

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

2.1.9

Generated by Doxygen 1.8.6

Thu Apr 17 2014 16:55:42

Contents

1	File Index	1
1.1	File List	1
2	File Documentation	3
2.1	csuper.h File Reference	3
2.1.1	Detailed Description	3
2.1.2	Macro Definition Documentation	3
2.1.2.1	_	3
2.1.2.2	CSUPER	3
2.2	display.c File Reference	4
2.2.1	Detailed Description	4
2.2.2	Function Documentation	4
2.2.2.1	printAllPoints	4
2.2.2.2	printCsuStruct	4
2.2.2.3	printDistributor	5
2.2.2.4	printGameConfig	5
2.2.2.5	printGameOver	5
2.2.2.6	printHeader	6
2.2.2.7	printLicense	6
2.2.2.8	printLigne	6
2.2.2.9	printNames	6
2.2.2.10	printPoints	7
2.2.2.11	printRanking	7
2.2.2.12	printStringThreeTabs	7
2.2.2.13	printTotalPoints	8
2.3	display.h File Reference	8
2.3.1	Detailed Description	8
2.3.2	Function Documentation	9
2.3.2.1	printAllPoints	9
2.3.2.2	printCsuStruct	10
2.3.2.3	printDistributor	10

2.3.2.4	printGameConfig	10
2.3.2.5	printGameOver	11
2.3.2.6	printHeader	11
2.3.2.7	printLicense	11
2.3.2.8	printLigne	11
2.3.2.9	printNames	12
2.3.2.10	printPoints	12
2.3.2.11	printRanking	12
2.3.2.12	printStringThreeTabs	13
2.3.2.13	printTotalPoints	13
2.4	interface.c File Reference	13
2.4.1	Detailed Description	13
2.4.2	Function Documentation	14
2.4.2.1	changeFilePath	14
2.4.2.2	deleteCsuFileNom	14
2.4.2.3	displayFile	14
2.4.2.4	displayFileLocale	15
2.4.2.5	listCsuFiles	16
2.4.2.6	loadGame	16
2.4.2.7	loadGameLocale	17
2.4.2.8	mainMenu	18
2.4.2.9	newGame	18
2.4.2.10	newGameConfig	19
2.4.2.11	play	19
2.4.2.12	preferencesMenu	20
2.4.2.13	printGameConfigFile	21
2.4.2.14	printListGameConfig	21
2.4.2.15	readFilePath	21
2.4.2.16	removeGameConfig	21
2.5	interface.h File Reference	22
2.5.1	Detailed Description	23
2.5.2	Enumeration Type Documentation	23
2.5.2.1	MainMenu	23
2.5.2.2	PreferencesMenu	23
2.5.3	Function Documentation	23
2.5.3.1	changeFilePath	23
2.5.3.2	deleteCsuFileNom	24
2.5.3.3	displayFile	24
2.5.3.4	displayFileLocale	25
2.5.3.5	listCsuFiles	26

2.5.3.6	loadGame	26
2.5.3.7	loadGameLocale	27
2.5.3.8	mainMenu	28
2.5.3.9	newGame	28
2.5.3.10	newGameConfig	29
2.5.3.11	play	29
2.5.3.12	preferencesMenu	30
2.5.3.13	printGameConfigFile	31
2.5.3.14	printListGameConfig	31
2.5.3.15	readFilePath	31
2.5.3.16	removeGameConfig	31
2.6	keyboarding.c File Reference	32
2.6.1	Detailed Description	32
2.6.2	Function Documentation	32
2.6.2.1	charKey	32
2.6.2.2	clean_stdin	33
2.6.2.3	doubleKey	33
2.6.2.4	floatKey	33
2.6.2.5	floatKeyNoComma	34
2.6.2.6	intKey	34
2.6.2.7	stringKey	34
2.6.2.8	systemPause	35
2.7	keyboarding.h File Reference	35
2.7.1	Detailed Description	36
2.7.2	Macro Definition Documentation	36
2.7.2.1	NB_CHARACTER_DOUB	36
2.7.2.2	NB_CHARACTER_FLOAT	36
2.7.2.3	NB_CHARACTER_INT	36
2.7.3	Function Documentation	36
2.7.3.1	charKey	36
2.7.3.2	clean_stdin	37
2.7.3.3	doubleKey	37
2.7.3.4	floatKey	37
2.7.3.5	floatKeyNoComma	38
2.7.3.6	intKey	38
2.7.3.7	stringKey	38
2.7.3.8	systemPause	39
2.8	main.c File Reference	39
2.8.1	Detailed Description	39
2.8.2	Function Documentation	40

2.8.2.1	main	40
2.9	main.h File Reference	40
2.9.1	Detailed Description	41
2.9.2	Function Documentation	41
2.9.2.1	main	41
2.10	menu.c File Reference	42
2.10.1	Detailed Description	43
2.10.2	Function Documentation	43
2.10.2.1	menuContinue	43
2.10.2.2	menuDelete	43
2.10.2.3	menuDistributor	44
2.10.2.4	menuFileName	44
2.10.2.5	menuGameConfig	45
2.10.2.6	menuNewPath	45
2.10.2.7	menuPlayerIndex	46
2.10.2.8	menuPlayersName	46
2.10.2.9	menuPlayersPoints	46
2.10.2.10	menuStartGame	47
2.11	menu.h File Reference	47
2.11.1	Detailed Description	48
2.11.2	Function Documentation	48
2.11.2.1	menuContinue	48
2.11.2.2	menuDelete	48
2.11.2.3	menuDistributor	49
2.11.2.4	menuFileName	49
2.11.2.5	menuGameConfig	50
2.11.2.6	menuNewPath	50
2.11.2.7	menuPlayerIndex	51
2.11.2.8	menuPlayersName	51
2.11.2.9	menuPlayersPoints	51
2.11.2.10	menuStartGame	52
Index		53

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

csuper.h	Include of csuper	3
display.c	Display the games	4
display.h	Display the games	8
interface.c	Graphical interface	13
interface.h	Graphical interface	22
keyboarding.c	Function of keyboarding	32
keyboarding.h	Function of keyboarding	35
main.c	Begin csuper	39
main.h	Begin csuper	40
menu.c	Menu functions	42
menu.h	Menu functions	47

Chapter 2

File Documentation

2.1 csuper.h File Reference

Include of csuper.

```
#include "../libcsuper/libcsuper.h"  
#include <locale.h>
```

Macros

- #define [CSUPER](#)
- #define [_\(STRING\)](#) gettext(STRING)

2.1.1 Detailed Description

Include of csuper.

Author

Remi BERTHO

Date

17/04/14

Version

2.2.0

2.1.2 Macro Definition Documentation

2.1.2.1 #define [_\(STRING \)](#) gettext(STRING)

Define the [_](#) for gettext.

2.1.2.2 #define [CSUPER](#)

Define that we compile csuper.

2.2 display.c File Reference

Display the games.

```
#include "display.h"
```

Functions

- void [printNames](#) (csuStruct *ptr_csu_struct, int *ptr_size_line)
- void [printLigne](#) (int size_line)
- void [printTotalPoints](#) (csuStruct *ptr_csu_struct)
- void [printDistributor](#) (csuStruct *ptr_csu_struct)
- void [printHeader](#) (csuStruct *ptr_csu_struct)
- void [printAllPoints](#) (csuStruct *ptr_csu_struct)
- void [printRanking](#) (csuStruct *ptr_csu_struct)
- void [printPoints](#) (csuStruct *ptr_csu_struct)
- void [printCsuStruct](#) (csuStruct *ptr_csu_struct)
- void [printGameOver](#) (csuStruct *ptr_csu_struct)
- void [printStringThreeTabs](#) (char *string)
- void [printLicense](#) ()
- void [printGameConfig](#) (game_config config)

2.2.1 Detailed Description

Display the games.

Author

Remi BERTHO

Date

17/04/14

Version

2.2.0

2.2.2 Function Documentation

2.2.2.1 void printAllPoints (csuStruct * ptr_csu_struct)

Print the scores of all player in each turn.

Parameters

in	*ptr_csu_struct	a pointer on a csu structure
----	-----------------	------------------------------

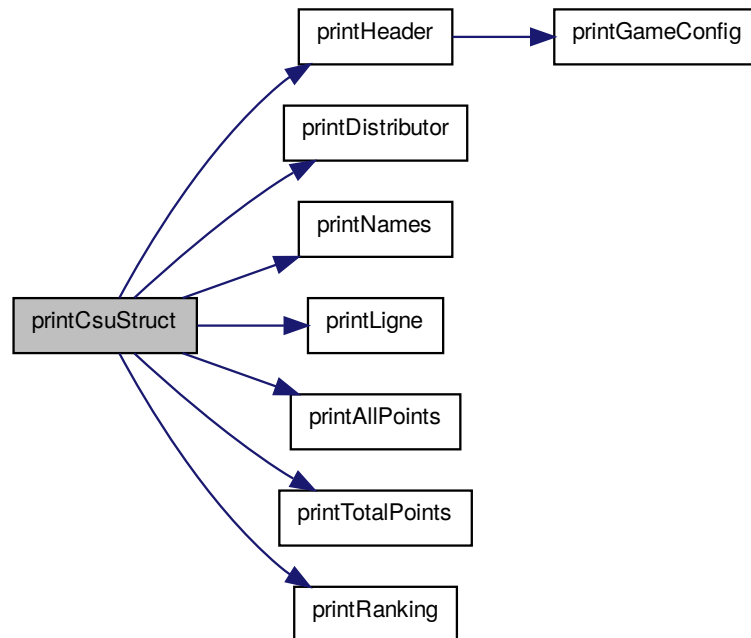
2.2.2.2 void printCsuStruct (csuStruct * ptr_csu_struct)

Print all data of the csu structure.

Parameters

in	<i>*ptr_csu_struct</i>	a pointer on a csu structure
----	------------------------	------------------------------

Here is the call graph for this function:



2.2.2.3 void printDistributor (csuStruct * ptr_csu_struct)

Display the distributor.

Parameters

in	<i>*ptr_csu_struct</i>	a pointer on a csu structure
----	------------------------	------------------------------

2.2.2.4 void printGameConfig (game_config config)

Print the game config

Parameters

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

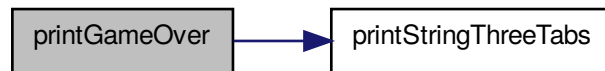
2.2.2.5 void printGameOver (csuStruct * ptr_csu_struct)

Display that the game is over and a podium.

Parameters

in	<i>*ptr_csu_struct</i>	a pointer on a csu structure
----	------------------------	------------------------------

Here is the call graph for this function:



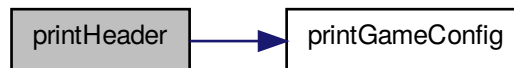
2.2.2.6 void printHeader (csuStruct * ptr_csu_struct)

Print the header of the structure

Parameters

in	<i>*ptr_csu_struct</i>	a pointer on a csu structure
----	------------------------	------------------------------

Here is the call graph for this function:



2.2.2.7 void printLicense ()

Print the license.

2.2.2.8 void printLigne (int size_line)

Print a line of - after a tabulation.

Parameters

in	<i>size_line</i>	the size of the line
----	------------------	----------------------

2.2.2.9 void printNames (csuStruct * ptr_csu_struct, int * ptr_size_line)

Print the payers names and calculate the size of a line.

Parameters

in	<i>*ptr_csu_struct</i>	a pointer on a csu structure
in, out	<i>*ptr_size_line</i>	a pointer to the size of the line, NULL if you doesn't need it

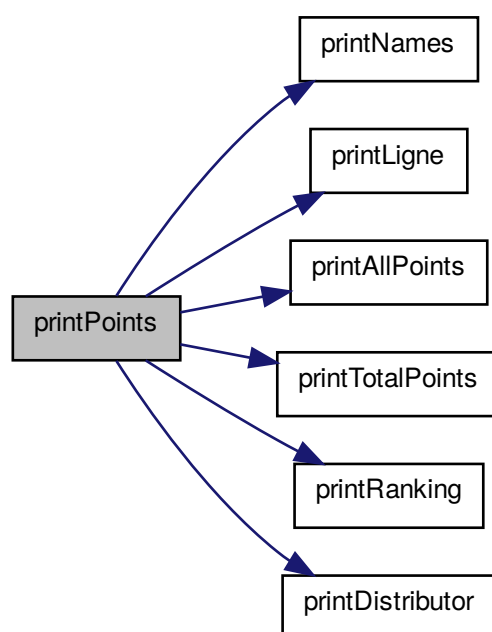
2.2.2.10 void printPoints (csuStruct * ptr_csu_struct)

Print the names, the points, the total points and the rank of each player.

Parameters

in	<i>*ptr_csu_struct</i>	a pointer on a csu structure
----	------------------------	------------------------------

Here is the call graph for this function:



2.2.2.11 void printRanking (csuStruct * ptr_csu_struct)

Print the rank of each player

Parameters

in	<i>*ptr_csu_struct</i>	a pointer on a csu structure
----	------------------------	------------------------------

2.2.2.12 void printStringThreeTabs (char * string)

Print a string center into a space of three tabulations.

Parameters

in	<i>*string</i>	a string
----	----------------	----------

2.2.2.13 void printTotalPoints (csuStruct * ptr_csu_struct)

Print the total score of the players.

Parameters

in	<i>*ptr_csu_struct</i>	a pointer on a csu structure
----	------------------------	------------------------------

2.3 display.h File Reference

Display the games.

```
#include <math.h>
#include <string.h>
#include "main.h"
#include "csuper.h"
```

Functions

- void [printNames](#) (csuStruct *ptr_csu_struct, int *ptr_size_ligne)
- void [printLigne](#) (int size_ligne)
- void [printTotalPoints](#) (csuStruct *ptr_csu_struct)
- void [printDistributor](#) (csuStruct *ptr_csu_struct)
- void [printHeader](#) (csuStruct *ptr_csu_struct)
- void [printAllPoints](#) (csuStruct *ptr_csu_struct)
- void [printRanking](#) (csuStruct *ptr_csu_struct)
- void [printPoints](#) (csuStruct *ptr_csu_struct)
- void [printCsuStruct](#) (csuStruct *ptr_csu_struct)
- void [printGameOver](#) (csuStruct *ptr_csu_struct)
- void [printStringThreeTabs](#) (char *string)
- void [printLicense](#) ()
- void [printGameConfig](#) (game_config config)

2.3.1 Detailed Description

Display the games.

Author

Remi BERTHO

Date

17/04/14

Version

2.2.0

2.3.2 Function Documentation

2.3.2.1 void printAllPoints (csuStruct * *ptr_csu_struct*)

Print the scores of all player in each turn.

Parameters

in	<i>*ptr_csu_struct</i>	a pointer on a csu structure
----	------------------------	------------------------------

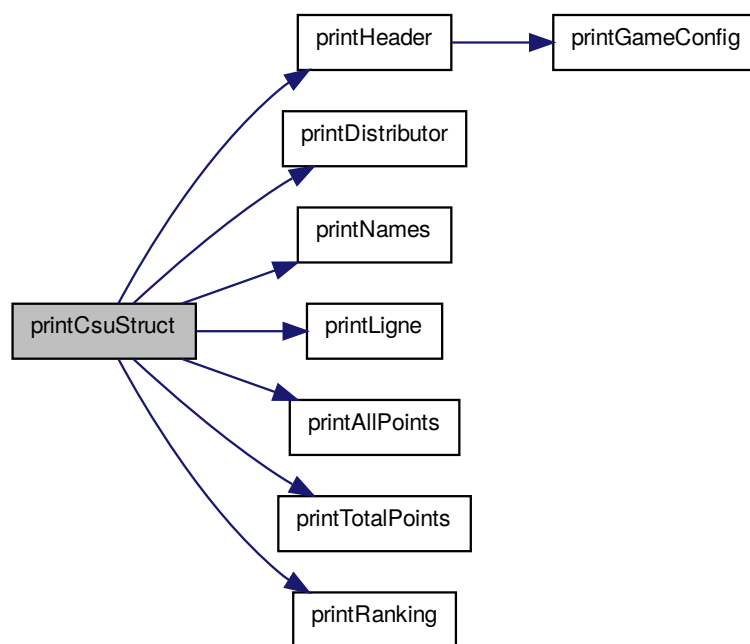
2.3.2.2 void printCsuStruct (csuStruct * ptr_csu_struct)

Print all data of the csu structure.

Parameters

in	<i>*ptr_csu_struct</i>	a pointer on a csu structure
----	------------------------	------------------------------

Here is the call graph for this function:

**2.3.2.3 void printDistributor (csuStruct * ptr_csu_struct)**

Display the distributor.

Parameters

in	<i>*ptr_csu_struct</i>	a pointer on a csu structure
----	------------------------	------------------------------

2.3.2.4 void printGameConfig (game_config config)

Print the game config

Parameters

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

2.3.2.5 void printGameOver (csuStruct * *ptr_csu_struct*)

Display that the game is over and a podium.

Parameters

in	* <i>ptr_csu_struct</i>	a pointer on a csu structure
----	-------------------------	------------------------------

Here is the call graph for this function:

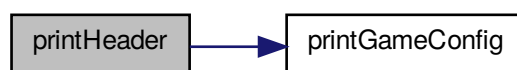
2.3.2.6 void printHeader (csuStruct * *ptr_csu_struct*)

Print the header of the structure

Parameters

in	* <i>ptr_csu_struct</i>	a pointer on a csu structure
----	-------------------------	------------------------------

Here is the call graph for this function:



2.3.2.7 void printLicense ()

Print the license.

2.3.2.8 void printLigne (int *size_line*)

Print a line of - after a tabulation.

Parameters

in	<i>size_line</i>	the size of the line
----	------------------	----------------------

2.3.2.9 void printNames (csuStruct * *ptr_csu_struct*, int * *ptr_size_line*)

Print the payers names and calculate the size of a line.

Parameters

in	* <i>ptr_csu_struct</i>	a pointer on a csu structure
in, out	* <i>ptr_size_line</i>	a pointer to the size of the line, NULL if you doesn't need it

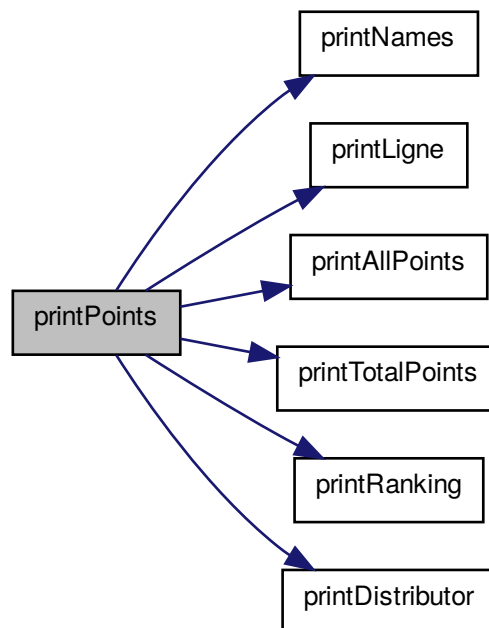
2.3.2.10 void printPoints (csuStruct * *ptr_csu_struct*)

Print the names, the points, the total points and the rank of each player.

Parameters

in	* <i>ptr_csu_struct</i>	a pointer on a csu structure
----	-------------------------	------------------------------

Here is the call graph for this function:

2.3.2.11 void printRanking (csuStruct * *ptr_csu_struct*)

Print the rank of each player

Parameters

in	<i>*ptr_csu_struct</i>	a pointer on a csu structure
----	------------------------	------------------------------

2.3.2.12 void printStringThreeTabs (char * *string*)

Print a string center into a space of three tabulations.

Parameters

in	<i>*string</i>	a string
----	----------------	----------

2.3.2.13 void printTotalPoints (csuStruct * *ptr_csu_struct*)

Print the total score of the players.

Parameters

in	<i>*ptr_csu_struct</i>	a pointer on a csu structure
----	------------------------	------------------------------

2.4 interface.c File Reference

Graphical interface.

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

Functions

- void [displayFile](#) ()
- void [deleteCsuFileNom](#) ()
- void [listCsuFiles](#) ()
- void [play](#) (csuStruct *ptr_csu_struct, char *file_name)
- void [newGame](#) ()
- void [loadGame](#) ()
- void [mainMenu](#) ()
- void [preferencesMenu](#) ()
- void [changeFilePath](#) ()
- void [readFilePath](#) ()
- void [loadGameLocale](#) (char *file_name)
- void [displayFileLocale](#) (char *file_name)
- void [newGameConfig](#) ()
- void [removeGameConfig](#) ()
- void [printListGameConfig](#) ()
- void [printGameConfigFile](#) ()

2.4.1 Detailed Description

Graphical interface.

Author

Remi BERTHO

Date

17/04/14

Version

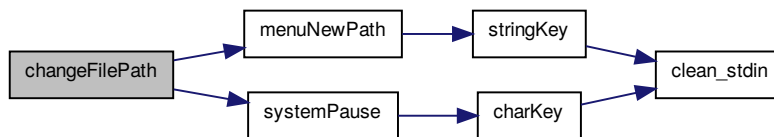
2.2.0

2.4.2 Function Documentation

2.4.2.1 void changeFilePath ()

Change the path which the file are saved.

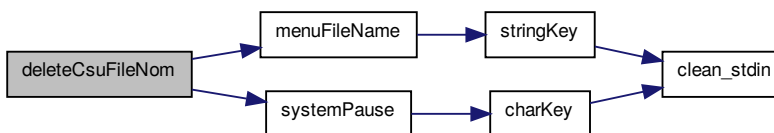
Here is the call graph for this function:



2.4.2.2 void deleteCsuFileNom ()

Ask a filename and deleted the file.

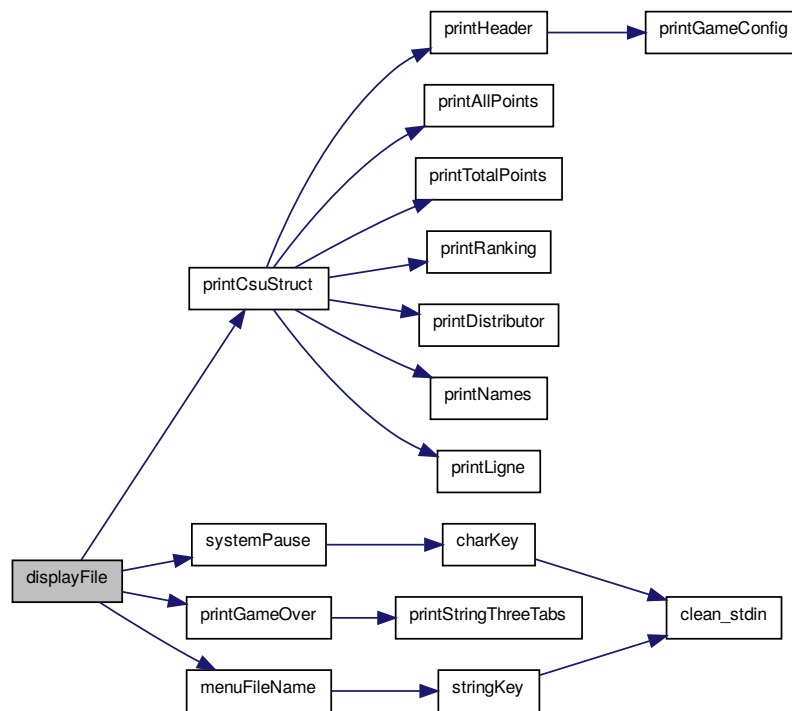
Here is the call graph for this function:



2.4.2.3 void displayFile ()

Ask a filename and display it.

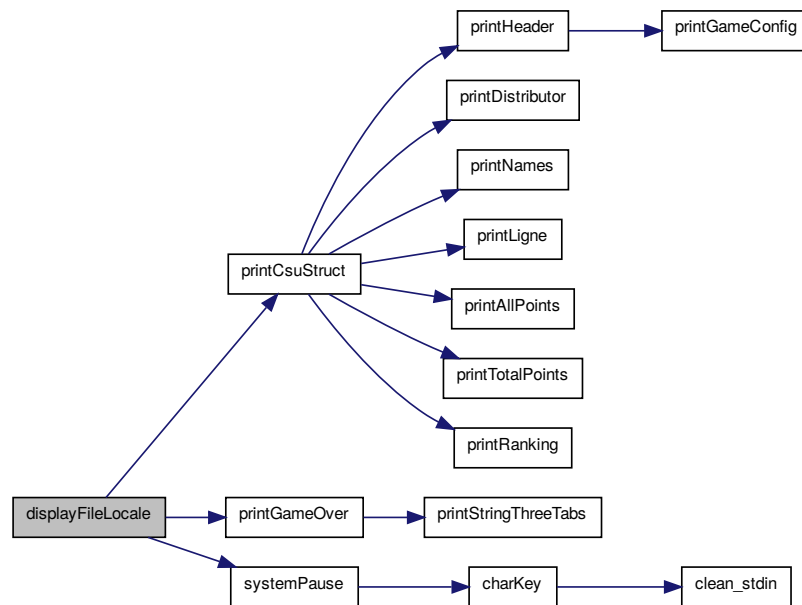
Here is the call graph for this function:



2.4.2.4 void displayFileLocale (char * file_name)

Display the file.

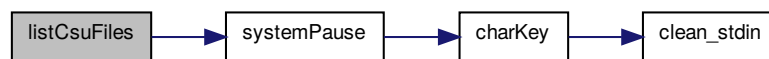
Here is the call graph for this function:



2.4.2.5 void listCsuFiles ()

List all the files with csu extension

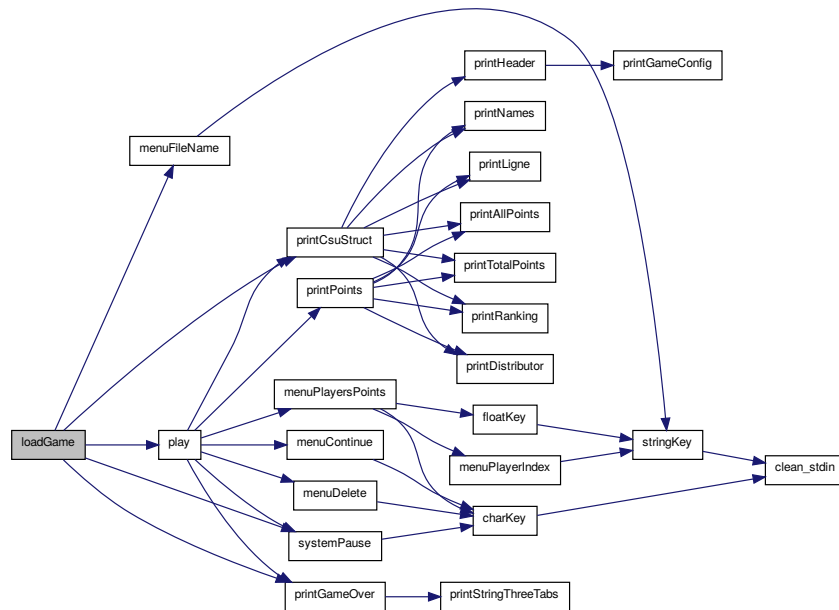
Here is the call graph for this function:



2.4.2.6 void loadGame ()

Load a game from a file and run the play function.

Here is the call graph for this function:



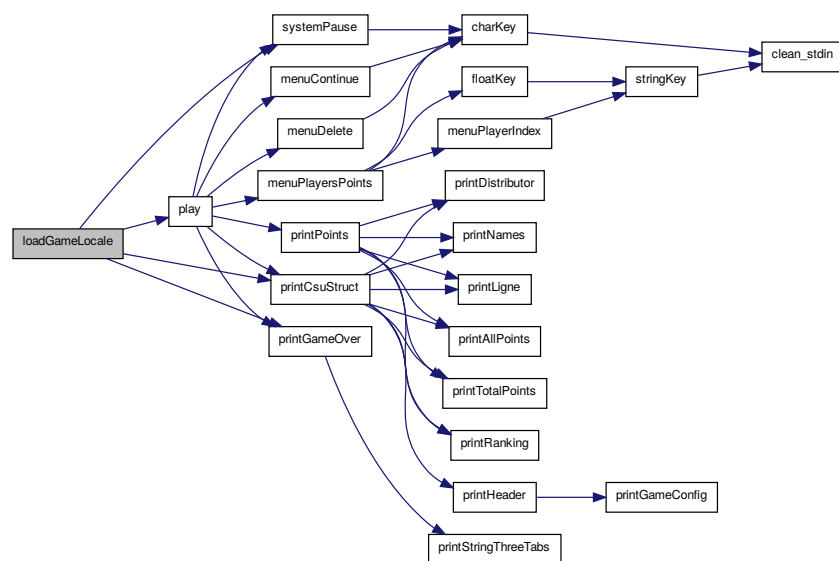
2.4.2.7 void loadGameLocale (char * file_name)

Load the file and run the play function

Parameters

in	file_name,the	filename
----	---------------	----------

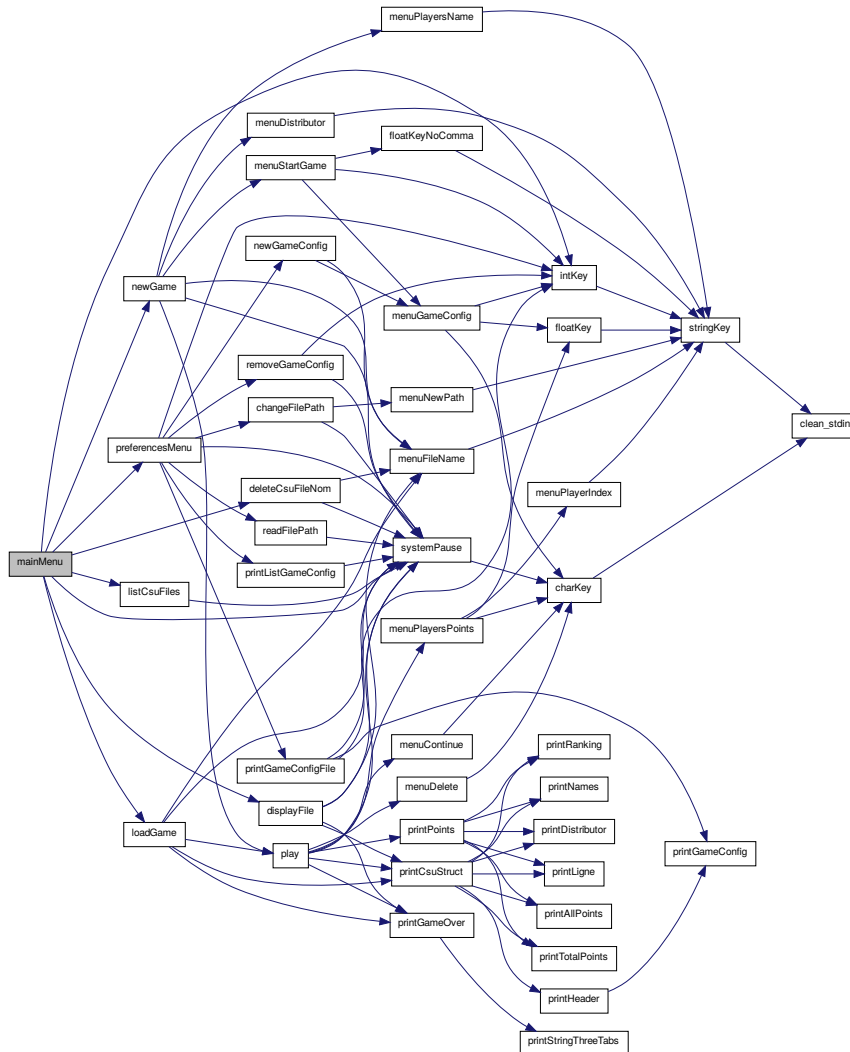
Here is the call graph for this function:



2.4.2.8 void mainMenu ()

Main menu of csuper.

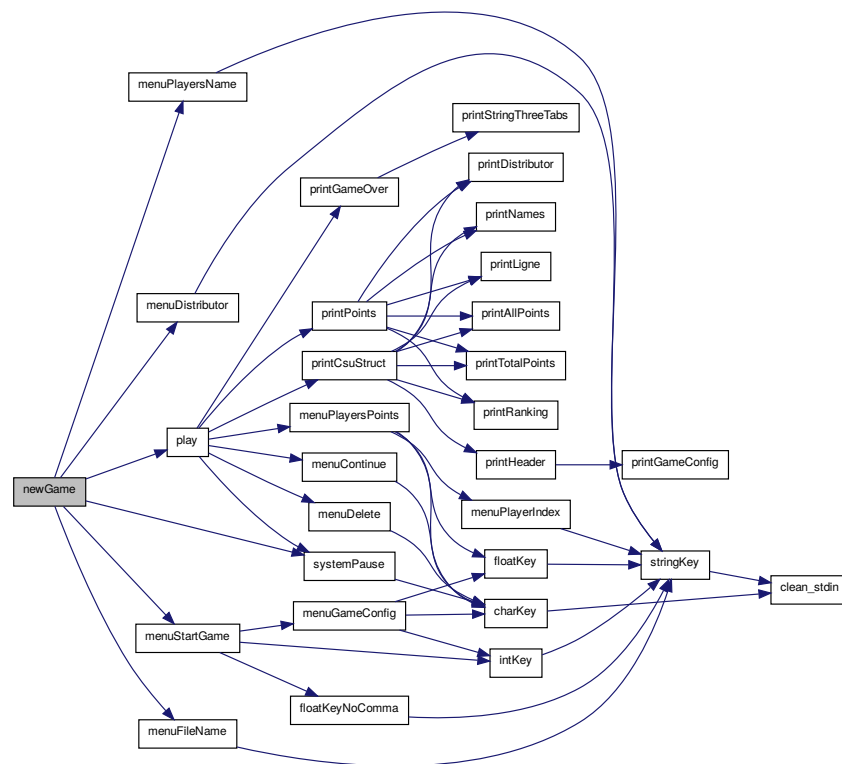
Here is the call graph for this function:



2.4.2.9 void newGame ()

Initialize a new game and run the play function.

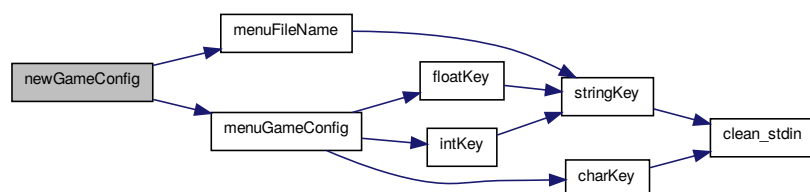
Here is the call graph for this function:



2.4.2.10 void newGameConfig ()

Add a new game configuration

Here is the call graph for this function:



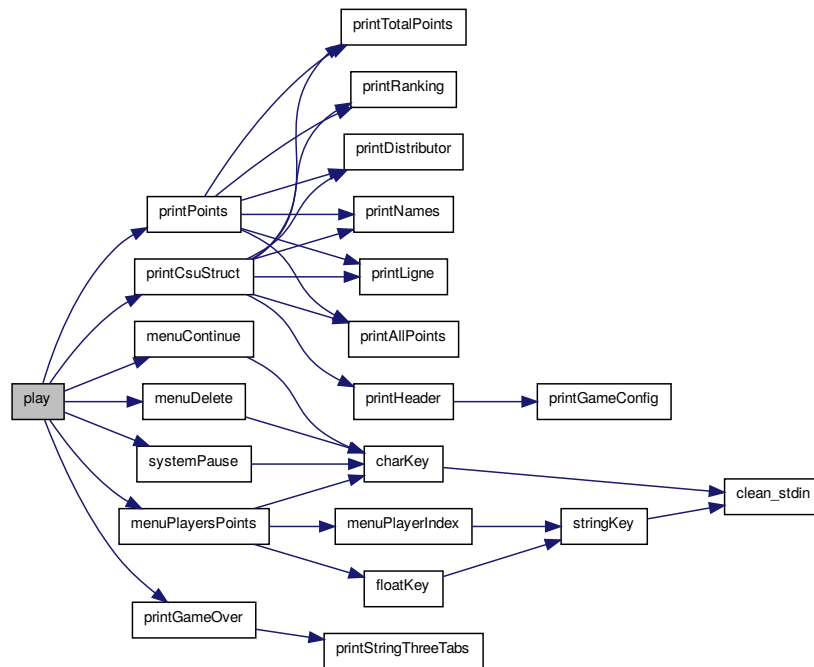
2.4.2.11 void play (csuStruct * ptr_csu_struct, char * file_name)

Count the points

Parameters

in, out	<i>*ptr_csu_struct</i>	a pointer on a csu structure
in	<i>*file_name</i>	the filename

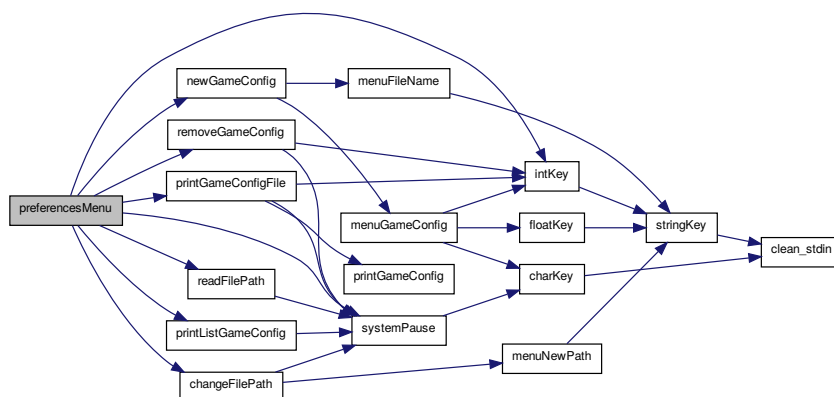
Here is the call graph for this function:



2.4.2.12 void preferencesMenu ()

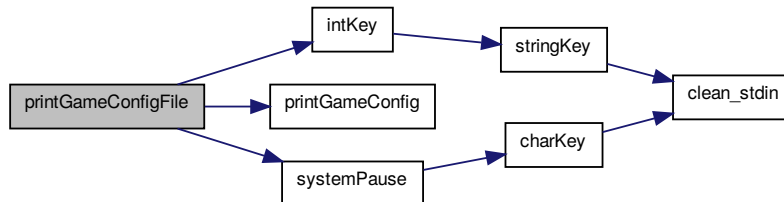
Preferences menu of csuper.

Here is the call graph for this function:



2.4.2.13 void printGameConfigFile ()

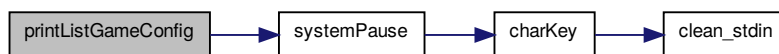
Here is the call graph for this function:



2.4.2.14 void printListGameConfig ()

Print the list of game configuration

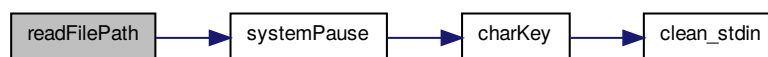
Here is the call graph for this function:



2.4.2.15 void readFilePath ()

Read the file path and display it.

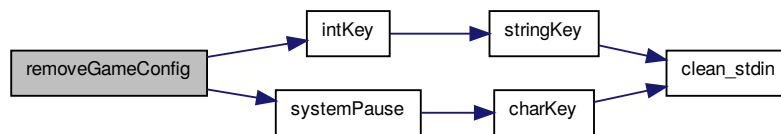
Here is the call graph for this function:



2.4.2.16 void removeGameConfig ()

Ask and remove a game configuration

Here is the call graph for this function:



2.5 interface.h File Reference

Graphical interface.

```
#include <dirent.h>
#include "menu.h"
#include "display.h"
#include "main.h"
#include "csuper.h"
```

Enumerations

- enum `MainMenu` {
`newMatch` =1, `loadMatch` =2, `printFile` =3, `deleteFile` =4,
`listFile` =5, `pref` =6, `quit` =7, `easterEggs` = 42 }
- enum `PreferencesMenu` {
`newPath` =1, `readPath` =2, `newGameConf` =3, `removeGameConf` =4,
`printListGameConf` =5, `printGameConf` =6, `backMainMenu` =7, `easterEggs2` = 42 }

Functions

- void `displayFile` ()
- void `deleteCsuFileNom` ()
- void `listCsuFiles` ()
- void `play` (csuStruct *ptr_csu_struct, char *file_name)
- void `newGame` ()
- void `loadGame` ()
- void `mainMenu` ()
- void `preferencesMenu` ()
- void `changeFilePath` ()
- void `readFilePath` ()
- void `loadGameLocale` (char *file_name)
- void `displayFileLocale` (char *file_name)
- void `newGameConfig` ()
- void `removeGameConfig` ()
- void `printListGameConfig` ()
- void `printGameConfigFile` ()

2.5.1 Detailed Description

Graphical interface.

Author

Remi BERTHO

Date

17/04/14

Version

2.2.0

2.5.2 Enumeration Type Documentation

2.5.2.1 enum MainMenu

Enumerator

newMatch
loadMatch
printFile
deleteFile
listFile
pref
quit
easterEggs

2.5.2.2 enum PreferencesMenu

Enumerator

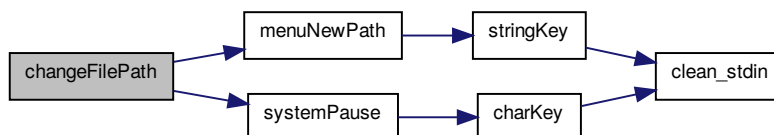
newPath
readPath
newGameConf
removeGameConf
printListGameConf
printGameConf
backMainMenu
easterEggs2

2.5.3 Function Documentation

2.5.3.1 void changeFilePath ()

Change the path which the file are saved.

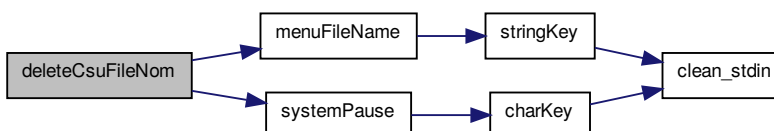
Here is the call graph for this function:



2.5.3.2 void deleteCsuFileNom ()

Ask a filename and deleted the file.

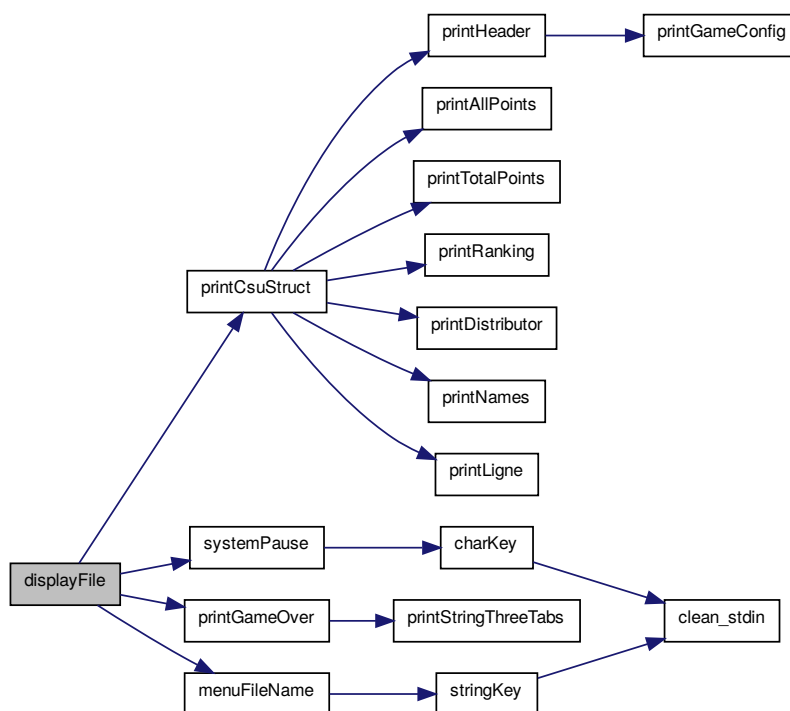
Here is the call graph for this function:



2.5.3.3 void displayFile ()

Ask a filename and display it.

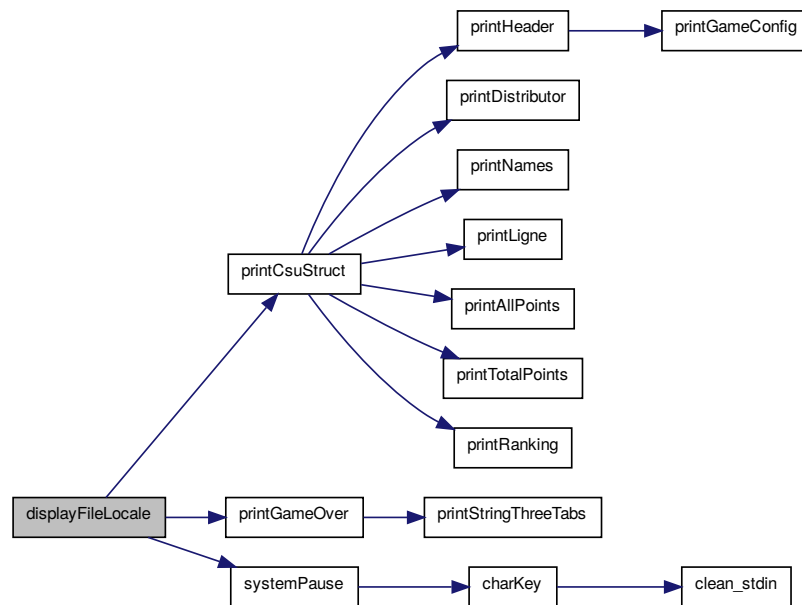
Here is the call graph for this function:



2.5.3.4 void displayFileLocale (char * file_name)

Display the file.

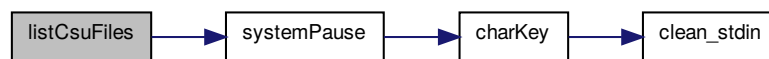
Here is the call graph for this function:



2.5.3.5 void listCsuFiles ()

List all the files with csu extension

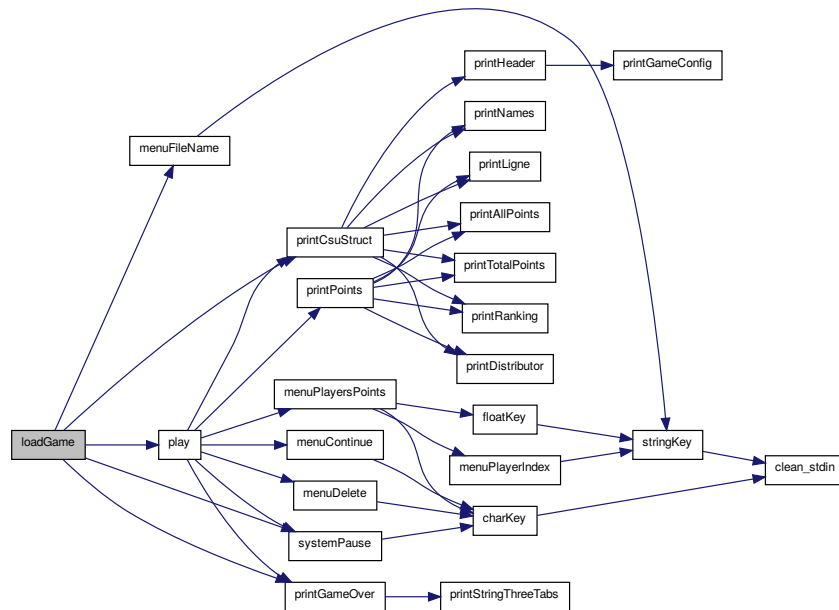
Here is the call graph for this function:



2.5.3.6 void loadGame ()

Load a game from a file and run the play function.

Here is the call graph for this function:



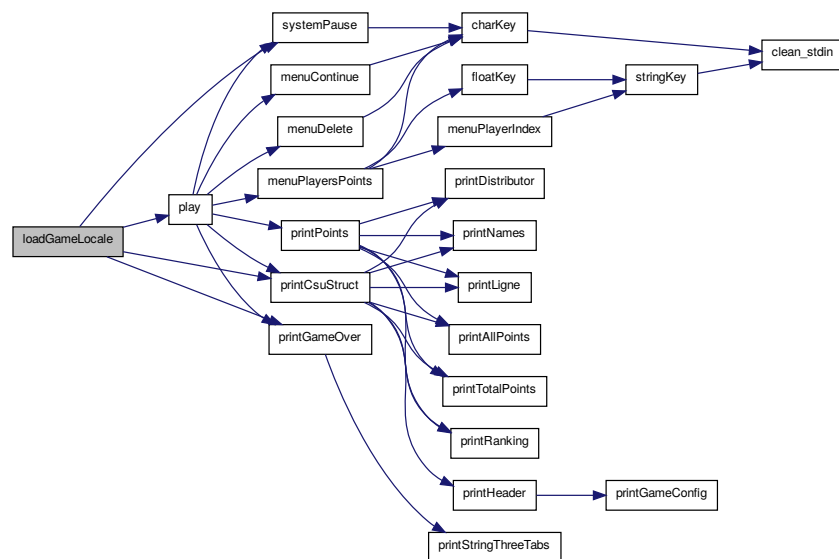
2.5.3.7 void loadGameLocale (char * file_name)

Load the file and run the play function

Parameters

in	file_name,the	filename
----	---------------	----------

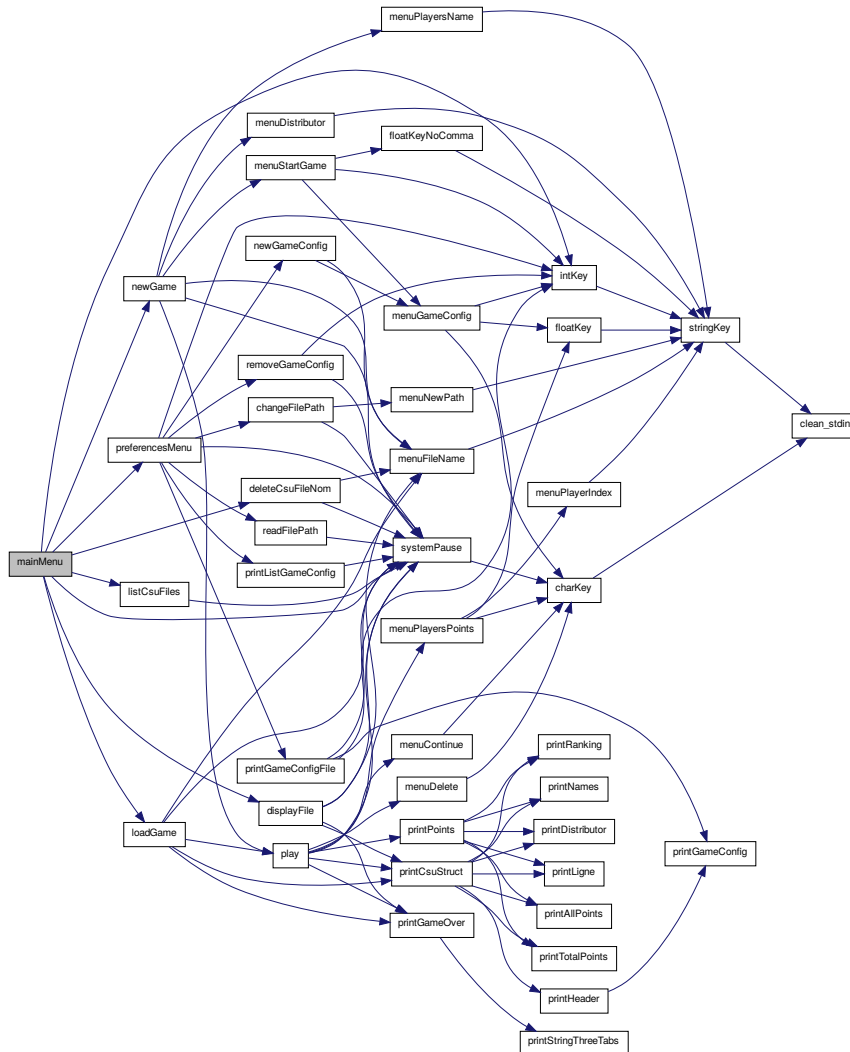
Here is the call graph for this function:



2.5.3.8 void mainMenu ()

Main menu of csuper.

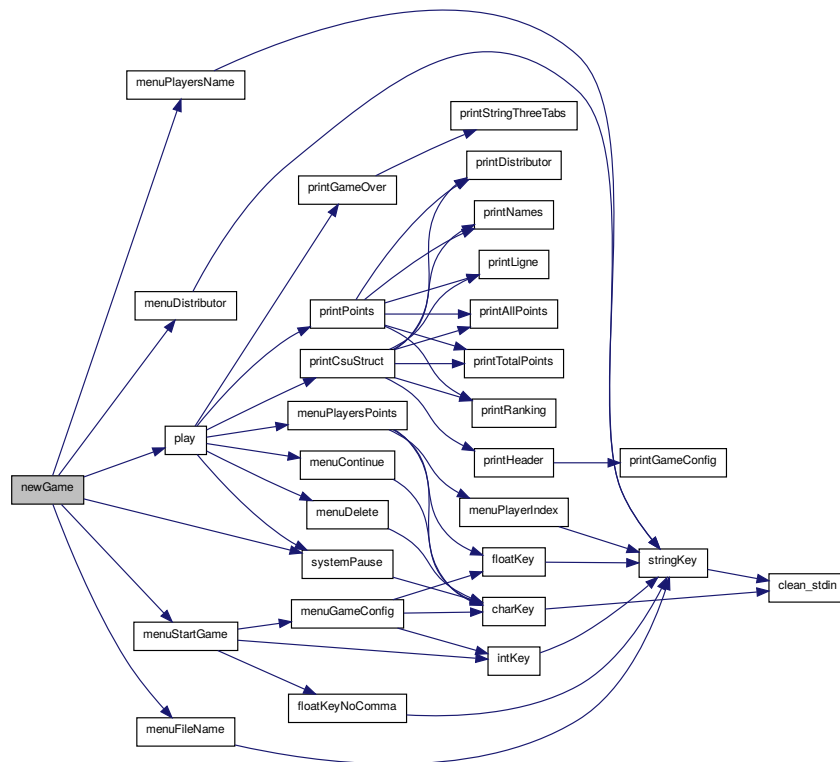
Here is the call graph for this function:



2.5.3.9 void newGame ()

Initialize a new game and run the play function.

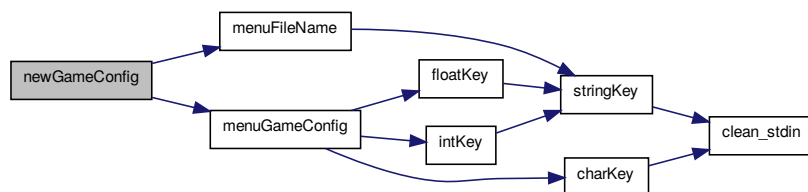
Here is the call graph for this function:



2.5.3.10 void newGameConfig ()

Add a new game configuration

Here is the call graph for this function:



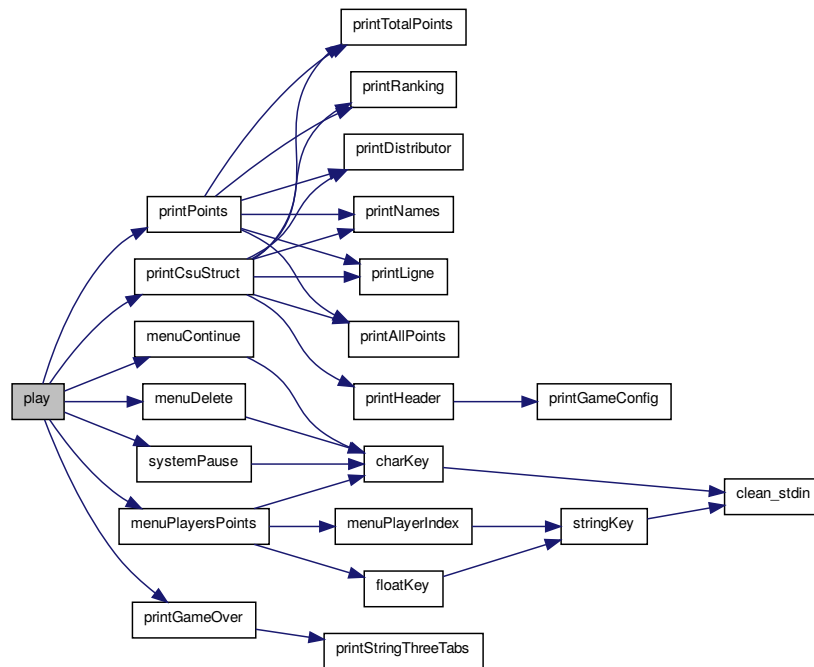
2.5.3.11 void play (csuStruct * ptr_csu_struct, char * file_name)

Count the points

Parameters

in, out	<i>*ptr_csu_struct</i>	a pointer on a csu structure
in	<i>*file_name</i>	the filename

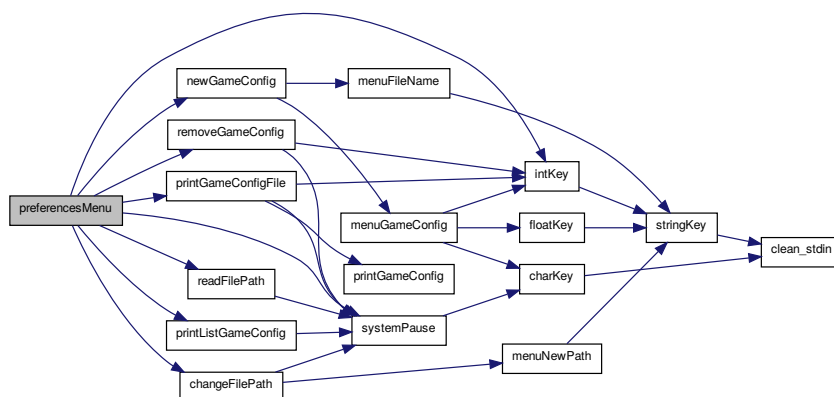
Here is the call graph for this function:



2.5.3.12 void preferencesMenu ()

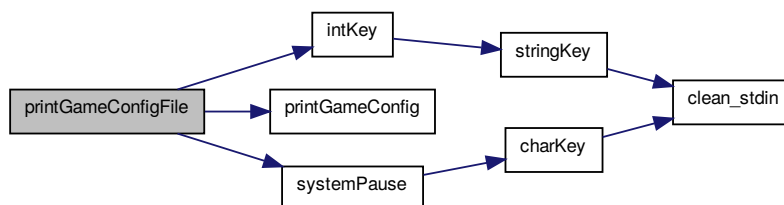
Preferences menu of csuper.

Here is the call graph for this function:



2.5.3.13 void printGameConfigFile ()

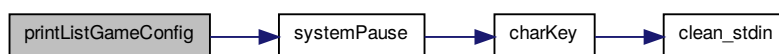
Here is the call graph for this function:



2.5.3.14 void printListGameConfig ()

Print the list of game configuration

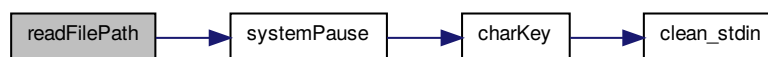
Here is the call graph for this function:



2.5.3.15 void readFilePath ()

Read the file path and display it.

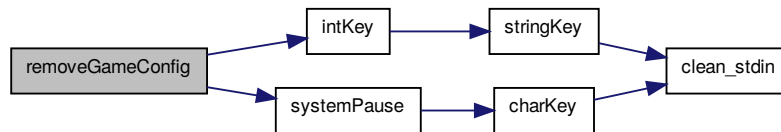
Here is the call graph for this function:



2.5.3.16 void removeGameConfig ()

Ask and remove a game configuration

Here is the call graph for this function:



2.6 keyboarding.c File Reference

Function of keyboarding.

```
#include "keyboarding.h"
```

Functions

- void `clean_stdin` (void)
- char * `stringKey` (char *string, int nb_char_plus_one)
- void `intKey` (int *nb)
- void `floatKey` (float *nb)
- void `floatKeyNoComma` (float *nb)
- void `doubleKey` (double *nb)
- char * `charKey` (char *c)
- void `systemPause` ()

2.6.1 Detailed Description

Function of keyboarding.

Author

Remi BERTHO

Date

17/04/14

Version

2.2.0

2.6.2 Function Documentation

2.6.2.1 char * charKey (char * c)

Do a keyboarding of a character.

Parameters

in, out	*c	the character
---------	----	---------------

Here is the call graph for this function:



2.6.2.2 void clean_stdin (void)

Clean the buffer of stdin.

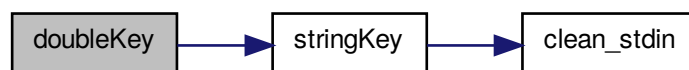
2.6.2.3 void doubleKey (double * nb)

Do a keyboarding of a double, put 0 if the keyboarding is not an double.

Parameters

in, out	*nb	the number
---------	-----	------------

Here is the call graph for this function:



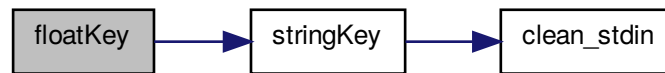
2.6.2.4 void floatKey (float * nb)

Do a keyboarding of an float, put 0 if the keyboarding is not an float.

Parameters

in, out	*nb	the number
---------	-----	------------

Here is the call graph for this function:



2.6.2.5 void floatKeyNoComma (float * *nb*)

Do a keyboarding of an float with no comma, put 0 if the keyboarding is not an float.

Parameters

<i>in, out</i>	<i>*nb</i>	the number
----------------	------------	------------

Here is the call graph for this function:



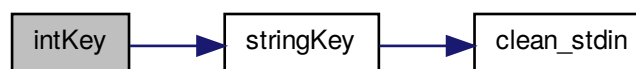
2.6.2.6 void * intKey (int * *nb*)

Do a keyboarding of an int, put 0 if the keyboarding is not an int.

Parameters

<i>in, out</i>	<i>*nb</i>	the number
----------------	------------	------------

Here is the call graph for this function:



2.6.2.7 char * stringKey (char * *string*, int *nb_char_plus_one*)

Do a keyboarding of a string with `nb-char_plus_one` minus 1 characters

Parameters

in, out	<i>*string</i>	a string
in	<i>nb_char_plus_one</i>	the number of characters that the function will read plus one

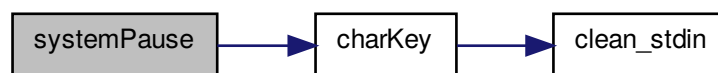
Here is the call graph for this function:



2.6.2.8 void systemPause ()

Ask the user do press enter to continue.

Here is the call graph for this function:



2.7 keyboarding.h File Reference

Function of keyboarding.

```
#include "csuper.h"
#include <locale.h>
#include <string.h>
#include <math.h>
#include "main.h"
```

Macros

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

Functions

- void `clean_stdin` (void)

- char * [stringKey](#) (char *string, int nb_char_plus_one)
- void [intKey](#) (int *nb)
- void [floatKey](#) (float *nb)
- void [floatKeyNoComma](#) (float *nb)
- void [doubleKey](#) (double *nb)
- char * [charKey](#) (char *c)
- void [systemPause](#) ()

2.7.1 Detailed Description

Function of keyboarding.

Author

Remi BERTHO

Date

17/04/14

Version

2.2.0

2.7.2 Macro Definition Documentation

2.7.2.1 #define NB_CHARACTER_DOUB 309

Define the number of characters that need a double

2.7.2.2 #define NB_CHARACTER_FLOAT 39

Define the number of characters that need a float.

2.7.2.3 #define NB_CHARACTER_INT 12

Define the number of characters that need an int.

2.7.3 Function Documentation

2.7.3.1 char* charKey (char * c)

Do a keyboarding of a character.

Parameters

in, out	*c	the character
---------	----	---------------

Here is the call graph for this function:



2.7.3.2 void clean_stdin (void)

Clean the buffer of stdin.

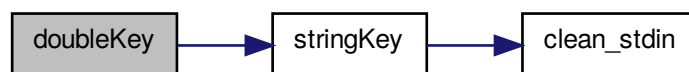
2.7.3.3 void doubleKey (double * nb)

Do a keyboarding of a double, put 0 if the keyboarding is not an double.

Parameters

in, out	*nb	the number
---------	-----	------------

Here is the call graph for this function:



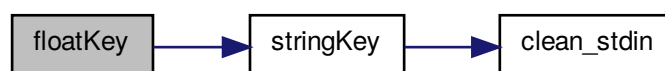
2.7.3.4 void floatKey (float * nb)

Do a keyboarding of an float, put 0 if the keyboarding is not an float.

Parameters

in, out	*nb	the number
---------	-----	------------

Here is the call graph for this function:



2.7.3.5 void floatKeyNoComma (float * *nb*)

Do a keyboarding of an float with no comma, put 0 if the keyboarding is not an float.

Parameters

<i>in, out</i>	<i>*nb</i>	the number
----------------	------------	------------

Here is the call graph for this function:



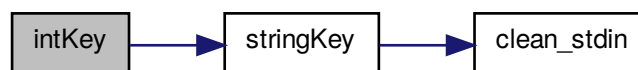
2.7.3.6 void intKey (int * *nb*)

Do a keyboarding of an int, put 0 if the keyboarding is not an int.

Parameters

<i>in, out</i>	<i>*nb</i>	the number
----------------	------------	------------

Here is the call graph for this function:



2.7.3.7 char* stringKey (char * *string*, int *nb_char_plus_one*)

Do a keyboarding of a string with *nb-char_plus_one* minus 1 characters

Parameters

<i>in, out</i>	<i>*string</i>	a string
<i>in</i>	<i>nb_char_plus_one</i>	the number of characters that the function will read plus one

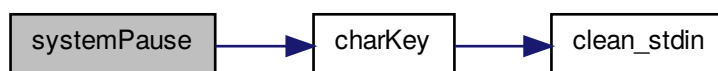
Here is the call graph for this function:



2.7.3.8 void systemPause ()

Ask the user do press enter to continue.

Here is the call graph for this function:



2.8 main.c File Reference

Begin csuper.

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

Functions

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

2.8.1 Detailed Description

Begin csuper.

Author

Remi BERTHO

Date

17/04/14

Version

2.2.0


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

Functions

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

2.9.1 Detailed Description

Begin csuper.

Author

Remi BERTHO

Date

17/04/14

Version

2.2.0

2.9.2 Function Documentation

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

Begin csuper.

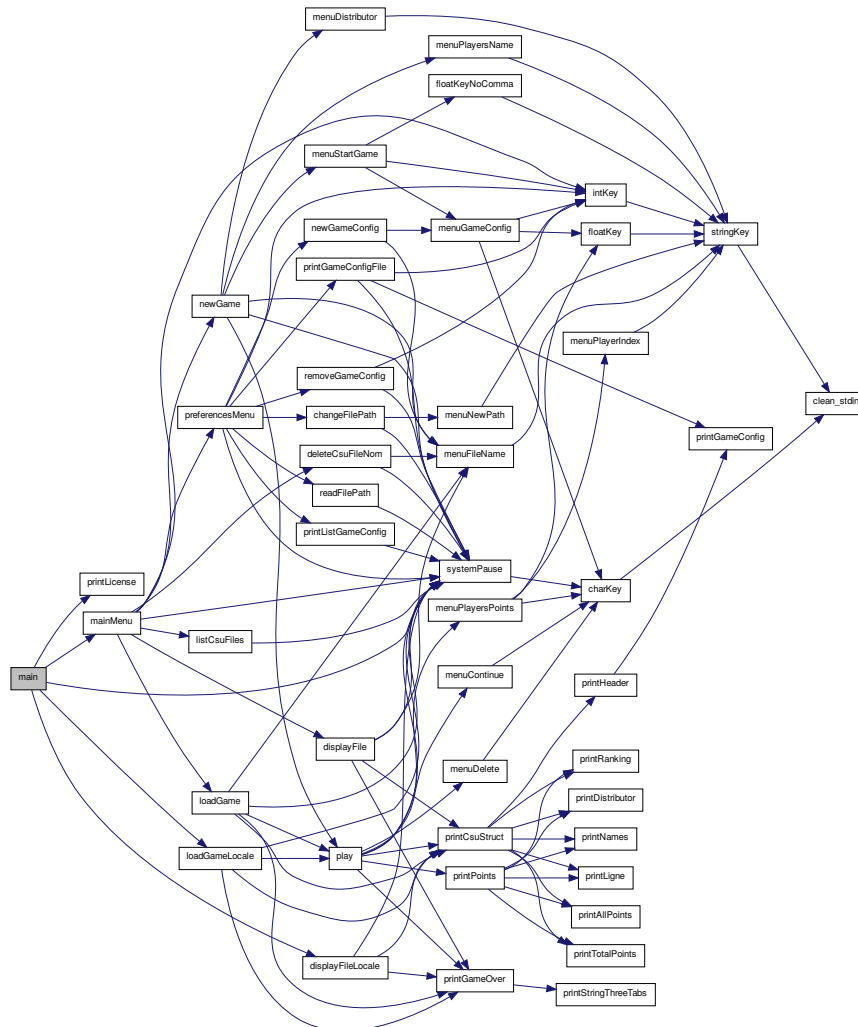
Parameters

in	<i>argc</i>	the number of argument.
in	<i>argv</i>	the array of argument.

Returns

EXIT_SUCCESS if everything is OK

Here is the call graph for this function:



2.10 menu.c File Reference

Menu functions.

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

Functions

- char * [menuFileName](#) (char file_name[SIZE_MAX_FILE_NAME])
- void [menuStartGame](#) (float *ptr_nb_player, game_config *ptr_config)
- void [menuGameConfig](#) (game_config *ptr_config)
- void [menuDistributor](#) (char *distributor_name)
- void [menuPlayersName](#) (csuStruct *ptr_csu_struct)

- void [menuPlayersPoints](#) (csuStruct *ptr_csu_struct)
- int [menuPlayerIndex](#) (csuStruct *ptr_csu_struct)
- int [menuContinue](#) ()
- int [menuDelete](#) ()
- void [menuNewPath](#) (char *new_path)

2.10.1 Detailed Description

Menu functions.

Author

Remi BERTHO

Date

17/04/14

Version

2.2.0

2.10.2 Function Documentation

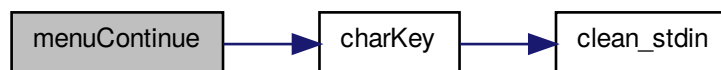
2.10.2.1 int menuContinue ()

Ask if we want to continue3

Returns

TRUE if we want to continue, FALSE otherwise

Here is the call graph for this function:



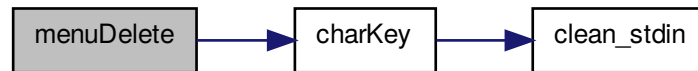
2.10.2.2 int menuDelete ()

Ask if we want to delete a file.

Returns

TRUE if we want to deleted the file, FALSE otherwise

Here is the call graph for this function:



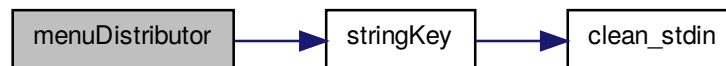
2.10.2.3 void menuDistributor (char * distributor_name)

Ask and save the distributor.

Parameters

<code>in, out</code>	<code>*distributor_ - name</code>	the name of the distributor.
----------------------	---------------------------------------	------------------------------

Here is the call graph for this function:



2.10.2.4 char * menuFileName (char file_name[SIZE_MAX_FILE_NAME])

Ask and save the filename.

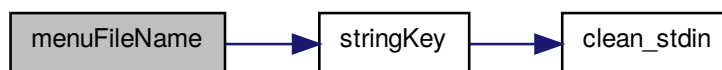
Parameters

<code>in, out</code>	<code>file_name</code>	the filename
----------------------	------------------------	--------------

Returns

filename

Here is the call graph for this function:



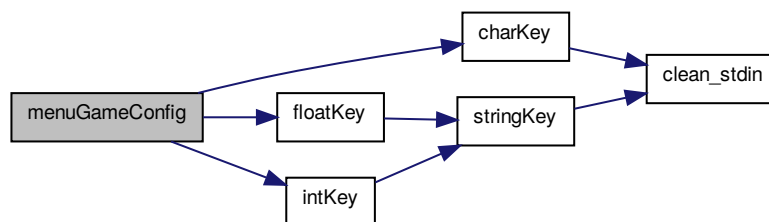
2.10.2.5 void menuGameConfig (game_config * ptr_config)

Ask and save a game configuration.

Parameters

in, out	<i>ptr_config</i>	a game configuration
---------	-------------------	----------------------

Here is the call graph for this function:



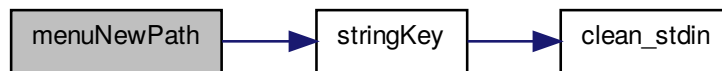
2.10.2.6 void menuNewPath (char * new_path)

Ask and save the new path

Parameters

in, out	<i>*new_path</i>	the new path
---------	------------------	--------------

Here is the call graph for this function:



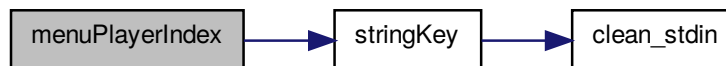
2.10.2.7 `int menuPlayerIndex (csuStruct * ptr_csu_struct)`

Ask a player name and save his index

Parameters

<code>in, out</code>	<code>*ptr_csu_struct</code>	a pointer on a csu structure
----------------------	------------------------------	------------------------------

Here is the call graph for this function:



2.10.2.8 `void menuPlayersName (csuStruct * ptr_csu_struct)`

Ask and save the name of the players.

Parameters

<code>in, out</code>	<code>*ptr_csu_struct</code>	a pointer on a csu structure
----------------------	------------------------------	------------------------------

Here is the call graph for this function:



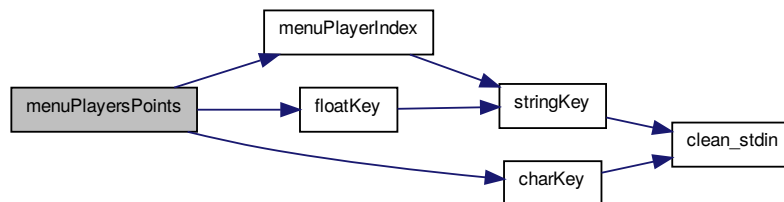
2.10.2.9 `void menuPlayersPoints (csuStruct * ptr_csu_struct)`

Begin a new turn, ask and save the new points and finish the turn

Parameters

in, out	<i>*ptr_csu_struct</i>	a pointer on a csu structure
---------	------------------------	------------------------------

Here is the call graph for this function:



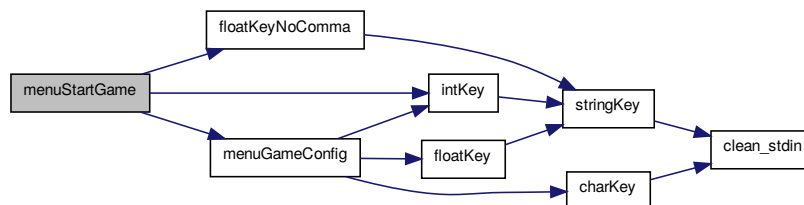
2.10.2.10 void menuStartGame (float * ptr_nb_player, game_config * ptr_config)

Ask and save the number of player and all the informations of a game configuration.

Parameters

in, out	<i>ptr_nb_player</i>	the number of player
in, out	<i>ptr_config</i>	the game configuration

Here is the call graph for this function:



2.11 menu.h File Reference

Menu functions.

```
#include "keyboarding.h"
#include "main.h"
#include "csuper.h"
```

Functions

- char * [menuFileName](#) (char file_name[SIZE_MAX_FILE_NAME])
- void [menuStartGame](#) (float *ptr_nb_player, game_config *ptr_config)
- void [menuGameConfig](#) (game_config *ptr_config)

- void `menuDistributor` (char *distributor_name)
- void `menuPlayersName` (csuStruct *ptr_csu_struct)
- void `menuPlayersPoints` (csuStruct *ptr_csu_struct)
- int `menuPlayerIndex` (csuStruct *ptr_csu_struct)
- int `menuContinue` ()
- int `menuDelete` ()
- void `menuNewPath` (char *new_path)

2.11.1 Detailed Description

Menu functions.

Author

Remi BERTHO

Date

17/04/14

Version

2.2.0

2.11.2 Function Documentation

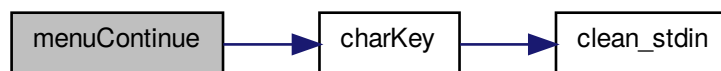
2.11.2.1 int menuContinue ()

Ask if we want to continue3

Returns

TRUE if we want to continue, FALSE otherwise

Here is the call graph for this function:



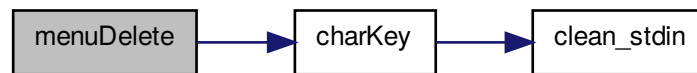
2.11.2.2 int menuDelete ()

Ask if we want to delete a file.

Returns

TRUE if we want to deleted the file, FALSE otherwise

Here is the call graph for this function:

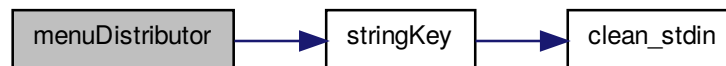
**2.11.2.3 void menuDistributor (char * distributor_name)**

Ask and save the distributor.

Parameters

<code>in, out</code>	<code>*distributor_ - name</code>	the name of the distributor.
----------------------	---------------------------------------	------------------------------

Here is the call graph for this function:

**2.11.2.4 char* menuFileName (char file_name[SIZE_MAX_FILE_NAME])**

Ask and save the filename.

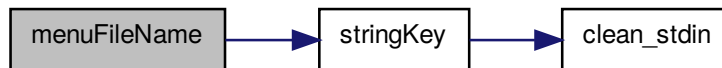
Parameters

<code>in, out</code>	<code>file_name</code>	the filename
----------------------	------------------------	--------------

Returns

filename

Here is the call graph for this function:



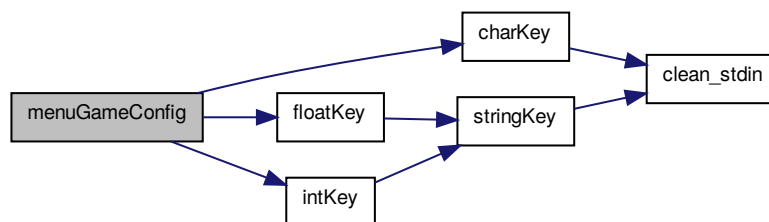
2.11.2.5 void menuGameConfig (game_config * ptr_config)

Ask and save a game configuration.

Parameters

in, out	<i>ptr_config</i>	a game configuration
---------	-------------------	----------------------

Here is the call graph for this function:



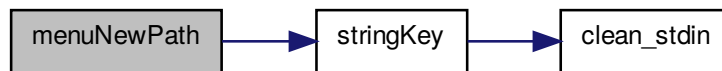
2.11.2.6 void menuNewPath (char * new_path)

Ask and save the new path

Parameters

in, out	<i>*new_path</i>	the new path
---------	------------------	--------------

Here is the call graph for this function:



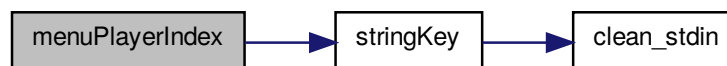
2.11.2.7 `int menuPlayerIndex (csuStruct * ptr_csu_struct)`

Ask a player name and save his index

Parameters

<code>in, out</code>	<code>*ptr_csu_struct</code>	a pointer on a csu structure
----------------------	------------------------------	------------------------------

Here is the call graph for this function:



2.11.2.8 `void menuPlayersName (csuStruct * ptr_csu_struct)`

Ask and save the name of the players.

Parameters

<code>in, out</code>	<code>*ptr_csu_struct</code>	a pointer on a csu structure
----------------------	------------------------------	------------------------------

Here is the call graph for this function:



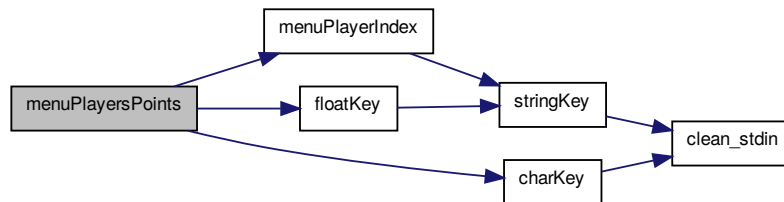
2.11.2.9 `void menuPlayersPoints (csuStruct * ptr_csu_struct)`

Begin a new turn, ask and save the new points and finish the turn

Parameters

<i>in, out</i>	<i>*ptr_csu_struct</i>	a pointer on a csu structure
----------------	------------------------	------------------------------

Here is the call graph for this function:



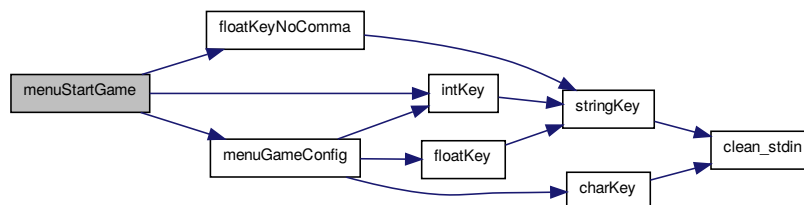
2.11.2.10 void menuStartGame (float * *ptr_nb_player*, game_config * *ptr_config*)

Ask and save the number of player and all the informations of a game configuration.

Parameters

<i>in, out</i>	<i>ptr_nb_player</i>	the number of player
<i>in, out</i>	<i>ptr_config</i>	the game configuration

Here is the call graph for this function:



Index

- - csuper.h, 3
- backMainMenu
 - interface.h, 23
- CSUPER
 - csuper.h, 3
- changeFilePath
 - interface.c, 14
 - interface.h, 23
- charKey
 - keyboarding.c, 32
 - keyboarding.h, 36
- clean_stdin
 - keyboarding.c, 33
 - keyboarding.h, 37
- csuper.h, 3
 - _, 3
 - CSUPER, 3
- deleteFile
 - interface.h, 23
- deleteCsuFileNom
 - interface.c, 14
 - interface.h, 24
- display.c, 4
 - printAllPoints, 4
 - printCsuStruct, 4
 - printDistributor, 5
 - printGameConfig, 5
 - printGameOver, 5
 - printHeader, 6
 - printLicense, 6
 - printLigne, 6
 - printNames, 6
 - printPoints, 7
 - printRanking, 7
 - printStringThreeTabs, 7
 - printTotalPoints, 8
- display.h, 8
 - printAllPoints, 9
 - printCsuStruct, 10
 - printDistributor, 10
 - printGameConfig, 10
 - printGameOver, 11
 - printHeader, 11
 - printLicense, 11
 - printLigne, 11
 - printNames, 12
 - printPoints, 12
 - printRanking, 12
 - printStringThreeTabs, 13
 - printTotalPoints, 13
- displayFile
 - interface.c, 14
 - interface.h, 24
- displayFileLocale
 - interface.c, 15
 - interface.h, 25
- doubleKey
 - keyboarding.c, 33
 - keyboarding.h, 37
- easterEggs
 - interface.h, 23
- easterEggs2
 - interface.h, 23
- floatKey
 - keyboarding.c, 33
 - keyboarding.h, 37
- floatKeyNoComma
 - keyboarding.c, 34
 - keyboarding.h, 38
- intKey
 - keyboarding.c, 34
 - keyboarding.h, 38
- interface.h
 - backMainMenu, 23
 - deleteFile, 23
 - easterEggs, 23
 - easterEggs2, 23
 - listFile, 23
 - loadMatch, 23
 - newGameConf, 23
 - newMatch, 23
 - newPath, 23
 - pref, 23
 - printFile, 23
 - printGameConf, 23
 - printListGameConf, 23
 - quit, 23
 - readPath, 23
 - removeGameConf, 23
- interface.c, 13
 - changeFilePath, 14
 - deleteCsuFileNom, 14
 - displayFile, 14

- displayFileLocale, 15
- listCsuFiles, 16
- loadGame, 16
- loadGameLocale, 17
- mainMenu, 17
- newGame, 18
- newGameConfig, 19
- play, 19
- preferencesMenu, 20
- printGameConfigFile, 20
- printListGameConfig, 21
- readFilePath, 21
- removeGameConfig, 21
- interface.h, 22
 - changeFilePath, 23
 - deleteCsuFileNom, 24
 - displayFile, 24
 - displayFileLocale, 25
 - listCsuFiles, 26
 - loadGame, 26
 - loadGameLocale, 27
 - MainMenu, 23
 - mainMenu, 27
 - newGame, 28
 - newGameConfig, 29
 - play, 29
 - PreferencesMenu, 23
 - preferencesMenu, 30
 - printGameConfigFile, 30
 - printListGameConfig, 31
 - readFilePath, 31
 - removeGameConfig, 31
- keyboarding.c, 32
 - charKey, 32
 - clean_stdin, 33
 - doubleKey, 33
 - floatKey, 33
 - floatKeyNoComma, 34
 - intKey, 34
 - stringKey, 34
 - systemPause, 35
- keyboarding.h, 35
 - charKey, 36
 - clean_stdin, 37
 - doubleKey, 37
 - floatKey, 37
 - floatKeyNoComma, 38
 - intKey, 38
 - NB_CARACT_DOUB, 36
 - NB_CARACT_FLOAT, 36
 - NB_CARACT_INT, 36
 - stringKey, 38
 - systemPause, 39
- listFile
 - interface.h, 23
- listCsuFiles
 - interface.c, 16
 - interface.h, 26
- loadMatch
 - interface.h, 23
- loadGame
 - interface.c, 16
 - interface.h, 26
- loadGameLocale
 - interface.c, 17
 - interface.h, 27
- main
 - main.c, 40
 - main.h, 41
- main.c, 39
 - main, 40
- main.h, 40
 - main, 41
- MainMenu
 - interface.h, 23
- mainMenu
 - interface.c, 17
 - interface.h, 27
- menu.c, 42
 - menuContinue, 43
 - menuDelete, 43
 - menuDistributor, 44
 - menuFileName, 44
 - menuGameConfig, 45
 - menuNewPath, 45
 - menuPlayerIndex, 46
 - menuPlayersName, 46
 - menuPlayersPoints, 46
 - menuStartGame, 47
- menu.h, 47
 - menuContinue, 48
 - menuDelete, 48
 - menuDistributor, 49
 - menuFileName, 49
 - menuGameConfig, 50
 - menuNewPath, 50
 - menuPlayerIndex, 51
 - menuPlayersName, 51
 - menuPlayersPoints, 51
 - menuStartGame, 52
- menuContinue
 - menu.c, 43
 - menu.h, 48
- menuDelete
 - menu.c, 43
 - menu.h, 48
- menuDistributor
 - menu.c, 44
 - menu.h, 49
- menuFileName
 - menu.c, 44
 - menu.h, 49
- menuGameConfig
 - menu.c, 45
 - menu.h, 50

- menuNewPath
 - menu.c, [45](#)
 - menu.h, [50](#)
- menuPlayerIndex
 - menu.c, [46](#)
 - menu.h, [51](#)
- menuPlayersName
 - menu.c, [46](#)
 - menu.h, [51](#)
- menuPlayersPoints
 - menu.c, [46](#)
 - menu.h, [51](#)
- menuStartGame
 - menu.c, [47](#)
 - menu.h, [52](#)
- NB_CARACT_DOUB
 - keyboarding.h, [36](#)
- NB_CARACT_FLOAT
 - keyboarding.h, [36](#)
- NB_CARACT_INT
 - keyboarding.h, [36](#)
- newGameConf
 - interface.h, [23](#)
- newMatch
 - interface.h, [23](#)
- newPath
 - interface.h, [23](#)
- newGame
 - interface.c, [18](#)
 - interface.h, [28](#)
- newGameConfig
 - interface.c, [19](#)
 - interface.h, [29](#)
- play
 - interface.c, [19](#)
 - interface.h, [29](#)
- pref
 - interface.h, [23](#)
- PreferencesMenu
 - interface.h, [23](#)
- preferencesMenu
 - interface.c, [20](#)
 - interface.h, [30](#)
- printFile
 - interface.h, [23](#)
- printGameConf
 - interface.h, [23](#)
- printListGameConf
 - interface.h, [23](#)
- printAllPoints
 - display.c, [4](#)
 - display.h, [9](#)
- printCsuStruct
 - display.c, [4](#)
 - display.h, [10](#)
- printDistributor
 - display.c, [5](#)
- display.h, [10](#)
- printGameConfig
 - display.c, [5](#)
 - display.h, [10](#)
- printGameConfigFile
 - interface.c, [20](#)
 - interface.h, [30](#)
- printGameOver
 - display.c, [5](#)
 - display.h, [11](#)
- printHeader
 - display.c, [6](#)
 - display.h, [11](#)
- printLicense
 - display.c, [6](#)
 - display.h, [11](#)
- printLigne
 - display.c, [6](#)
 - display.h, [11](#)
- printListGameConfig
 - interface.c, [21](#)
 - interface.h, [31](#)
- printNames
 - display.c, [6](#)
 - display.h, [12](#)
- printPoints
 - display.c, [7](#)
 - display.h, [12](#)
- printRanking
 - display.c, [7](#)
 - display.h, [12](#)
- printStringThreeTabs
 - display.c, [7](#)
 - display.h, [13](#)
- printTotalPoints
 - display.c, [8](#)
 - display.h, [13](#)
- quit
 - interface.h, [23](#)
- readPath
 - interface.h, [23](#)
- readFilePath
 - interface.c, [21](#)
 - interface.h, [31](#)
- removeGameConf
 - interface.h, [23](#)
- removeGameConfig
 - interface.c, [21](#)
 - interface.h, [31](#)
- stringKey
 - keyboarding.c, [34](#)
 - keyboarding.h, [38](#)
- systemPause
 - keyboarding.c, [35](#)
 - keyboarding.h, [39](#)