

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

## 2.1.6

Generated by Doxygen 1.8.6

Wed Mar 26 2014 20:35:49



# Contents

<b>1</b>	<b>Data Structure Index</b>	<b>1</b>
1.1	Data Structures . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Data Structure Documentation</b>	<b>5</b>
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	number_after_comma . . . . .	6
3.1.2.10	point . . . . .	6
3.1.2.11	point_tot . . . . .	6
3.1.2.12	position . . . . .	6
3.1.2.13	sens_premier . . . . .	6
3.1.2.14	taille_max_nom . . . . .	6
3.1.2.15	tour_par_tour . . . . .	7
3.1.2.16	use_distributor . . . . .	7
3.1.2.17	version . . . . .	7
<b>4</b>	<b>File Documentation</b>	<b>9</b>
4.1	affichage.c File Reference . . . . .	9
4.1.1	Detailed Description . . . . .	10
4.1.2	Function Documentation . . . . .	10

4.1.2.1	<a href="#">afficherChaineTroisTab</a>	10
4.1.2.2	<a href="#">afficherDistribue</a>	10
4.1.2.3	<a href="#">afficherEnTete</a>	10
4.1.2.4	<a href="#">afficherLicense</a>	11
4.1.2.5	<a href="#">afficherLigne</a>	11
4.1.2.6	<a href="#">afficherNom</a>	11
4.1.2.7	<a href="#">afficherPartieFinie</a>	11
4.1.2.8	<a href="#">afficherPosition</a>	12
4.1.2.9	<a href="#">afficherScore</a>	12
4.1.2.10	<a href="#">afficherScoreEntier</a>	13
4.1.2.11	<a href="#">afficherScoreTotal</a>	13
4.1.2.12	<a href="#">afficherStruct</a>	13
4.2	<a href="#">affichage.h File Reference</a>	14
4.2.1	<a href="#">Detailed Description</a>	15
4.2.2	<a href="#">Function Documentation</a>	16
4.2.2.1	<a href="#">afficherChaineTroisTab</a>	16
4.2.2.2	<a href="#">afficherDistribue</a>	16
4.2.2.3	<a href="#">afficherEnTete</a>	16
4.2.2.4	<a href="#">afficherLicense</a>	16
4.2.2.5	<a href="#">afficherLigne</a>	16
4.2.2.6	<a href="#">afficherNom</a>	17
4.2.2.7	<a href="#">afficherPartieFinie</a>	17
4.2.2.8	<a href="#">afficherPosition</a>	17
4.2.2.9	<a href="#">afficherScore</a>	17
4.2.2.10	<a href="#">afficherScoreEntier</a>	18
4.2.2.11	<a href="#">afficherScoreTotal</a>	18
4.2.2.12	<a href="#">afficherStruct</a>	19
4.3	<a href="#">argument_main.c File Reference</a>	19
4.3.1	<a href="#">Detailed Description</a>	20
4.3.2	<a href="#">Function Documentation</a>	20
4.3.2.1	<a href="#">searchArgument</a>	20
4.4	<a href="#">argument_main.h File Reference</a>	21
4.4.1	<a href="#">Detailed Description</a>	22
4.4.2	<a href="#">Macro Definition Documentation</a>	22
4.4.2.1	<a href="#">CHAINE_LECTURE_FICHER</a>	22
4.4.2.2	<a href="#">CHAINE_LECTURE_FICHER_RED</a>	22
4.4.2.3	<a href="#">CHAINE_OUVERTURE_FICHER</a>	22
4.4.2.4	<a href="#">CHAINE_OUVERTURE_FICHER_RED</a>	22
4.4.2.5	<a href="#">LECTURE_FICHER</a>	22
4.4.2.6	<a href="#">OUVERTURE_FICHER</a>	22

4.4.3	Function Documentation	22
4.4.3.1	searchArgument	22
4.5	emplacement_fichier.c File Reference	23
4.5.1	Detailed Description	24
4.5.2	Function Documentation	24
4.5.2.1	changerCheminFichier	24
4.5.2.2	creationPreferences	25
4.5.2.3	lectureCheminFichier	25
4.5.2.4	lecturePreferences	25
4.6	emplacement_fichier.h File Reference	26
4.6.1	Detailed Description	27
4.6.2	Macro Definition Documentation	28
4.6.2.1	NOM_DOSSIER	28
4.6.2.2	NOM_FICHIER	28
4.6.2.3	PORTABLE	28
4.6.3	Function Documentation	28
4.6.3.1	changerCheminFichier	28
4.6.3.2	creationPreferences	28
4.6.3.3	lectureCheminFichier	29
4.6.3.4	lecturePreferences	29
4.7	fichier.c File Reference	30
4.7.1	Detailed Description	31
4.7.2	Function Documentation	32
4.7.2.1	ecrireFichier	32
4.7.2.2	lireFichier	32
4.7.2.3	nouveauScore	33
4.7.2.4	ouvrirFichierExtension	34
4.7.2.5	renommerFichier	35
4.7.2.6	supprimerFichier	35
4.8	fichier.h File Reference	36
4.8.1	Detailed Description	37
4.8.2	Macro Definition Documentation	37
4.8.2.1	EXTENSION_FICHIER	37
4.8.2.2	TAILLE_MAX_NOM_FICHIER	37
4.8.2.3	TYPE_FICHIER	37
4.8.3	Function Documentation	37
4.8.3.1	ecrireFichier	37
4.8.3.2	lireFichier	38
4.8.3.3	nouveauScore	39
4.8.3.4	ouvrirFichierExtension	39

4.8.3.5	renommerFichier	40
4.8.3.6	supprimerFichier	40
4.9	fonction.c File Reference	40
4.9.1	Detailed Description	41
4.9.2	Function Documentation	42
4.9.2.1	ajoutExtension	42
4.9.2.2	compareFlottantCroissant	42
4.9.2.3	compareFlottantDecroissant	42
4.9.2.4	fermerFichier	42
4.9.2.5	lireTailleFichier	42
4.9.2.6	mauvais_choix	43
4.9.2.7	myAlloc	43
4.9.2.8	myRealloc	43
4.9.2.9	ouvrirFichier	43
4.9.2.10	systemEfface	43
4.9.2.11	systemPause	43
4.10	fonction.h File Reference	44
4.10.1	Detailed Description	44
4.10.2	Macro Definition Documentation	45
4.10.2.1	FAUX	45
4.10.2.2	VRAI	45
4.10.3	Function Documentation	45
4.10.3.1	ajoutExtension	45
4.10.3.2	compareFlottantCroissant	45
4.10.3.3	compareFlottantDecroissant	45
4.10.3.4	fermerFichier	45
4.10.3.5	lireTailleFichier	46
4.10.3.6	mauvais_choix	46
4.10.3.7	myAlloc	46
4.10.3.8	myRealloc	46
4.10.3.9	ouvrirFichier	46
4.10.3.10	systemEfface	46
4.10.3.11	systemPause	47
4.11	interface.c File Reference	47
4.11.1	Detailed Description	48
4.11.2	Function Documentation	48
4.11.2.1	afficheFichier	48
4.11.2.2	afficheFichierLocale	49
4.11.2.3	chargerPartie	50
4.11.2.4	chargerPartieLocale	51

4.11.2.5	<a href="#">jouer</a>	52
4.11.2.6	<a href="#">lireCheminFichier</a>	53
4.11.2.7	<a href="#">listerFichier</a>	54
4.11.2.8	<a href="#">menuPrincipal</a>	54
4.11.2.9	<a href="#">nouveauCheminFichier</a>	55
4.11.2.10	<a href="#">nouvellePartie</a>	56
4.11.2.11	<a href="#">supprimerFichierNom</a>	57
4.12	<a href="#">interface.h File Reference</a>	58
4.12.1	<a href="#">Detailed Description</a>	59
4.12.2	<a href="#">Enumeration Type Documentation</a>	59
4.12.2.1	<a href="#">Menu</a>	59
4.12.3	<a href="#">Function Documentation</a>	59
4.12.3.1	<a href="#">afficheFichier</a>	59
4.12.3.2	<a href="#">afficheFichierLocale</a>	60
4.12.3.3	<a href="#">chargerPartie</a>	61
4.12.3.4	<a href="#">chargerPartieLocale</a>	62
4.12.3.5	<a href="#">jouer</a>	63
4.12.3.6	<a href="#">lireCheminFichier</a>	64
4.12.3.7	<a href="#">listerFichier</a>	65
4.12.3.8	<a href="#">menuPrincipal</a>	65
4.12.3.9	<a href="#">nouveauCheminFichier</a>	66
4.12.3.10	<a href="#">nouvellePartie</a>	67
4.12.3.11	<a href="#">supprimerFichierNom</a>	68
4.13	<a href="#">main.c File Reference</a>	69
4.13.1	<a href="#">Detailed Description</a>	70
4.13.2	<a href="#">Function Documentation</a>	70
4.13.2.1	<a href="#">main</a>	70
4.14	<a href="#">main.h File Reference</a>	71
4.14.1	<a href="#">Detailed Description</a>	72
4.14.2	<a href="#">Function Documentation</a>	72
4.14.2.1	<a href="#">main</a>	72
4.15	<a href="#">menu.c File Reference</a>	74
4.15.1	<a href="#">Detailed Description</a>	74
4.15.2	<a href="#">Function Documentation</a>	75
4.15.2.1	<a href="#">menuContinuer</a>	75
4.15.2.2	<a href="#">menuDebutPartie</a>	75
4.15.2.3	<a href="#">menuDistribue</a>	75
4.15.2.4	<a href="#">menuNomFichier</a>	76
4.15.2.5	<a href="#">menuNomJoueur</a>	76
4.15.2.6	<a href="#">menuNouveauChemin</a>	77

4.15.2.7	menuNumJoueur	77
4.15.2.8	menuPointsJoueur	77
4.15.2.9	menuSupprimer	78
4.16	menu.h File Reference	78
4.16.1	Detailed Description	79
4.16.2	Function Documentation	80
4.16.2.1	menuContinuer	80
4.16.2.2	menuDebutPartie	80
4.16.2.3	menuDistribue	80
4.16.2.4	menuNomFichier	81
4.16.2.5	menuNomJoueur	81
4.16.2.6	menuNouveauChemin	82
4.16.2.7	menuNumJoueur	83
4.16.2.8	menuPointsJoueur	83
4.16.2.9	menuSupprimer	84
4.17	saisie_clavier.c File Reference	84
4.17.1	Detailed Description	85
4.17.2	Function Documentation	85
4.17.2.1	clean_stdin	85
4.17.2.2	saisieClavierCaractere	86
4.17.2.3	saisieClavierChaine	86
4.17.2.4	saisieClavierDouble	86
4.17.2.5	saisieClavierEntier	87
4.17.2.6	saisieClavierFlottant	87
4.17.2.7	saisieClavierFlottantSansVirgule	87
4.18	saisie_clavier.h File Reference	88
4.18.1	Detailed Description	89
4.18.2	Macro Definition Documentation	89
4.18.2.1	NB_CARACT_DOUB	89
4.18.2.2	NB_CARACT_FLOT	89
4.18.2.3	NB_CARACT_INT	89
4.18.3	Function Documentation	89
4.18.3.1	clean_stdin	89
4.18.3.2	saisieClavierCaractere	89
4.18.3.3	saisieClavierChaine	90
4.18.3.4	saisieClavierDouble	90
4.18.3.5	saisieClavierEntier	91
4.18.3.6	saisieClavierFlottant	92
4.18.3.7	saisieClavierFlottantSansVirgule	92
4.19	structure.c File Reference	92



4.19.1 Detailed Description . . . . .	93
4.19.2 Function Documentation . . . . .	94
4.19.2.1 ajoutDistribueStruct . . . . .	94
4.19.2.2 calculPosition . . . . .	94
4.19.2.3 creerFichierStruct . . . . .	95
4.19.2.4 debNouvTour . . . . .	96
4.19.2.5 depScoreMax . . . . .	96
4.19.2.6 fermeeFichierStruct . . . . .	96
4.19.2.7 finNouvTour . . . . .	97
4.19.2.8 maxNbTour . . . . .	97
4.19.2.9 rechercheNumJoueur . . . . .	97
4.20 structure.h File Reference . . . . .	97
4.20.1 Detailed Description . . . . .	98
4.20.2 Macro Definition Documentation . . . . .	99
4.20.2.1 TAILLE_MAX_NOM . . . . .	99
4.20.2.2 VERSION . . . . .	99
4.20.3 Function Documentation . . . . .	99
4.20.3.1 ajoutDistribueStruct . . . . .	99
4.20.3.2 calculPosition . . . . .	99
4.20.3.3 creerFichierStruct . . . . .	100
4.20.3.4 debNouvTour . . . . .	101
4.20.3.5 depScoreMax . . . . .	101
4.20.3.6 fermeeFichierStruct . . . . .	101
4.20.3.7 finNouvTour . . . . .	101
4.20.3.8 maxNbTour . . . . .	102
4.20.3.9 rechercheNumJoueur . . . . .	103
<b>Index</b>	<b>104</b>



# Chapter 1

## Data Structure Index

### 1.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">Fichier_Jeu</a> . . . . .	5
---------------------------------------	---



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">affichage.c</a>	Fonctions qui gerent l'affichage des scores du logiciel . . . . .	9
<a href="#">affichage.h</a>	Prototypes des fonctions qui gerent l'affichage des scores du logiciel . . . . .	14
<a href="#">argument_main.c</a>	Lancement du programme . . . . .	19
<a href="#">argument_main.h</a>	Lancement du programme . . . . .	21
<a href="#">emplacement_fichier.c</a>	Fonctions qui l'emplacement des fichiers sauvegardes . . . . .	23
<a href="#">emplacement_fichier.h</a>	Prototypes des fonctions qui l'emplacement des fichiers sauvegardes . . . . .	26
<a href="#">fichier.c</a>	Fonction de gestion des fichiers . . . . .	30
<a href="#">fichier.h</a>	Prototypes des fonction de gestion des fichiers . . . . .	36
<a href="#">fonction.c</a>	Fonctions essentielles au fonctionnement du programme . . . . .	40
<a href="#">fonction.h</a>	Prototypes des fonctions essentielles au fonctionnement du programme . . . . .	44
<a href="#">interface.c</a>	Fonctions qui gerent l'interface graphique du logiciel . . . . .	47
<a href="#">interface.h</a>	Prototypes des fonctions qui gerent l'interface graphique du logiciel . . . . .	58
<a href="#">main.c</a>	Lancement du programme . . . . .	69
<a href="#">main.h</a>	Lancement du programme . . . . .	71
<a href="#">menu.c</a>	Fonctions qui gerent les menus demandant de rentrer des valeurs du logiciel . . . . .	74
<a href="#">menu.h</a>	Prototypes des fonctions qui gerent les menus demandant de rentrer des valeurs du logiciel . . . . .	78
<a href="#">saisie_clavier.c</a>	Fonctions de saisie clavier . . . . .	84
<a href="#">saisie_clavier.h</a>	Prototypes des fonctions de saisie clavier . . . . .	88
<a href="#">structure.c</a>	Fonction de gestion des fichiers de la struction contenant les informations . . . . .	92

[structure.h](#)

Prototypes des fonction de gestion des fichiers de la struction contenant les informations . . . [97](#)

## 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 [tour\\_par\\_tour](#)
- char [use\\_distributeur](#)
- char [number\\_after\\_comma](#)
- char \*\* [nom\\_joueur](#)
- float \* [point\\_tot](#)
- float \* [position](#)
- float \* [nb\\_tour](#)
- float [distribue](#)
- float \*\* [point](#)

#### 3.1.1 Detailed Description

Type representant un fichier .jeu

#### 3.1.2 Field Documentation

##### 3.1.2.1 float annee

Annee de creation de la structure.

##### 3.1.2.2 float distribue

Numero de la personne qui doit distribuer.

### 3.1.2.3 float jour

Jour de creation de la structure.

### 3.1.2.4 float mois

Mois de creation de la structure.

### 3.1.2.5 float nb\_joueur

Nombre de joueurs.

### 3.1.2.6 float nb\_max

Nombre maximum que peut prendre un joueur.

### 3.1.2.7 float\* nb\_tour

Nombre de tour dans le jeu par joueur.

### 3.1.2.8 char\*\* nom\_joueur

Tableau contenant tout les noms de joueurs.

### 3.1.2.9 char number\_after\_comma

Le nombre de chiffres apres la virgule dans l'affichage

### 3.1.2.10 float\*\* point

Tableau contenat les points de chaque joueur a chaque tour.

### 3.1.2.11 float\* point\_tot

Tableau contenant tout les points totaux des joueurs.

### 3.1.2.12 float\* position

Tableau contenant la position des joueurs.

### 3.1.2.13 char sens\_premier

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

### 3.1.2.14 float taille\_max\_nom

Taille maximum que peut prendre un nom de joueur.



## 3.1.2.15 char tour\_par\_tour

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

## 3.1.2.16 char use\_distributor

Vaut 1 si on utilise un distributeur, 0 sinon

## 3.1.2.17 float version

Version de la structure.

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

- [structure.h](#)



## Chapter 4

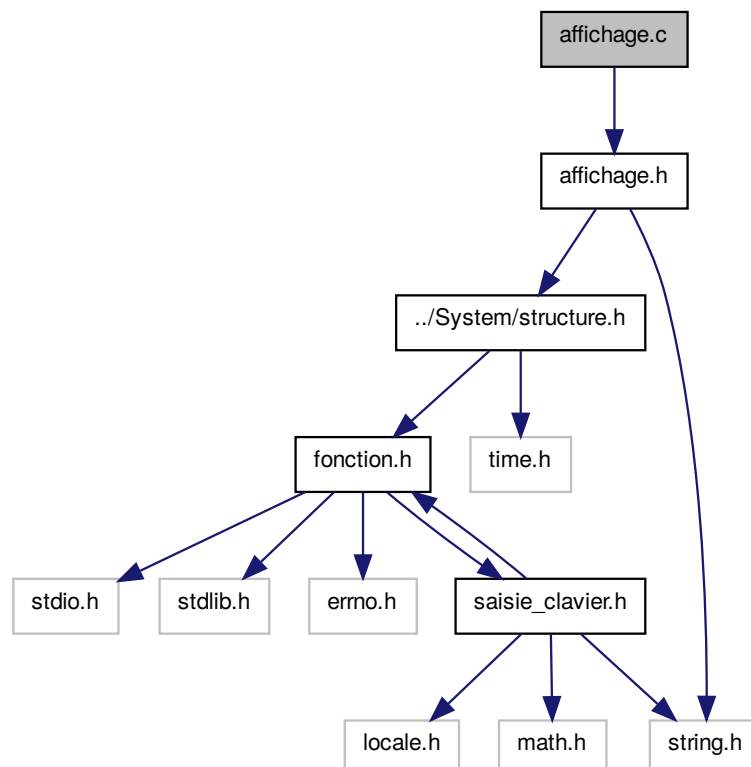
# File Documentation

### 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
----	-----------------	-------------------------

#### 4.1.2.2 void [afficherDistribue](#) ( [Fichier\\_Jeu](#) \* *ptr\_struct\_fichier* )

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

##### Parameters

in	* <i>ptr_struct_fichier</i>	un pointeur sur la structure <a href="#">Fichier_Jeu</a>
----	-----------------------------	--

#### 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 <a href="#">Fichier_Jeu</a>
----	----------------------------	--

Here is the call graph for this function:



## 4.1.2.4 void afficherLicense ( )

Affiche la license

4.1.2.5 void afficherLigne ( int *taille\_ligne* )

Affiche *taille\_ligne* - apres une tabulation

## Parameters

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

4.1.2.6 void afficherNom ( [Fichier\\_Jeu](#) \* *ptr\_struct\_fichier*, int \* *ptr\_taille\_ligne* )

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

## Parameters

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

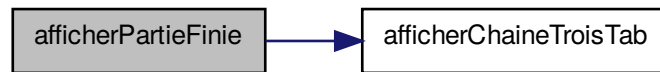
4.1.2.7 void afficherPartieFinie ( [Fichier\\_Jeu](#) \* *ptr\_struct\_fichier* )

Affiche un podium des resultats

## Parameters

in	<i>*ptr_struct_fichier</i>	un pointeur sur la structure <a href="#">Fichier_Jeu</a>
----	----------------------------	--

Here is the call graph for this function:



#### 4.1.2.8 void afficherPosition ( Fichier\_Jeu \* ptr\_struct\_fichier )

Affiche les positions des joueurs

##### Parameters

in	<i>*ptr_struct_fichier</i>	un pointeur sur la structure <a href="#">Fichier_Jeu</a>
----	----------------------------	--

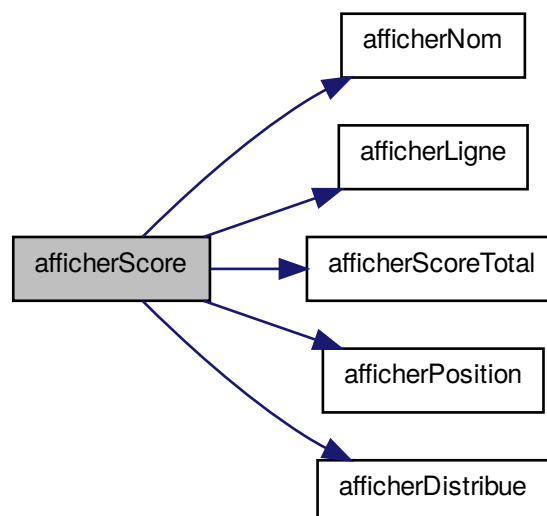
#### 4.1.2.9 void afficherScore ( Fichier\_Jeu \* ptr\_struct\_fichier )

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

##### Parameters

in	<i>*ptr_struct_fichier</i>	un pointeur sur la structure <a href="#">Fichier_Jeu</a>
----	----------------------------	--

Here is the call graph for this function:



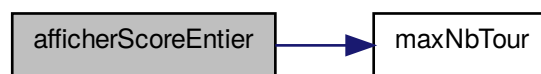
#### 4.1.2.10 void afficherScoreEntier ( Fichier\_Jeu \* ptr\_struct\_fichier )

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

##### Parameters

in	*ptr_struct_fichier	un pointeur sur la structure <a href="#">Fichier_Jeu</a>
----	---------------------	--

Here is the call graph for this function:



#### 4.1.2.11 void afficherScoreTotal ( Fichier\_Jeu \* ptr\_struct\_fichier )

Affiche le score total des joueurs

##### Parameters

in	*ptr_struct_fichier	un pointeur sur la structure <a href="#">Fichier_Jeu</a>
----	---------------------	--

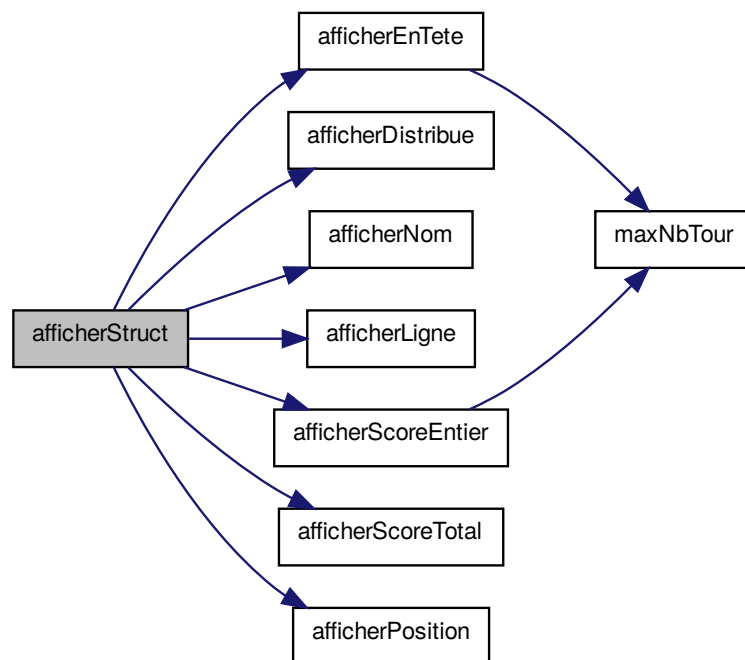
#### 4.1.2.12 void afficherStruct ( Fichier\_Jeu \* ptr\_struct\_fichier )

Affiche toutes les donnees d'une structure [Fichier\\_Jeu](#)

##### Parameters

in	*ptr_struct_fichier	un pointeur sur la structure <a href="#">Fichier_Jeu</a>
----	---------------------	--

Here is the call graph for this function:



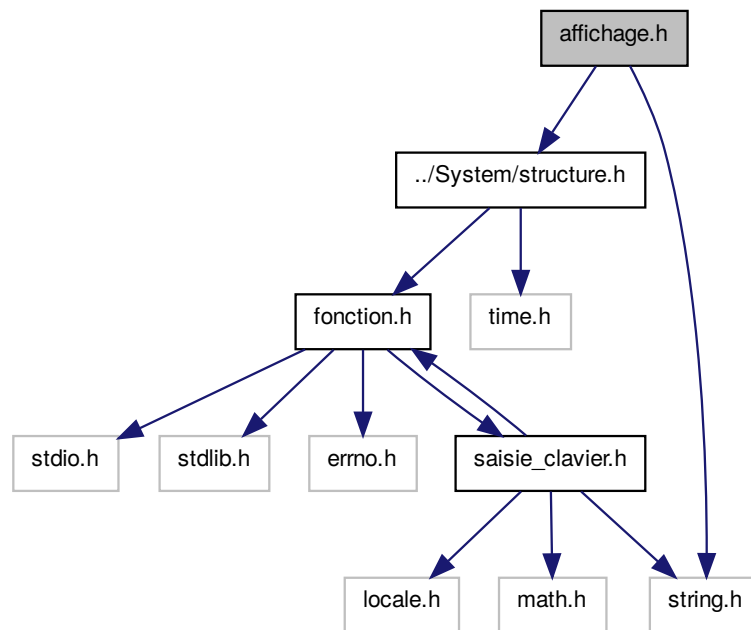
## 4.2 affichage.h File Reference

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

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



Include dependency graph for affichage.h:



## 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 gèrent l'affichage des scores du logiciel.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

## 4.2.2 Function Documentation

### 4.2.2.1 void afficherChaineTroisTab ( char \* *chaine* )

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

Parameters

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

### 4.2.2.2 void afficherDistribue ( Fichier\_Jeu \* *ptr\_struct\_fichier* )

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

Parameters

in	* <i>ptr_struct_fichier</i>	un pointeur sur la structure <a href="#">Fichier_Jeu</a>
----	-----------------------------	--

### 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 <a href="#">Fichier_Jeu</a>
----	-----------------------------	--

Here is the call graph for this function:



### 4.2.2.4 void afficherLicense ( )

Affiche la license

### 4.2.2.5 void afficherLigne ( int *taille\_ligne* )

Affiche *taille\_ligne* - apres une tabulation

## Parameters

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

**4.2.2.6 void afficherNom ( Fichier\_Jeu \* ptr\_struct\_fichier, int \* ptr\_taille\_ligne )**

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

## Parameters

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

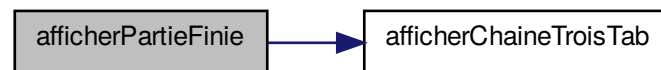
**4.2.2.7 void afficherPartieFinie ( Fichier\_Jeu \* ptr\_struct\_fichier )**

Affiche un podium des resultats

## Parameters

in	<i>*ptr_struct_fichier</i>	un pointeur sur la structure <a href="#">Fichier_Jeu</a>
----	----------------------------	--

Here is the call graph for this function:

**4.2.2.8 void afficherPosition ( Fichier\_Jeu \* ptr\_struct\_fichier )**

Affiche les positions des joueurs

## Parameters

in	<i>*ptr_struct_fichier</i>	un pointeur sur la structure <a href="#">Fichier_Jeu</a>
----	----------------------------	--

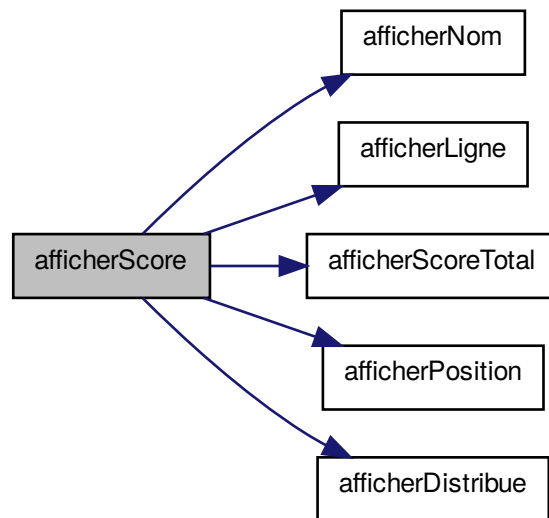
**4.2.2.9 void afficherScore ( Fichier\_Jeu \* ptr\_struct\_fichier )**

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

## Parameters

in	<i>*ptr_struct_fichier</i>	un pointeur sur la structure <a href="#">Fichier_Jeu</a>
----	----------------------------	--

Here is the call graph for this function:



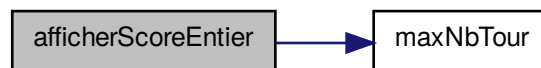
#### 4.2.2.10 void afficherScoreEntier ( Fichier\_Jeu \* ptr\_struct\_fichier )

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

##### Parameters

in	<i>*ptr_struct_fichier</i>	un pointeur sur la structure <a href="#">Fichier_Jeu</a>
----	----------------------------	--

Here is the call graph for this function:



#### 4.2.2.11 void afficherScoreTotal ( Fichier\_Jeu \* ptr\_struct\_fichier )

Affiche le score total des joueurs

## Parameters

in	<i>*ptr_struct_fichier</i>	un pointeur sur la structure <a href="#">Fichier_Jeu</a>
----	----------------------------	--

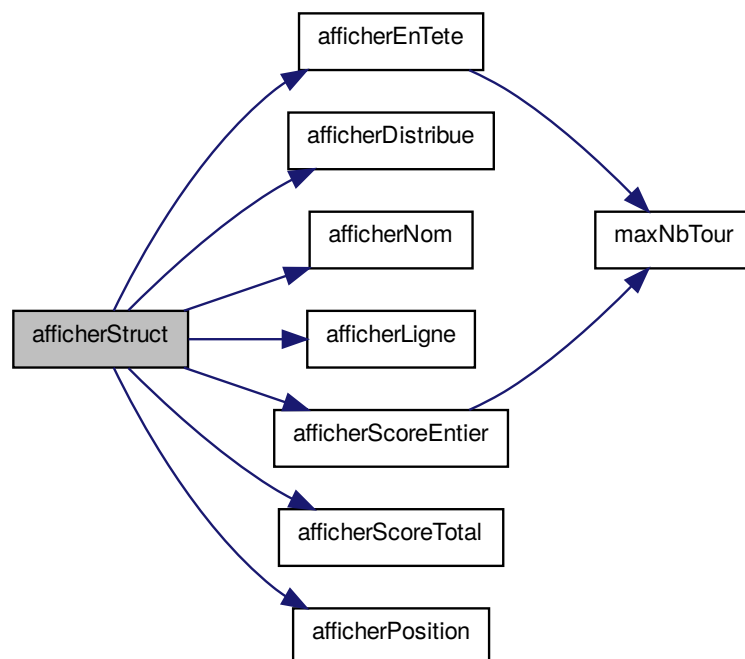
## 4.2.2.12 void afficherStruct ( Fichier\_Jeu \* ptr\_struct\_fichier )

Affiche toutes les donnes d'une structure [Fichier\\_Jeu](#)

## Parameters

in	<i>*ptr_struct_fichier</i>	un pointeur sur la structure <a href="#">Fichier_Jeu</a>
----	----------------------------	--

Here is the call graph for this function:

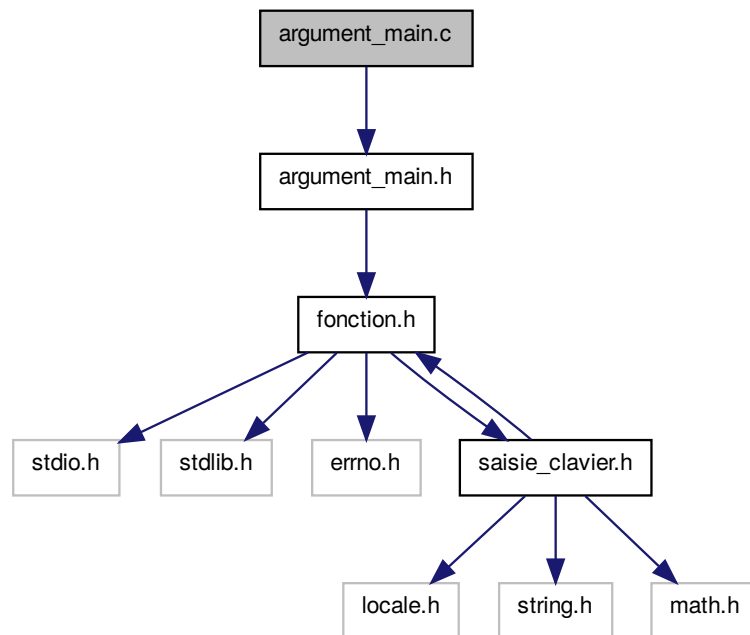


## 4.3 argument\_main.c File Reference

Lancement du programme.

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

Include dependency graph for argument\_main.c:



## Functions

- int [searchArgument](#) (int argc, char \*argv[], int \*fonction, int \*emplacement\_fichier)

### 4.3.1 Detailed Description

Lancement du programme.

#### Author

Remi BERTHO

#### Date

21/03/14

#### Version

2.1.0

### 4.3.2 Function Documentation

4.3.2.1 int `searchArgument` ( int *argc*, char \* *argv*[], int \* *fonction*, int \* *emplacement\_fichier* )

Lance le programme

## Parameters

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

## Returns

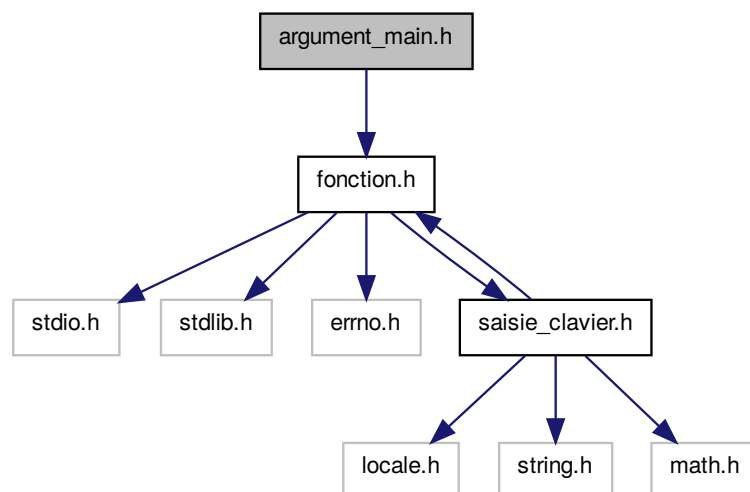
VRAI si la fonction a trouve un argument, FAUX sinon

## 4.4 argument\_main.h File Reference

Lancement du programme.

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

Include dependency graph for argument\_main.h:



## Macros

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

## Functions

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

#### 4.4.1 Detailed Description

Lancement du programme.

Author

Remi BERTHO

Date

21/03/14

Version

2.1.0

#### 4.4.2 Macro Definition Documentation

##### 4.4.2.1 #define CHAINE\_LECTURE\_FICHIER "--read"

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

##### 4.4.2.2 #define CHAINE\_LECTURE\_FICHIER\_RED "-r"

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

##### 4.4.2.3 #define CHAINE\_OUVERTURE\_FICHIER "--open"

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

##### 4.4.2.4 #define CHAINE\_OUVERTURE\_FICHIER\_RED "-o"

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

##### 4.4.2.5 #define LECTURE\_FICHIER 0

Definit l'appel a la lecture du fichier a 0

##### 4.4.2.6 #define OUVERTURE\_FICHIER 1

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

#### 4.4.3 Function Documentation

##### 4.4.3.1 int searchArgument ( int *argc*, char \* *argv*[], int \* *fonction*, int \* *emplacement\_fichier* )

Lance le programme



## Parameters

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

## Returns

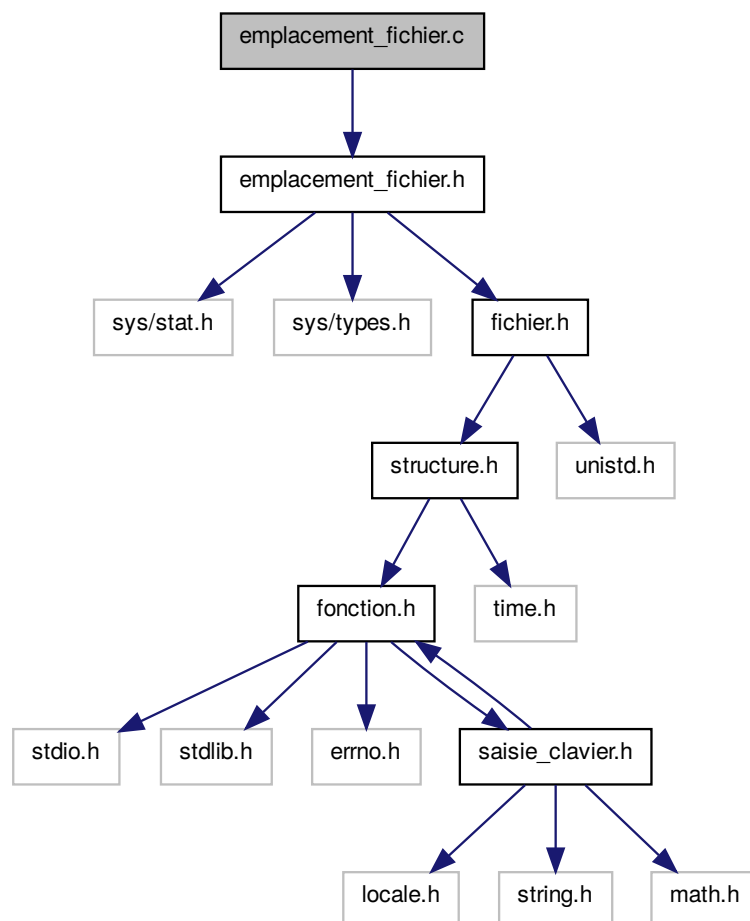
VRAI si la fonction a trouve un argument, FAUX sinon

## 4.5 emplacement\_fichier.c File Reference

Fonctions qui l'emplacement des fichiers sauvegardes.

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

Include dependency graph for emplacement\_fichier.c:



## Functions

- int [creationPreferences](#) ()
- int [lecturePreferences](#) (char \*nom\_fichier)
- int [lectureCheminFichier](#) (char \*nom\_fichier)
- int [changerCheminFichier](#) (char \*nouveauChemin)

### 4.5.1 Detailed Description

Fonctions qui l'emplacement des fichiers sauvegardes.

#### Author

Remi BERTHO

#### Date

13/02/14

#### Version

2.0

### 4.5.2 Function Documentation

#### 4.5.2.1 int [changerCheminFichier](#) ( char \* *nouveauChemin* )

Changer le chemin de sauvegarde des fichier par nouveauChemin

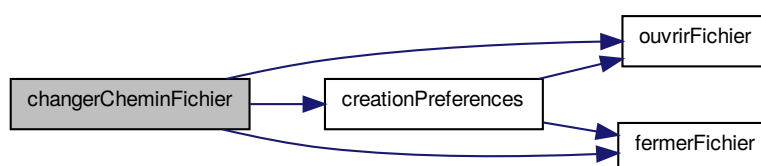
##### Parameters

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

##### Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



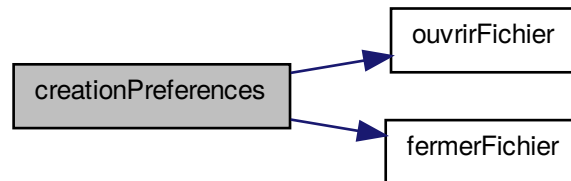
## 4.5.2.2 void creationPreferences ( )

Cree le dossier et le fichier qui va contenir les preferences

## Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



## 4.5.2.3 int lectureCheminFichier ( char \* nom\_fichier )

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

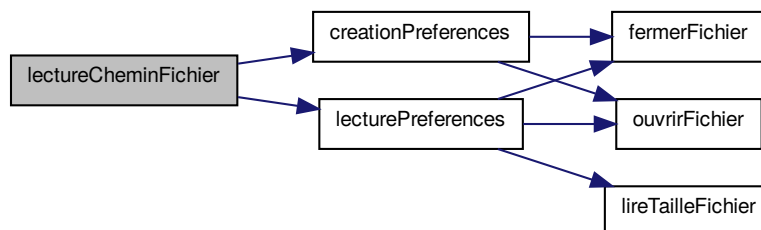
## Parameters

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

## Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



## 4.5.2.4 int lecturePreferences ( char \* nom\_fichier )

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

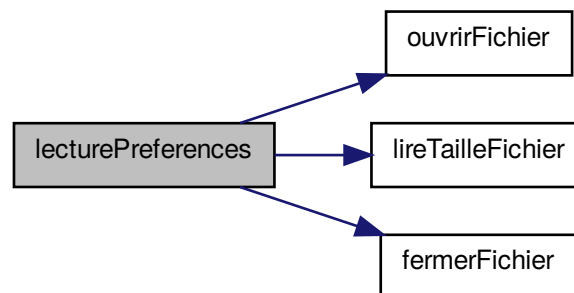
**Parameters**

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

**Returns**

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:

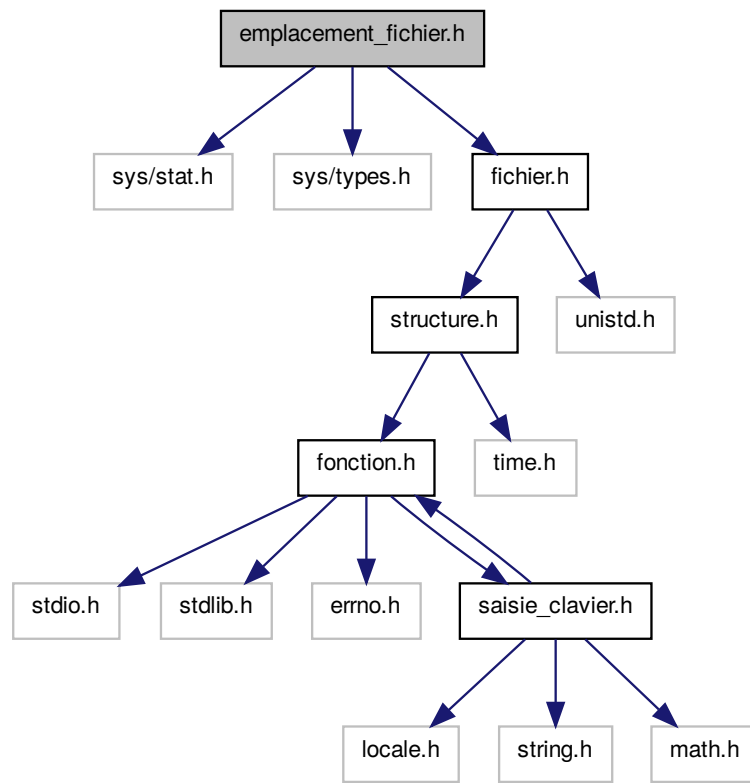


## 4.6 emplacement\_fichier.h File Reference

Prototypes des fonctions qui l'emplacement des fichiers sauvegardes.

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

Include dependency graph for emplacement\_fichier.h:



## Macros

- `#define` `PORTABLE`
- `#define` `NOM_FICHIER` `"preferences.txt"`
- `#define` `NOM_DOSSIER` `".csuper"`

## Functions

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

### 4.6.1 Detailed Description

Prototypes des fonctions qui l'emplacement des fichiers sauvegardes.

#### Author

Remi BERTHO

## Date

13/02/14

## Version

2.0

## 4.6.2 Macro Definition Documentation

### 4.6.2.1 #define NOM\_DOSSIER ".csuper"

Definit NOM\_DOSSIER a ".csuper"

### 4.6.2.2 #define NOM\_FICHIER "preferences.txt"

Definit NOM\_FICHIER a "preferences.txt"

### 4.6.2.3 #define PORTABLE

Definit PORTABLE

## 4.6.3 Function Documentation

### 4.6.3.1 int changerCheminFichier ( char \* *nouveauChemin* )

Changer le chemin de sauvegarde des fichier par nouveauChemin

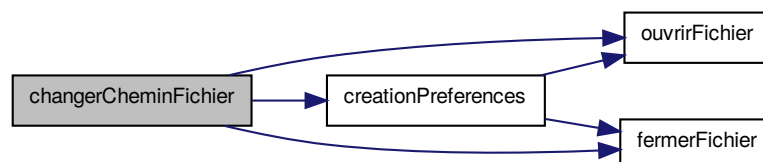
#### Parameters

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

#### Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



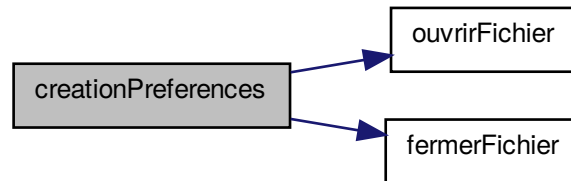
### 4.6.3.2 int creationPreferences ( )

Cree le dossier et le fichier qui va contenir les preferences

## Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



## 4.6.3.3 int lectureCheminFichier ( char \* nom\_fichier )

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

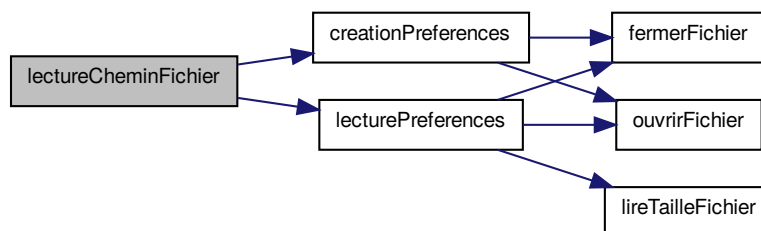
## Parameters

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

## Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



## 4.6.3.4 int lecturePreferences ( char \* nom\_fichier )

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

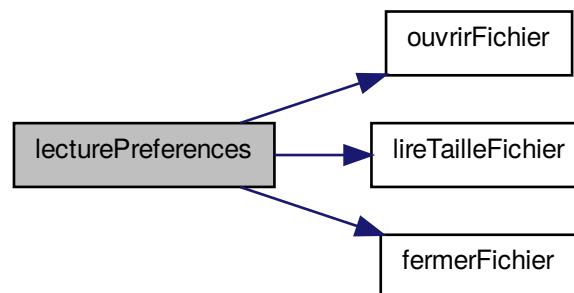
**Parameters**

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

**Returns**

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



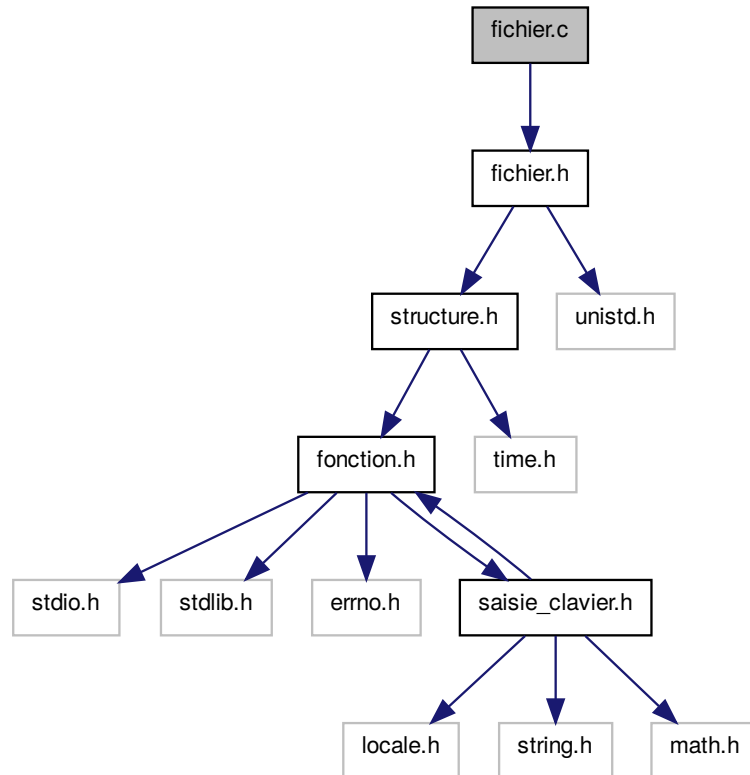
## 4.7 fichier.c File Reference

Fonction de gestion des fichiers.



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

Include dependency graph for fichier.c:



## Functions

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

### 4.7.1 Detailed Description

Fonction de gestion des fichiers.

#### Author

Remi BERTHO

#### Date

09/03/14

## Version

2.1.0

## 4.7.2 Function Documentation

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

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

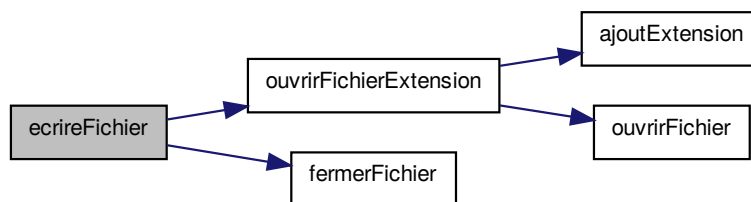
## Parameters

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

## Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:

4.7.2.2 `Fichier_Jeu * lireFichier ( char * nom )`

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

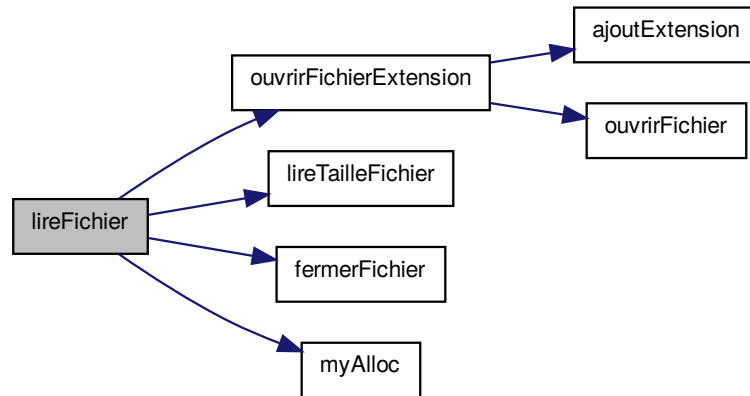
## Parameters

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

## Returns

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

Here is the call graph for this function:



#### 4.7.2.3 void nouveauScore ( char \* nom, Fichier\_Jeu \* ptr\_struct\_fichier )

Mets a jour le fichier avec les nouveaux scores

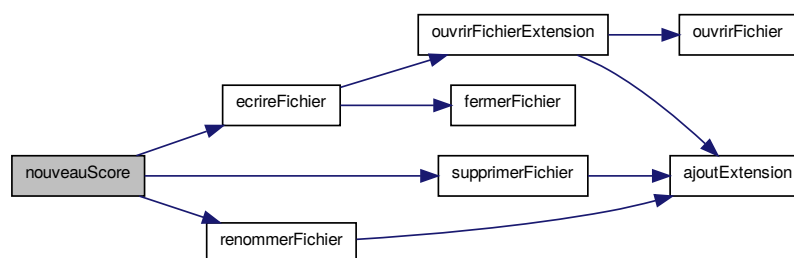
## Parameters

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

## Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



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

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

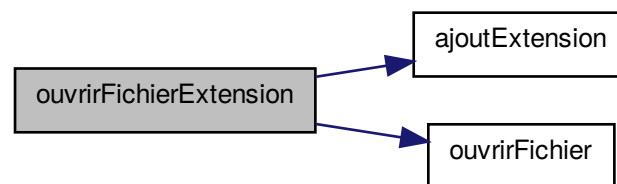
## Parameters

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

## Returns

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

Here is the call graph for this function:



#### 4.7.2.5 int renommerFichier ( char \* *nom\_ancien*, char \* *nom\_nouveau* )

Renomme le fichier dont le nom est en parametre

## Parameters

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

## Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



#### 4.7.2.6 int supprimerFichier ( char \* *nom* )

Supprime le fichier dont le nom est en parametre

## Parameters

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

## Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



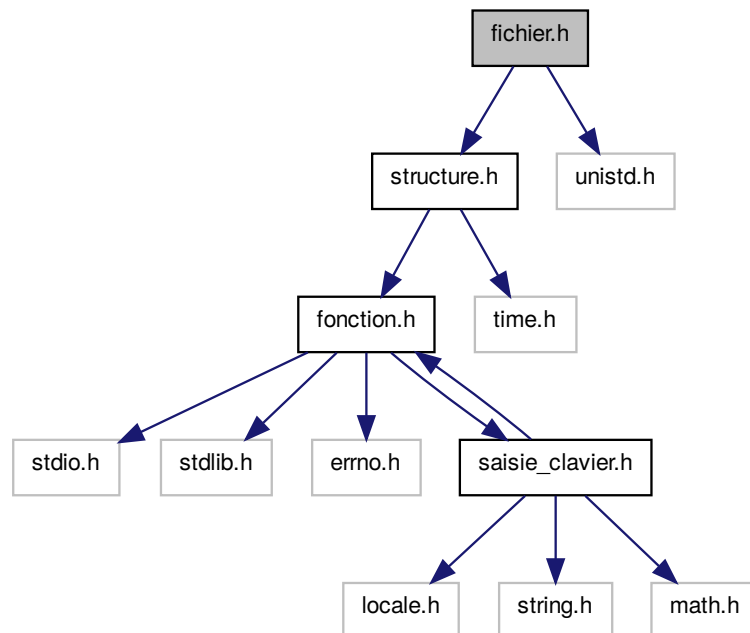
## 4.8 fichier.h File Reference

Prototypes des fonction de gestion des fichiers.

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

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

Include dependency graph for fichier.h:



## Macros

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

## Functions

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

### 4.8.1 Detailed Description

Prototypes des fonction de gestion des fichiers.

#### Author

Remi BERTHO

#### Date

13/02/14

#### Version

2.0

### 4.8.2 Macro Definition Documentation

#### 4.8.2.1 `#define EXTENSION_FICHIER "csu"`

Definit l'extension du fichier a "csu"

#### 4.8.2.2 `#define TAILLE_MAX_NOM_FICHIER 250`

Definit la taille max d'un nom a 250

#### 4.8.2.3 `#define TYPE_FICHIER "CompteurScoreUniversel"`

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

### 4.8.3 Function Documentation

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

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

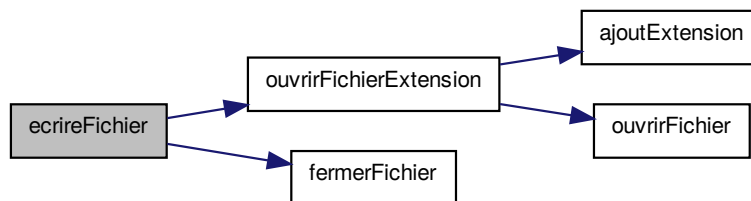
## Parameters

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

## Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



## 4.8.3.2 Fichier\_Jeu\* lireFichier ( char \* nom )

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

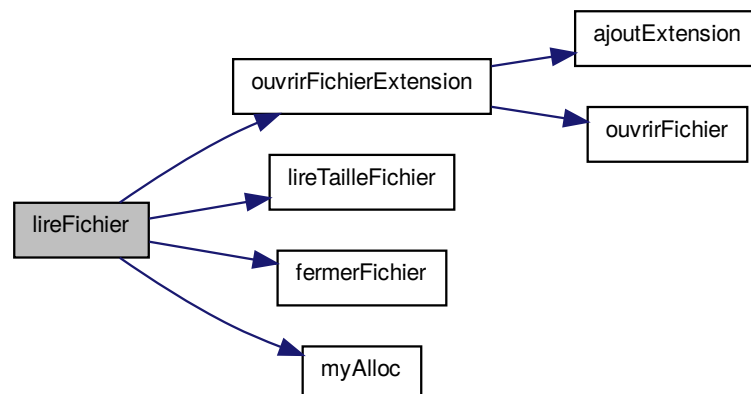
## Parameters

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

## Returns

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

Here is the call graph for this function:





#### 4.8.3.3 int nouveauScore ( char \* nom, Fichier\_Jeu \* ptr\_struct\_fichier )

Mets a jour le fichier avec les nouveaux scores

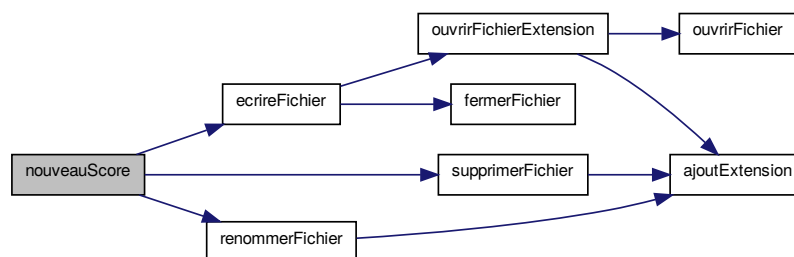
##### Parameters

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

##### Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



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

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

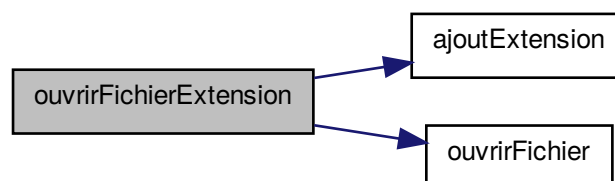
##### Parameters

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

##### Returns

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

Here is the call graph for this function:



#### 4.8.3.5 int renommerFichier ( char \* *nom\_ancien*, char \* *nom\_nouveau* )

Renomme le fichier dont le nom est en parametre

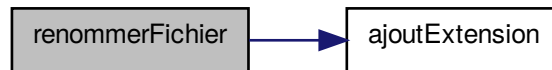
##### Parameters

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

##### Returns

VRAI si tout s'est bien passe, FAUX sinon

Here is the call graph for this function:



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

Supprime le fichier dont le nom est en parametre

##### Parameters

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

##### Returns

VRAI si tout s'est bien passe, FAUX sinon

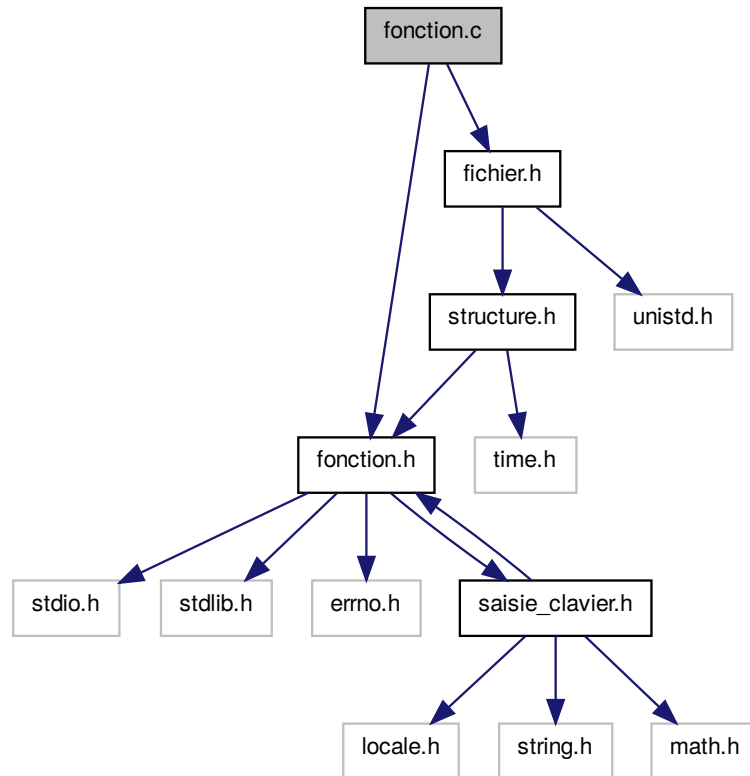
Here is the call graph for this function:



## 4.9 fonction.c File Reference

Fonctions essentielles au fonctionnement du programme.

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



## Functions

- void [mauvais\\_choix](#) ()
- void [systemPause](#) ()
- void [systemEfface](#) ()
- int [compareFlottantCroissant](#) (void const \*a, void const \*b)
- int [compareFlottantDecroissant](#) (void const \*a, void const \*b)
- FILE \* [ouvrirFichier](#) (char nom[], char mode[])
- int [fermerFichier](#) (FILE \*ptr\_fichier)
- int [lireTailleFichier](#) (FILE \*ptr\_fichier)
- void \* [myAlloc](#) (int taille\_alloue)
- void [myRealloc](#) (void \*\*ptr, int taille\_alloue)
- void [ajoutExtension](#) (char \*nom\_fichier)

### 4.9.1 Detailed Description

Fonctions essentielles au fonctionnement du programme.

## Author

Remi BERTHO

## Date

13/02/14

## Version

2.0

## 4.9.2 Function Documentation

### 4.9.2.1 void ajoutExtension ( char \* *nom\_fichier* )

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

## Parameters

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

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

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

## Parameters

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

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

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

## Parameters

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

### 4.9.2.4 int fermerFichier ( FILE \* *ptr\_fichier* )

Ferme le fichier

## Parameters

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

## Returns

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

### 4.9.2.5 int lireTailleFichier ( FILE \* *ptr\_fichier* )

Lis la taille du fichier

## Parameters

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

## Returns

entier ayant la taille du fichier

## 4.9.2.6 void mauvais\_choix ( )

Affiche un message d'erreur.

4.9.2.7 void \* myAlloc ( int *taille\_alloue* )

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

## Parameters

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

## Returns

un pointeur sur la structure alloué

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

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

## Parameters

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

## Returns

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

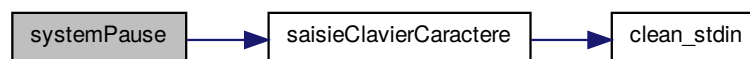
## 4.9.2.10 void systemEfface ( )

Efface la console de l'utilisateur.

## 4.9.2.11 void systemPause ( )

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

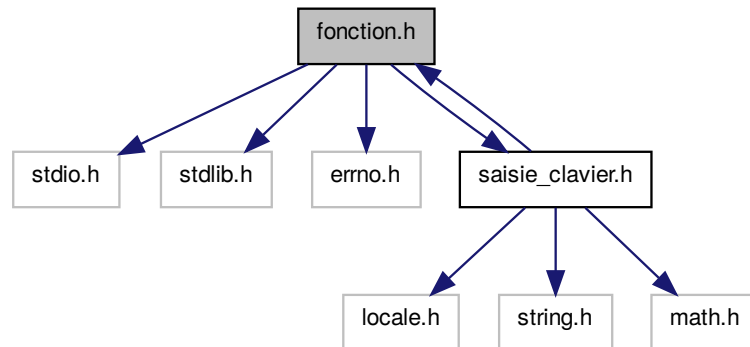
Here is the call graph for this function:



## 4.10 fonction.h File Reference

Prototypes des fonctions essentielles au fonctionnement du programme.

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include "saisie_clavier.h"
Include dependency graph for fonction.h:
```



### Macros

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

### Functions

- void `mauvais_choix` ()
- void `systemPause` ()
- void `systemEfface` ()
- int `compareFlottantDecroissant` (void const \*a, void const \*b)
- int `compareFlottantCroissant` (void const \*a, void const \*b)
- FILE \* `ouvrirFichier` (char nome[], char mode[])
- int `fermerFichier` (FILE \*ptr\_fichier)
- int `lireTailleFichier` (FILE \*ptr\_fichier)
- void \* `myAlloc` (int taille\_alloue)
- void `myRealloc` (void \*\*ptr, int taille\_alloue)
- void `ajoutExtension` (char \*nom\_fichier)

### 4.10.1 Detailed Description

Prototypes des fonctions essentielles au fonctionnement du programme.

#### Author

Remi BERTHO

## Date

13/02/14

## Version

2.0

## 4.10.2 Macro Definition Documentation

### 4.10.2.1 #define FAUX 0

Definit FAUX a 0

### 4.10.2.2 #define VRAI 1

Definit VRAI a 1

## 4.10.3 Function Documentation

### 4.10.3.1 void ajoutExtension ( char \* *nom\_fichier* )

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

## Parameters

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

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

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

## Parameters

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

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

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

## Parameters

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

### 4.10.3.4 int fermerFichier ( FILE \* *ptr\_fichier* )

Ferme le fichier

**Parameters**

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

**Returns**

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

**4.10.3.5 int lireTailleFichier ( FILE \* *ptr\_fichier* )**

Lis la taille du fichier

**Parameters**

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

**Returns**

entier ayant la taille du fichier

**4.10.3.6 void mauvais\_choix ( )**

Affiche un message d'erreur.

**4.10.3.7 void\* myAlloc ( int *taille\_alloue* )**

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

**Parameters**

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

**Returns**

un pointeur sur la structure alloué

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

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

**Parameters**

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

**Returns**

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

**4.10.3.10 void systemEfface ( )**

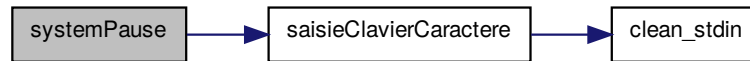
Efface la console de l'utilisateur.



## 4.10.3.11 void systemPause ( )

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

Here is the call graph for this function:

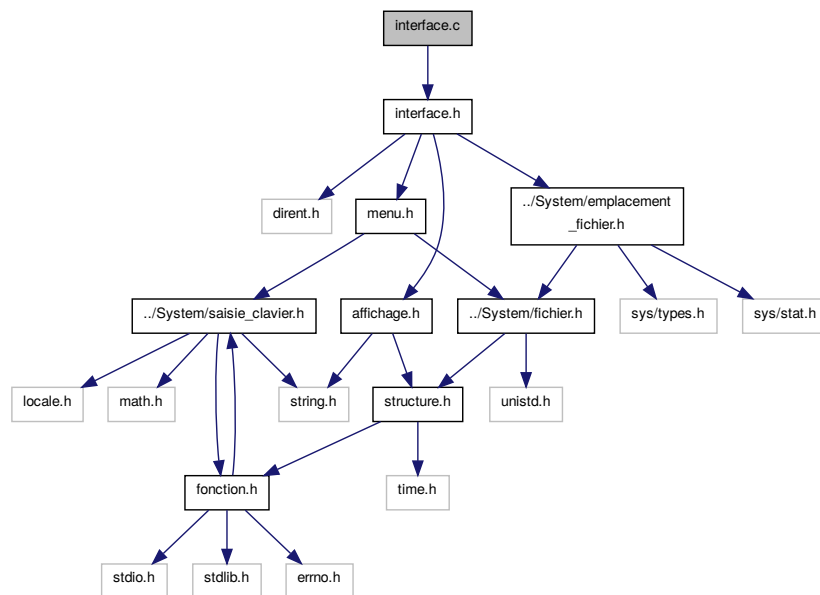


## 4.11 interface.c File Reference

Fonctions qui gerent l'interface graphique du logiciel.

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

Include dependency graph for interface.c:



## Functions

- void [afficheFichier](#) ()
- void [supprimerFichierNom](#) ()
- void [listerFichier](#) ()
- void [jouer](#) (Fichier\_Jeu \*ptr\_struct\_fichier, char \*nom\_fichier)
- void [nouvellePartie](#) ()
- void [chargerPartie](#) ()
- void [menuPrincipal](#) ()
- void [nouveauCheminFichier](#) ()

- void `lireCheminFichier` ()
- void `chargerPartieLocale` (char \*nom\_fichier)
- void `afficheFichierLocale` (char \*nom\_fichier)

#### 4.11.1 Detailed Description

Fonctions qui gerent l'interface graphique du logiciel.

##### Author

Remi BERTHO

##### Date

09/03/14

##### Version

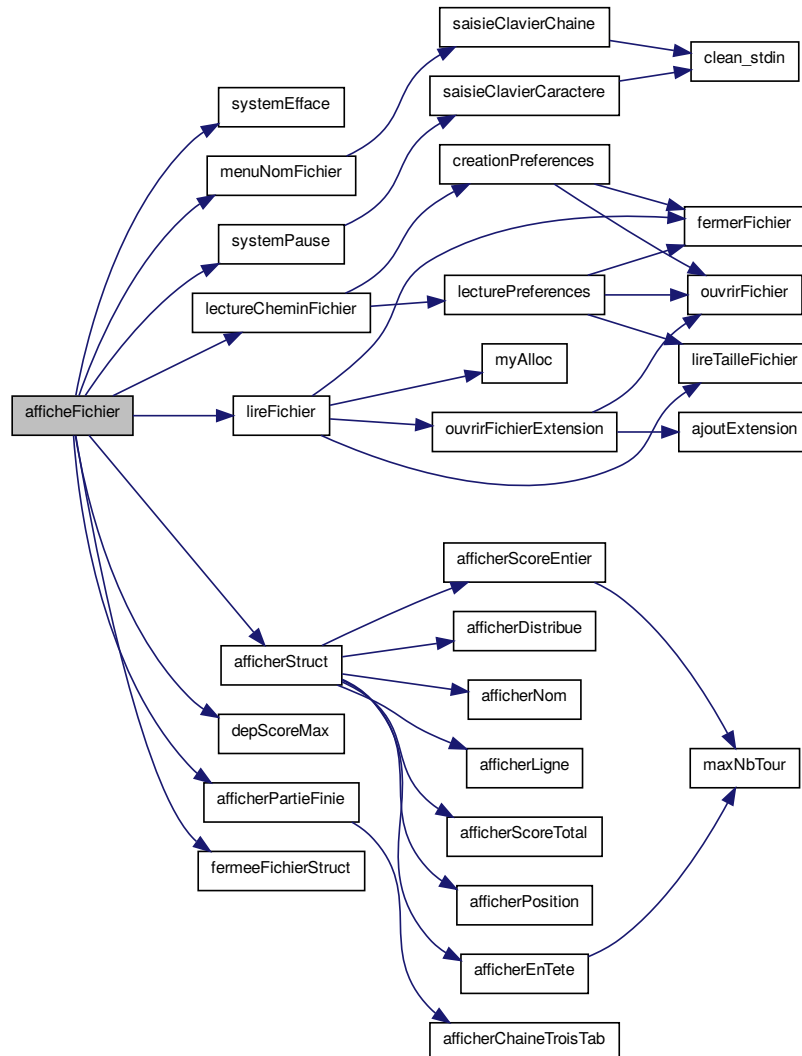
2.1.0

#### 4.11.2 Function Documentation

##### 4.11.2.1 void `afficheFichier` ( )

Demande le nom d'un fichier et l'affiche

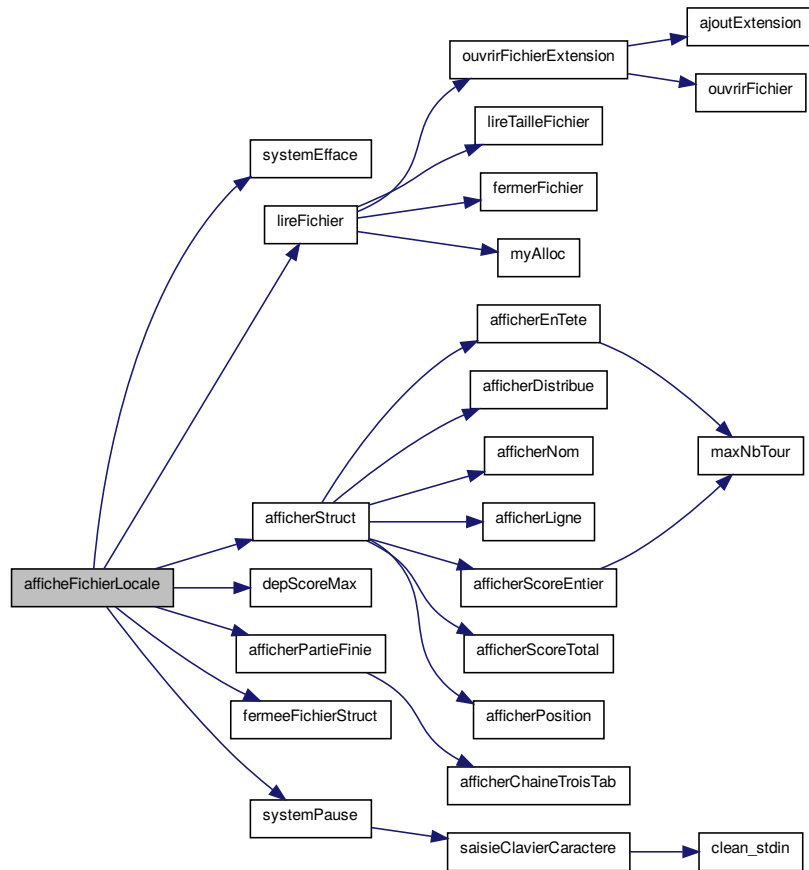
Here is the call graph for this function:



#### 4.11.2.2 void afficheFichierLocale ( char \* nom\_fichier )

Affiche le fichier dont le nom a été donné en paramètre

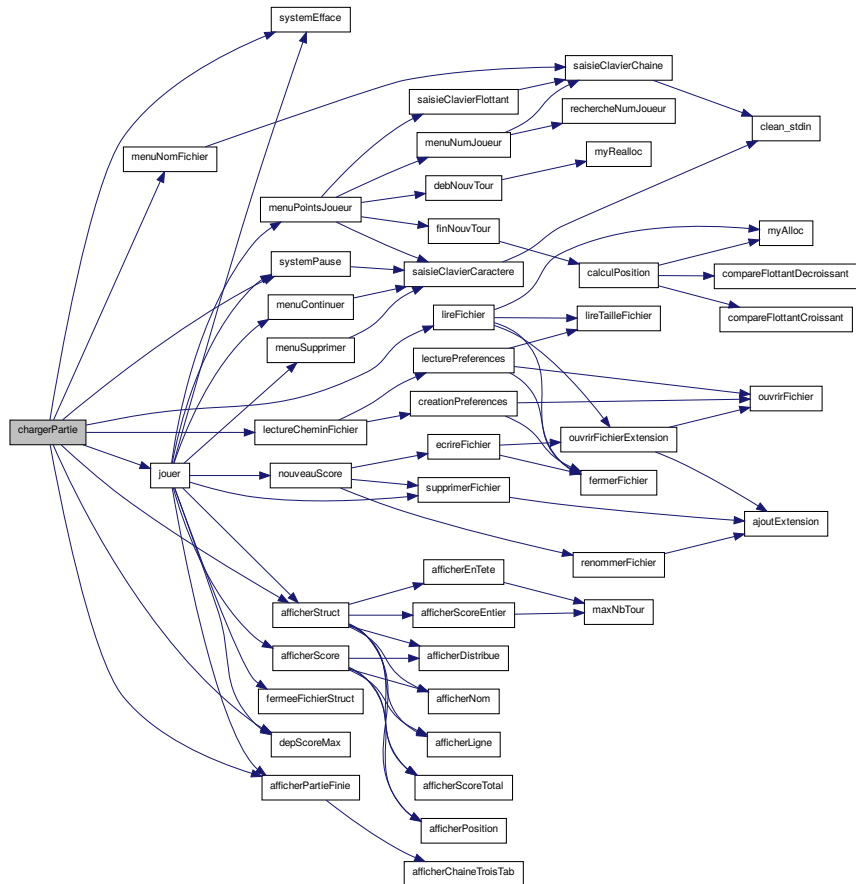
Here is the call graph for this function:



#### 4.11.2.3 void chargerPartie ( )

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

Here is the call graph for this function:



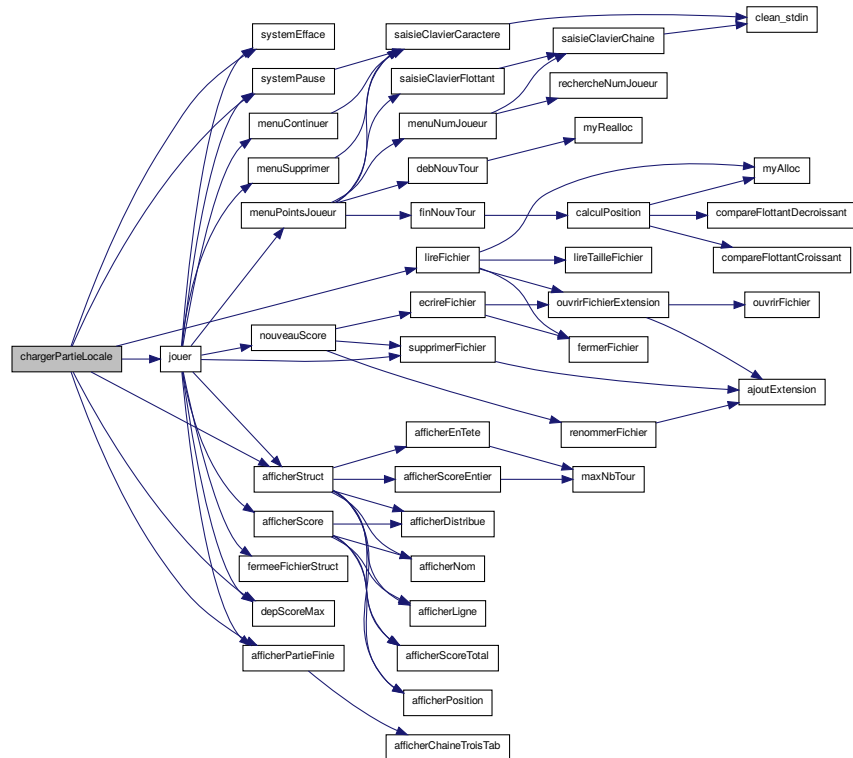
#### 4.11.2.4 void chargerPartieLocale ( char \* nom\_fichier )

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

##### Parameters

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

Here is the call graph for this function:



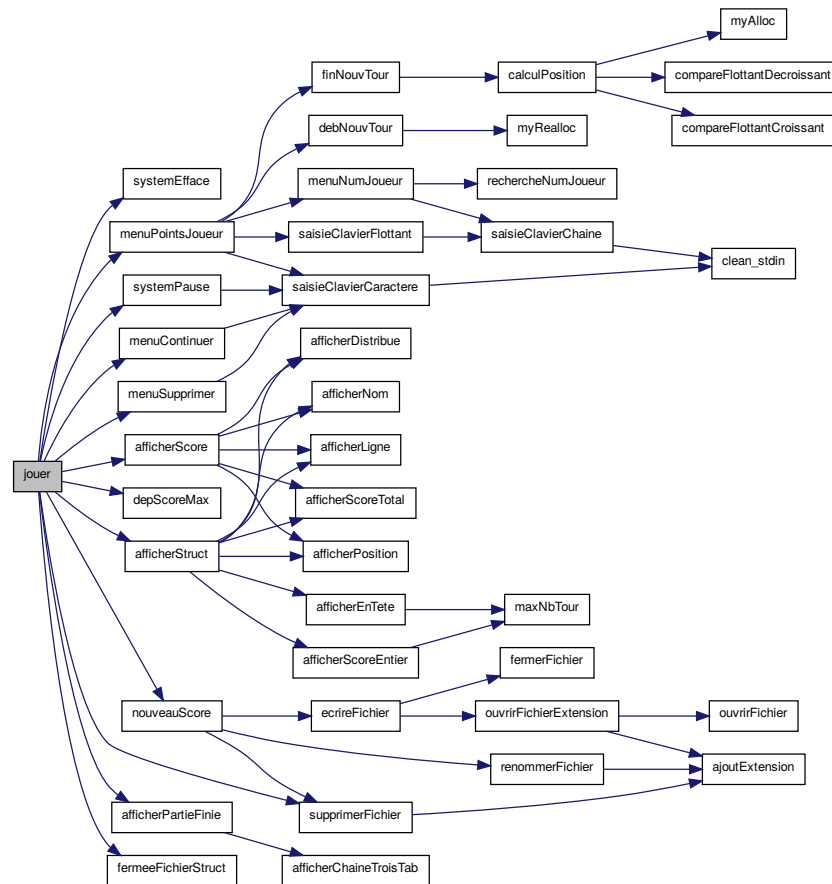
#### 4.11.2.5 void jouer ( Fichier\_Jeu \* ptr\_struct\_fichier, char \* nom\_fichier )

Lance la partie de comptage jusqu'a la fin

##### Parameters

in, out	*ptr_struct_fichier	la structure <a href="#">Fichier_Jeu</a>
in	*nom_fichier	le nom du fichier dans lequel on stocke les informations

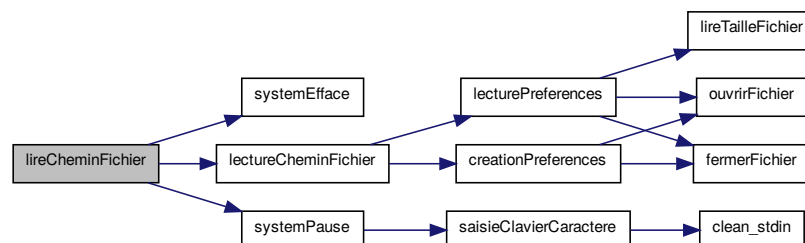
Here is the call graph for this function:



#### 4.11.2.6 void lireCheminFichier ( )

Lis le chemin de fichier et l'affiche

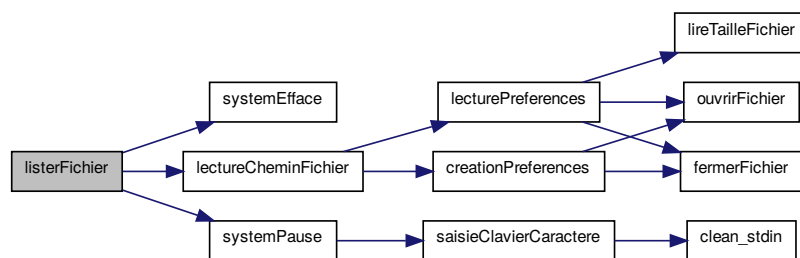
Here is the call graph for this function:



#### 4.11.2.7 void listerFichier ( )

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

Here is the call graph for this function:

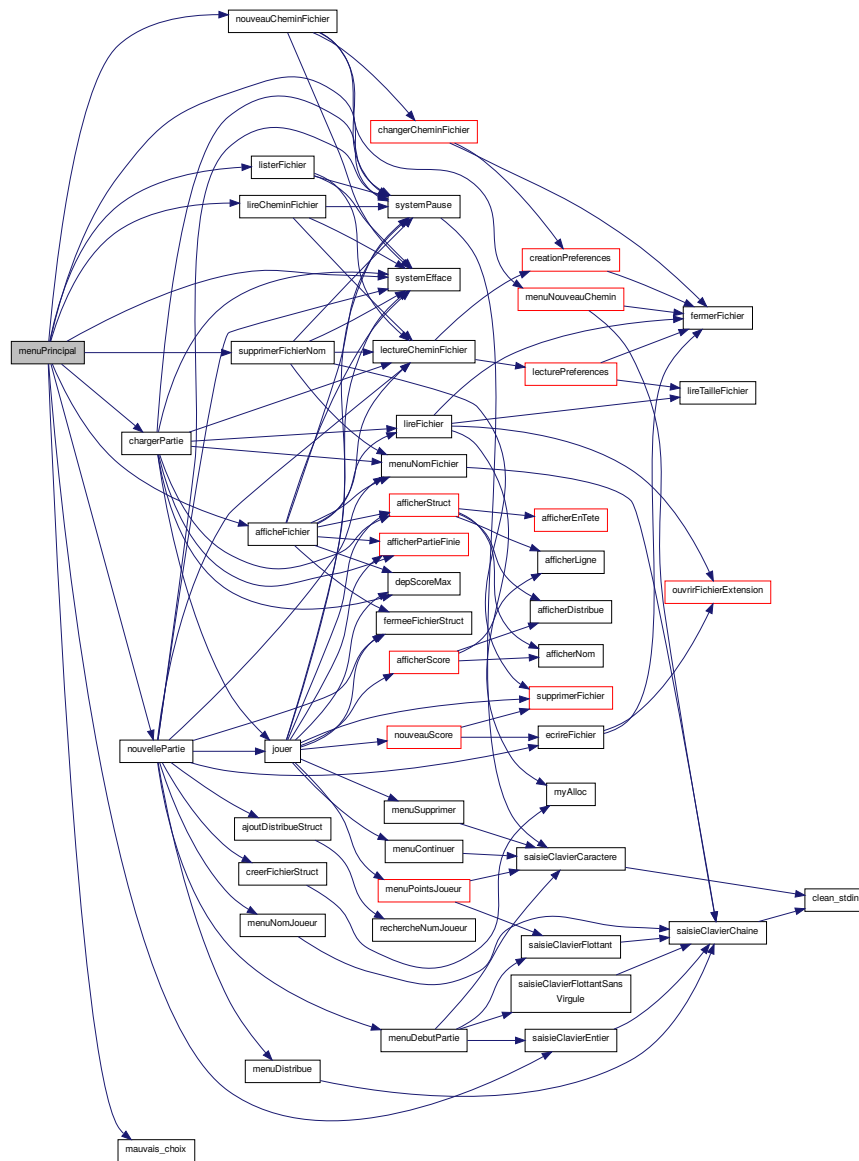


#### 4.11.2.8 void menuPrincipal ( )

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



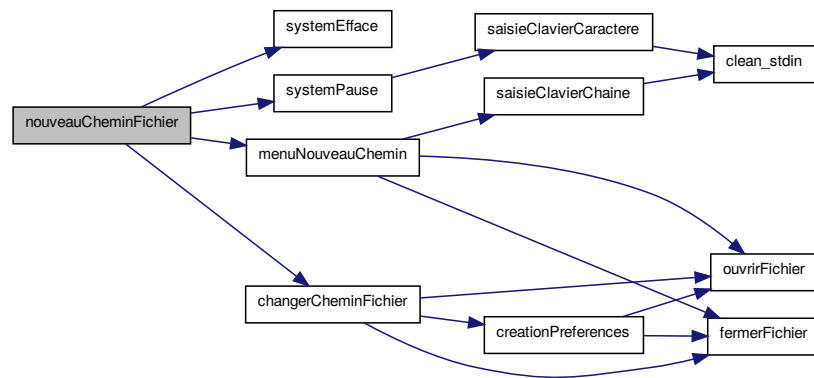
Here is the call graph for this function:



#### 4.11.2.9 void nouveauCheminFichier ( )

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

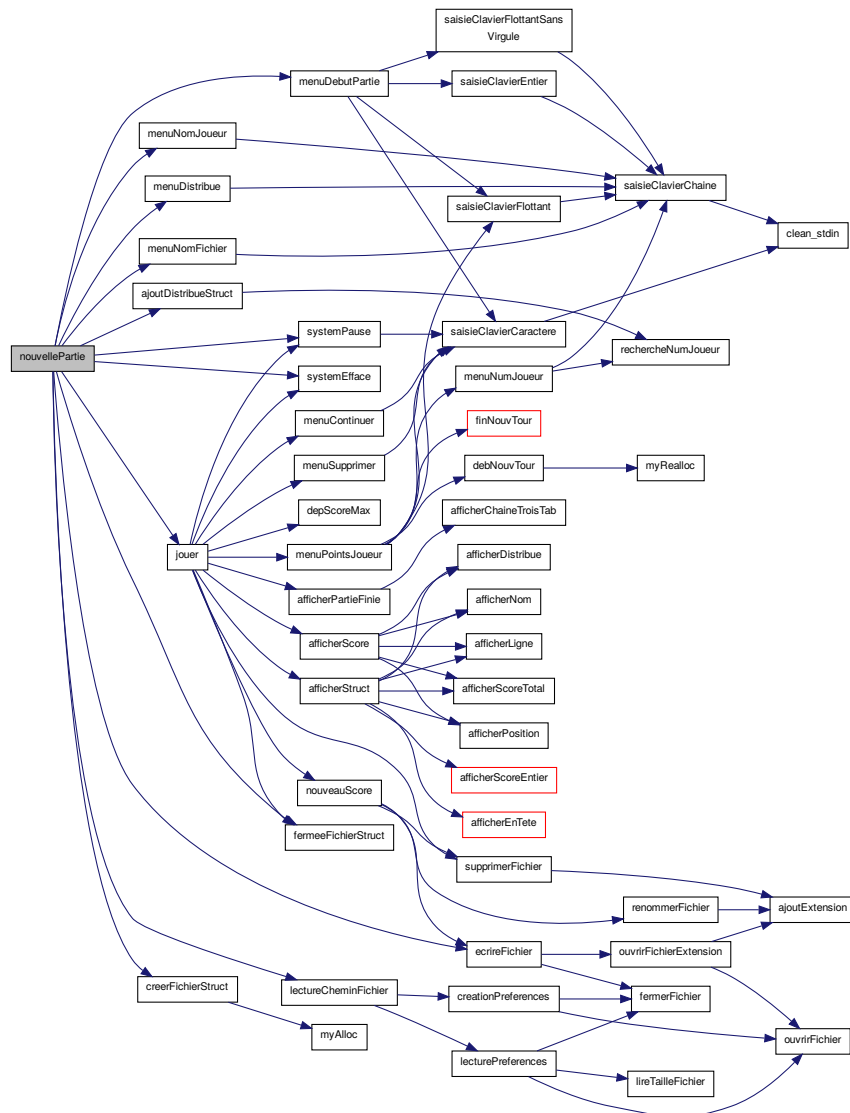
Here is the call graph for this function:



#### 4.11.2.10 void nouvellePartie ( )

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

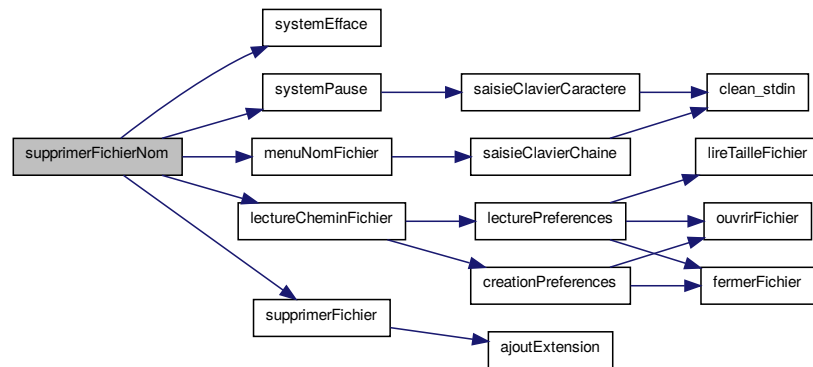
Here is the call graph for this function:



#### 4.11.2.11 void supprimerFichierNom ( )

Demande le nom d'un fichier et le supprime

Here is the call graph for this function:

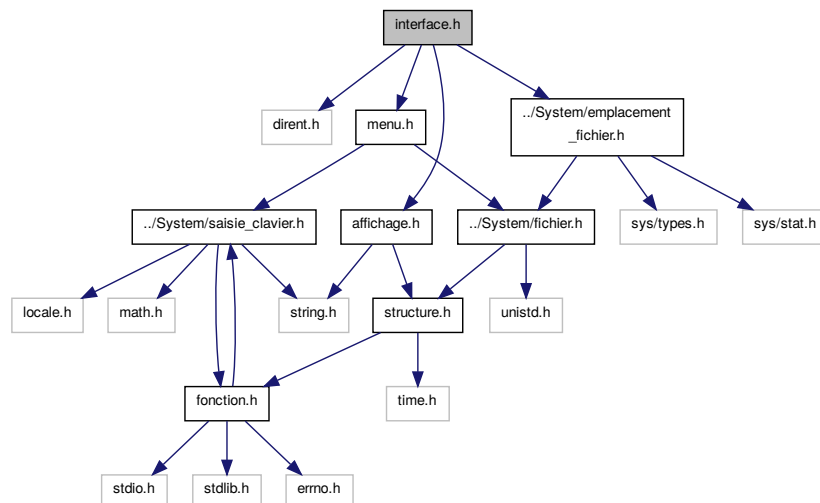


## 4.12 interface.h File Reference

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

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

Include dependency graph for interface.h:



## Enumerations

- enum `Menu` {  
`nouvPart` =1, `charPart` =2, `affFich` =3, `supprFich` =4,  
`listFich` =5, `quit` =6, `easterEggs` = 42 }

## Functions

- void [afficheFichier](#) ()
- void [supprimerFichierNom](#) ()
- void [listerFichier](#) ()
- void [jouer](#) ([Fichier\\_Jeu](#) \*ptr\_struct\_fichier, char \*nom\_fichier)
- void [nouvellePartie](#) ()
- void [chargerPartie](#) ()
- void [menuPrincipal](#) ()
- void [nouveauCheminFichier](#) ()
- void [lireCheminFichier](#) ()
- void [chargerPartieLocale](#) (char \*nom\_fichier)
- void [afficheFichierLocale](#) (char \*nom\_fichier)

### 4.12.1 Detailed Description

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

#### Author

Remi BERTHO

#### Date

09/03/14

#### Version

2.1.0

### 4.12.2 Enumeration Type Documentation

#### 4.12.2.1 enum Menu

##### Enumerator

***nouvPart***

***charPart***

***affFich***

***supprFich***

***listFich***

***quit***

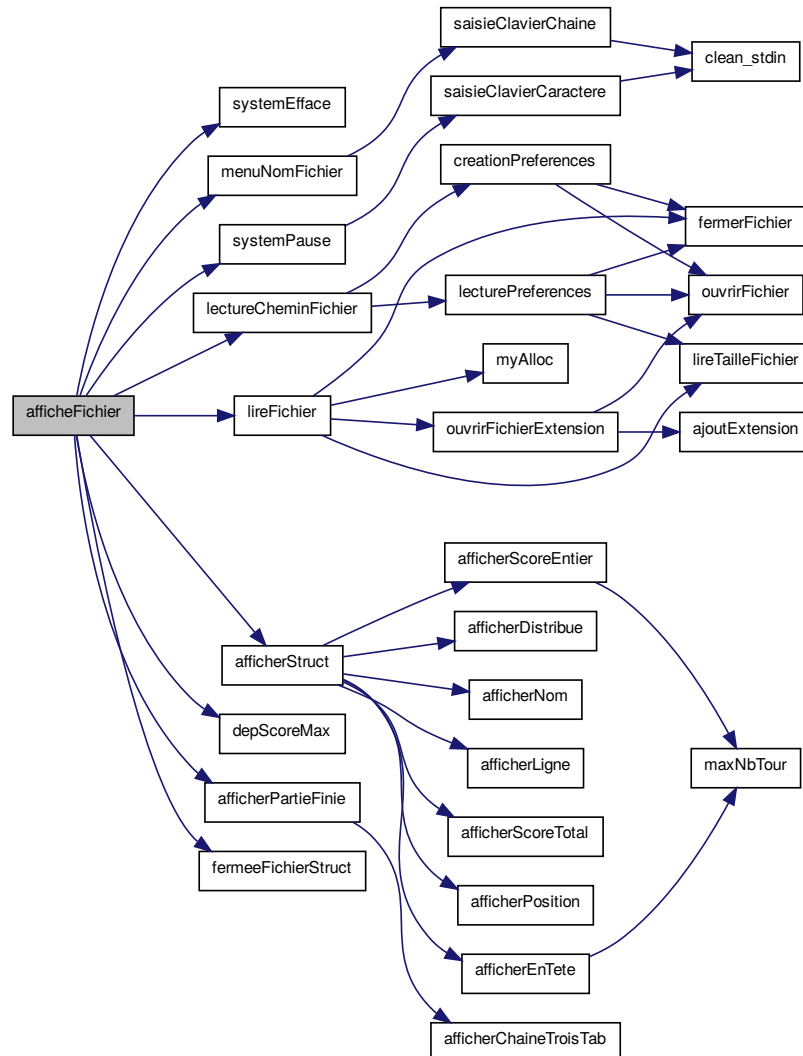
***easterEggs***

### 4.12.3 Function Documentation

#### 4.12.3.1 void afficheFichier ( )

Demande le nom d'un fichier et l'affiche

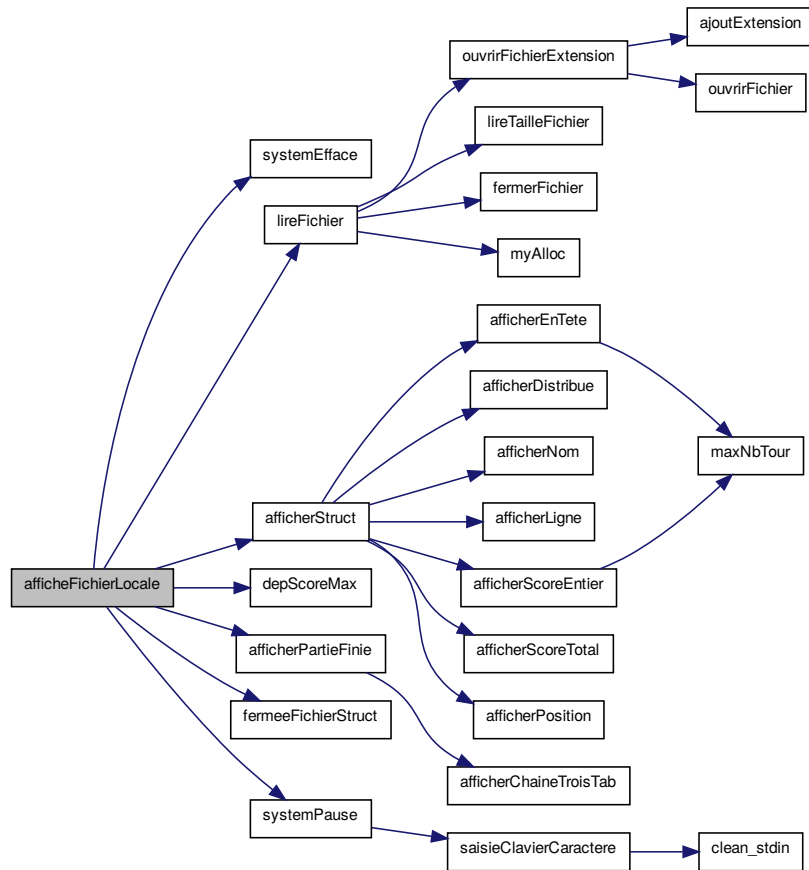
Here is the call graph for this function:



#### 4.12.3.2 void afficheFichierLocale ( char \* nom\_fichier )

Affiche le fichier dont le nom a été donné en paramètre

Here is the call graph for this function:



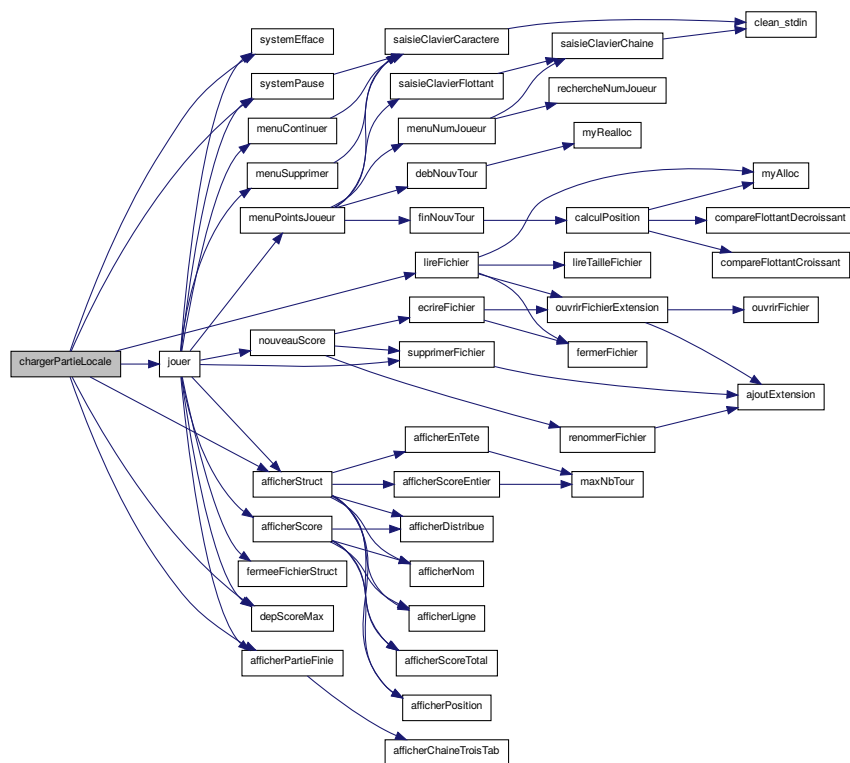
#### 4.12.3.3 void chargerPartie ( )

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





Here is the call graph for this function:



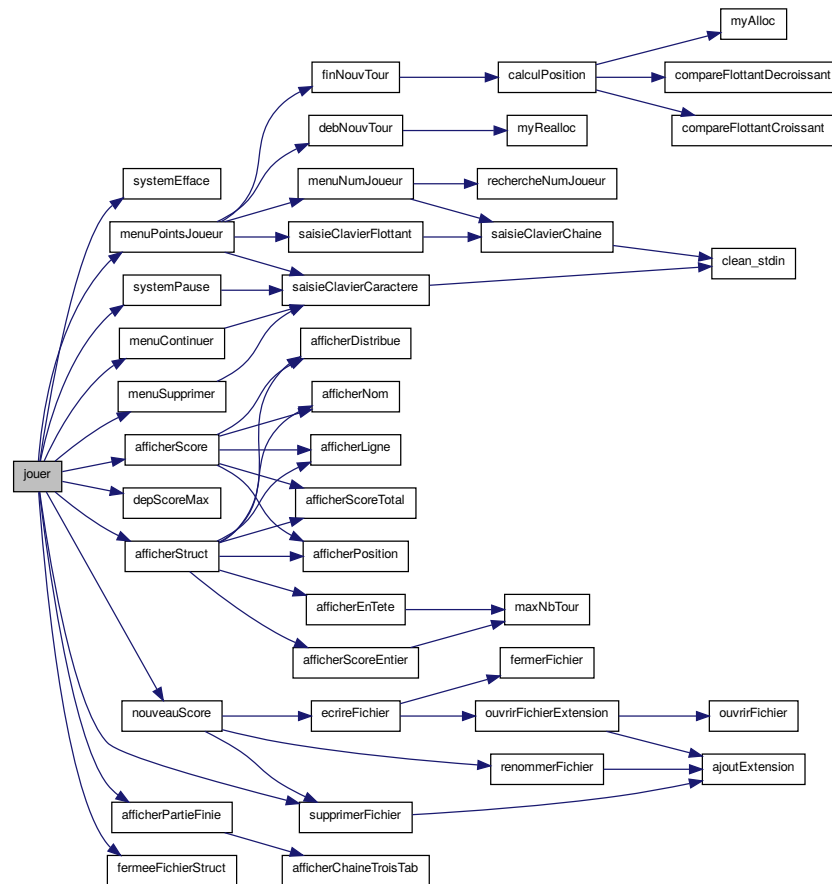
#### 4.12.3.5 void jouer ( Fichier\_Jeu \* ptr\_struct\_fichier, char \* nom\_fichier )

Lance la partie de comptage jusqu'a la fin

##### Parameters

in, out	*ptr_struct_fichier	la structure <a href="#">Fichier_Jeu</a>
in	*nom_fichier	le nom du fichier dans lequel on stocke les informations

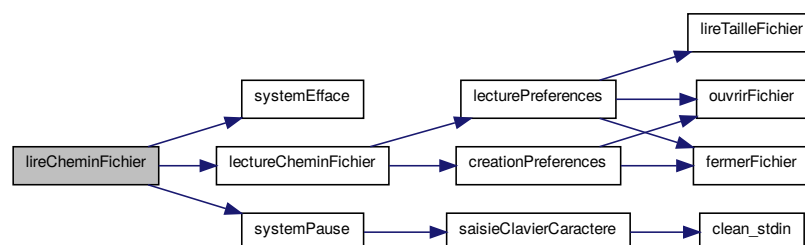
Here is the call graph for this function:



#### 4.12.3.6 void lireCheminFichier ( )

Lis le chemin de fichier et l'affiche

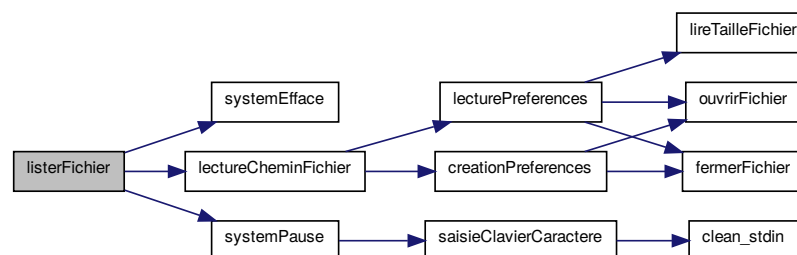
Here is the call graph for this function:



## 4.12.3.7 void listerFichier ( )

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

Here is the call graph for this function:

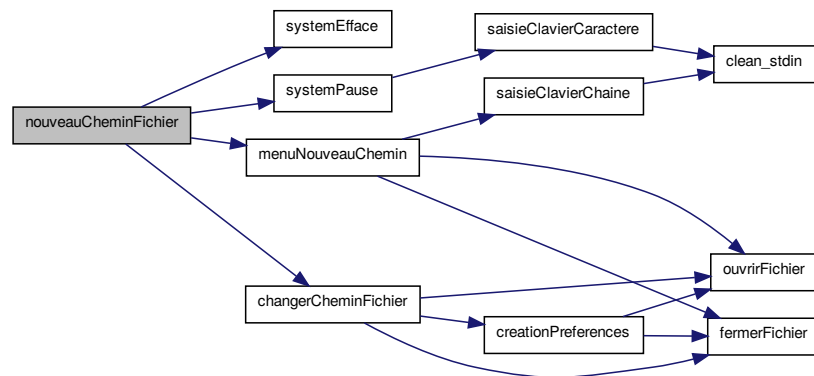


## 4.12.3.8 void menuPrincipal ( )

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



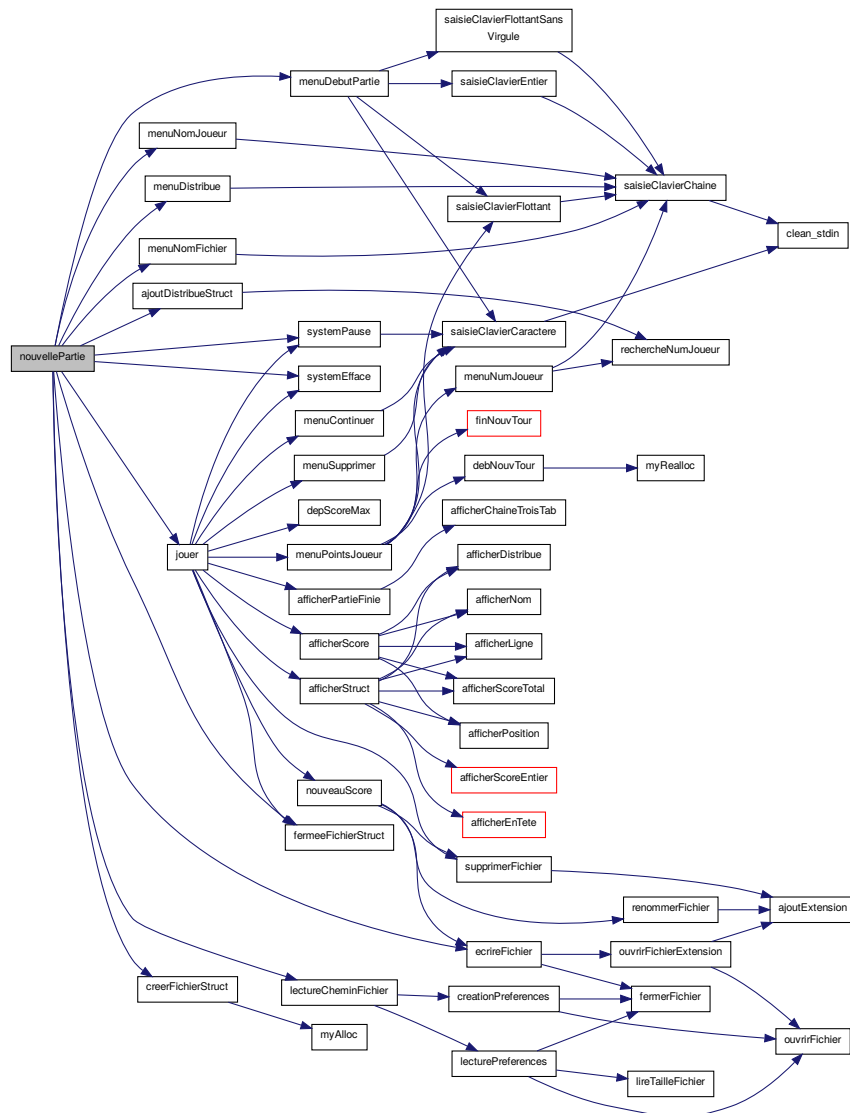
Here is the call graph for this function:



#### 4.12.3.10 void nouvellePartie ( )

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

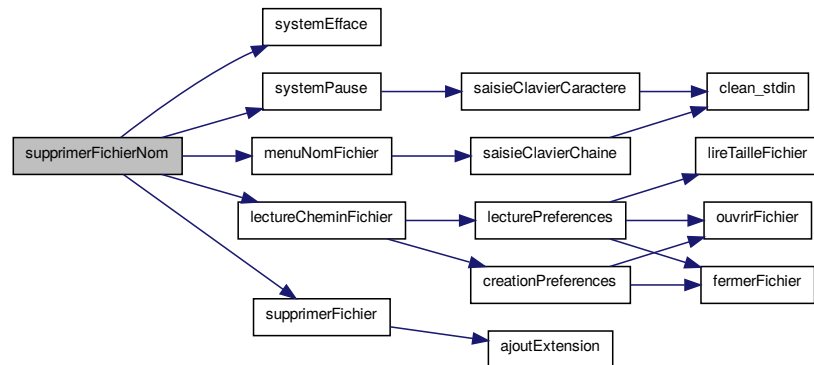
Here is the call graph for this function:



#### 4.12.3.11 void supprimerFichierNom ( )

Demande le nom d'un fichier et le supprime

Here is the call graph for this function:

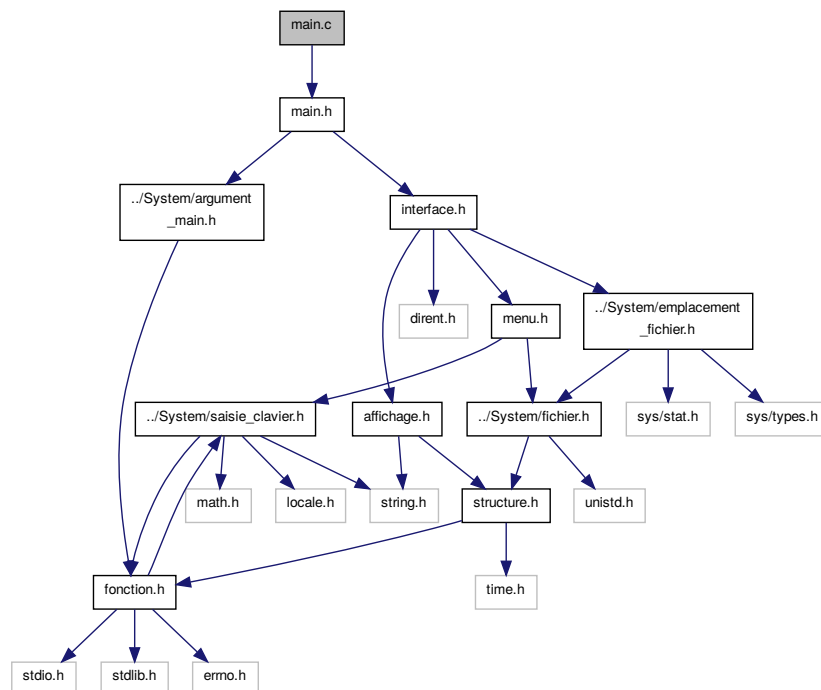


## 4.13 main.c File Reference

Lancement du programme.

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

Include dependency graph for main.c:



## Functions

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

### 4.13.1 Detailed Description

Lancement du programme.

#### Author

Remi BERTHO

#### Date

09/03/14

#### Version

2.1.0

### 4.13.2 Function Documentation

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

Lance le programme

#### Parameters

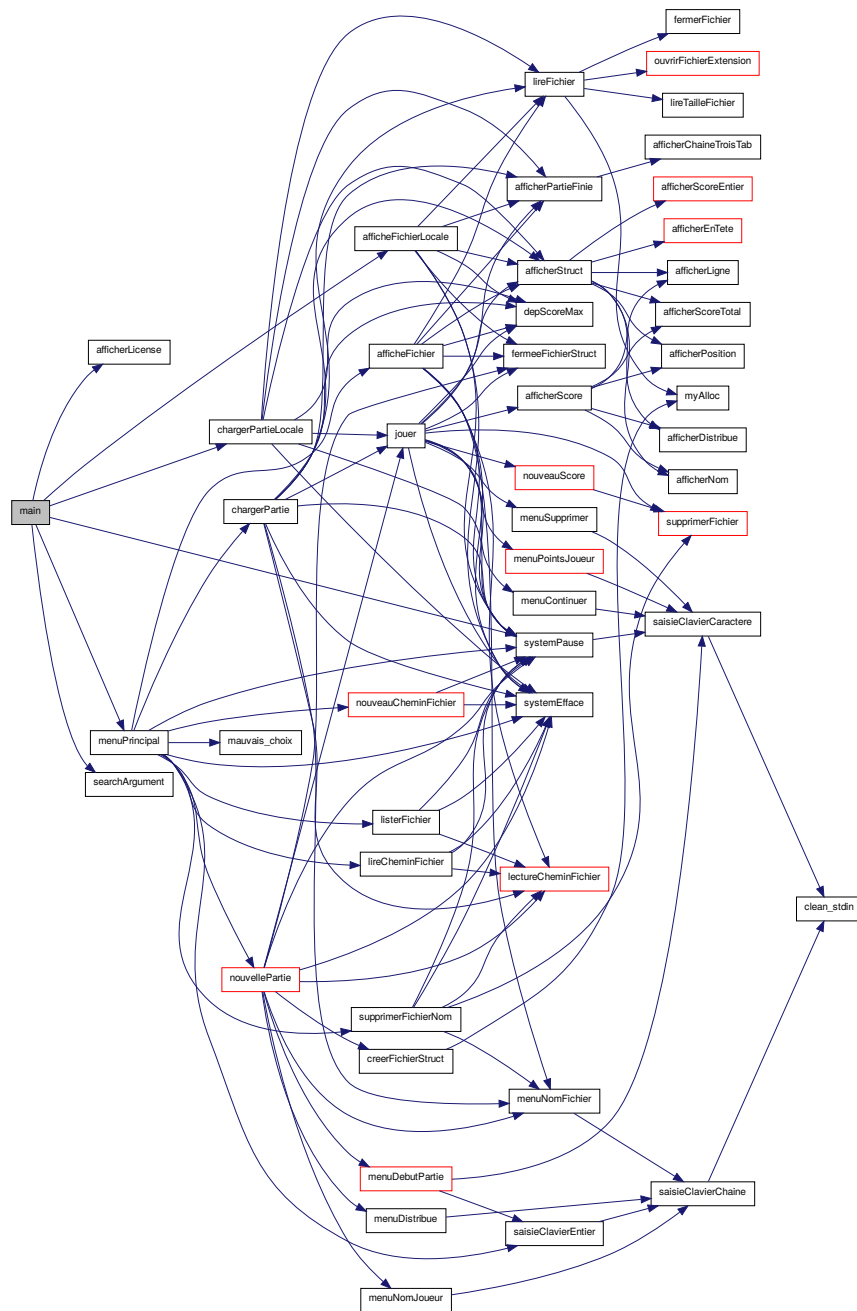
<code>in</code>	<code><i>argc</i></code>	le nombre d 'argument
<code>in</code>	<code><i>argv</i></code>	le tableau des arguments

#### Returns

0 si tout s'est bien passe



Here is the call graph for this function:

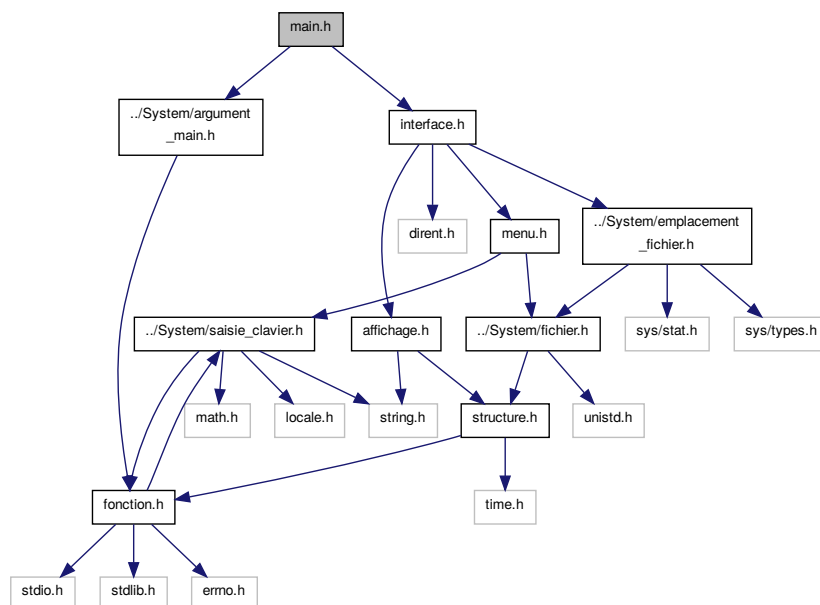


## 4.14 main.h File Reference

Lancement du programme.

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

Include dependency graph for main.h:



## Functions

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

### 4.14.1 Detailed Description

Lancement du programme.

Author

Remi BERTHO

Date

09/03/14

Version

2.1.0

### 4.14.2 Function Documentation

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

Lance le programme

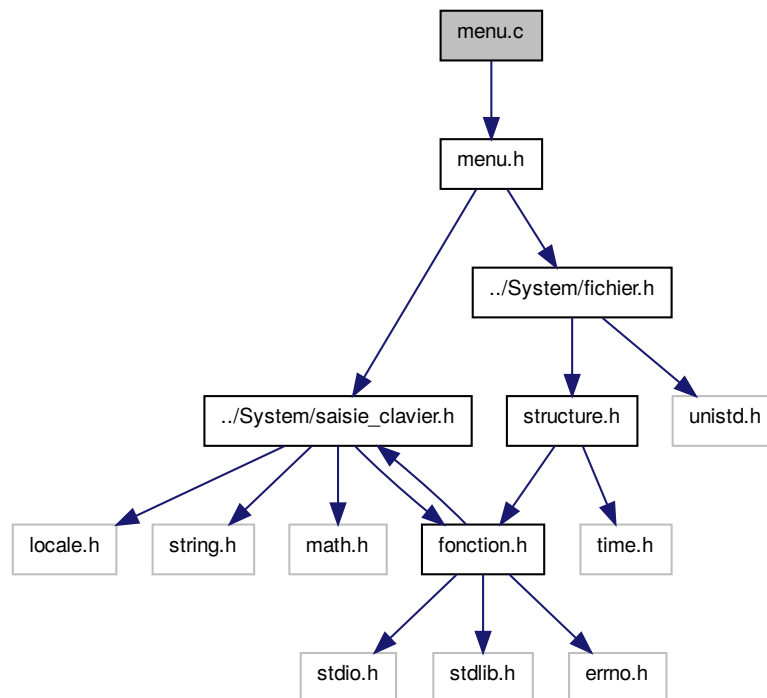


## 4.15 menu.c File Reference

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

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

Include dependency graph for menu.c:



### Functions

- char \* [menuNomFichier](#) (char nom\_fichier[TAILLE\_MAX\_NOM\_FICHIER])
- void [menuDebutPartie](#) (float \*ptr\_nb\_joueur, float \*ptr\_nb\_max, char \*ptr\_sens\_premier, char \*ptr\_tour\_par\_tour, char \*ptr\_use\_distributor, char \*ptr\_number\_after\_comma)
- void [menuDistribue](#) (char \*nom\_distribue)
- void [menuNomJoueur](#) (Fichier\_Jeu \*ptr\_struct\_fichier)
- void [menuPointsJoueur](#) (Fichier\_Jeu \*ptr\_struct\_fichier)
- int [menuNumJoueur](#) (Fichier\_Jeu \*ptr\_struct\_fichier)
- int [menuContinuer](#) ()
- int [menuSupprimer](#) ()
- void [menuNouveauChemin](#) (char \*nouveauChemin)

### 4.15.1 Detailed Description

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

Author

Remi BERTHO

## Date

09/03/14

## Version

2.1.0

## 4.15.2 Function Documentation

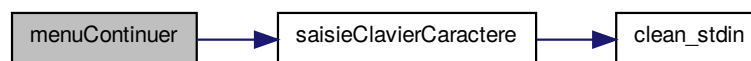
4.15.2.1 `int menuContinuer ( )`

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

## Returns

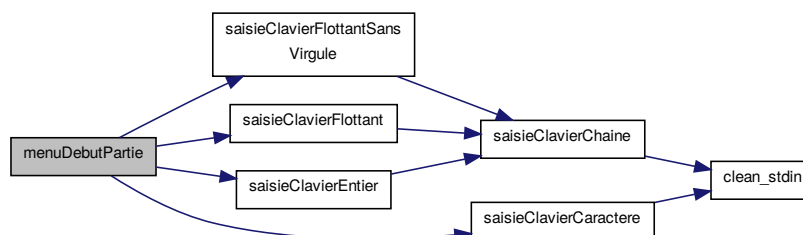
VRAI si l'on veut continuer, FAUX sinon

Here is the call graph for this function:



4.15.2.2 `void menuDebutPartie ( float * ptr_nb_joueur, float * ptr_nb_max, char * ptr_sens_premier, char * ptr_tour_par_tour, char * ptr_use_distributor, char * ptr_number_after_comma )`

Here is the call graph for this function:

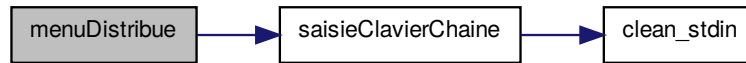
4.15.2.3 `void menuDistribue ( char * nom_distribue )`

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

## Parameters

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

Here is the call graph for this function:



#### 4.15.2.4 char \* menuNomFichier ( char nom\_fichier[TAILLE\_MAX\_NOM\_FICHIER] )

Demande et enregistre le nom du fichier

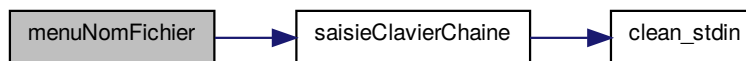
## Parameters

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

## Returns

la chaine de caractere contenant le nom du fichier

Here is the call graph for this function:



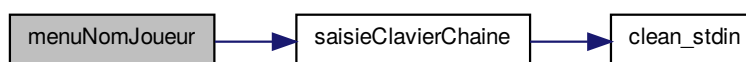
#### 4.15.2.5 void menuNomJoueur ( Fichier\_Jeu \* ptr\_struct\_fichier )

Demande et enregistre le nom des joueurs

## Parameters

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

Here is the call graph for this function:



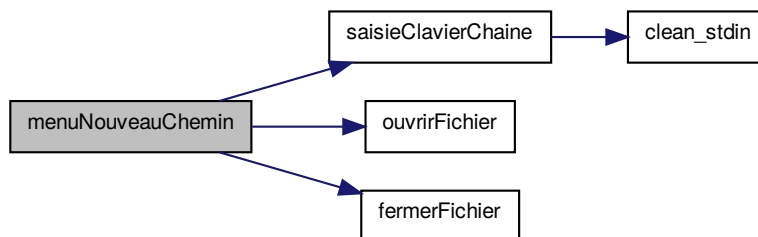
#### 4.15.2.6 void menuNouveauChemin ( char \* *nouveauChemin* )

Demande et enregistre le nouveau chemin

##### Parameters

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

Here is the call graph for this function:



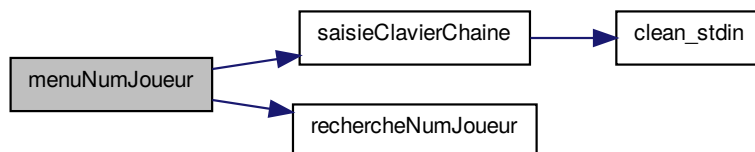
#### 4.15.2.7 int menuNumJoueur ( Fichier\_Jeu \* *ptr\_struct\_fichier* )

Demande et enregistre le numero du joueur

##### Parameters

in, out	* <i>ptr_struct_fichier</i>	la structure <a href="#">Fichier_Jeu</a>
---------	-----------------------------	--

Here is the call graph for this function:



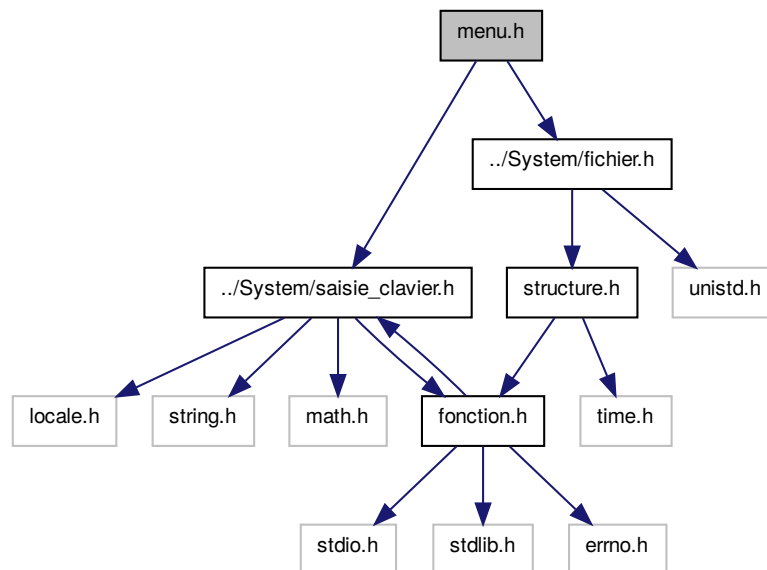
#### 4.15.2.8 void menuPointsJoueur ( Fichier\_Jeu \* *ptr\_struct\_fichier* )

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





Include dependency graph for menu.h:



## Functions

- char \* [menuNomFichier](#) (char nom\_fichier[[TAILLE\\_MAX\\_NOM\\_FICHIER](#)])
- void [menuDebutPartie](#) (float \*ptr\_nb\_joueur, float \*ptr\_nb\_max, char \*ptr\_sens\_premier, char \*ptr\_tour\_par\_tour, char \*ptr\_use\_distributor, char \*ptr\_number\_after\_comma)
- void [menuDistribue](#) (char \*nom\_distribue)
- void [menuNomJoueur](#) ([Fichier\\_Jeu](#) \*ptr\_struct\_fichier)
- void [menuPointsJoueur](#) ([Fichier\\_Jeu](#) \*ptr\_struct\_fichier)
- int [menuNumJoueur](#) ([Fichier\\_Jeu](#) \*ptr\_struct\_fichier)
- int [menuContinuer](#) ()
- int [menuSupprimer](#) ()
- void [menuNouveauChemin](#) (char \*nouveauChemin)

### 4.16.1 Detailed Description

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

#### Author

Remi BERTHO

#### Date

13/02/14

#### Version

2.0

## 4.16.2 Function Documentation

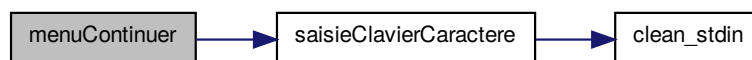
### 4.16.2.1 int menuContinuer ( )

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

#### Returns

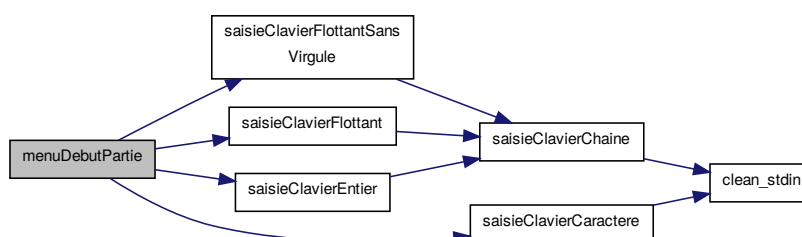
VRAI si l'on veut continuer, FAUX sinon

Here is the call graph for this function:



### 4.16.2.2 void menuDebutPartie ( float \* *ptr\_nb\_joueur*, float \* *ptr\_nb\_max*, char \* *ptr\_sens\_premier*, char \* *ptr\_tour\_par\_tour*, char \* *ptr\_use\_distributor*, char \* *ptr\_number\_after\_comma* )

Here is the call graph for this function:



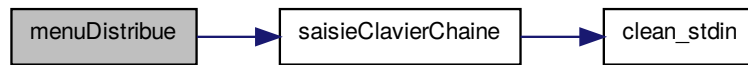
### 4.16.2.3 void menuDistribue ( char \* *nom\_distribue* )

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

#### Parameters

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

Here is the call graph for this function:



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

Demande et enregistre le nom du fichier

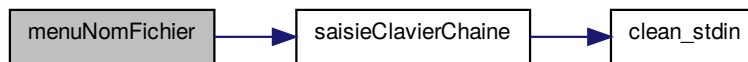
##### Parameters

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

##### Returns

la chaine de caractere contenant le nom du fichier

Here is the call graph for this function:



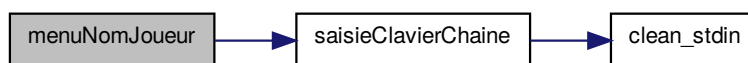
#### 4.16.2.5 `void menuNomJoueur ( Fichier_Jeu * ptr_struct_fichier )`

Demande et enregistre le nom des joueurs

##### Parameters

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

Here is the call graph for this function:



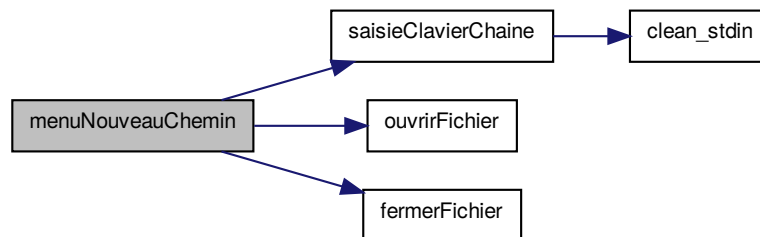
4.16.2.6 void menuNouveauChemin ( char \* *nouveauChemin* )

Demande et enregistre le nouveau chemin

## Parameters

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

Here is the call graph for this function:



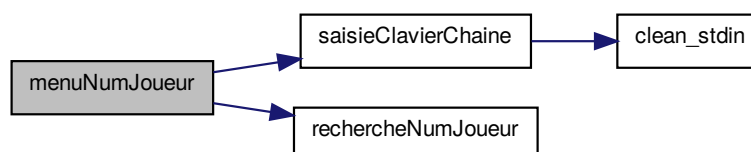
#### 4.16.2.7 int menuNumJoueur ( Fichier\_Jeu \* ptr\_struct\_fichier )

Demande et enregistre le numero du joueur

## Parameters

in, out	<i>*ptr_struct_fichier</i>	la structure <a href="#">Fichier_Jeu</a>
---------	----------------------------	--

Here is the call graph for this function:



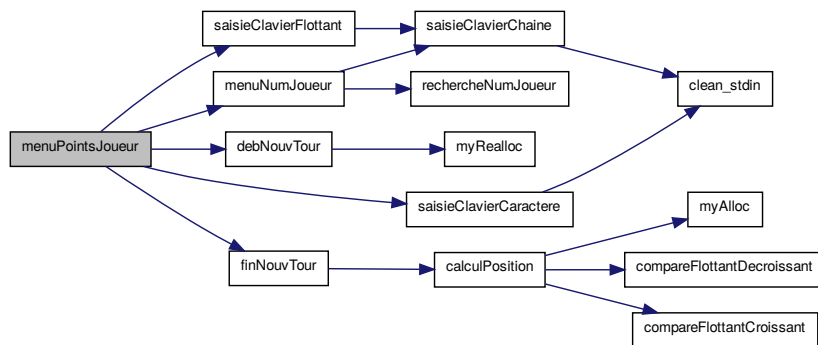
#### 4.16.2.8 void menuPointsJoueur ( Fichier\_Jeu \* ptr\_struct\_fichier )

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

## Parameters

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

Here is the call graph for this function:



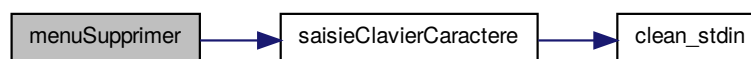
#### 4.16.2.9 int menuSupprimer ( )

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

##### Returns

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

Here is the call graph for this function:

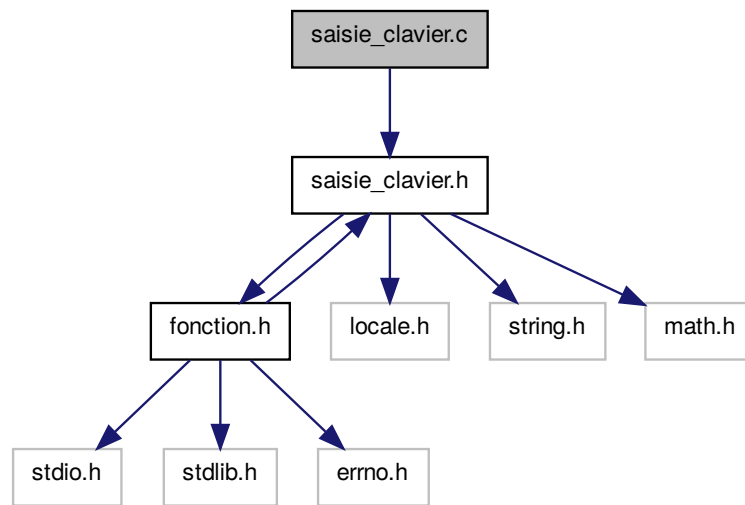


## 4.17 saisie\_clavier.c File Reference

Fonctions de saisie clavier.

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

Include dependency graph for saisie\_clavier.c:



## Functions

- void `clean_stdin` (void)
- char \* `saisieClavierChaine` (char \*chaine, int nb\_caract\_plus\_un)
- void `saisieClavierEntier` (int \*nb)
- void `saisieClavierFlottant` (float \*nb)
- void `saisieClavierFlottantSansVirgule` (float \*nb)
- void `saisieClavierDouble` (double \*nb)
- char \* `saisieClavierCaractere` (char \*c)

### 4.17.1 Detailed Description

Fonctions de saisie clavier.

Author

Remi BERTHO

Date

13/02/14

Version

2.0

### 4.17.2 Function Documentation

#### 4.17.2.1 void `clean_stdin` ( void )

Vide la cache de stdin.

#### 4.17.2.2 `char * saisieClavierCaractere ( char * c )`

Fait une saisie clavier d'un caractere.

##### Parameters

<code>in, out</code>	<code>*c</code>	le caractere que l'on veut saisir
----------------------	-----------------	-----------------------------------

Here is the call graph for this function:



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

Fait une saisie clavier de chaine de caractere au clavier dans chaine sur nb-caract\_plus\_un moins 1 caractere

##### Parameters

<code>in, out</code>	<code>*chaine</code>	une chaine de caractere
<code>in</code>	<code>nb_caract_plus-un</code>	le nombre de caractere moins un que la fonction va lire

Here is the call graph for this function:



#### 4.17.2.4 `void saisieClavierDouble ( double * nb )`

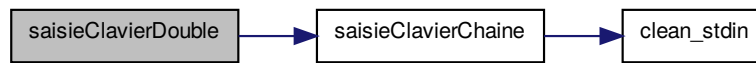
Fait une saisie clavier d'un double, met 0 si l'entree n'est pas un chiffre.

##### Parameters

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



Here is the call graph for this function:



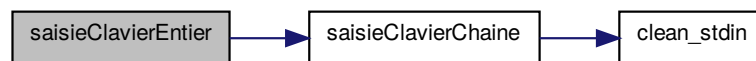
#### 4.17.2.5 void \* saisieClavierEntier ( int \* nb )

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

##### Parameters

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

Here is the call graph for this function:



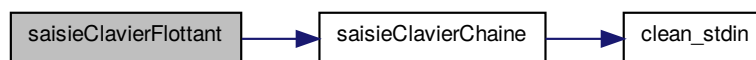
#### 4.17.2.6 void saisieClavierFlottant ( float \* nb )

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

##### Parameters

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

Here is the call graph for this function:



#### 4.17.2.7 void saisieClavierFlottantSansVirgule ( float \* nb )

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

## Parameters

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

Here is the call graph for this function:

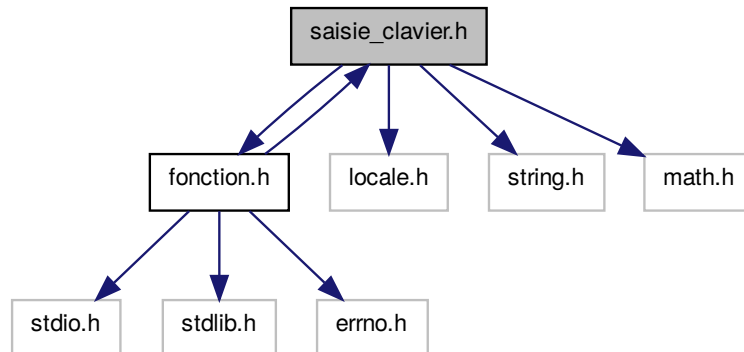


## 4.18 saisie\_clavier.h File Reference

Prototypes des fonctions de saisie clavier.

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

Include dependency graph for saisie\_clavier.h:



### Macros

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

### Functions

- void `clean_stdin` (void)
- char \* `saisieClavierChaine` (char \*chaine, int nb\_caract\_plus\_un)
- void `saisieClavierEntier` (int \*nb)

- void [saisieClavierFlottant](#) (float \*nb)
- void [saisieClavierFlottantSansVirgule](#) (float \*nb)
- void [saisieClavierDouble](#) (double \*nb)
- char \* [saisieClavierCaractere](#) (char \*c)

### 4.18.1 Detailed Description

Prototypes des fonctions de saisie clavier.

#### Author

Remi BERTHO

#### Date

13/02/14

#### Version

2.0

### 4.18.2 Macro Definition Documentation

#### 4.18.2.1 `#define NB_CHARACTER DOUB 309`

Definit NB\_CHARACTER\_DOUB a 309

#### 4.18.2.2 `#define NB_CHARACTER_FLOT 39`

Definit NB\_CHARACTER\_FLOT a 39

#### 4.18.2.3 `#define NB_CHARACTER_INT 12`

Definit NB\_CHARACTER\_INT a 12

### 4.18.3 Function Documentation

#### 4.18.3.1 `void clean_stdin ( void )`

Vide la cache de stdin.

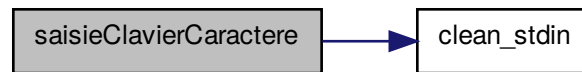
#### 4.18.3.2 `char* saisieClavierCaractere ( char * c )`

Fait une saisie clavier d'un caractere.

#### Parameters

<code>in, out</code>	<code>*c</code>	le caractere que l'on veut saisir
----------------------	-----------------	-----------------------------------

Here is the call graph for this function:



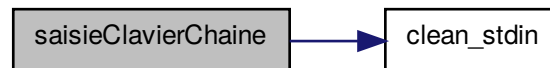
#### 4.18.3.3 `char* saisieClavierChaine ( char * chaine, int nb_caract_plus_un )`

Fait une saisie clavier de chaine de caractere au clavier dans chaine sur nb-caract\_plus\_un moins 1 caractere

##### Parameters

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

Here is the call graph for this function:



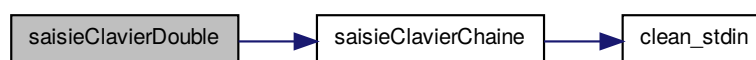
#### 4.18.3.4 `void saisieClavierDouble ( double * nb )`

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

##### Parameters

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

Here is the call graph for this function:



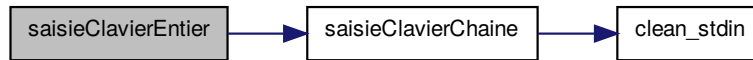
#### 4.18.3.5 void saisieClavierEntier ( int \* nb )

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

**Parameters**

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

Here is the call graph for this function:

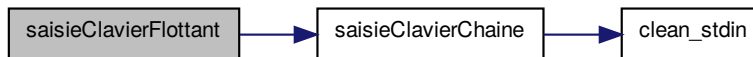
**4.18.3.6 void saisieClavierFlottant ( float \* nb )**

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

**Parameters**

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

Here is the call graph for this function:

**4.18.3.7 void saisieClavierFlottantSansVirgule ( float \* nb )**

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

**Parameters**

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

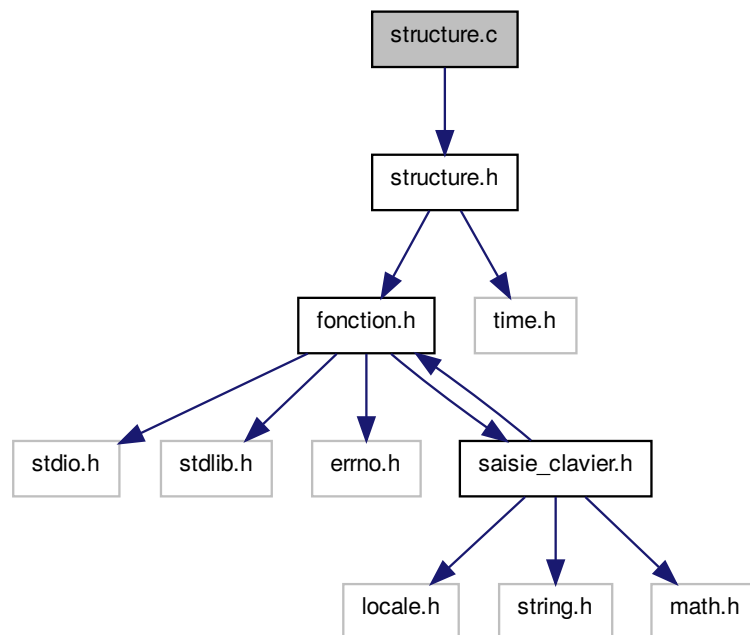
Here is the call graph for this function:

**4.19 structure.c File Reference**

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

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

Include dependency graph for structure.c:



## Functions

- [Fichier\\_Jeu](#) \* [creerFichierStruct](#) (float nb\_joueur, float nb\_max, char sens\_premier, char tour\_par\_tour, char use\_distributor, char number\_after\_comma)
- void [fermeeFichierStruct](#) ([Fichier\\_Jeu](#) \*ptr\_struct\_fichier)
- void [debNouvTour](#) ([Fichier\\_Jeu](#) \*ptr\_struct\_fichier, int num\_joueur)
- void [finNouvTour](#) ([Fichier\\_Jeu](#) \*ptr\_struct\_fichier, int num\_joueur)
- void [calculPosition](#) ([Fichier\\_Jeu](#) \*ptr\_struct\_fichier)
- void [ajoutDistribueStruct](#) ([Fichier\\_Jeu](#) \*ptr\_struct\_fichier, char \*nom\_distribue)
- int [depScoreMax](#) ([Fichier\\_Jeu](#) \*ptr\_struct\_fichier)
- int [maxNbTour](#) ([Fichier\\_Jeu](#) \*ptr\_struct\_fichier)
- int [rechercheNumJoueur](#) ([Fichier\\_Jeu](#) \*ptr\_struct\_fichier, char \*nom\_personne)

### 4.19.1 Detailed Description

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

Author

Remi BERTHO

Date

09/03/14

## Version

2.1.0

## 4.19.2 Function Documentation

## 4.19.2.1 void ajoutDistribueStruct ( Fichier\_Jeu \* ptr\_struct\_fichier, char \* nom\_distribue )

Ajoute la personne qui distribue dans la structure.

## Parameters

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

Here is the call graph for this function:



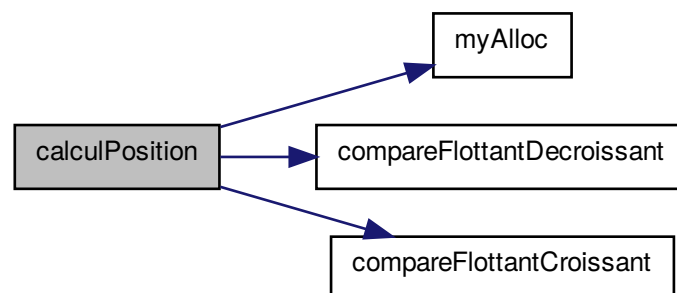
## 4.19.2.2 void calculPosition ( Fichier\_Jeu \* ptr\_struct\_fichier )

Calcule les positions des joueurs

## Parameters

in, out	<i>*ptr_struct_fichier</i>	un pointeur sur la structure <a href="#">Fichier_Jeu</a>
---------	----------------------------	--

Here is the call graph for this function:





4.19.2.3 **Fichier\_Jeu** \* creerFichierStruct ( float *nb\_joueur*, float *nb\_max*, char *sens\_premier*, char *tour\_par\_tour*, char *use\_distributeur*, char *number\_after\_comma* )

Cree une structure Fichier\_jeu a partir des donnees de la fonction.

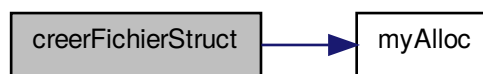
**Parameters**

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
in	<i>use_distributeur</i>	permet de savoir si l'on utilise un distributeur ou pas
in	<i>number_after_comma</i>	nombre de chiffres apres la virgule

**Returns**

un pointeur sur le structure [Fichier\\_Jeu](#) cree

Here is the call graph for this function:



#### 4.19.2.4 void debNouvTour ( Fichier\_Jeu \* ptr\_struct\_fichier, int num\_joueur )

Here is the call graph for this function:



#### 4.19.2.5 int depScoreMax ( Fichier\_Jeu \* ptr\_struct\_fichier )

Verifie si quelqu'un a depasse le score maximum

**Parameters**

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

**Returns**

VRAI s'il y a un depassement, FAUX sinon

#### 4.19.2.6 void fermeeFichierStruct ( Fichier\_Jeu \* ptr\_struct\_fichier )

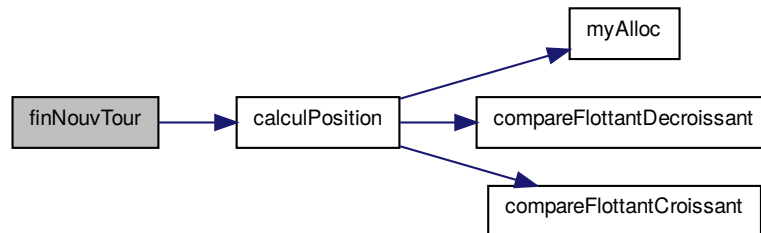
Desalloue la memoire attribuee a la structure [Fichier\\_Jeu](#) mis en parametre

## Parameters

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

## 4.19.2.7 void finNouvTour ( Fichier\_Jeu \* ptr\_struct\_fichier, int num\_joueur )

Here is the call graph for this function:



## 4.19.2.8 int maxNbTour ( Fichier\_Jeu \* ptr\_struct\_fichier )

Cherche le maximum de nombre de tour

## Parameters

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

## Returns

le nombre maximum de nombre de tour

## 4.19.2.9 int rechercheNumJoueur ( Fichier\_Jeu \* ptr\_struct\_fichier, char \* nom\_personne )

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

## Parameters

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

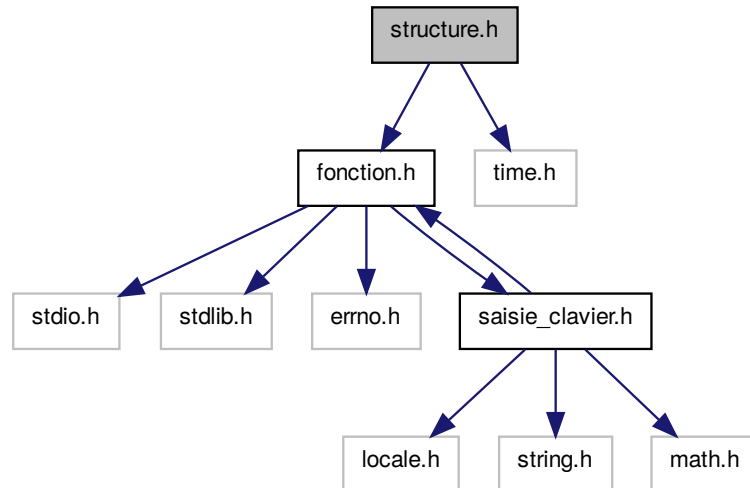
## Returns

l'indice du tableau, -1 si non trouve

## 4.20 structure.h File Reference

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

```
#include "fonction.h"
#include <time.h>
Include dependency graph for structure.h:
```



## Data Structures

- struct [Fichier\\_Jeu](#)

## Macros

- `#define` [TAILLE\\_MAX\\_NOM](#) 30
- `#define` [VERSION](#) 1.4

## Functions

- [Fichier\\_Jeu](#) \* [creerFichierStruct](#) (float nb\_joueur, float nb\_max, char sens\_premier, char tour\_par\_tour, char use\_distributeur, char number\_after\_comma)
- void [fermeeFichierStruct](#) ([Fichier\\_Jeu](#) \*ptr\_struct\_fichier)
- void [debNouvTour](#) ([Fichier\\_Jeu](#) \*ptr\_struct\_fichier, int num\_joueur)
- void [finNouvTour](#) ([Fichier\\_Jeu](#) \*ptr\_struct\_fichier, int num\_joueur)
- void [calculPosition](#) ([Fichier\\_Jeu](#) \*ptr\_struct\_fichier)
- void [ajoutDistribueStruct](#) ([Fichier\\_Jeu](#) \*ptr\_struct\_fichier, char \*nom\_distribue)
- int [depScoreMax](#) ([Fichier\\_Jeu](#) \*ptr\_struct\_fichier)
- int [maxNbTour](#) ([Fichier\\_Jeu](#) \*ptr\_struct\_fichier)
- int [rechercheNumJoueur](#) ([Fichier\\_Jeu](#) \*ptr\_struct\_fichier, char \*nom\_personne)

### 4.20.1 Detailed Description

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

## Author

Remi BERTHO

## Date

09/03/14

## Version

2.1.0

## 4.20.2 Macro Definition Documentation

### 4.20.2.1 #define TAILLE\_MAX\_NOM 30

Definit la taille max d'un nom a 30

### 4.20.2.2 #define VERSION 1.4

Definit la version a 1.4

## 4.20.3 Function Documentation

### 4.20.3.1 void ajoutDistribueStruct ( Fichier\_Jeu \* *ptr\_struct\_fichier*, char \* *nom\_distribue* )

Ajoute la personne qui distribue dans la structure.

**Parameters**

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

Here is the call graph for this function:



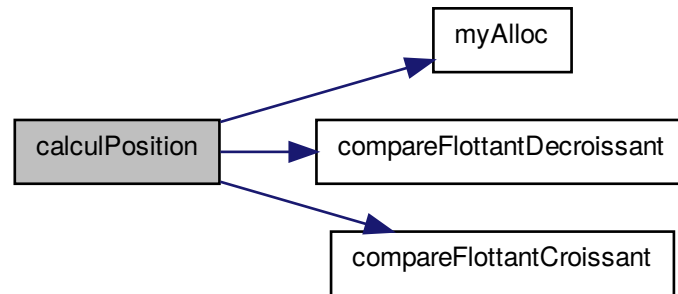
### 4.20.3.2 void calculPosition ( Fichier\_Jeu \* *ptr\_struct\_fichier* )

Calcule les positions des joueurs

## Parameters

in, out	<i>*ptr_struct_ - fichier</i>	un pointeur sur la structure <a href="#">Fichier_Jeu</a>
---------	-----------------------------------	--

Here is the call graph for this function:



#### 4.20.3.3 `Fichier_Jeu*` `creerFichierStruct` ( float *nb\_joueur*, float *nb\_max*, char *sens\_premier*, char *tour\_par\_tour*, char *use\_distributeur*, char *number\_after\_comma* )

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
in	<i>use_distributeur</i>	permet de savoir si l'on utilise un distributeur ou pas
in	<i>number_after_comma</i>	nombre de chiffres apres la virgule

## Returns

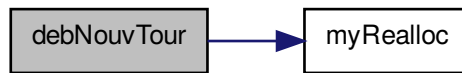
un pointeur sur le structure [Fichier\\_Jeu](#) creee

Here is the call graph for this function:



4.20.3.4 void debNouvTour ( Fichier\_Jeu \* ptr\_struct\_fichier, int num\_joueur )

Here is the call graph for this function:



4.20.3.5 int depScoreMax ( Fichier\_Jeu \* ptr\_struct\_fichier )

Verifie si quelqu'un a depasse le score maximum

#### Parameters

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

#### Returns

VRAI s'il y a un depassement, FAUX sinon

4.20.3.6 void fermeeFichierStruct ( Fichier\_Jeu \* ptr\_struct\_fichier )

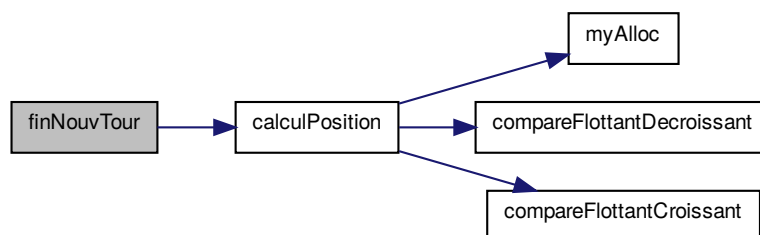
Desalloue la memoire attribuee a la structure Fichier\_Jeu mis en parametre

#### Parameters

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

4.20.3.7 void finNouvTour ( Fichier\_Jeu \* ptr\_struct\_fichier, int num\_joueur )

Here is the call graph for this function:



4.20.3.8 int maxNbTour ( Fichier\_Jeu \* ptr\_struct\_fichier )

Cherche le maximum de nombre de tour



## Parameters

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

## Returns

le nombre maximum de nombre de tour

**4.20.3.9 int rechercheNumJoueur ( Fichier\_Jeu \* ptr\_struct\_fichier, char \* nom\_personne )**

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

## Parameters

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

## Returns

l'indice du tableau, -1 si non trouve

# Index

affFich  
    interface.h, 59

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

affichage.h, 14  
    afficherChaineTroisTab, 16  
    afficherDistribue, 16  
    afficherEnTete, 16  
    afficherLicense, 16  
    afficherLigne, 16  
    afficherNom, 17  
    afficherPartieFinie, 17  
    afficherPosition, 17  
    afficherScore, 17  
    afficherScoreEntier, 18  
    afficherScoreTotal, 18  
    afficherStruct, 19

afficheFichier  
    interface.c, 48  
    interface.h, 59

afficheFichierLocale  
    interface.c, 49  
    interface.h, 60

afficherChaineTroisTab  
    affichage.c, 10  
    affichage.h, 16

afficherDistribue  
    affichage.c, 10  
    affichage.h, 16

afficherEnTete  
    affichage.c, 10  
    affichage.h, 16

afficherLicense  
    affichage.c, 11  
    affichage.h, 16

afficherLigne  
    affichage.c, 11  
    affichage.h, 16

afficherNom  
    affichage.c, 11  
    affichage.h, 17

afficherPartieFinie  
    affichage.c, 11  
    affichage.h, 17

afficherPosition  
    affichage.c, 12  
    affichage.h, 17

afficherScore  
    affichage.c, 12  
    affichage.h, 17

afficherScoreEntier  
    affichage.c, 13  
    affichage.h, 18

afficherScoreTotal  
    affichage.c, 13  
    affichage.h, 18

afficherStruct  
    affichage.c, 13  
    affichage.h, 19

ajoutDistribueStruct  
    structure.c, 94  
    structure.h, 99

ajoutExtension  
    fonction.c, 42  
    fonction.h, 45

annee  
    Fichier\_Jeu, 5

argument\_main.c, 19  
    searchArgument, 20

argument\_main.h, 21  
    LECTURE\_FICHER, 22  
    OUVERTURE\_FICHER, 22  
    searchArgument, 22

calculPosition  
    structure.c, 94  
    structure.h, 99

changerCheminFichier  
    emplacement\_fichier.c, 24  
    emplacement\_fichier.h, 28

charPart  
    interface.h, 59

chargerPartie  
    interface.c, 50  
    interface.h, 61

chargerPartieLocale  
    interface.c, 51  
    interface.h, 62

- clean\_stdin
  - saisie\_clavier.c, 85
  - saisie\_clavier.h, 89
- compareFlottantCroissant
  - fonction.c, 42
  - fonction.h, 45
- compareFlottantDecroissant
  - fonction.c, 42
  - fonction.h, 45
- creationPreferences
  - emplacement\_fichier.c, 24
  - emplacement\_fichier.h, 28
- creerFichierStruct
  - structure.c, 94
  - structure.h, 100
- debNouvTour
  - structure.c, 96
  - structure.h, 100
- depScoreMax
  - structure.c, 96
  - structure.h, 101
- distribue
  - Fichier\_Jeu, 5
- EXTENSION\_FICHER
  - fichier.h, 37
- easterEggs
  - interface.h, 59
- ecrireFichier
  - fichier.c, 32
  - fichier.h, 37
- emplacement\_fichier.c, 23
  - changerCheminFichier, 24
  - creationPreferences, 24
  - lectureCheminFichier, 25
  - lecturePreferences, 25
- emplacement\_fichier.h, 26
  - changerCheminFichier, 28
  - creationPreferences, 28
  - lectureCheminFichier, 29
  - lecturePreferences, 29
  - NOM\_DOSSIER, 28
  - NOM\_FICHER, 28
  - PORTABLE, 28
- FAUX
  - fonction.h, 45
- fermeeFichierStruct
  - structure.c, 96
  - structure.h, 101
- fermerFichier
  - fonction.c, 42
  - fonction.h, 45
- fichier.c, 30
  - ecrireFichier, 32
  - lireFichier, 32
  - nouveauScore, 33
  - ouvrirFichierExtension, 33
  - renommerFichier, 35
  - supprimerFichier, 35
- fichier.h, 36
  - EXTENSION\_FICHER, 37
  - ecrireFichier, 37
  - lireFichier, 38
  - nouveauScore, 39
  - ouvrirFichierExtension, 39
  - renommerFichier, 39
  - supprimerFichier, 40
  - TYPE\_FICHER, 37
- Fichier\_Jeu, 5
  - annee, 5
  - distribue, 5
  - jour, 5
  - mois, 6
  - nb\_joueur, 6
  - nb\_max, 6
  - nb\_tour, 6
  - nom\_joueur, 6
  - number\_after\_comma, 6
  - point, 6
  - point\_tot, 6
  - position, 6
  - sens\_premier, 6
  - taille\_max\_nom, 6
  - tour\_par\_tour, 6
  - use\_distributor, 7
  - version, 7
- finNouvTour
  - structure.c, 97
  - structure.h, 101
- fonction.c, 40
  - ajoutExtension, 42
  - compareFlottantCroissant, 42
  - compareFlottantDecroissant, 42
  - fermerFichier, 42
  - lireTailleFichier, 42
  - mauvais\_choix, 43
  - myAlloc, 43
  - myRealloc, 43
  - ouvrirFichier, 43
  - systemEfface, 43
  - systemPause, 43
- fonction.h, 44
  - ajoutExtension, 45
  - compareFlottantCroissant, 45
  - compareFlottantDecroissant, 45
  - FAUX, 45
  - fermerFichier, 45
  - lireTailleFichier, 46
  - mauvais\_choix, 46
  - myAlloc, 46
  - myRealloc, 46
  - ouvrirFichier, 46
  - systemEfface, 46
  - systemPause, 46
  - VRAI, 45

- interface.h
  - affFich, 59
  - charPart, 59
  - easterEggs, 59
  - listFich, 59
  - nouvPart, 59
  - quit, 59
  - supprFich, 59
- interface.c, 47
  - afficheFichier, 48
  - afficheFichierLocale, 49
  - chargerPartie, 50
  - chargerPartieLocale, 51
  - jouer, 52
  - lireCheminFichier, 53
  - listerFichier, 53
  - menuPrincipal, 54
  - nouveauCheminFichier, 55
  - nouvellePartie, 56
  - supprimerFichierNom, 57
- interface.h, 58
  - afficheFichier, 59
  - afficheFichierLocale, 60
  - chargerPartie, 61
  - chargerPartieLocale, 62
  - jouer, 63
  - lireCheminFichier, 64
  - listerFichier, 64
  - Menu, 59
  - menuPrincipal, 65
  - nouveauCheminFichier, 66
  - nouvellePartie, 67
  - supprimerFichierNom, 68
- jouer
  - interface.c, 52
  - interface.h, 63
- jour
  - Fichier\_Jeu, 5
- LECTURE\_FICHER
  - argument\_main.h, 22
- lectureCheminFichier
  - emplacement\_fichier.c, 25
  - emplacement\_fichier.h, 29
- lecturePreferences
  - emplacement\_fichier.c, 25
  - emplacement\_fichier.h, 29
- lireCheminFichier
  - interface.c, 53
  - interface.h, 64
- lireFichier
  - fichier.c, 32
  - fichier.h, 38
- lireTailleFichier
  - fonction.c, 42
  - fonction.h, 46
- listFich
  - interface.h, 59
- listerFichier
  - interface.c, 53
  - interface.h, 64
- main
  - main.c, 70
  - main.h, 72
- main.c, 69
  - main, 70
- main.h, 71
  - main, 72
- mauvais\_choix
  - fonction.c, 43
  - fonction.h, 46
- maxNbTour
  - structure.c, 97
  - structure.h, 101
- Menu
  - interface.h, 59
- menu.c, 74
  - menuContinuer, 75
  - menuDebutPartie, 75
  - menuDistribue, 75
  - menuNomFichier, 76
  - menuNomJoueur, 76
  - menuNouveauChemin, 76
  - menuNumJoueur, 77
  - menuPointsJoueur, 77
  - menuSupprimer, 78
- menu.h, 78
  - menuContinuer, 80
  - menuDebutPartie, 80
  - menuDistribue, 80
  - menuNomFichier, 81
  - menuNomJoueur, 81
  - menuNouveauChemin, 81
  - menuNumJoueur, 83
  - menuPointsJoueur, 83
  - menuSupprimer, 84
- menuContinuer
  - menu.c, 75
  - menu.h, 80
- menuDebutPartie
  - menu.c, 75
  - menu.h, 80
- menuDistribue
  - menu.c, 75
  - menu.h, 80
- menuNomFichier
  - menu.c, 76
  - menu.h, 81
- menuNomJoueur
  - menu.c, 76
  - menu.h, 81
- menuNouveauChemin
  - menu.c, 76
  - menu.h, 81
- menuNumJoueur
  - menu.c, 77

- menu.h, 83
- menuPointsJoueur
  - menu.c, 77
  - menu.h, 83
- menuPrincipal
  - interface.c, 54
  - interface.h, 65
- menuSupprimer
  - menu.c, 78
  - menu.h, 84
- mois
  - Fichier\_Jeu, 6
- myAlloc
  - fonction.c, 43
  - fonction.h, 46
- myRealloc
  - fonction.c, 43
  - fonction.h, 46
- NB\_CARACT\_DOUB
  - saisie\_clavier.h, 89
- NB\_CARACT\_FLOT
  - saisie\_clavier.h, 89
- NB\_CARACT\_INT
  - saisie\_clavier.h, 89
- NOM\_DOSSIER
  - emplacement\_fichier.h, 28
- NOM\_FICHIER
  - emplacement\_fichier.h, 28
- nb\_joueur
  - Fichier\_Jeu, 6
- nb\_max
  - Fichier\_Jeu, 6
- nb\_tour
  - Fichier\_Jeu, 6
- nom\_joueur
  - Fichier\_Jeu, 6
- nouvPart
  - interface.h, 59
- nouveauCheminFichier
  - interface.c, 55
  - interface.h, 66
- nouveauScore
  - fichier.c, 33
  - fichier.h, 39
- nouvellePartie
  - interface.c, 56
  - interface.h, 67
- number\_after\_comma
  - Fichier\_Jeu, 6
- OUVERTURE\_FICHIER
  - argument\_main.h, 22
- ouvrirFichier
  - fonction.c, 43
  - fonction.h, 46
- ouvrirFichierExtension
  - fichier.c, 33
  - fichier.h, 39
- PORTABLE
  - emplacement\_fichier.h, 28
- point
  - Fichier\_Jeu, 6
- point\_tot
  - Fichier\_Jeu, 6
- position
  - Fichier\_Jeu, 6
- quit
  - interface.h, 59
- rechercheNumJoueur
  - structure.c, 97
  - structure.h, 103
- renommerFichier
  - fichier.c, 35
  - fichier.h, 39
- saisie\_clavier.c, 84
  - clean\_stdin, 85
  - saisieClavierCaractere, 85
  - saisieClavierChaine, 86
  - saisieClavierDouble, 86
  - saisieClavierEntier, 87
  - saisieClavierFlottant, 87
  - saisieClavierFlottantSansVirgule, 87
- saisie\_clavier.h, 88
  - clean\_stdin, 89
  - NB\_CARACT\_DOUB, 89
  - NB\_CARACT\_FLOT, 89
  - NB\_CARACT\_INT, 89
  - saisieClavierCaractere, 89
  - saisieClavierChaine, 90
  - saisieClavierDouble, 90
  - saisieClavierEntier, 90
  - saisieClavierFlottant, 92
  - saisieClavierFlottantSansVirgule, 92
- saisieClavierCaractere
  - saisie\_clavier.c, 85
  - saisie\_clavier.h, 89
- saisieClavierChaine
  - saisie\_clavier.c, 86
  - saisie\_clavier.h, 90
- saisieClavierDouble
  - saisie\_clavier.c, 86
  - saisie\_clavier.h, 90
- saisieClavierEntier
  - saisie\_clavier.c, 87
  - saisie\_clavier.h, 90
- saisieClavierFlottant
  - saisie\_clavier.c, 87
  - saisie\_clavier.h, 92
- saisieClavierFlottantSansVirgule
  - saisie\_clavier.c, 87
  - saisie\_clavier.h, 92
- searchArgument
  - argument\_main.c, 20
  - argument\_main.h, 22

- sens\_premier
  - Fichier\_Jeu, 6
- structure.c, 92
  - ajoutDistribueStruct, 94
  - calculPosition, 94
  - creerFichierStruct, 94
  - debNouvTour, 96
  - depScoreMax, 96
  - fermeeFichierStruct, 96
  - finNouvTour, 97
  - maxNbTour, 97
  - rechercheNumJoueur, 97
- structure.h, 97
  - ajoutDistribueStruct, 99
  - calculPosition, 99
  - creerFichierStruct, 100
  - debNouvTour, 100
  - depScoreMax, 101
  - fermeeFichierStruct, 101
  - finNouvTour, 101
  - maxNbTour, 101
  - rechercheNumJoueur, 103
  - TAILLE\_MAX\_NOM, 99
  - VERSION, 99
- supprFich
  - interface.h, 59
- supprimerFichier
  - fichier.c, 35
  - fichier.h, 40
- supprimerFichierNom
  - interface.c, 57
  - interface.h, 68
- systemEfface
  - fonction.c, 43
  - fonction.h, 46
- systemPause
  - fonction.c, 43
  - fonction.h, 46
- TAILLE\_MAX\_NOM
  - structure.h, 99
- TYPE\_FICHIER
  - fichier.h, 37
- taille\_max\_nom
  - Fichier\_Jeu, 6
- tour\_par\_tour
  - Fichier\_Jeu, 6
- use\_distributor
  - Fichier\_Jeu, 7
- VERSION
  - structure.h, 99
- VRAI
  - fonction.h, 45
- version
  - Fichier\_Jeu, 7