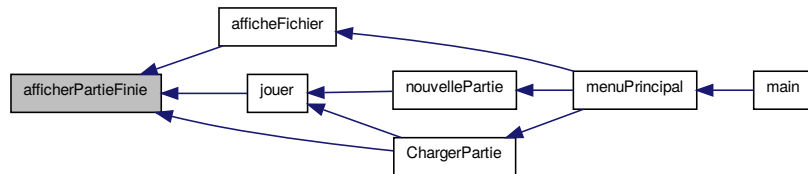
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



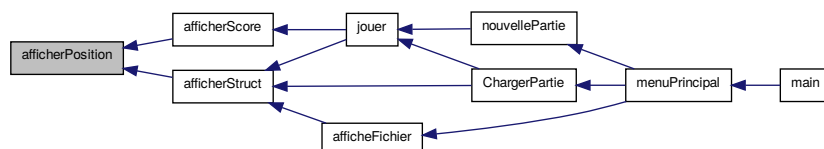
4.2.2.8 void afficherPosition (Fichier_Jeu * ptr_struct_fichier)

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:



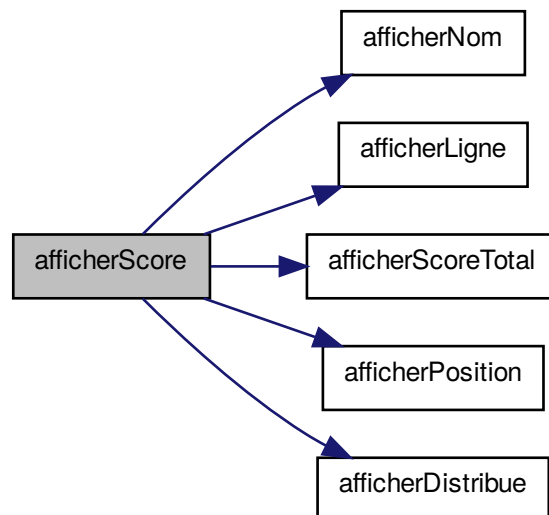
4.2.2.9 void afficherScore (Fichier_Jeu * ptr_struct_fichier)

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

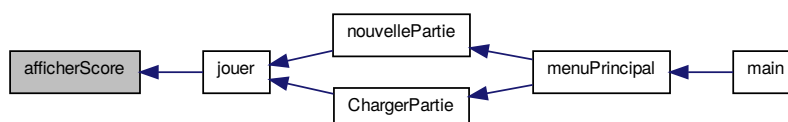
Parameters

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:



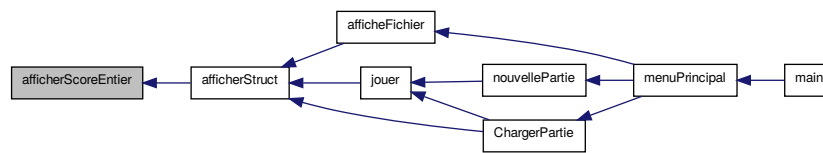
4.2.2.10 void afficherScoreEntier (Fichier_Jeu * ptr_struct_fichier)

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 caller graph for this function:



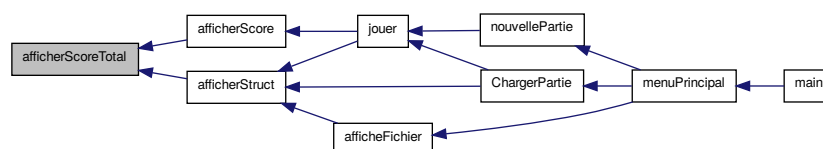
4.2.2.11 void afficherScoreTotal (Fichier_Jeu * ptr_struct_fichier)

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:



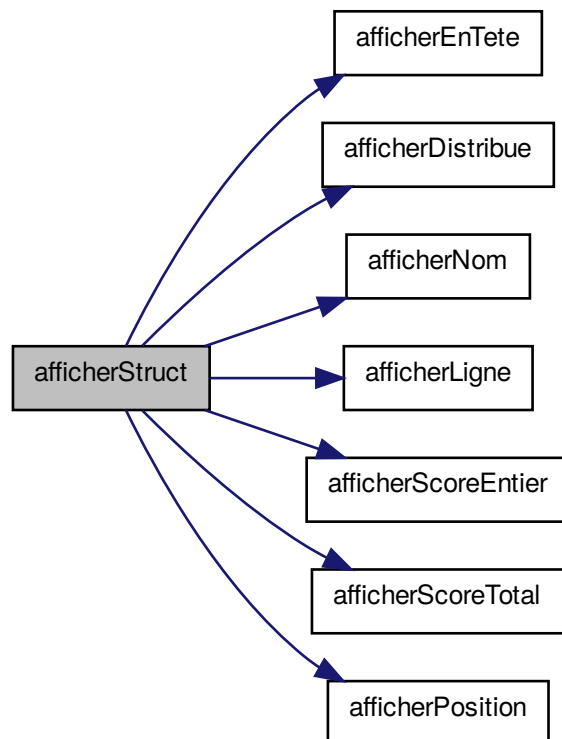
4.2.2.12 void afficherStruct (Fichier_Jeu * ptr_struct_fichier)

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

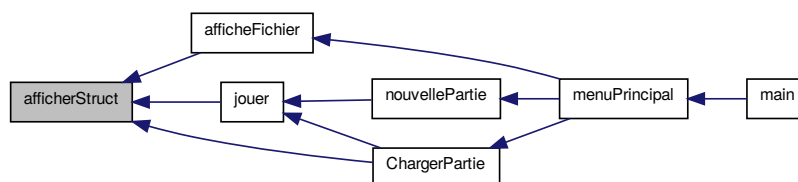
Parameters

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:

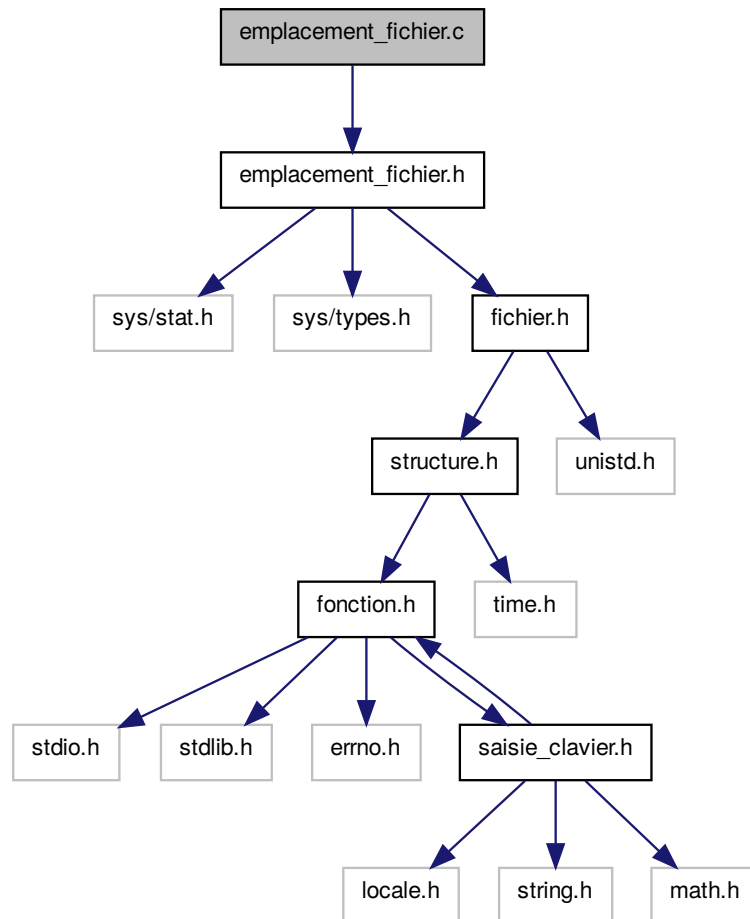


4.3 emplacement_fichier.c File Reference

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)

4.3.1 Detailed Description

Fonctions qui l'emplacement des fichiers sauvegardes.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.3.2 Function Documentation

4.3.2.1 `int changerCheminFichier (char * nouveauChemin)`

Changer le chemin de sauvegarde des fichier par nouveauChemin

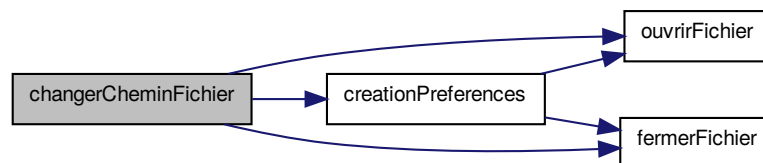
Parameters

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

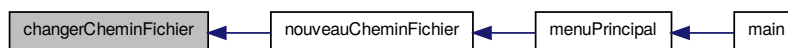
Returns

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:



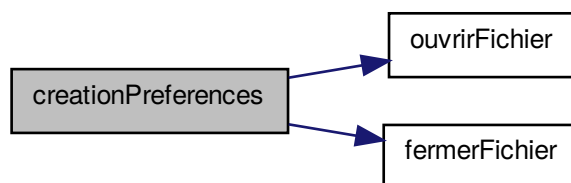
4.3.2.2 `void creationPreferences ()`

Cree le dossier et le fichier qui va contenir les preferences

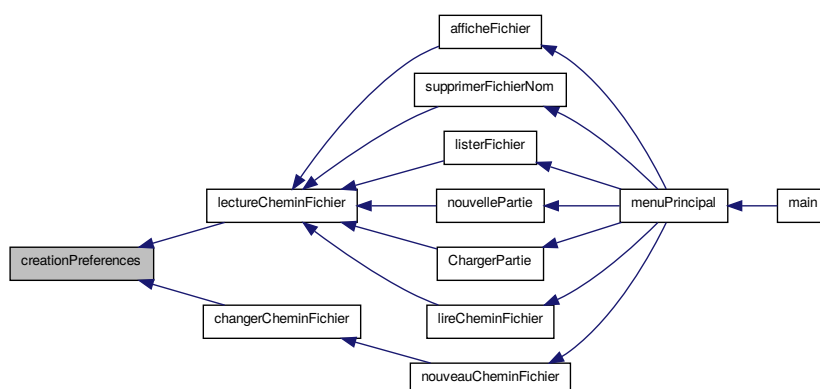
Returns

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:

**4.3.2.3 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

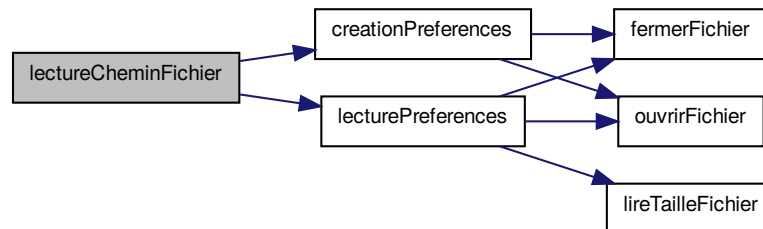
Parameters

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

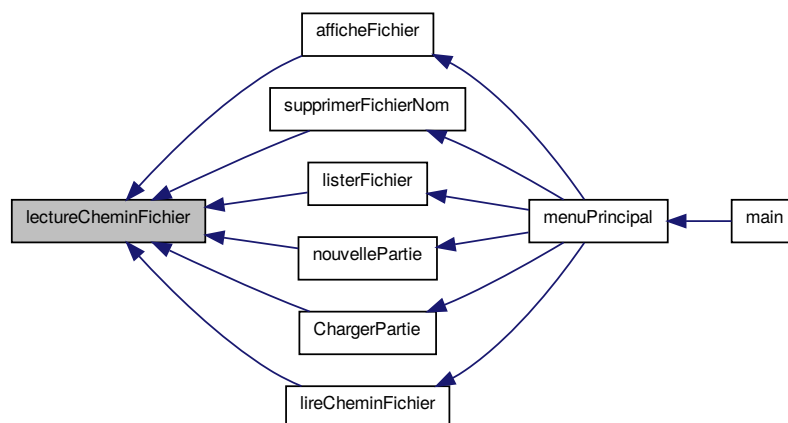
Returns

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:



4.3.2.4 int lecturePreferences (char * *nom_fichier*)

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

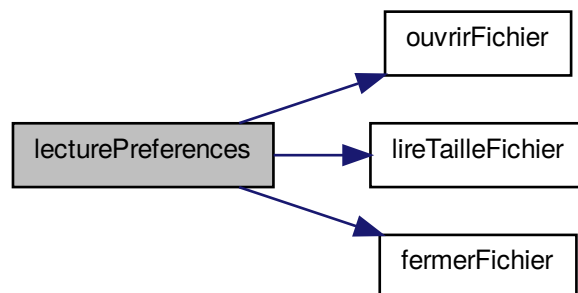
Parameters

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

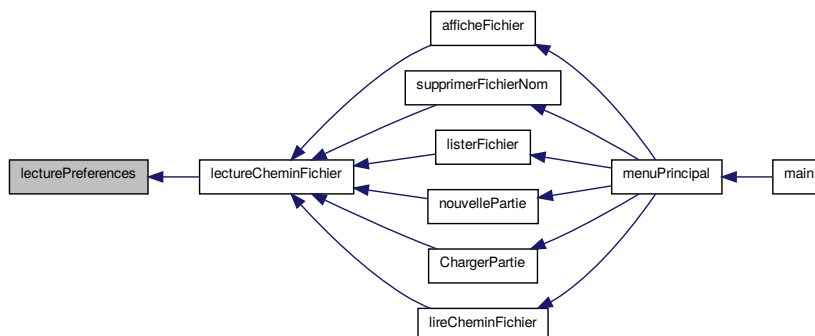
Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



Here is the caller graph for this function:

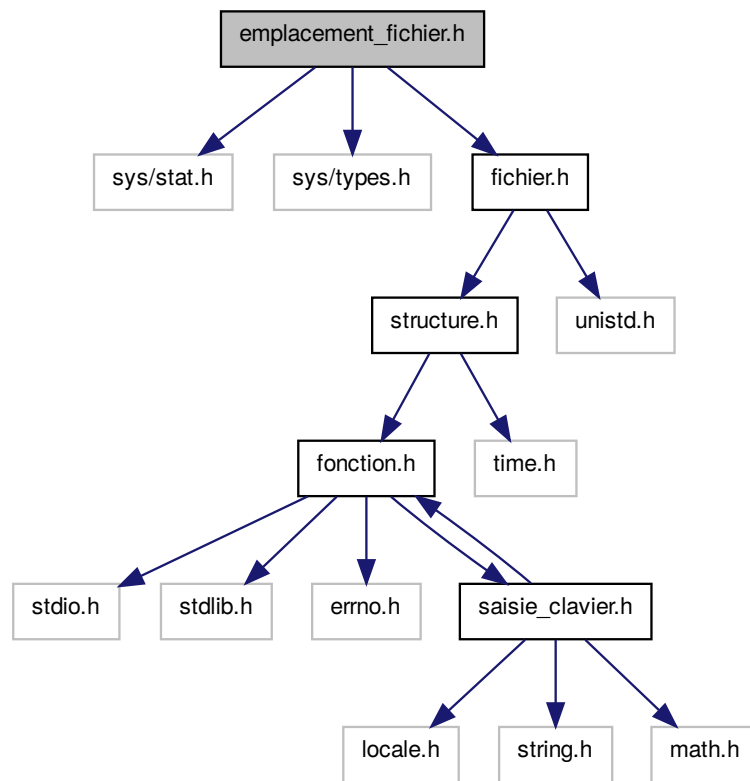


4.4 emplacement_fichier.h File Reference

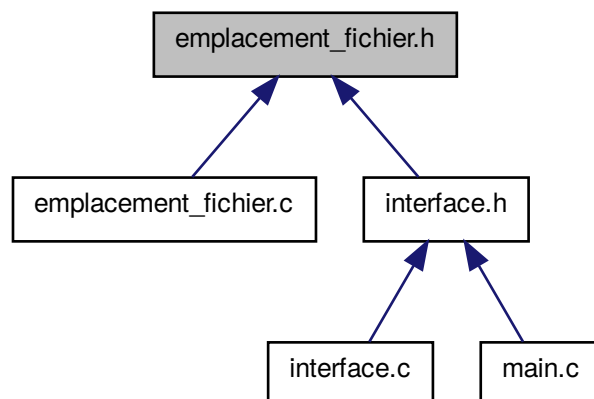
Prototypes des fonctions qui l'emplacement des fichiers sauvegardes.

```
#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 `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)

4.4.1 Detailed Description

Prototypes des fonctions qui l'emplacement des fichiers sauvegardes.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.4.2 Macro Definition Documentation

4.4.2.1 #define `NOM_DOSSIER` ".csuper"

Definit `NOM_DOSSIER` a ".csuper"

4.4.2.2 #define `NOM_FICHIER` "preferences.txt"

Definit `NOM_FICHIER` a "preferences.txt"

4.4.3 Function Documentation

4.4.3.1 int `changerCheminFichier` (char * *nouveauChemin*)

Changer le chemin de sauvegarde des fichier par nouveauChemin

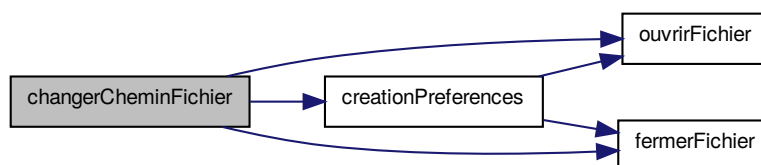
Parameters

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

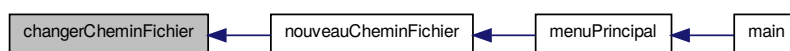
Returns

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:



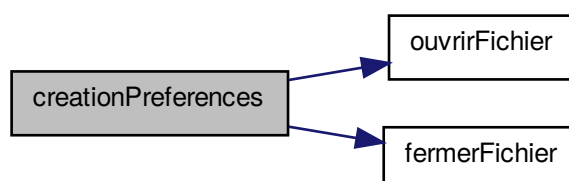
4.4.3.2 int creationPreferences ()

Cree le dossier et le fichier qui va contenir les preferences

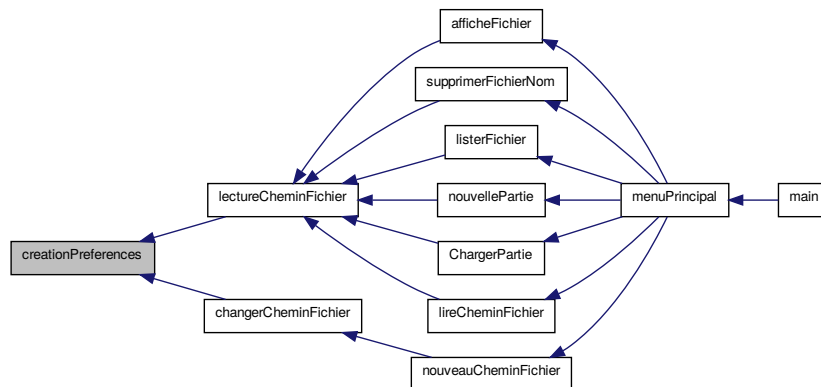
Returns

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:



4.4.3.3 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

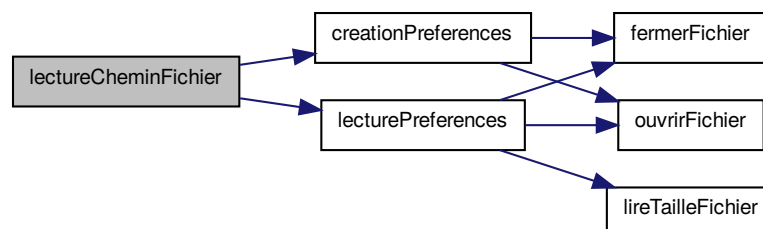
Parameters

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

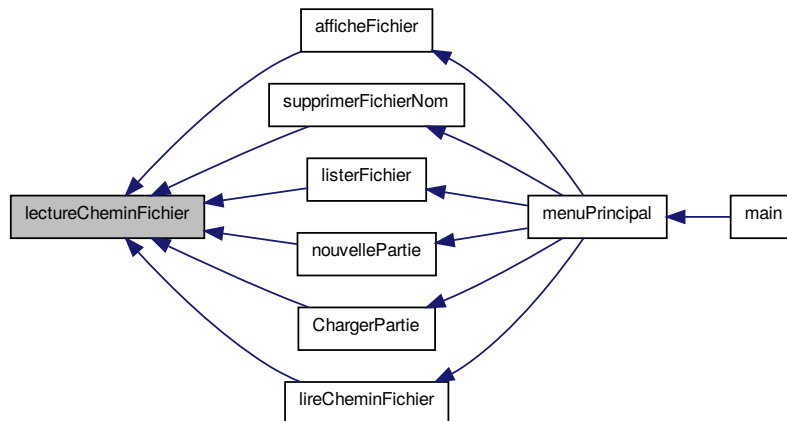
Returns

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:



4.4.3.4 int lecturePreferences (char * nom_fichier)

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

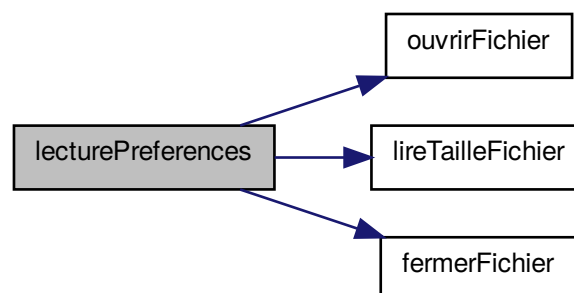
Parameters

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

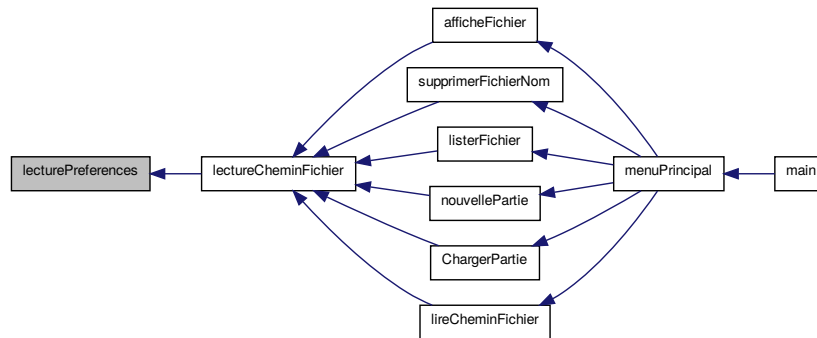
Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



Here is the caller graph for this function:

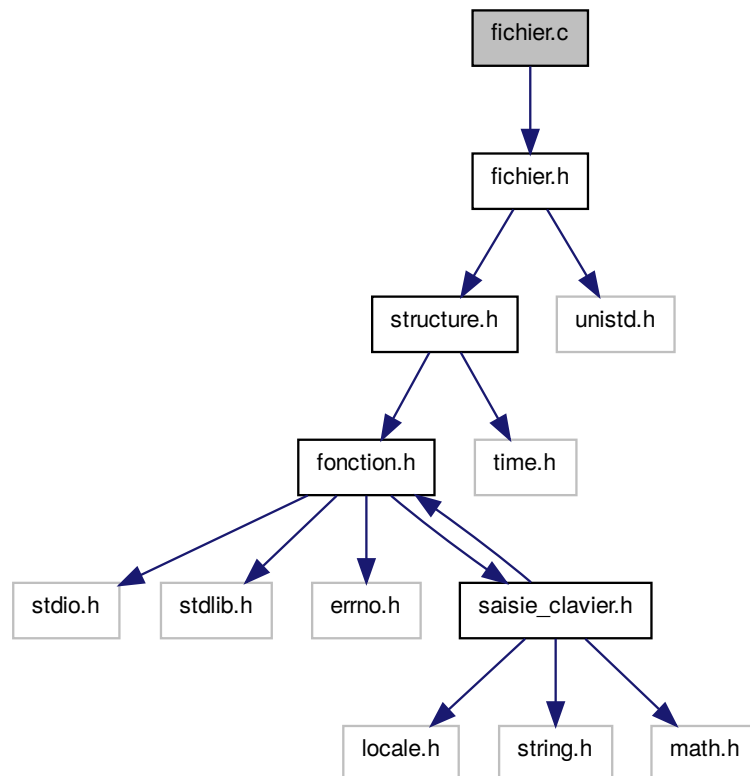


4.5 fichier.c File Reference

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)

4.5.1 Detailed Description

Fonction de gestion des fichiers.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.5.2 Function Documentation

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

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

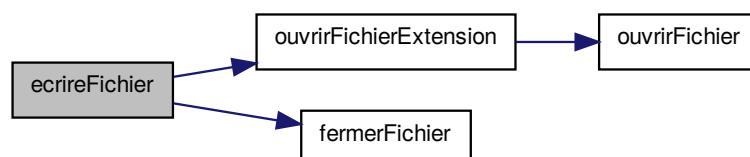
Parameters

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

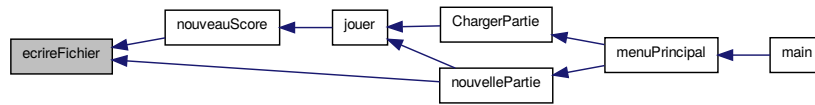
Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



Here is the caller graph for this function:



4.5.2.2 Fichier_Jeu * lireFichier (char * nom)

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

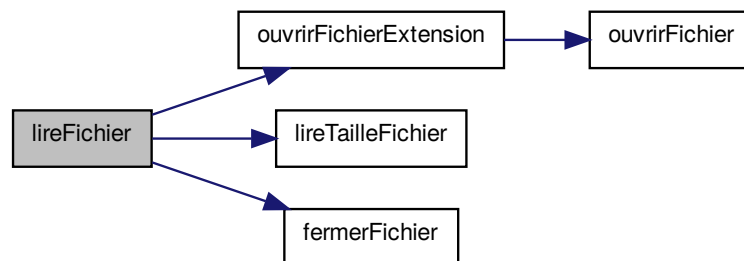
Parameters

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

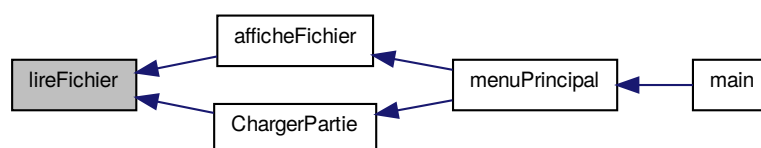
Returns

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:



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

Mets a jour le fichier avec les nouveaux scores

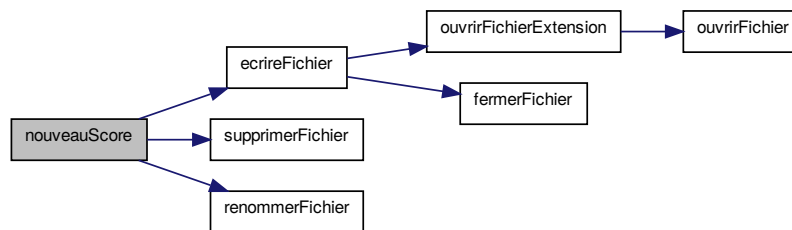
Parameters

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

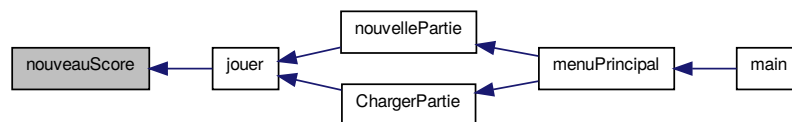
Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



Here is the caller graph for this function:



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

Ouvre un fichier a partir de son nom (nom[]) et du mod voulu (mode[]) en y ajouter l'extension du fichier

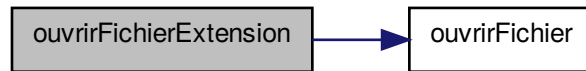
Parameters

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

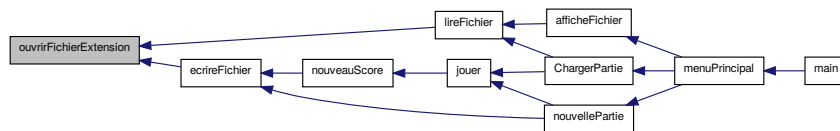
Returns

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:



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

Renomme le fichier dont le nom est en parametre

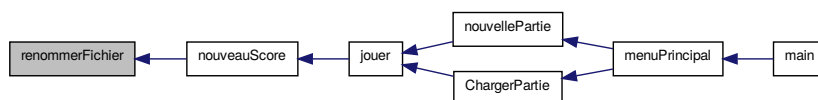
Parameters

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:



4.5.2.6 int supprimerFichier (char * *nom*)

Supprime le fichier dont le nom est en parametre

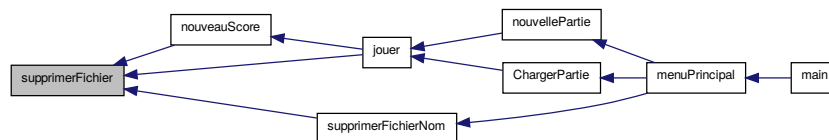
Parameters

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

Returns

VRAI si tout s'est bien passe, FAUX sinon

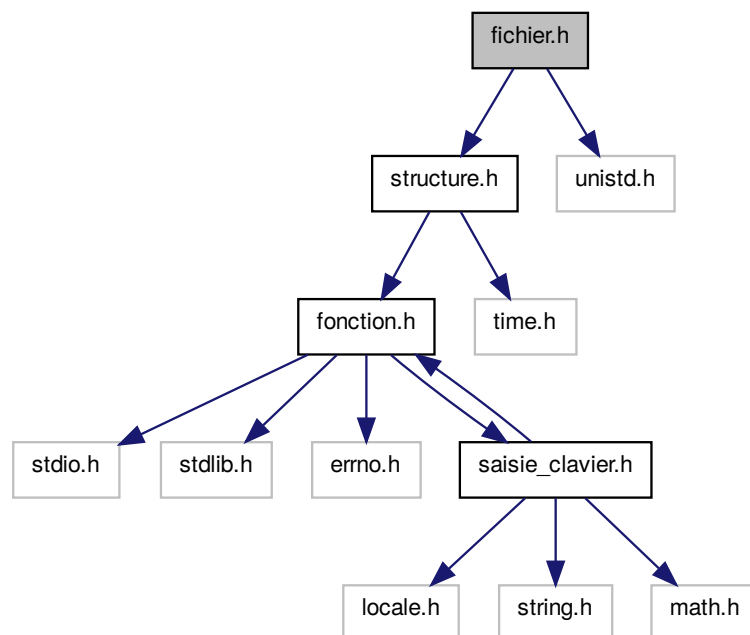
Here is the caller graph for this function:



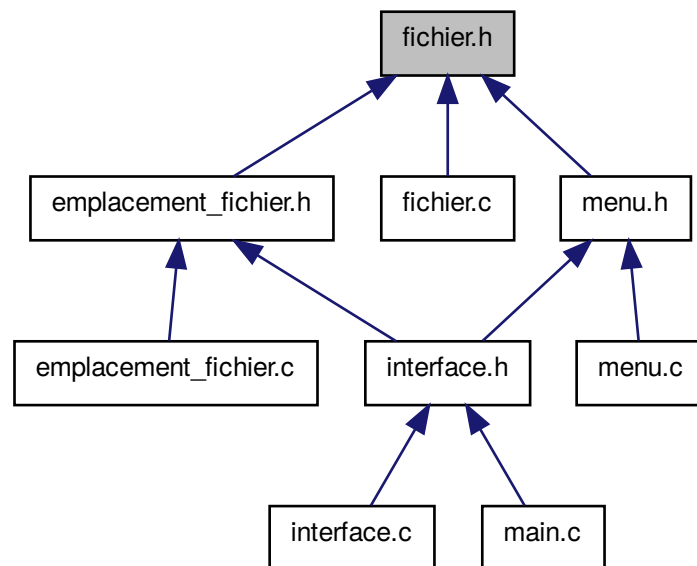
4.6 fichier.h File Reference

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)`

4.6.1 Detailed Description

Prototypes des fonction de gestion des fichiers.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.6.2 Macro Definition Documentation

4.6.2.1 #define EXTENSION_FICHIER "csu"

Definit l'extension du fichier a "csu"

4.6.2.2 #define TAILLE_MAX_NOM_FICHIER 250

Definit la taille max d'un nom a 250

4.6.2.3 #define TYPE_FICHIER "CompteurScoreUniversel"

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

4.6.3 Function Documentation

4.6.3.1 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

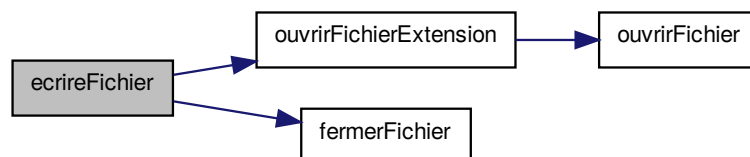
Parameters

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

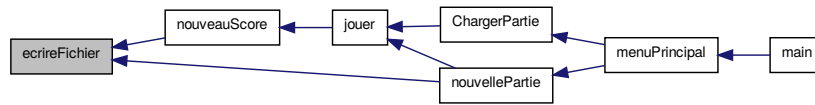
Returns

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:



4.6.3.2 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

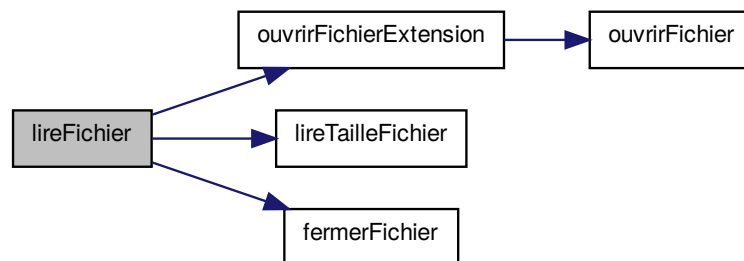
Parameters

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

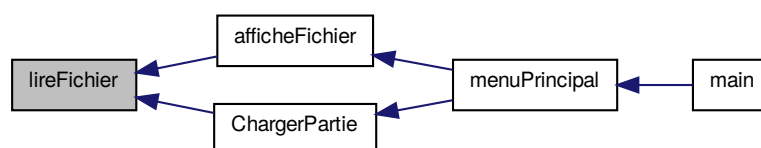
Returns

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:



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

Mets a jour le fichier avec les nouveaux scores

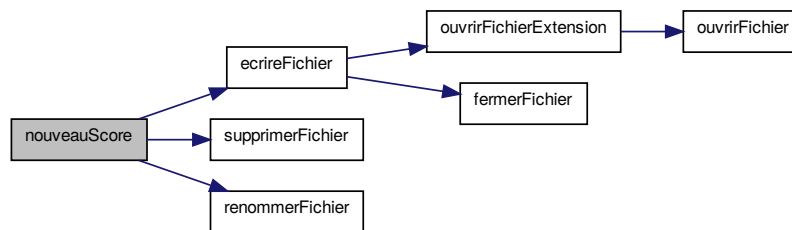
Parameters

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

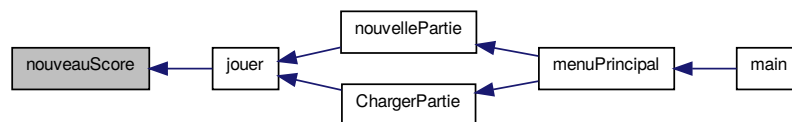
Returns

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:



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

Ouvre un fichier a partir de son nom (nom[]) et du mod voulu (mode[]) en y ajouter l'extension du fichier

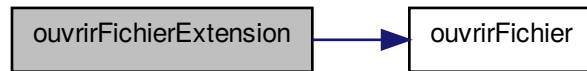
Parameters

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

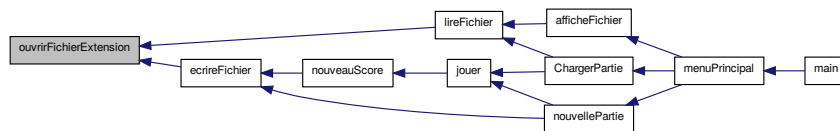
Returns

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:



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

Renomme le fichier dont le nom est en parametre

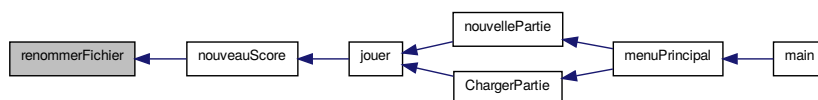
Parameters

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:



4.6.3.6 int supprimerFichier (char * *nom*)

Supprime le fichier dont le nom est en parametre

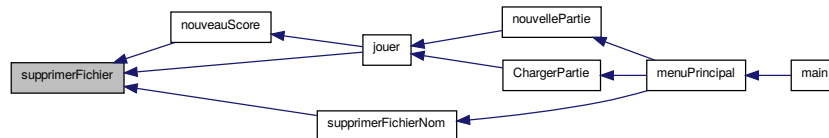
Parameters

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

Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the caller graph for this function:

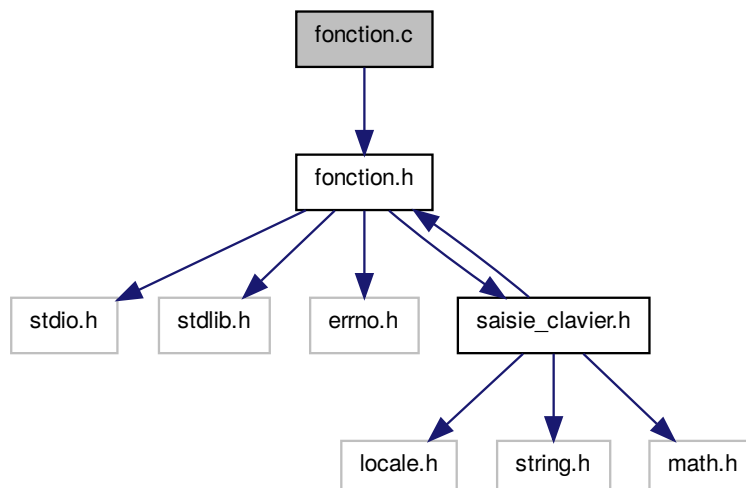


4.7 fonction.c File Reference

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)

4.7.1 Detailed Description

Fonctions essentielles au fonctionnement du programme.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.7.2 Function Documentation

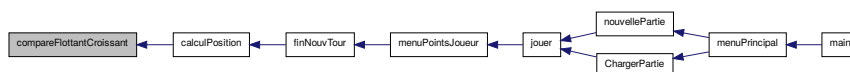
4.7.2.1 int compareFlottantCroissant (void const * *a*, void const * *b*)

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:



4.7.2.2 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:



4.7.2.3 int fermerFichier (FILE * *ptr_fichier*)

Ferme le fichier

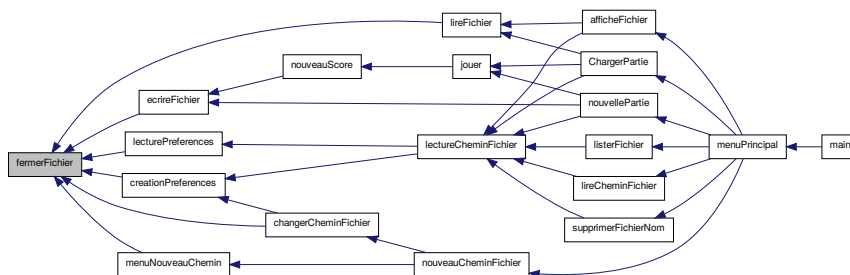
Parameters

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:



4.7.2.4 int lireTailleFichier (FILE * ptr_fichier)

Lis la taille du fichier

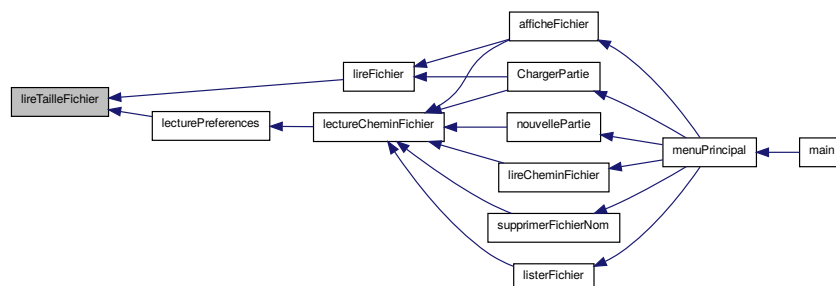
Parameters

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

Returns

entier ayant la taille du fichier

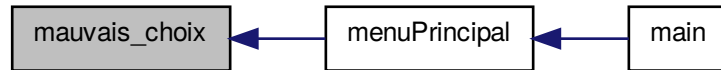
Here is the caller graph for this function:



4.7.2.5 void mauvais_choix ()

Affiche un message d'erreur.

Here is the caller graph for this function:



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

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

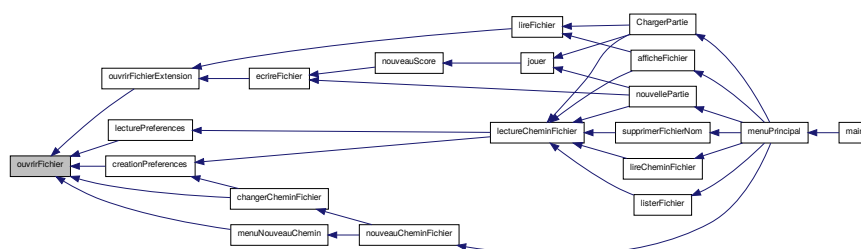
Parameters

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

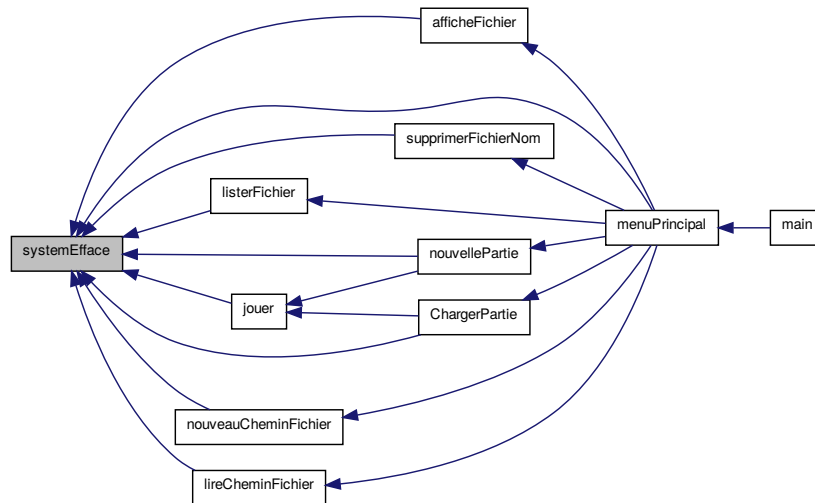
Here is the caller graph for this function:



4.7.2.7 void systemEfface ()

Efface la console de l'utilisateur.

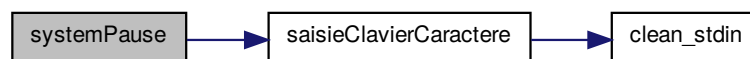
Here is the caller graph for this function:



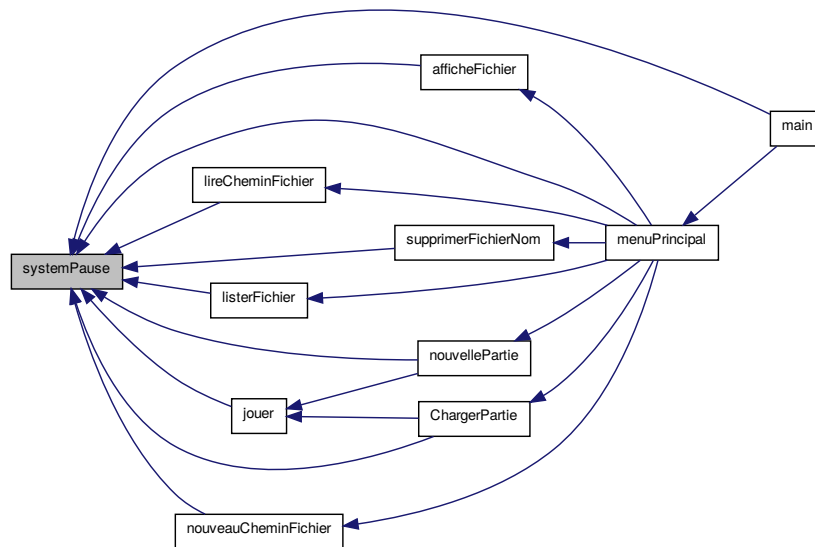
4.7.2.8 void systemPause ()

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:



4.8 fonction.h File Reference

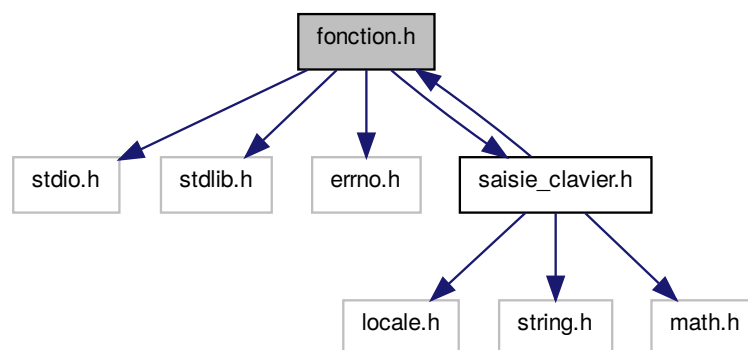
Prototypes des fonctions essentielles au fonctionnement du programme.

```

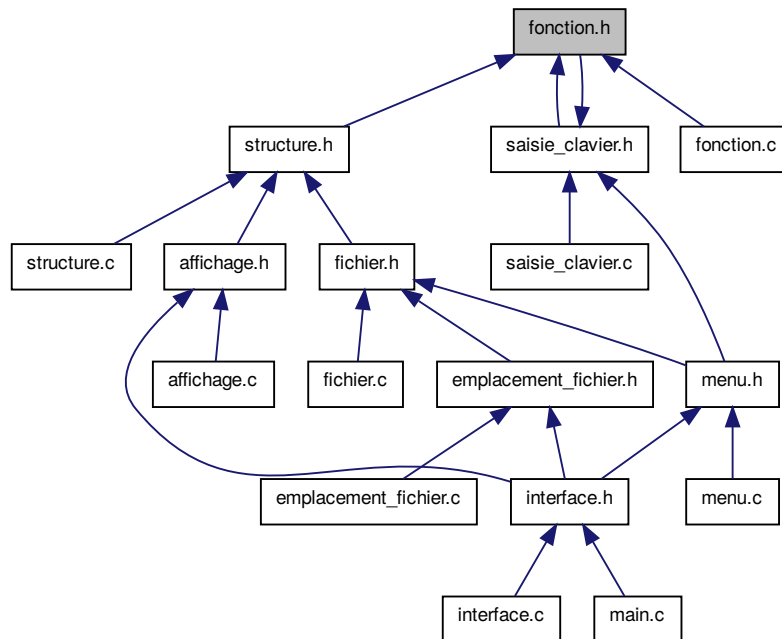
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include "saisie_clavier.h"

```

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)

4.8.1 Detailed Description

Prototypes des fonctions essentielles au fonctionnement du programme.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.8.2 Macro Definition Documentation

4.8.2.1 #define FAUX 0

Definit FAUX a 0

4.8.2.2 #define VRAI 1

Definit VRAI a 1

4.8.3 Function Documentation

4.8.3.1 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:

4.8.3.2 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

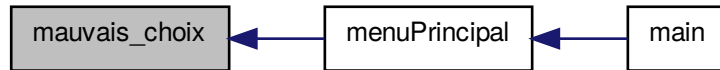
in	<i>*a</i>	un pointeur sur un flottant
in	<i>*b</i>	un pointeur sur un flottant

Here is the caller graph for this function:

4.8.3.3 int fermerFichier (FILE * *ptr_fichier*)

Ferme le fichier

Here is the caller graph for this function:



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

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

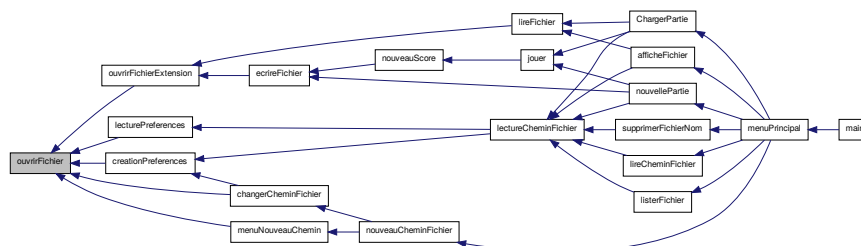
Parameters

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

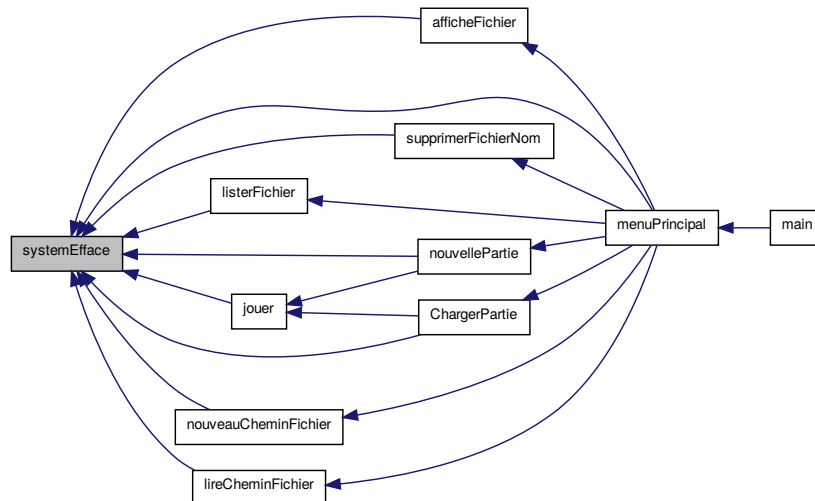
Here is the caller graph for this function:



4.8.3.7 void systemEfface ()

Efface la console de l'utilisateur.

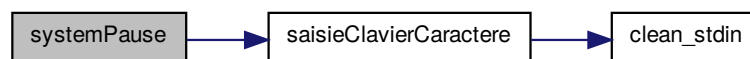
Here is the caller graph for this function:



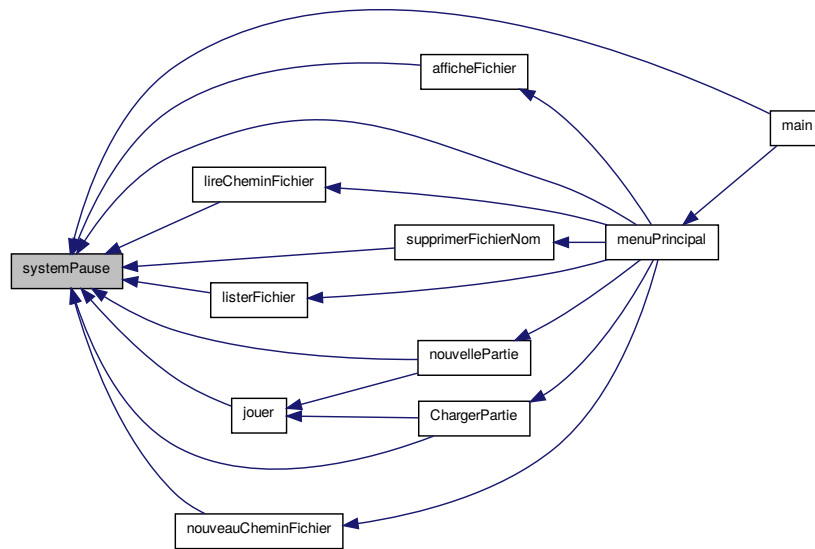
4.8.3.8 void systemPause ()

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:

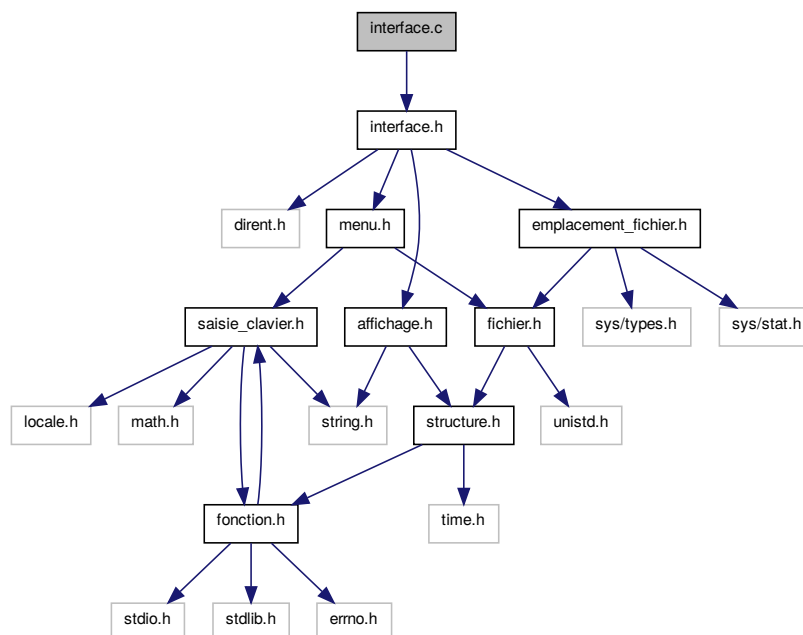


4.9 interface.c File Reference

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](#) ()

4.9.1 Detailed Description

Fonctions qui gerent l'interface graphique du logiciel.

Author

Remi BERTHO

Date

13/02/14

Version

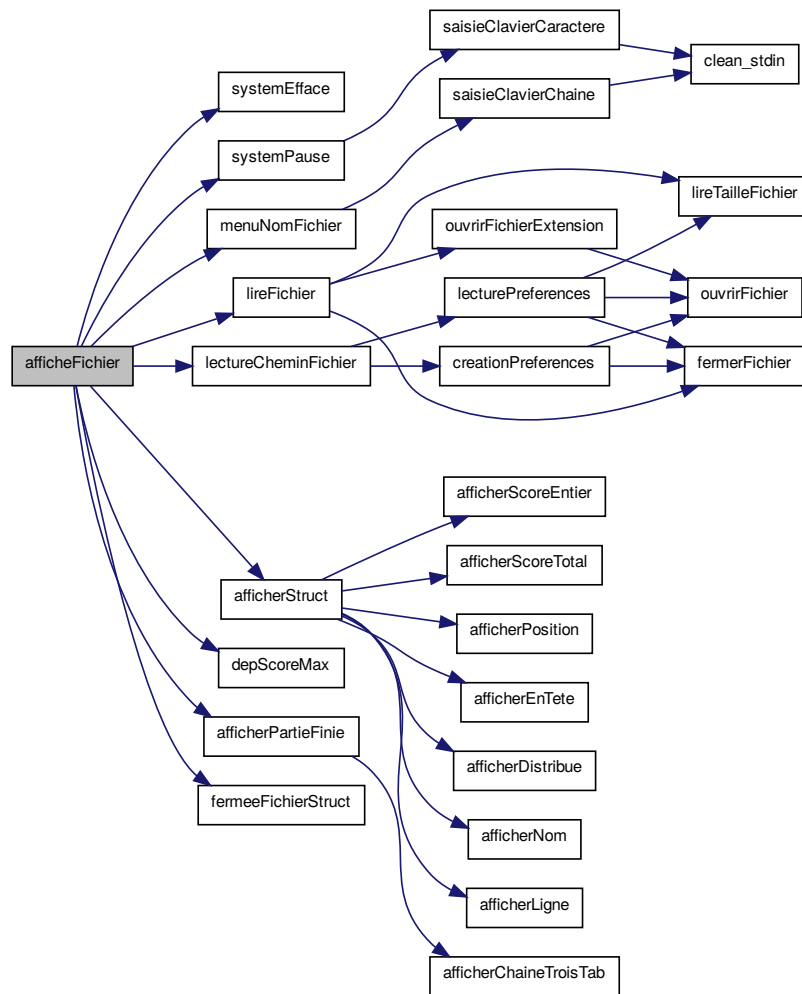
2.0

4.9.2 Function Documentation

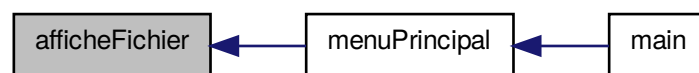
4.9.2.1 void [afficheFichier](#) ()

Demande le nom d'un fichier et l'affiche

Here is the call graph for this function:



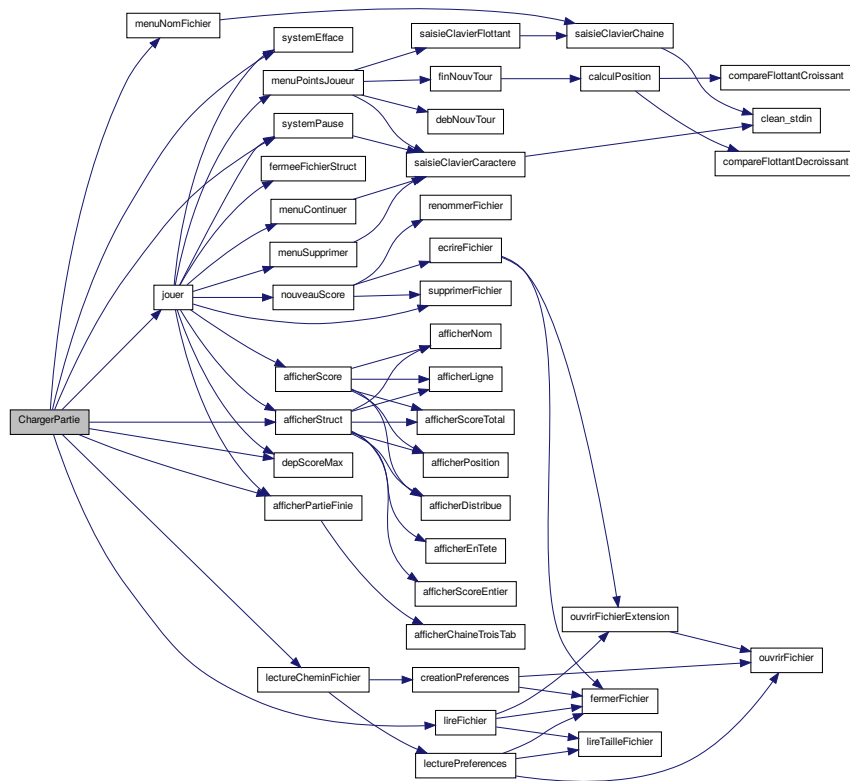
Here is the caller graph for this function:



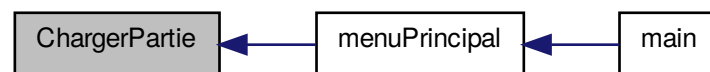
4.9.2.2 void ChargerPartie ()

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:



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

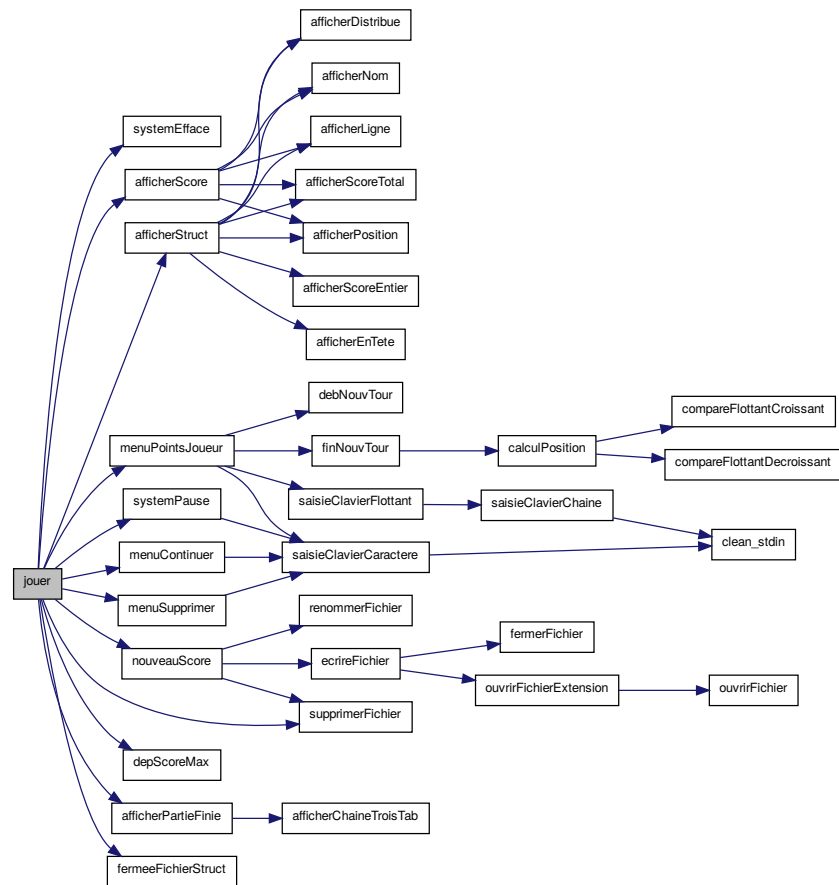
Lance la partie de comptage jusqu'a la fin

Parameters

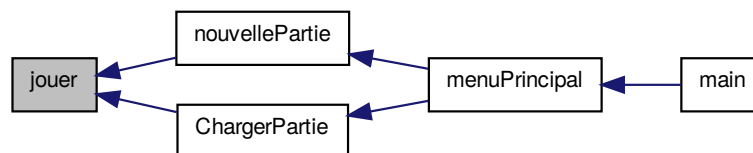
in, out	*ptr_struct_fichier	la structure Fichier_Jeu
---------	---------------------	--

in	*nom_fichier	le nom du fichier dans lequel on stocke les informations
----	--------------	--

Here is the call graph for this function:



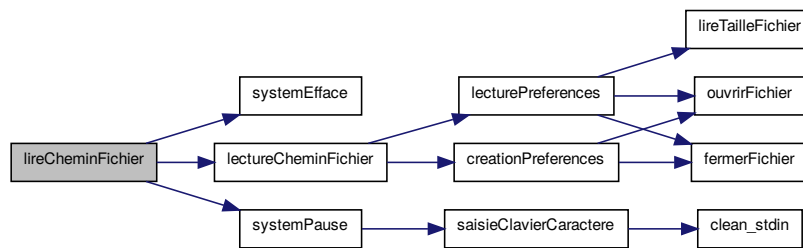
Here is the caller graph for this function:



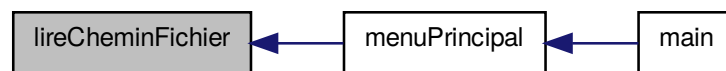
4.9.2.4 void lireCheminFichier ()

Lis le chemin de fichier et l'affiche

Here is the call graph for this function:



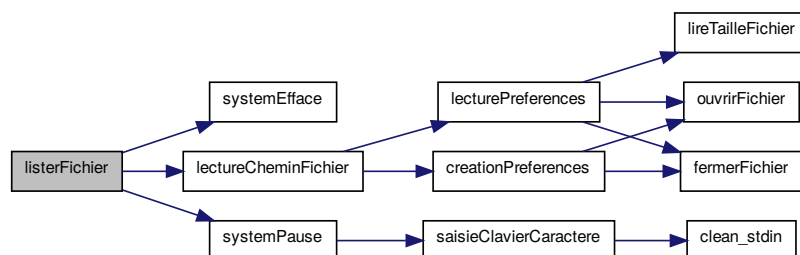
Here is the caller graph for this function:



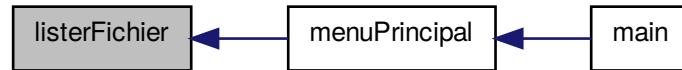
4.9.2.5 void listerFichier ()

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

Here is the call graph for this function:



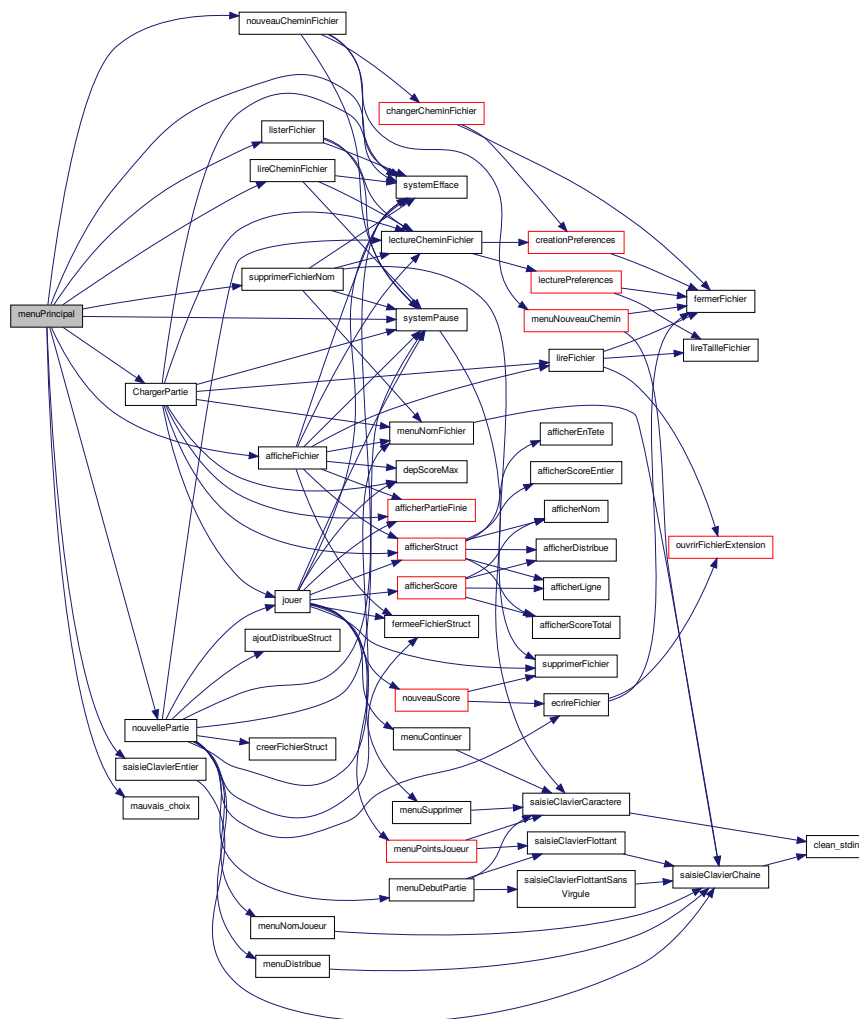
Here is the caller graph for this function:



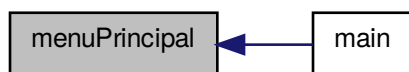
4.9.2.6 void menuPrincipal ()

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

Here is the call graph for this function:



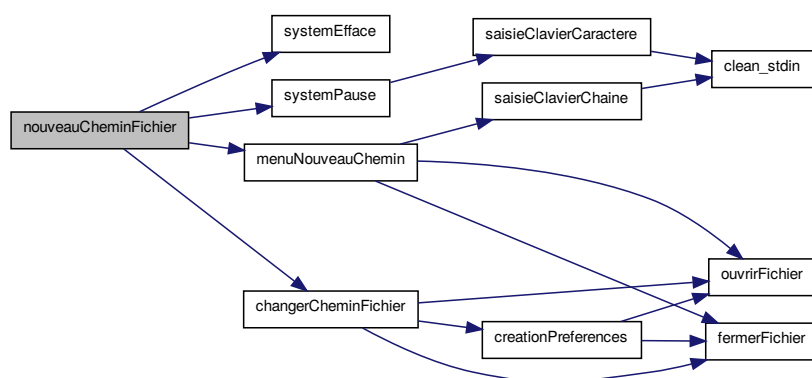
Here is the caller graph for this function:



4.9.2.7 void nouveauCheminFichier ()

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

Here is the call graph for this function:



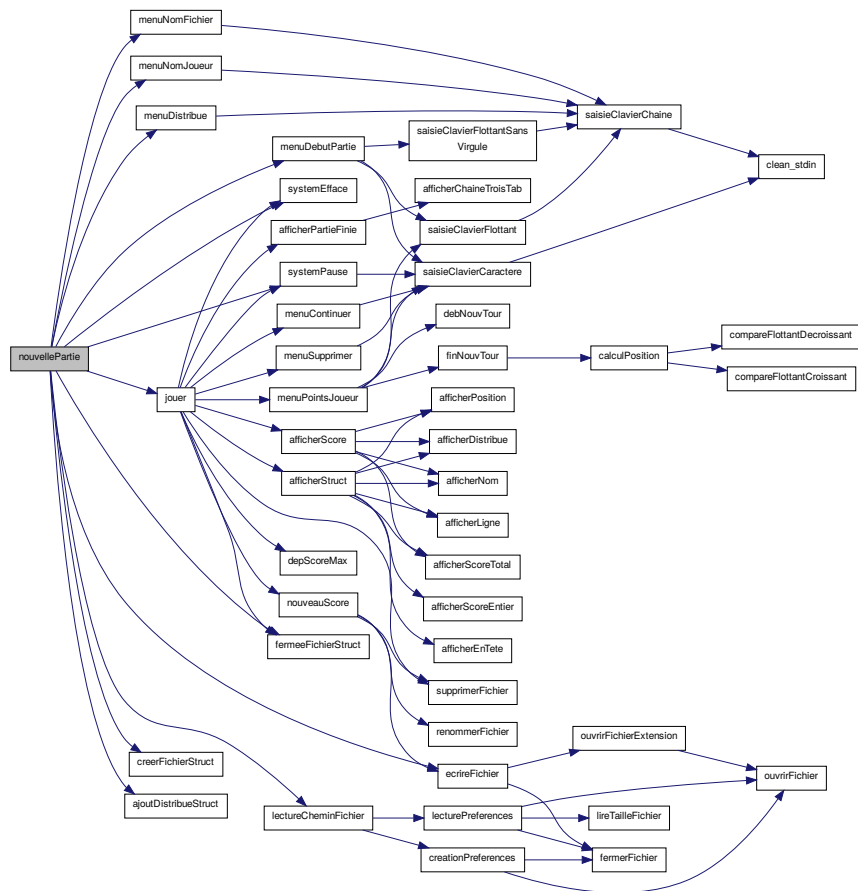
Here is the caller graph for this function:



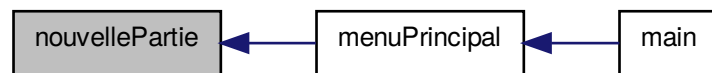
4.9.2.8 void nouvellePartie ()

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

Here is the call graph for this function:



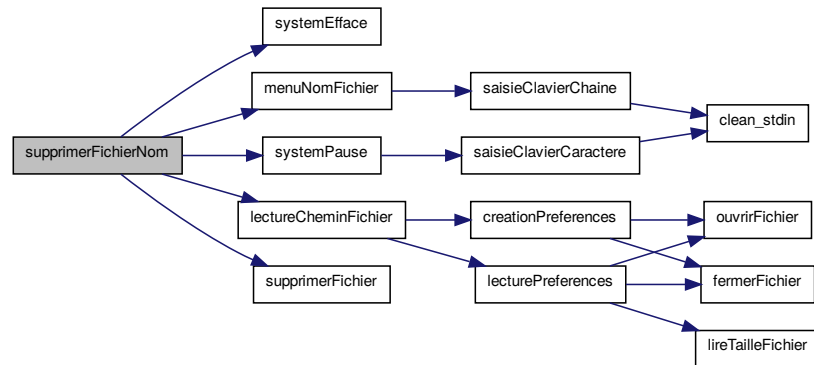
Here is the caller graph for this function:



4.9.2.9 void supprimerFichierNom ()

Demande le nom d'un fichier et le supprime

Here is the call graph for this function:



Here is the caller graph for this function:



4.10 interface.h File Reference

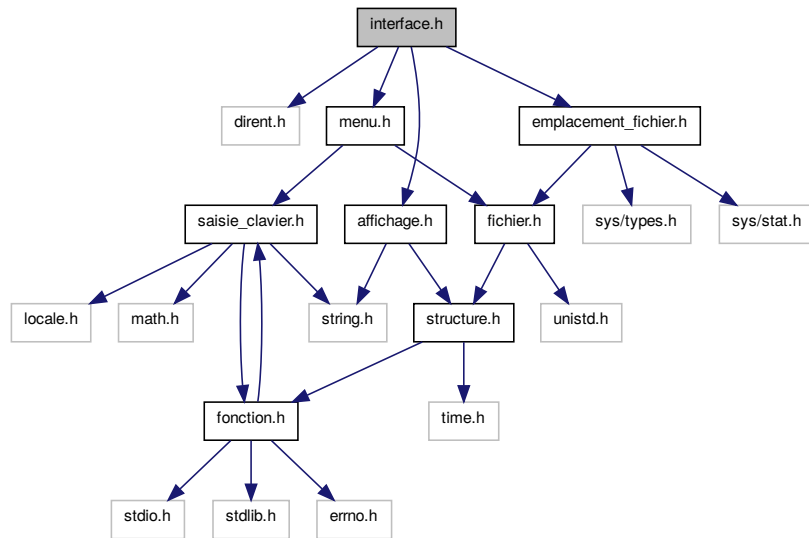
Prototypes des fonctions qui gerent l'interface graphique du logiciel.

```

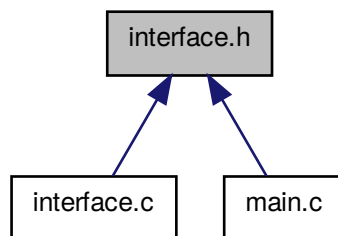
#include <dirent.h>
#include "menu.h"
#include "affichage.h"
#include "emplacement_fichier.h"

```

Include dependency graph for interface.h:



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



Enumerations

- enum `Menu` {
`nouvPart` =1, `charPart` =2, `affFich` =3, `supprFich` =4,
`listFich` =5, `nouvChem` =6, `lireChem` =7, `quit` =8 }

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](#) ()

4.10.1 Detailed Description

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

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.10.2 Enumeration Type Documentation

4.10.2.1 enum Menu

Enumerator

nouvPart

charPart

affFich

supprFich

listFich

nouvChem

lireChem

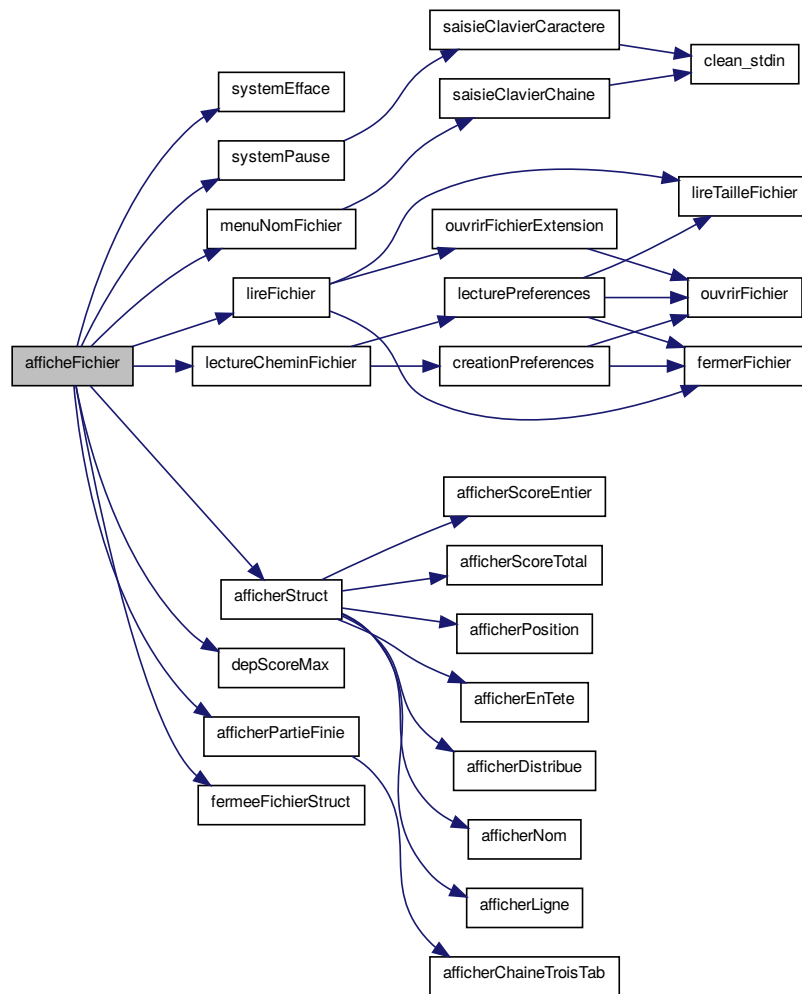
quit

4.10.3 Function Documentation

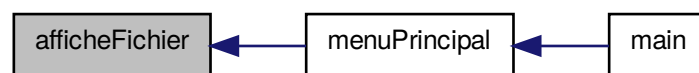
4.10.3.1 void afficheFichier ()

Demande le nom d'un fichier et l'affiche

Here is the call graph for this function:



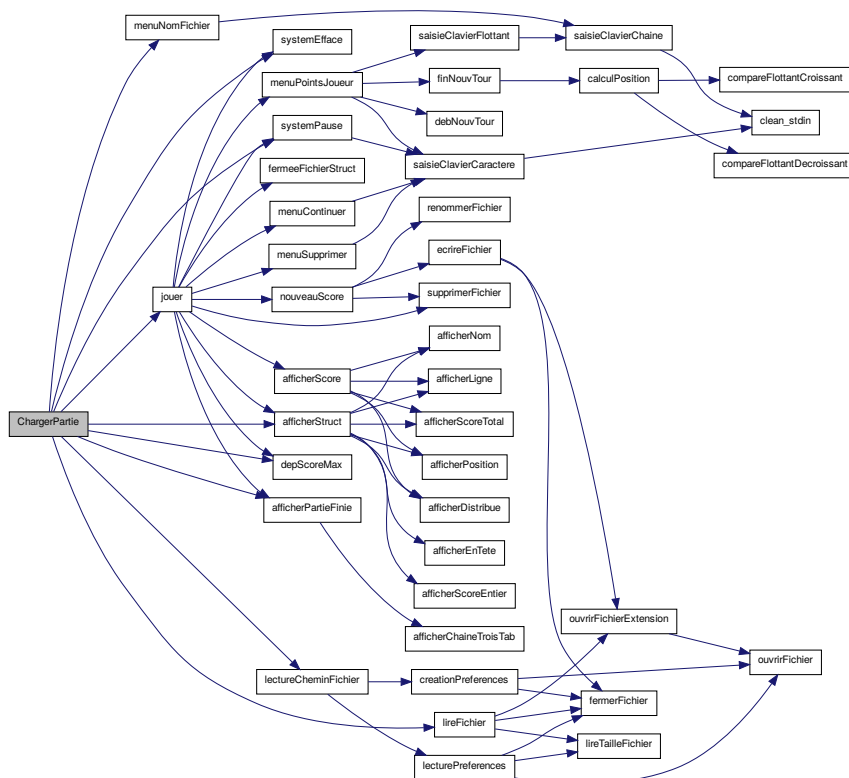
Here is the caller graph for this function:



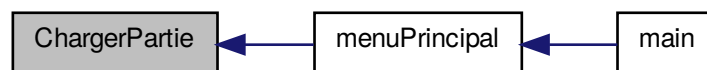
4.10.3.2 void ChargerPartie ()

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:



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

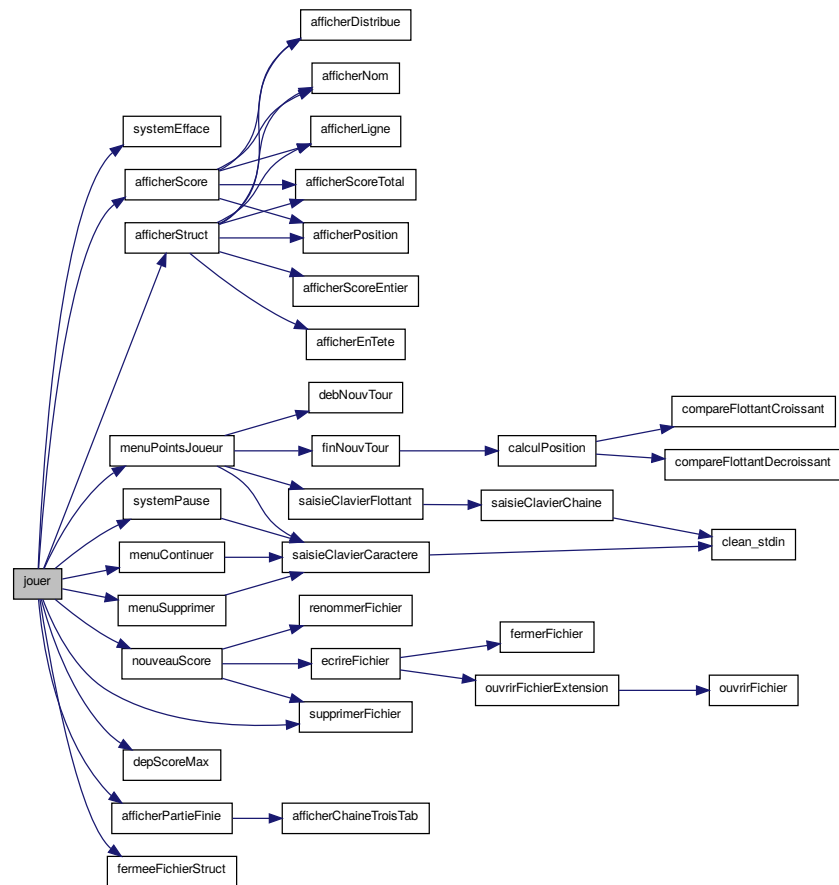
Lance la partie de comptage jusqu'à la fin

Parameters

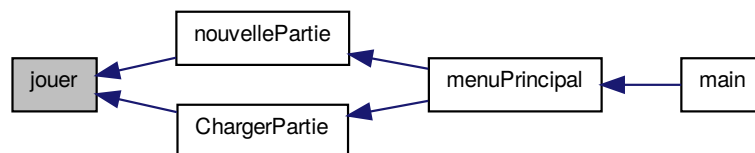
in, out	<i>*ptr_struct_fichier</i>	la structure Fichier_Jeu
---------	----------------------------	--

in	*nom_fichier	le nom du fichier dans lequel on stocke les informations
----	--------------	--

Here is the call graph for this function:



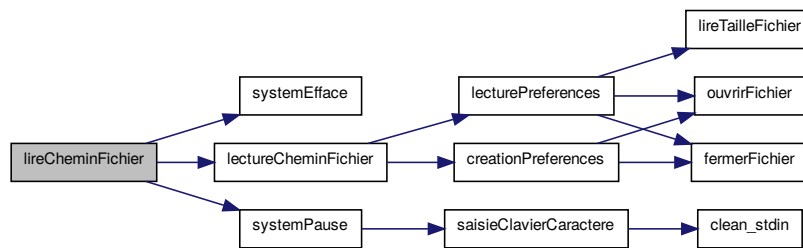
Here is the caller graph for this function:



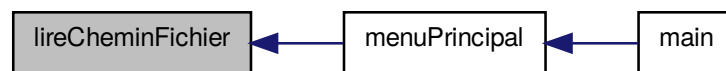
4.10.3.4 void lireCheminFichier ()

Lis le chemin de fichier et l'affiche

Here is the call graph for this function:



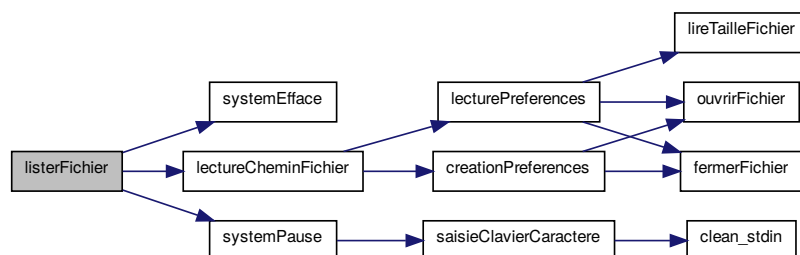
Here is the caller graph for this function:



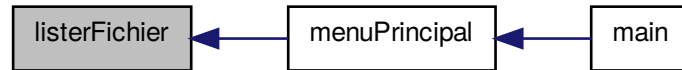
4.10.3.5 void listerFichier ()

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

Here is the call graph for this function:



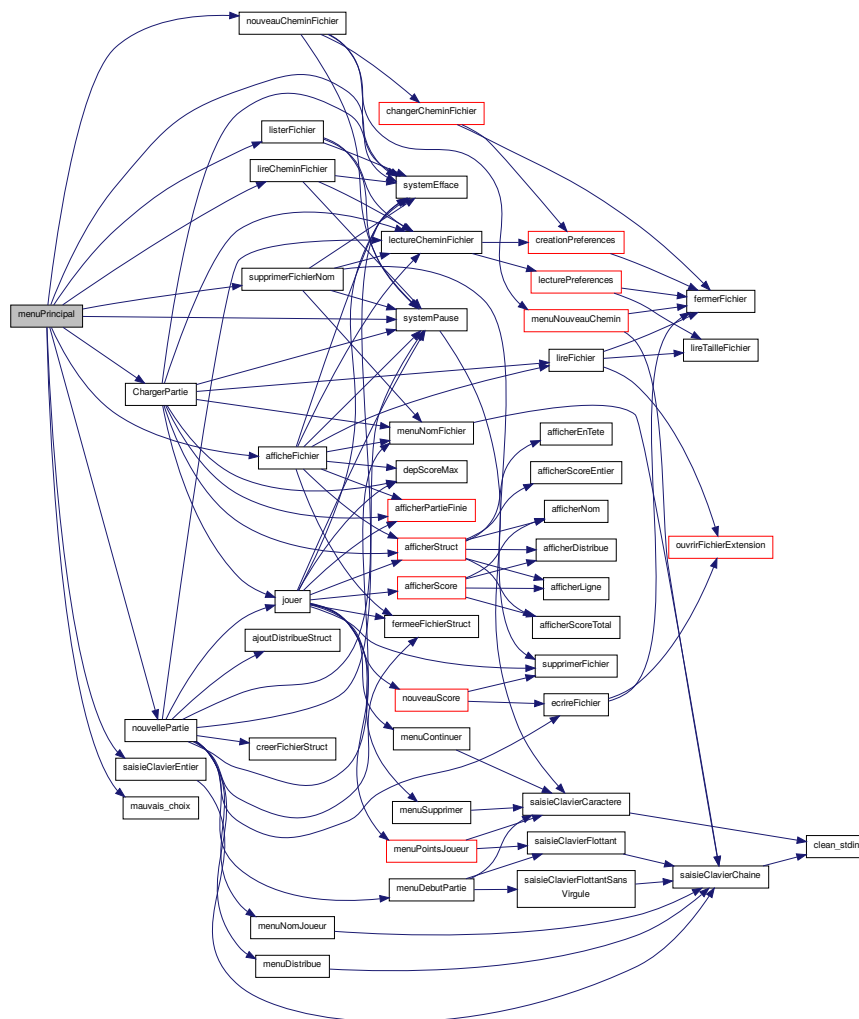
Here is the caller graph for this function:



4.10.3.6 void menuPrincipal ()

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

Here is the call graph for this function:



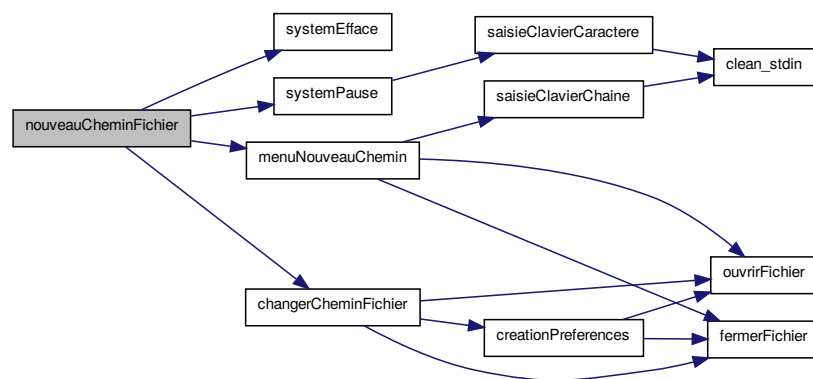
Here is the caller graph for this function:



4.10.3.7 void nouveauCheminFichier ()

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

Here is the call graph for this function:



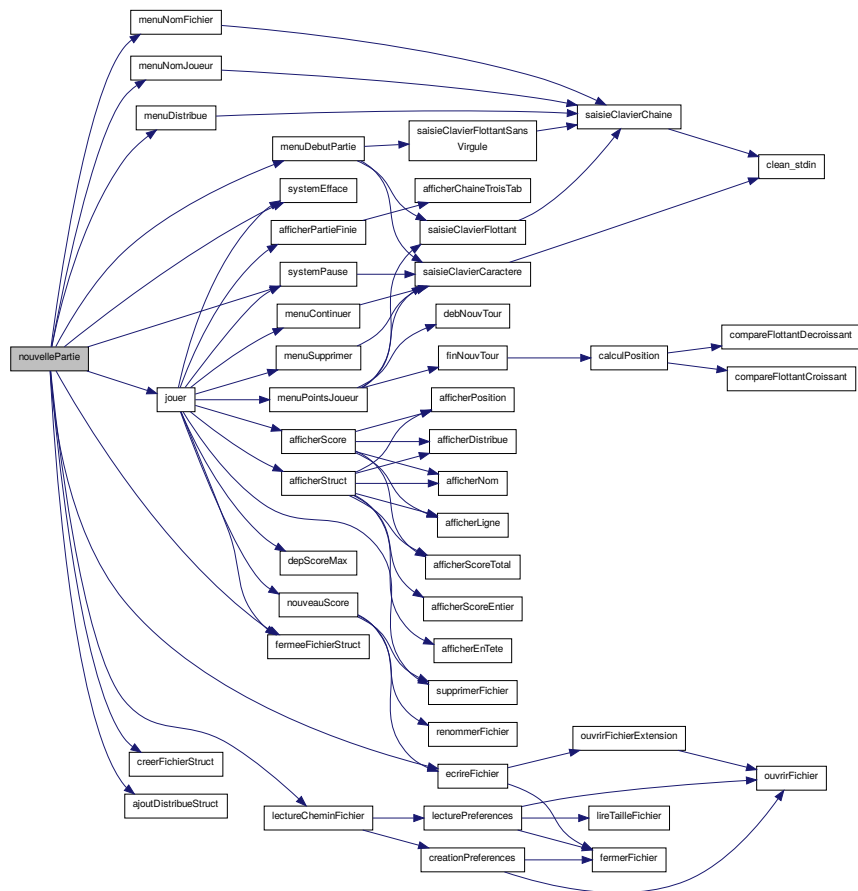
Here is the caller graph for this function:



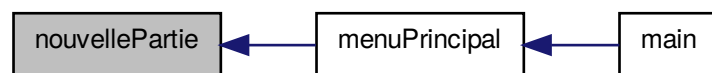
4.10.3.8 void nouvellePartie ()

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

Here is the call graph for this function:



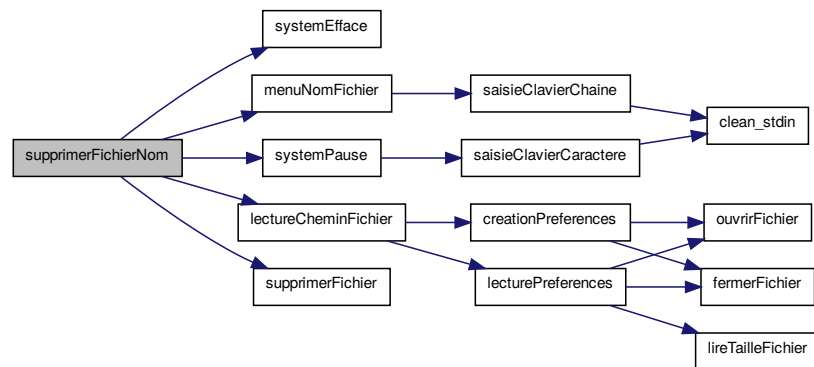
Here is the caller graph for this function:



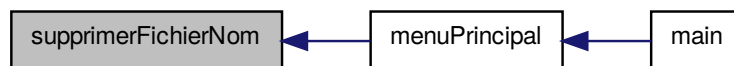
4.10.3.9 void supprimerFichierNom ()

Demande le nom d'un fichier et le supprime

Here is the call graph for this function:



Here is the caller graph for this function:

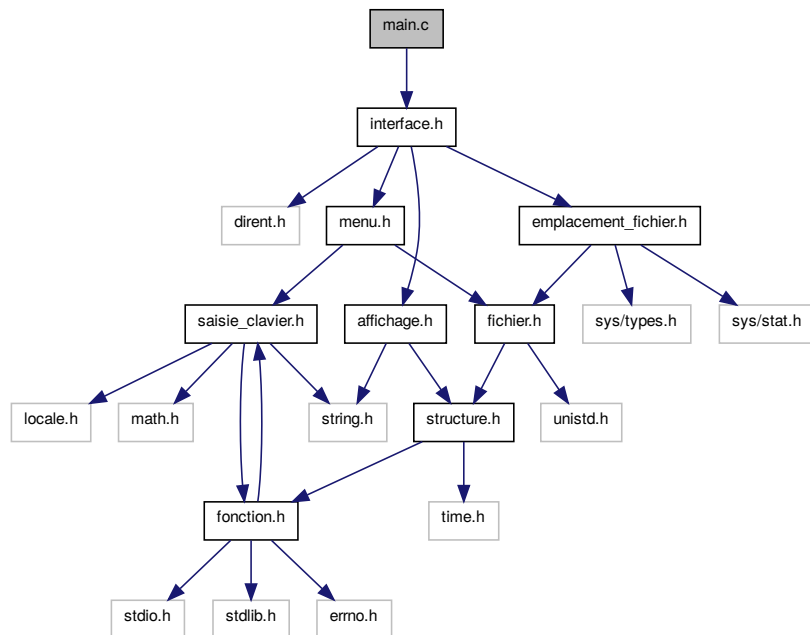


4.11 main.c File Reference

Lancement du programme.

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

Include dependency graph for main.c:



Functions

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

4.11.1 Detailed Description

Lancement du programme.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

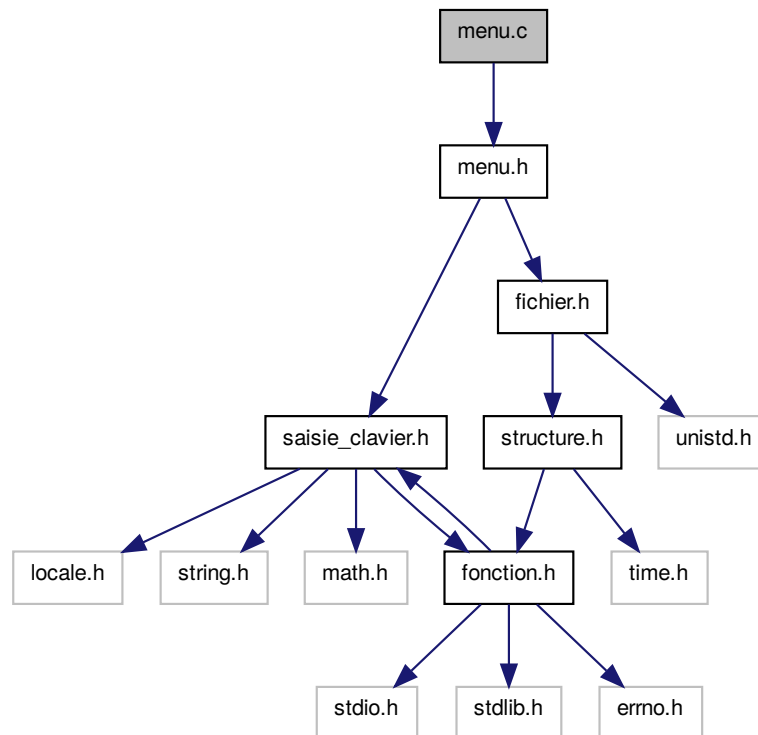
4.11.2 Function Documentation

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

Lance le programme


```
#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)

4.12.1 Detailed Description

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

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.12.2 Function Documentation

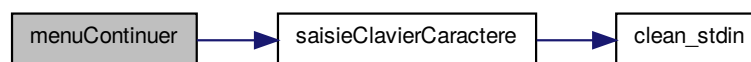
4.12.2.1 int menuContinuer ()

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

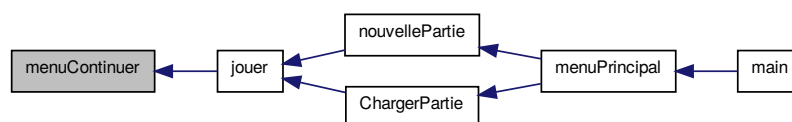
Returns

VRAI si l'on veut continuer, FAUX sinon

Here is the call graph for this function:



Here is the caller graph for this function:



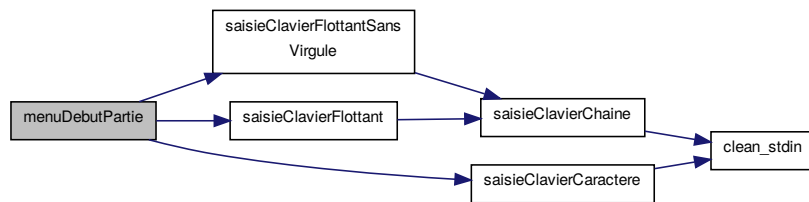
4.12.2.2 void menuDebutPartie (float * ptr_nb_joueur, float * ptr_nb_max, char * ptr_sens_premier)

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:



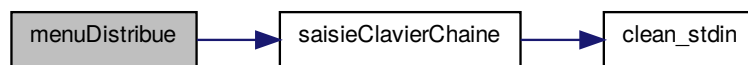
4.12.2.3 void menuDistribue (char * nom_distribue)

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

Parameters

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



4.12.2.4 `char * menuNomFichier (char nom_fichier[TAILLE_MAX_NOM_FICHIER])`

Demande et enregistre le nom du fichier

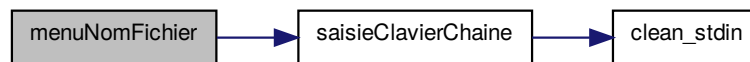
Parameters

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

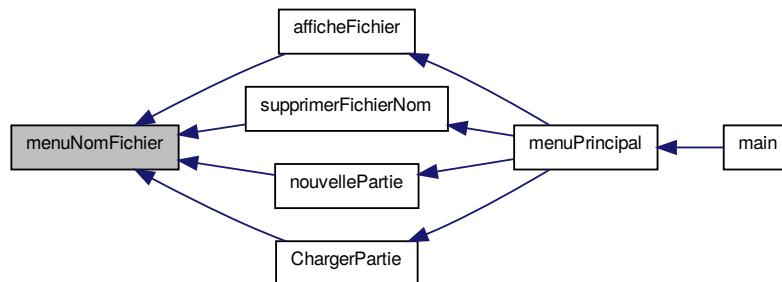
Returns

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:



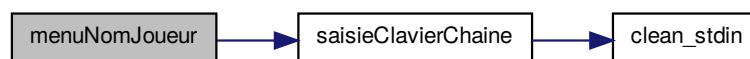
4.12.2.5 void menuNomJoueur (Fichier_Jeu * ptr_struct_fichier)

Demande et enregistre le nom des joueurs

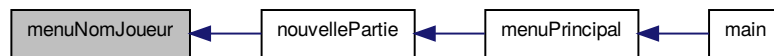
Parameters

<i>in, out</i>	<i>*ptr_struct_fichier</i>	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:



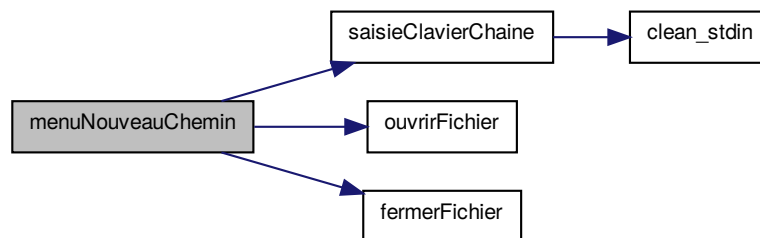
4.12.2.6 void menuNouveauChemin (char * *nouveauChemin*)

Demande et enregistre le nouveau chemin

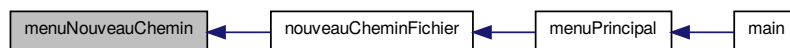
Parameters

<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:



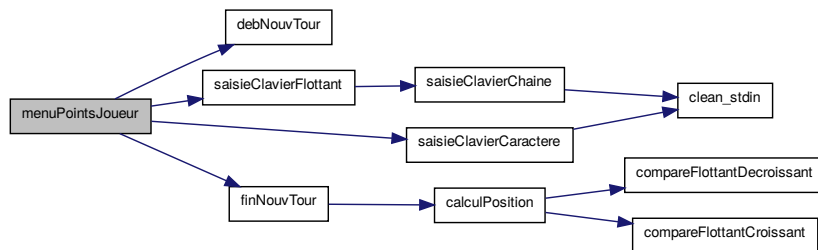
4.12.2.7 void menuPointsJoueur (Fichier_Jeu * *ptr_struct_fichier*)

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

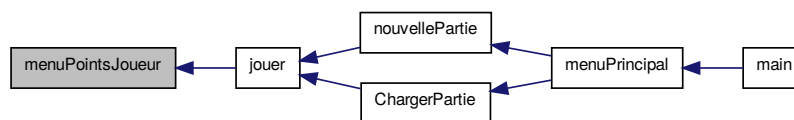
Parameters

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:



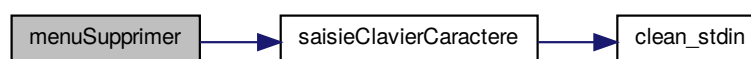
4.12.2.8 int menuSupprimer ()

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

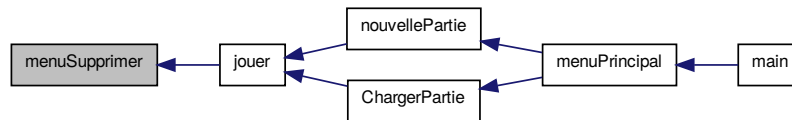
Returns

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:



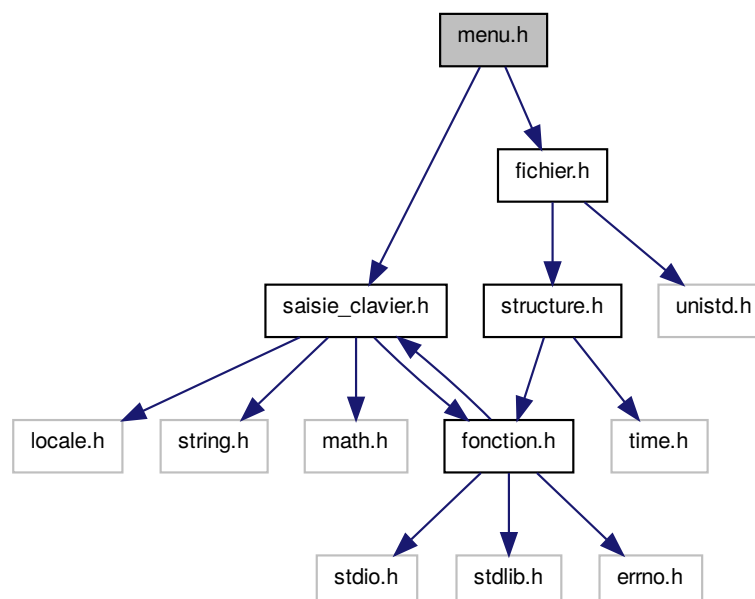
4.13 menu.h File Reference

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

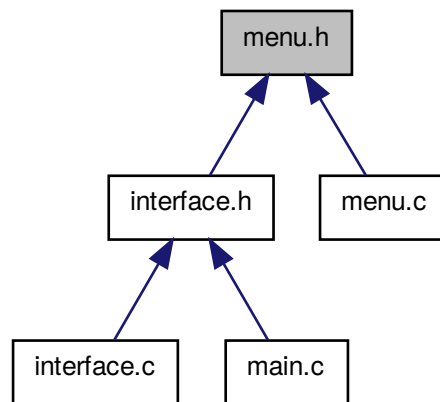
```
#include "saisie_clavier.h"
```

```
#include "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)

4.13.1 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

4.13.2 Function Documentation

4.13.2.1 int menuContinuer ()

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

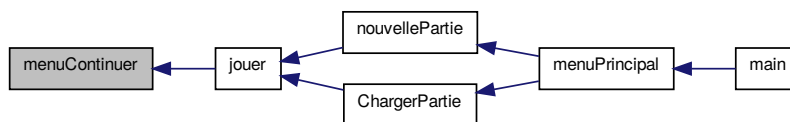
Returns

VRAI si l'on veut continuer, FAUX sinon

Here is the call graph for this function:



Here is the caller graph for this function:



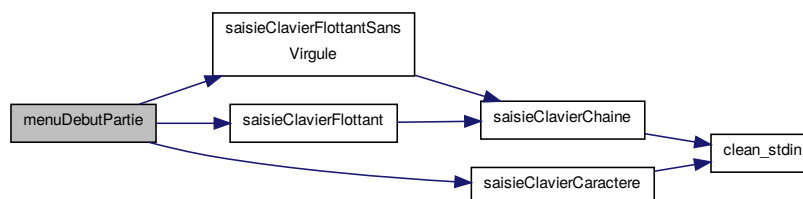
4.13.2.2 void menuDebutPartie (float * ptr_nb_joueur, float * ptr_nb_max, char * ptr_sens_premier)

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

Parameters

in, out	*ptr_nb_joueur	le nombre de joueur
in, out	*ptr_nb_max	le nombre maximum
in, out	*ptr_sens_ - premier	definit le sens du premier

Here is the call graph for this function:



Here is the caller graph for this function:



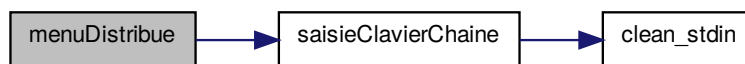
4.13.2.3 void menuDistribue (char * nom_distribue)

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

Parameters

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



4.13.2.4 char* menuNomFichier (char nom_fichier[TAILLE_MAX_NOM_FICHIER])

Demande et enregistre le nom du fichier

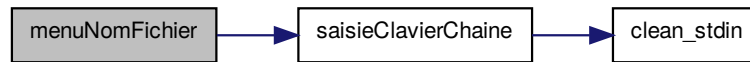
Parameters

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

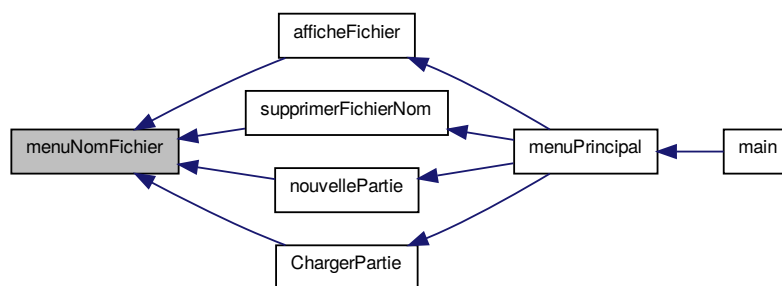
Returns

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:



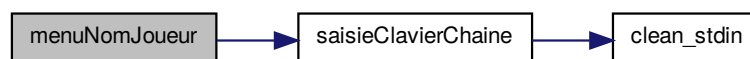
4.13.2.5 void menuNomJoueur (Fichier_Jeu * ptr_struct_fichier)

Demande et enregistre le nom des joueurs

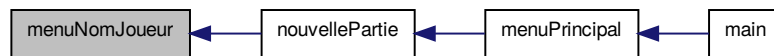
Parameters

<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:



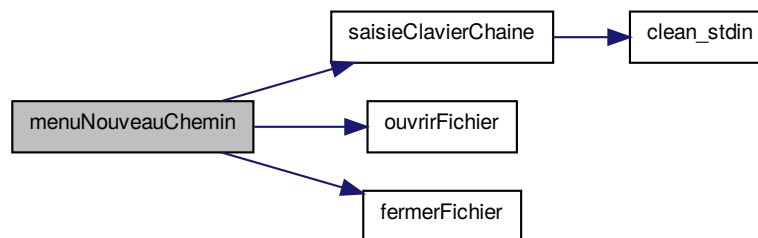
4.13.2.6 void menuNouveauChemin (char * *nouveauChemin*)

Demande et enregistre le nouveau chemin

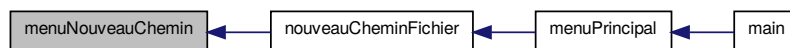
Parameters

<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:



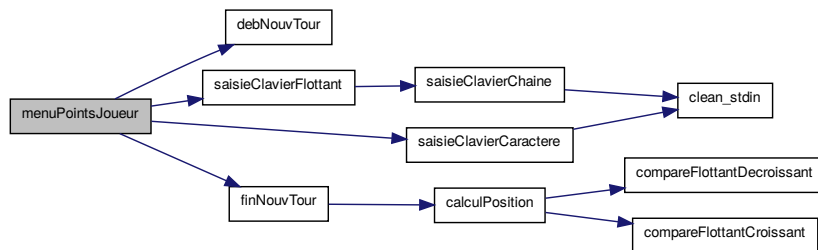
4.13.2.7 void menuPointsJoueur (Fichier_Jeu * *ptr_struct_fichier*)

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

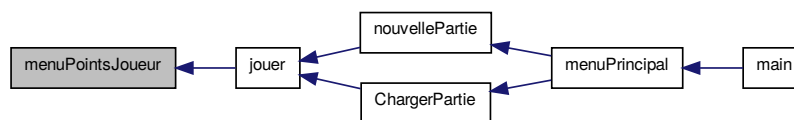
Parameters

<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:



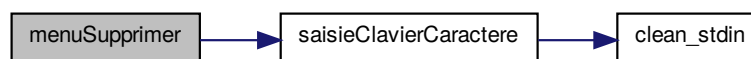
4.13.2.8 `int menuSupprimer ()`

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

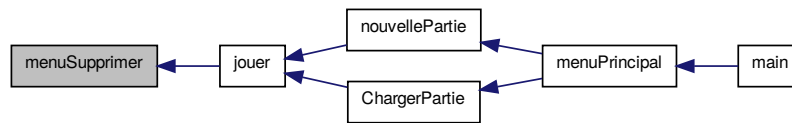
Returns

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:

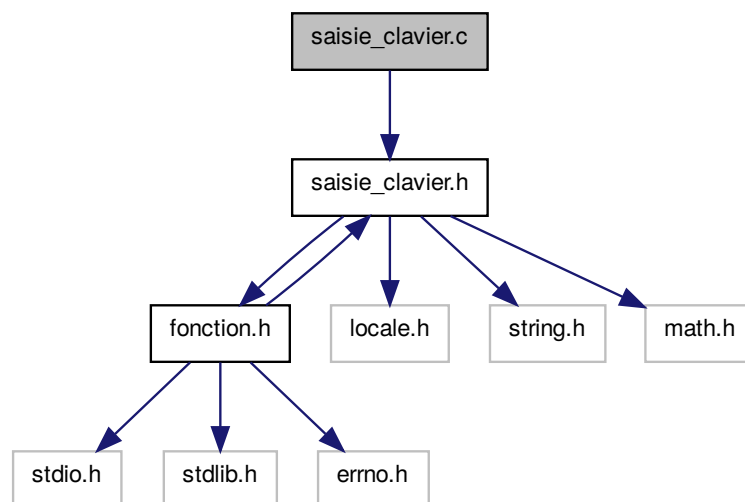


4.14 saisie_clavier.c File Reference

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)

4.14.1 Detailed Description

Fonctions de saisie clavier.

Author

Remi BERTHO

Date

13/02/14

Version

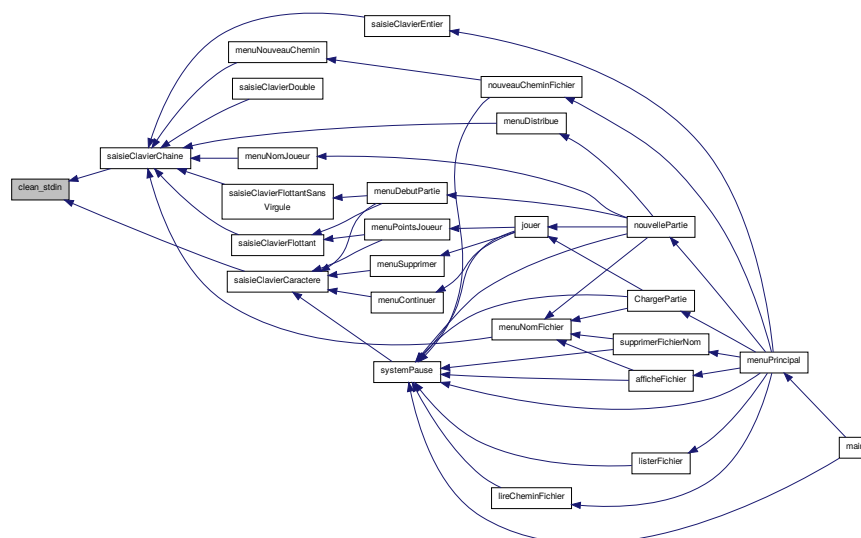
2.0

4.14.2 Function Documentation

4.14.2.1 void clean_stdin (void)

Vide la cache de stdin.

Here is the caller graph for this function:



4.14.2.2 char * saisieClavierCaractere (char * c)

Fait une saisie clavier d'un caractere.

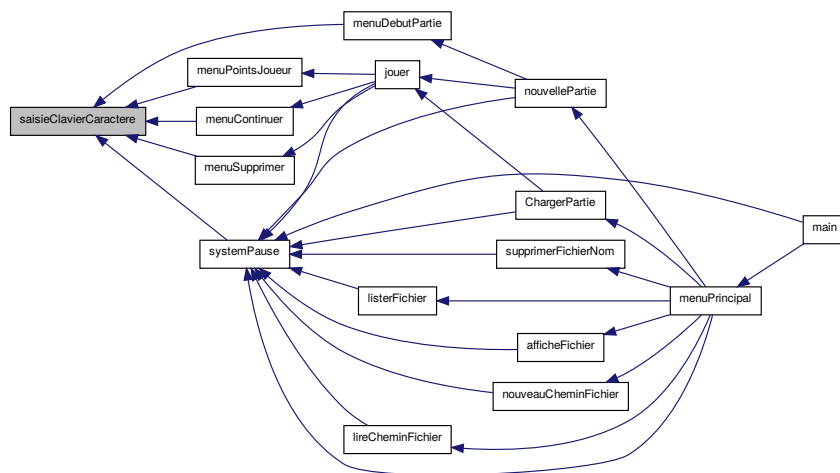
Parameters

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:



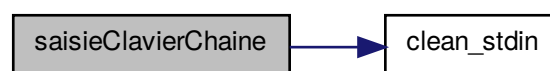
4.14.2.3 char * saisieClavierChaine (char * chaine, int nb_caract_plus_un)

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

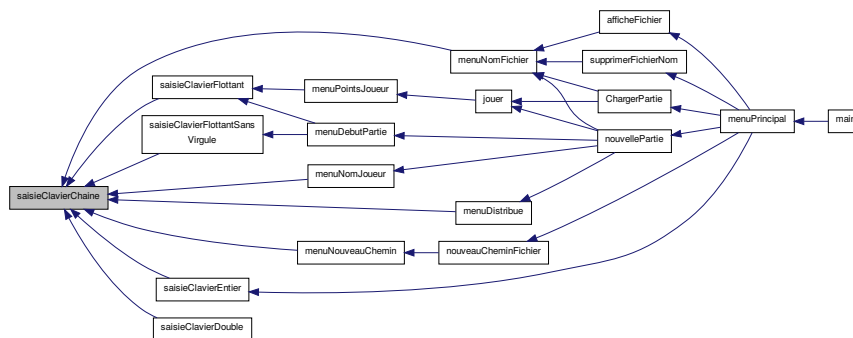
Parameters

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:



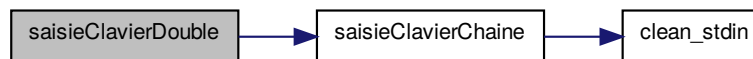
4.14.2.4 void saisieClavierDouble (double * nb)

Fait une saisie clavier d'un double.

Parameters

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

Here is the call graph for this function:



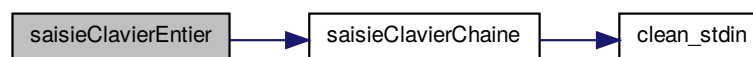
4.14.2.5 void * saisieClavierEntier (int * nb)

Fait une saisie clavier d'un entier.

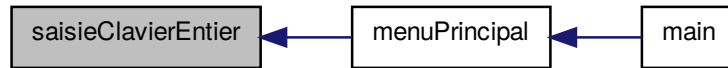
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



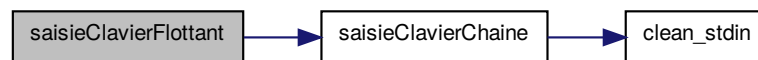
4.14.2.6 void saisieClavierFlottant (float * nb)

Fait une saisie clavier d'un flottant.

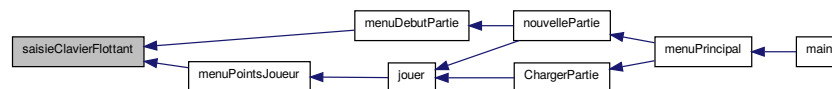
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



4.14.2.7 void saisieClavierFlottantSansVirgule (float * nb)

Fait une saisie clavier d'un flottant.

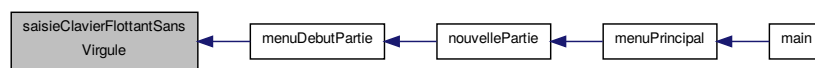
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:

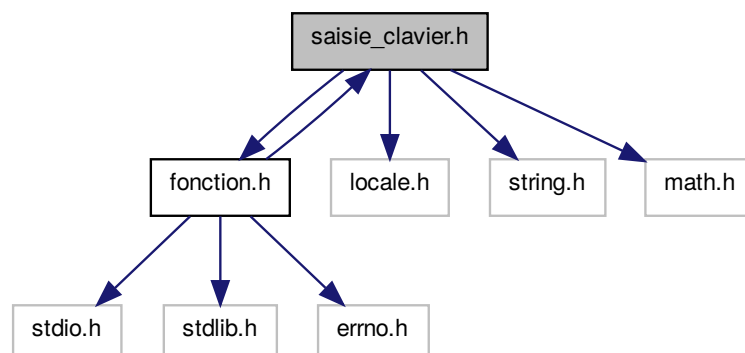


4.15 saisie_clavier.h File Reference

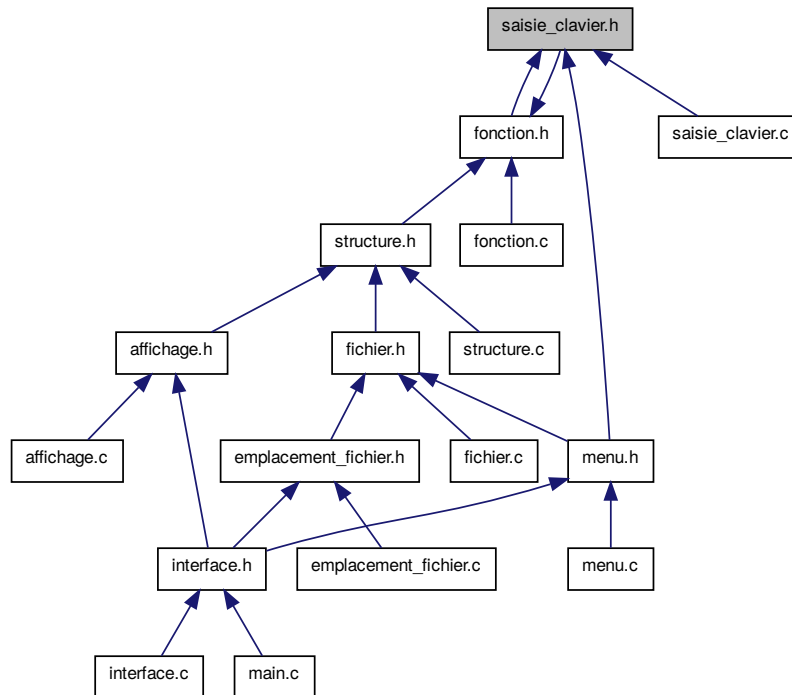
Prototypes des fonctions de saisie clavier.

```
#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)

4.15.1 Detailed Description

Prototypes des fonctions de saisie clavier.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.15.2 Macro Definition Documentation

4.15.2.1 `#define NB_CARACT_DOUB 309`

Definit NB_CARACT_DOUB a 309

4.15.2.2 `#define NB_CARACT_FLOT 39`

Definit NB_CARACT_FLOT a 39

4.15.2.3 `#define NB_CARACT_INT 12`

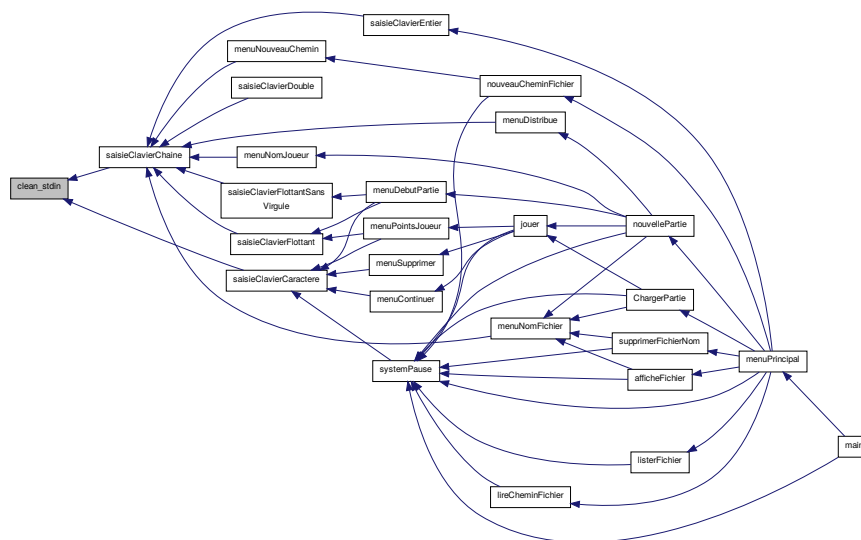
Definit NB_CARACT_INT a 12

4.15.3 Function Documentation

4.15.3.1 `void clean_stdin (void)`

Vide la cache de stdin.

Here is the caller graph for this function:

4.15.3.2 `char* saisieClavierCaractere (char * c)`

Fait une saisie clavier d'un caractere.

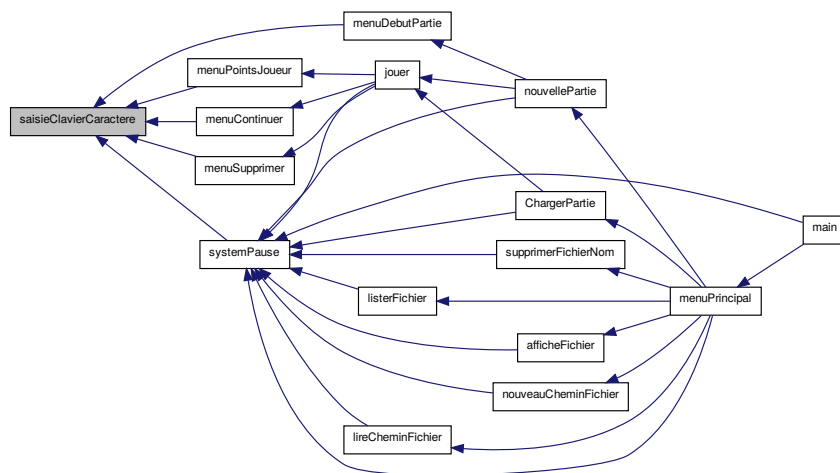
Parameters

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:



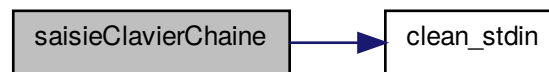
4.15.3.3 char* saisieClavierChaine (char * chaine, int nb_caract_plus_un)

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

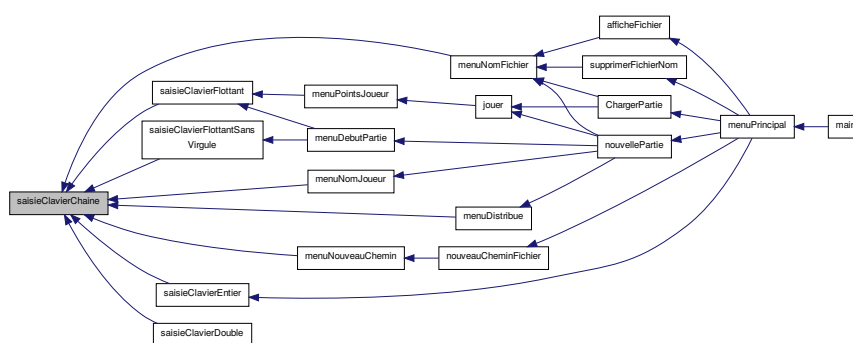
Parameters

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:



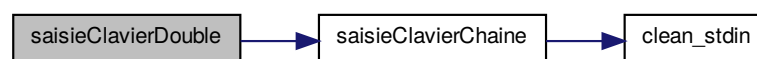
4.15.3.4 void saisieClavierDouble (double * nb)

Fait une saisie clavier d'un double.

Parameters

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

Here is the call graph for this function:



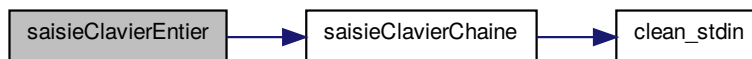
4.15.3.5 void saisieClavierEntier (int * nb)

Fait une saisie clavier d'un entier.

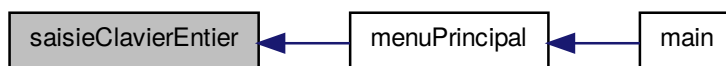
Parameters

<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:



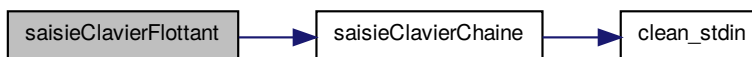
4.15.3.6 void saisieClavierFlottant (float * nb)

Fait une saisie clavier d'un flottant.

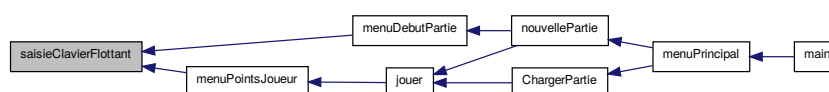
Parameters

<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:



4.15.3.7 void saisieClavierFlottantSansVirgule (float * *nb*)

Fait une saisie clavier d'un flottant.

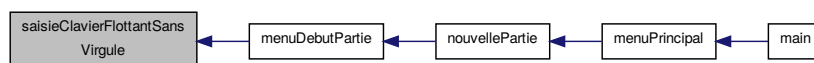
Parameters

<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:

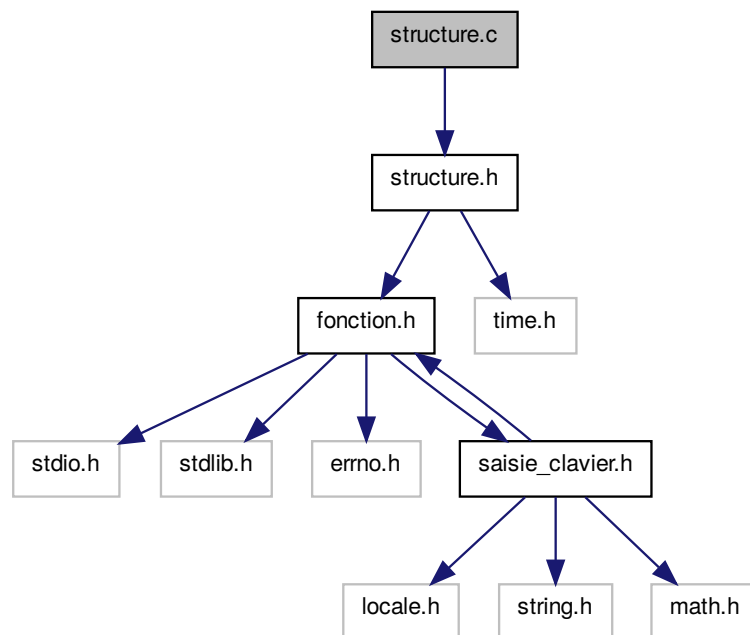


4.16 structure.c File Reference

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`)
- void `finNouvTour` (`Fichier_Jeu *ptr_struct_fichier`)
- void `calculPosition` (`Fichier_Jeu *ptr_struct_fichier`)
- void `ajoutDistribueStruct` (`Fichier_Jeu *ptr_struct_fichier`, char *nom_distribue)
- int `depScoreMax` (`Fichier_Jeu *ptr_struct_fichier`)

4.16.1 Detailed Description

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

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.16.2 Function Documentation

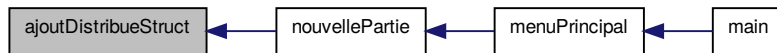
4.16.2.1 void ajoutDistribueStruct (Fichier_Jeu * *ptr_struct_fichier*, char * *nom_distribue*)

Ajoute la personne qui distribue dans la structure.

Parameters

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

Here is the caller graph for this function:



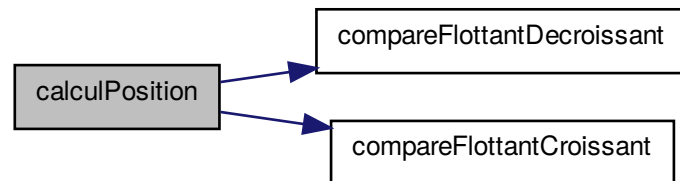
4.16.2.2 void calculPosition (Fichier_Jeu * ptr_struct_fichier)

Calcule les positions des joueurs

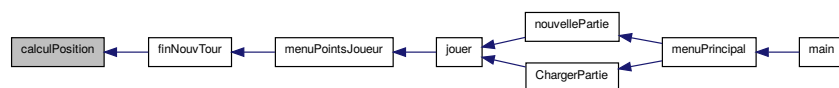
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



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

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

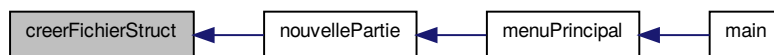
Parameters

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 caller graph for this function:



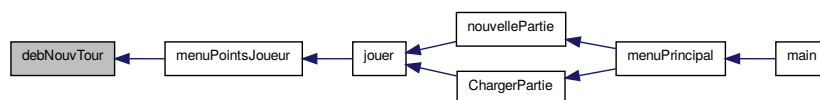
4.16.2.4 void debNouvTour (Fichier_Jeu * ptr_struct_fichier)

Realloue l'espace memoire dedie au points sur la structure [Fichier_Jeu](#) mis en parametre

Parameters

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

Here is the caller graph for this function:



4.16.2.5 int depScoreMax (Fichier_Jeu * ptr_struct_fichier)

Verifie si quelqu'un a depasse le score maximum

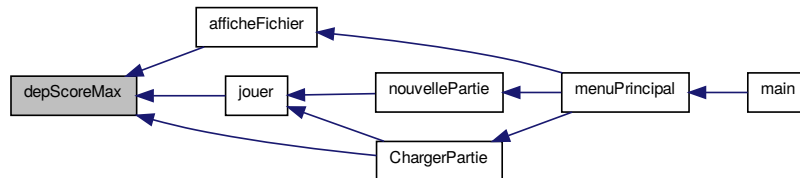
Parameters

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

Returns

VRAI s'il y a un depassement, FAUX sinon

Here is the caller graph for this function:



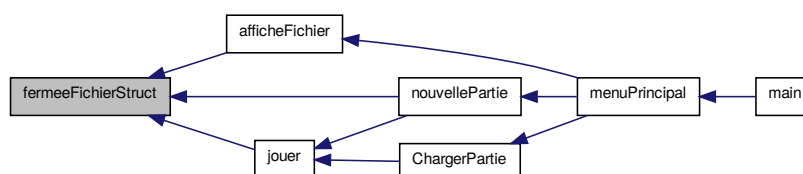
4.16.2.6 void fermeeFichierStruct (Fichier_Jeu * ptr_struct_fichier)

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

Parameters

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

Here is the caller graph for this function:



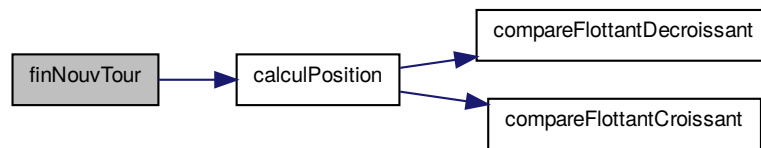
4.16.2.7 void finNouvTour (Fichier_Jeu * ptr_struct_fichier)

Met a jour les points totaux, le nombre de tour, la personne qui doit distribuer et les positions

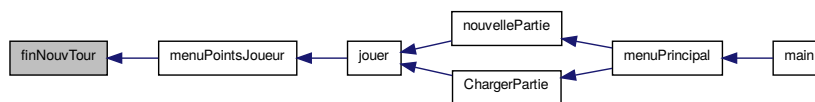
Parameters

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

Here is the call graph for this function:



Here is the caller graph for this function:



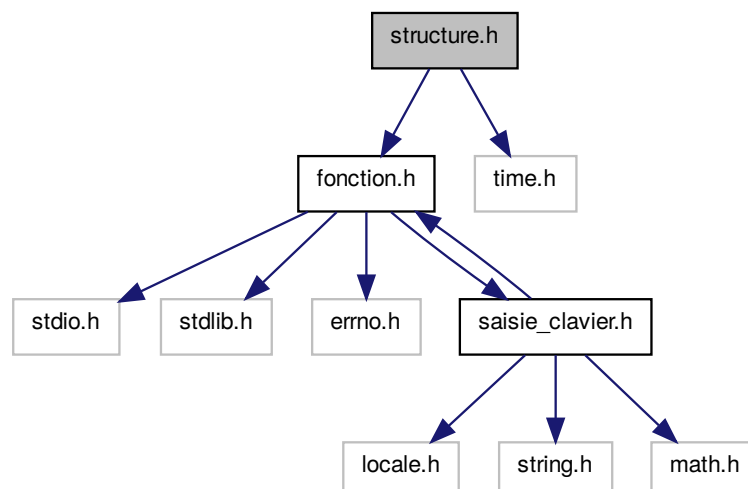
4.17 structure.h File Reference

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

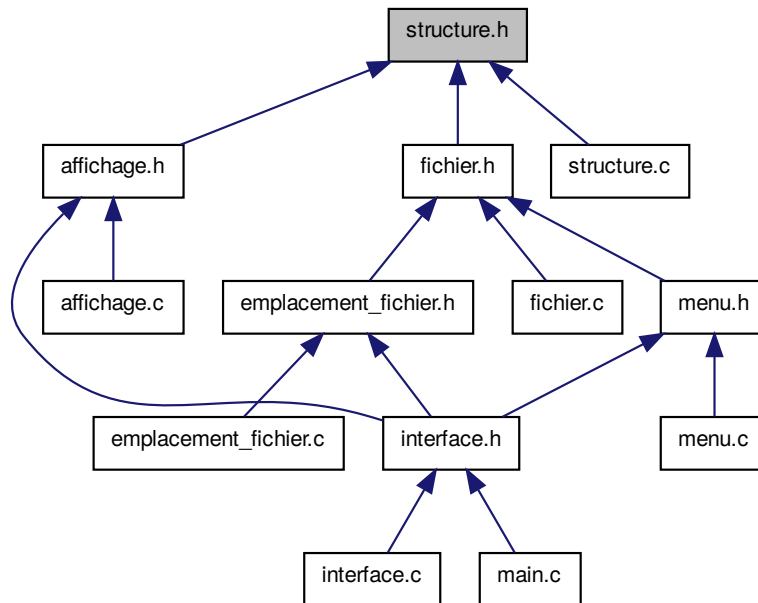
```
#include "fonction.h"
```

```
#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](#) 1.3

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)
- void [finNouvTour](#) ([Fichier_Jeu](#) *ptr_struct_fichier)
- void [calculPosition](#) ([Fichier_Jeu](#) *ptr_struct_fichier)
- void [ajoutDistribueStruct](#) ([Fichier_Jeu](#) *ptr_struct_fichier, char *nom_distribue)
- int [depScoreMax](#) ([Fichier_Jeu](#) *ptr_struct_fichier)

4.17.1 Detailed Description

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

Author

Remi BERTHO

Date

13/02/14

Version

2.0

4.17.2 Macro Definition Documentation

4.17.2.1 #define TAILLE_MAX_NOM 30

Definit la taille max d'un nom a 30

4.17.2.2 #define VERSION 1.3

Definit la version a 1.3

4.17.3 Function Documentation

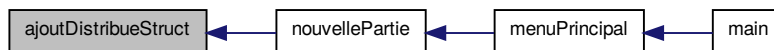
4.17.3.1 void ajoutDistribueStruct (Fichier_Jeu * ptr_struct_fichier, char * nom_distribue)

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:



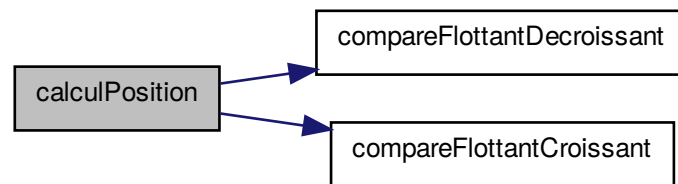
4.17.3.2 void calculPosition (Fichier_Jeu * ptr_struct_fichier)

Calcule les positions des joueurs

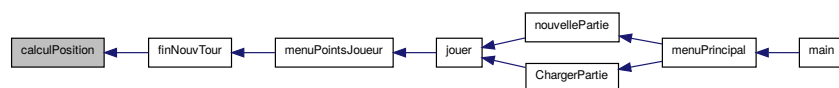
Parameters

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:



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

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

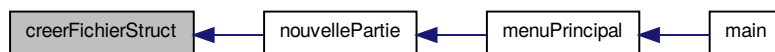
Parameters

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 caller graph for this function:



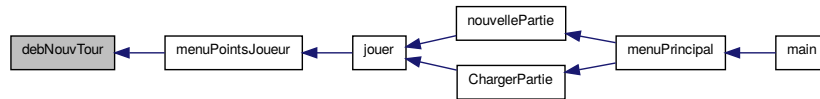
4.17.3.4 void debNouvTour (Fichier_Jeu * ptr_struct_fichier)

Realloue l'espace memoire dedie au points sur la structure [Fichier_Jeu](#) mis en parametre

Parameters

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

Here is the caller graph for this function:



4.17.3.5 int depScoreMax (Fichier_Jeu * ptr_struct_fichier)

Verifie si quelqu'un a depasse le score maximum

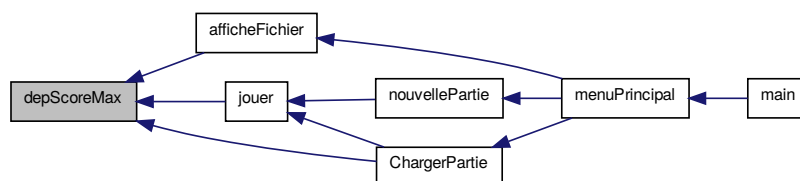
Parameters

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:



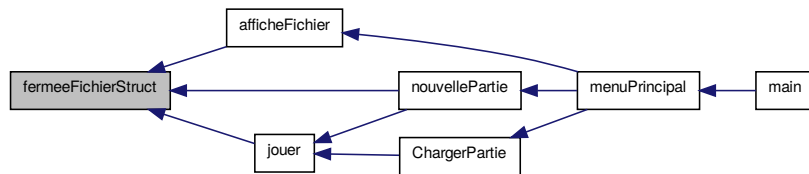
4.17.3.6 void fermeeFichierStruct (Fichier_Jeu * ptr_struct_fichier)

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

Parameters

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

Here is the caller graph for this function:



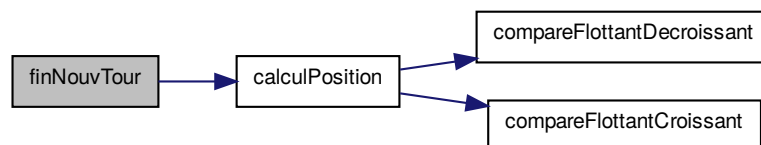
4.17.3.7 void finNouvTour (Fichier_Jeu * ptr_struct_fichier)

Met a jour les points totaux, le nombre de tour, la personne qui doit distribuer et les positions

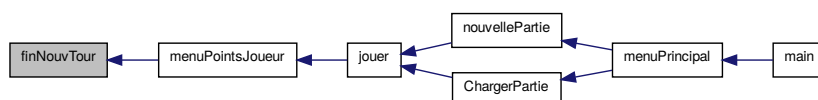
Parameters

<i>in, out</i>	<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:



Index

affFich
 interface.h, 73

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

affichage.h, 17
 afficherChaineTroisTab, 19
 afficherDistribue, 19
 afficherEnTete, 20
 afficherLicense, 20
 afficherLigne, 20
 afficherNom, 22
 afficherPartieFinie, 22
 afficherPosition, 23
 afficherScore, 23
 afficherScoreEntier, 24
 afficherScoreTotal, 25
 afficherStruct, 25

afficheFichier
 interface.c, 63
 interface.h, 73

afficherChaineTroisTab
 affichage.c, 10
 affichage.h, 19

afficherDistribue
 affichage.c, 10
 affichage.h, 19

afficherEnTete
 affichage.c, 11
 affichage.h, 20

afficherLicense
 affichage.c, 11
 affichage.h, 20

afficherLigne
 affichage.c, 11
 affichage.h, 20

afficherNom
 affichage.c, 13
 affichage.h, 22

afficherPartieFinie
 affichage.c, 13
 affichage.h, 22

afficherPosition
 affichage.c, 14
 affichage.h, 23

afficherScore
 affichage.c, 14
 affichage.h, 23

afficherScoreEntier
 affichage.c, 15
 affichage.h, 24

afficherScoreTotal
 affichage.c, 16
 affichage.h, 25

afficherStruct
 affichage.c, 16
 affichage.h, 25

ajoutDistribueStruct
 structure.c, 112
 structure.h, 118

annee
 Fichier_Jeu, 5

calculPosition
 structure.c, 113
 structure.h, 118

changerCheminFichier
 emplacement_fichier.c, 28
 emplacement_fichier.h, 33

charPart
 interface.h, 73

ChargerPartie
 interface.c, 64
 interface.h, 74

clean_stdin
 saisie_clavier.c, 99
 saisie_clavier.h, 105

compareFlottantCroissant
 fonction.c, 51
 fonction.h, 58

compareFlottantDecroissant
 fonction.c, 51
 fonction.h, 58

creationPreferences
 emplacement_fichier.c, 28
 emplacement_fichier.h, 34

creerFichierStruct
 structure.c, 113
 structure.h, 119

- debNouvTour
 - structure.c, 114
 - structure.h, 119
- depScoreMax
 - structure.c, 114
 - structure.h, 120
- distribue
 - Fichier_Jeu, 5
- EXTENSION_FICHER
 - fichier.h, 45
- ecrireFichier
 - fichier.c, 38
 - fichier.h, 45
- emplacement_fichier.c, 26
 - changerCheminFichier, 28
 - creationPreferences, 28
 - lectureCheminFichier, 29
 - lecturePreferences, 30
- emplacement_fichier.h, 31
 - changerCheminFichier, 33
 - creationPreferences, 34
 - lectureCheminFichier, 35
 - lecturePreferences, 36
 - NOM_DOSSIER, 33
 - NOM_FICHER, 33
- FAUX
 - fonction.h, 58
- fermeeFichierStruct
 - structure.c, 115
 - structure.h, 120
- fermerFichier
 - fonction.c, 51
 - fonction.h, 58
- fichier.c, 37
 - ecrireFichier, 38
 - lireFichier, 39
 - nouveauScore, 39
 - ouvrirFichierExtension, 41
 - renommerFichier, 42
 - supprimerFichier, 42
- fichier.h, 43
 - EXTENSION_FICHER, 45
 - ecrireFichier, 45
 - lireFichier, 46
 - nouveauScore, 46
 - ouvrirFichierExtension, 48
 - renommerFichier, 49
 - supprimerFichier, 49
 - TYPE_FICHER, 45
- 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, 115
 - structure.h, 121
- fonction.c, 50
 - compareFlottantCroissant, 51
 - compareFlottantDecroissant, 51
 - fermerFichier, 51
 - lireTailleFichier, 53
 - mauvais_choix, 53
 - ouvrirFichier, 54
 - systemEfface, 54
 - systemPause, 55
- fonction.h, 56
 - compareFlottantCroissant, 58
 - compareFlottantDecroissant, 58
 - FAUX, 58
 - fermerFichier, 58
 - lireTailleFichier, 59
 - mauvais_choix, 59
 - ouvrirFichier, 60
 - systemEfface, 60
 - systemPause, 61
 - VRAI, 58
- interface.h
 - affFich, 73
 - charPart, 73
 - lireChem, 73
 - listFich, 73
 - nouvChem, 73
 - nouvPart, 73
 - quit, 73
 - supprFich, 73
- interface.c, 62
 - afficheFichier, 63
 - ChargerPartie, 64
 - jouer, 65
 - lireCheminFichier, 66
 - listerFichier, 67
 - menuPrincipal, 68
 - nouveauCheminFichier, 69
 - nouvellePartie, 69
 - supprimerFichierNom, 70
- interface.h, 71
 - afficheFichier, 73
 - ChargerPartie, 74
 - jouer, 75
 - lireCheminFichier, 76
 - listerFichier, 77
 - Menu, 73
 - menuPrincipal, 78
 - nouveauCheminFichier, 79

- nouvellePartie, 79
- supprimerFichierNom, 80
- jouer
 - interface.c, 65
 - interface.h, 75
- jour
 - Fichier_Jeu, 5
- lectureCheminFichier
 - emplacement_fichier.c, 29
 - emplacement_fichier.h, 35
- lecturePreferences
 - emplacement_fichier.c, 30
 - emplacement_fichier.h, 36
- lireChem
 - interface.h, 73
- lireCheminFichier
 - interface.c, 66
 - interface.h, 76
- lireFichier
 - fichier.c, 39
 - fichier.h, 46
- lireTailleFichier
 - fonction.c, 53
 - fonction.h, 59
- listFich
 - interface.h, 73
- listerFichier
 - interface.c, 67
 - interface.h, 77
- main
 - main.c, 82
- main.c, 81
 - main, 82
- mauvais_choix
 - fonction.c, 53
 - fonction.h, 59
- Menu
 - interface.h, 73
- menu.c, 83
 - menuContinuer, 85
 - menuDebutPartie, 85
 - menuDistribue, 86
 - menuNomFichier, 86
 - menuNomJoueur, 88
 - menuNouveauChemin, 89
 - menuPointsJoueur, 89
 - menuSupprimer, 90
- menu.h, 91
 - menuContinuer, 92
 - menuDebutPartie, 93
 - menuDistribue, 94
 - menuNomFichier, 94
 - menuNomJoueur, 95
 - menuNouveauChemin, 96
 - menuPointsJoueur, 96
 - menuSupprimer, 97
- menuContinuer
 - menu.c, 85
 - menu.h, 92
- menuDebutPartie
 - menu.c, 85
 - menu.h, 93
- menuDistribue
 - menu.c, 86
 - menu.h, 94
- menuNomFichier
 - menu.c, 86
 - menu.h, 94
- menuNomJoueur
 - menu.c, 88
 - menu.h, 95
- menuNouveauChemin
 - menu.c, 89
 - menu.h, 96
- menuPointsJoueur
 - menu.c, 89
 - menu.h, 96
- menuPrincipal
 - interface.c, 68
 - interface.h, 78
- menuSupprimer
 - menu.c, 90
 - menu.h, 97
- mois
 - Fichier_Jeu, 6
- NB_CARACT_DOUB
 - saisie_clavier.h, 105
- NB_CARACT_FLOT
 - saisie_clavier.h, 105
- NB_CARACT_INT
 - saisie_clavier.h, 105
- NOM_DOSSIER
 - emplacement_fichier.h, 33
- NOM_FICHER
 - emplacement_fichier.h, 33
- nb_joueur
 - Fichier_Jeu, 6
- nb_max
 - Fichier_Jeu, 6
- nb_tour
 - Fichier_Jeu, 6
- nom_joueur
 - Fichier_Jeu, 6
- nouvChem
 - interface.h, 73
- nouvPart
 - interface.h, 73
- nouveauCheminFichier
 - interface.c, 69
 - interface.h, 79
- nouveauScore
 - fichier.c, 39
 - fichier.h, 46
- nouvellePartie

- interface.c, 69
- interface.h, 79
- ouvrirFichier
 - fonction.c, 54
 - fonction.h, 60
- ouvrirFichierExtension
 - fichier.c, 41
 - fichier.h, 48
- point
 - Fichier_Jeu, 6
- point_tot
 - Fichier_Jeu, 6
- position
 - Fichier_Jeu, 6
- quit
 - interface.h, 73
- renommerFichier
 - fichier.c, 42
 - fichier.h, 49
- saisie_clavier.c, 98
 - clean_stdin, 99
 - saisieClavierCaractere, 99
 - saisieClavierChaine, 100
 - saisieClavierDouble, 101
 - saisieClavierEntier, 101
 - saisieClavierFlottant, 102
 - saisieClavierFlottantSansVirgule, 102
- saisie_clavier.h, 103
 - clean_stdin, 105
 - NB_CARACT_DOUB, 105
 - NB_CARACT_FLOT, 105
 - NB_CARACT_INT, 105
 - saisieClavierCaractere, 105
 - saisieClavierChaine, 106
 - saisieClavierDouble, 107
 - saisieClavierEntier, 107
 - saisieClavierFlottant, 108
 - saisieClavierFlottantSansVirgule, 108
- saisieClavierCaractere
 - saisie_clavier.c, 99
 - saisie_clavier.h, 105
- saisieClavierChaine
 - saisie_clavier.c, 100
 - saisie_clavier.h, 106
- saisieClavierDouble
 - saisie_clavier.c, 101
 - saisie_clavier.h, 107
- saisieClavierEntier
 - saisie_clavier.c, 101
 - saisie_clavier.h, 107
- saisieClavierFlottant
 - saisie_clavier.c, 102
 - saisie_clavier.h, 108
- saisieClavierFlottantSansVirgule
 - saisie_clavier.c, 102
 - saisie_clavier.h, 108
- saisie_clavier.c, 102
- saisie_clavier.h, 108
- sens_premier
 - Fichier_Jeu, 6
- structure.c, 110
 - ajoutDistribueStruct, 112
 - calculPosition, 113
 - creerFichierStruct, 113
 - debNouvTour, 114
 - depScoreMax, 114
 - fermeeFichierStruct, 115
 - finNouvTour, 115
- structure.h, 116
 - ajoutDistribueStruct, 118
 - calculPosition, 118
 - creerFichierStruct, 119
 - debNouvTour, 119
 - depScoreMax, 120
 - fermeeFichierStruct, 120
 - finNouvTour, 121
 - TAILLE_MAX_NOM, 118
 - VERSION, 118
- supprFich
 - interface.h, 73
- supprimerFichier
 - fichier.c, 42
 - fichier.h, 49
- supprimerFichierNom
 - interface.c, 70
 - interface.h, 80
- systemEfface
 - fonction.c, 54
 - fonction.h, 60
- systemPause
 - fonction.c, 55
 - fonction.h, 61
- TAILLE_MAX_NOM
 - structure.h, 118
- TYPE_FICHER
 - fichier.h, 45
- taille_max_nom
 - Fichier_Jeu, 6
- VERSION
 - structure.h, 118
- VRAI
 - fonction.h, 58
- version
 - Fichier_Jeu, 6