

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

## 2.1.9

Generated by Doxygen 1.8.6

Tue Apr 22 2014 09:59:26



# Contents

<b>1</b>	<b>File Index</b>	<b>1</b>
1.1	File List	1
<b>2</b>	<b>File Documentation</b>	<b>3</b>
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	color	4
2.2.2.2	printAllPoints	4
2.2.2.3	printCsuStruct	5
2.2.2.4	printDistributor	5
2.2.2.5	printGameConfig	6
2.2.2.6	printGameOver	6
2.2.2.7	printHeader	6
2.2.2.8	printLicense	7
2.2.2.9	printLigne	7
2.2.2.10	printNames	7
2.2.2.11	printPoints	7
2.2.2.12	printRanking	8
2.2.2.13	printSpecial	8
2.2.2.14	printStringThreeTabs	9
2.2.2.15	printTotalPoints	9
2.3	display.h File Reference	9
2.3.1	Detailed Description	10
2.3.2	Enumeration Type Documentation	10
2.3.2.1	backgroundColor	10

2.3.2.2	foregroundColor	11
2.3.2.3	Writing	11
2.3.3	Function Documentation	11
2.3.3.1	color	11
2.3.3.2	printAllPoints	11
2.3.3.3	printCsuStruct	12
2.3.3.4	printDistributor	12
2.3.3.5	printGameConfig	13
2.3.3.6	printGameOver	14
2.3.3.7	printHeader	14
2.3.3.8	printLicense	14
2.3.3.9	printLigne	14
2.3.3.10	printNames	15
2.3.3.11	printPoints	15
2.3.3.12	printRanking	16
2.3.3.13	printSpecial	16
2.3.3.14	printStringThreeTabs	16
2.3.3.15	printTotalPoints	16
2.4	interface.c File Reference	17
2.4.1	Detailed Description	17
2.4.2	Function Documentation	18
2.4.2.1	changeFilePath	18
2.4.2.2	deleteCsuFileNom	18
2.4.2.3	displayFile	18
2.4.2.4	displayFileLocale	19
2.4.2.5	listCsuFiles	20
2.4.2.6	loadGame	20
2.4.2.7	loadGameLocale	20
2.4.2.8	mainMenu	21
2.4.2.9	newGame	22
2.4.2.10	newGameConfig	23
2.4.2.11	play	23
2.4.2.12	preferencesMenu	24
2.4.2.13	printGameConfigFile	25
2.4.2.14	printListGameConfig	25
2.4.2.15	readFilePath	25
2.4.2.16	removeGameConfig	25
2.5	interface.h File Reference	26
2.5.1	Detailed Description	27
2.5.2	Enumeration Type Documentation	27

2.5.2.1	MainMenu	27
2.5.2.2	PreferencesMenu	27
2.5.3	Function Documentation	27
2.5.3.1	changeFilePath	27
2.5.3.2	deleteCsuFileNom	28
2.5.3.3	displayFile	28
2.5.3.4	displayFileLocale	29
2.5.3.5	listCsuFiles	30
2.5.3.6	loadGame	30
2.5.3.7	loadGameLocale	30
2.5.3.8	mainMenu	31
2.5.3.9	newGame	32
2.5.3.10	newGameConfig	33
2.5.3.11	play	33
2.5.3.12	preferencesMenu	34
2.5.3.13	printGameConfigFile	35
2.5.3.14	printListGameConfig	35
2.5.3.15	readFilePath	35
2.5.3.16	removeGameConfig	35
2.6	keyboarding.c File Reference	36
2.6.1	Detailed Description	36
2.6.2	Function Documentation	36
2.6.2.1	charKey	36
2.6.2.2	clean_stdin	37
2.6.2.3	doubleKey	37
2.6.2.4	floatKey	37
2.6.2.5	floatKeyNoComma	38
2.6.2.6	intKey	38
2.6.2.7	stringKey	38
2.6.2.8	systemPause	39
2.7	keyboarding.h File Reference	39
2.7.1	Detailed Description	40
2.7.2	Macro Definition Documentation	40
2.7.2.1	NB_CARACT_DOUB	40
2.7.2.2	NB_CARACT_FLOAT	40
2.7.2.3	NB_CARACT_INT	40
2.7.3	Function Documentation	40
2.7.3.1	charKey	40
2.7.3.2	clean_stdin	41
2.7.3.3	doubleKey	41

2.7.3.4	floatKey	41
2.7.3.5	floatKeyNoComma	42
2.7.3.6	intKey	42
2.7.3.7	stringKey	42
2.7.3.8	systemPause	43
2.8	main.c File Reference	43
2.8.1	Detailed Description	43
2.8.2	Function Documentation	44
2.8.2.1	main	44
2.9	main.h File Reference	44
2.9.1	Detailed Description	45
2.9.2	Function Documentation	45
2.9.2.1	main	45
2.10	menu.c File Reference	46
2.10.1	Detailed Description	47
2.10.2	Function Documentation	47
2.10.2.1	menuContinue	47
2.10.2.2	menuDelete	47
2.10.2.3	menuDistributor	48
2.10.2.4	menuFileName	48
2.10.2.5	menuGameConfig	49
2.10.2.6	menuNewPath	49
2.10.2.7	menuPlayerIndex	50
2.10.2.8	menuPlayersName	50
2.10.2.9	menuPlayersPoints	50
2.10.2.10	menuStartGame	51
2.11	menu.h File Reference	51
2.11.1	Detailed Description	52
2.11.2	Function Documentation	52
2.11.2.1	menuContinue	52
2.11.2.2	menuDelete	52
2.11.2.3	menuDistributor	53
2.11.2.4	menuFileName	53
2.11.2.5	menuGameConfig	54
2.11.2.6	menuNewPath	54
2.11.2.7	menuPlayerIndex	55
2.11.2.8	menuPlayersName	55
2.11.2.9	menuPlayersPoints	55
2.11.2.10	menuStartGame	56







# Chapter 1

## File Index

### 1.1 File List

Here is a list of all files with brief descriptions:

<a href="#">csuper.h</a>	Include of csuper . . . . .	3
<a href="#">display.c</a>	Display the games . . . . .	4
<a href="#">display.h</a>	Display the games . . . . .	9
<a href="#">interface.c</a>	Graphical interface . . . . .	17
<a href="#">interface.h</a>	Graphical interface . . . . .	26
<a href="#">keyboarding.c</a>	Function of keyboarding . . . . .	36
<a href="#">keyboarding.h</a>	Function of keyboarding . . . . .	39
<a href="#">main.c</a>	Begin csuper . . . . .	43
<a href="#">main.h</a>	Begin csuper . . . . .	44
<a href="#">menu.c</a>	Menu functions . . . . .	46
<a href="#">menu.h</a>	Menu functions . . . . .	51



## 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)
- void [printSpecial](#) (char \*string, int nb\_arg,...)
- void [color](#) (int color)

### 2.2.1 Detailed Description

Display the games.

#### Author

Remi BERTHO

#### Date

22/04/14

#### Version

2.2.0

### 2.2.2 Function Documentation

#### 2.2.2.1 void color ( int *color* )

Chose a color for printing. Work only under Unix, under Windows do nothing.

##### Parameters

<i>in</i>	<i>color</i>	the code of the color (see foregroundColor, backgroundColor, Writing)
-----------	--------------	---

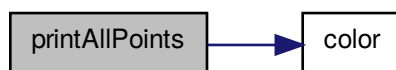
#### 2.2.2.2 void printAllPoints ( csuStruct \* *ptr\_csu\_struct* )

Print the scores of all player in each turn.

## Parameters

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

Here is the call graph for this function:



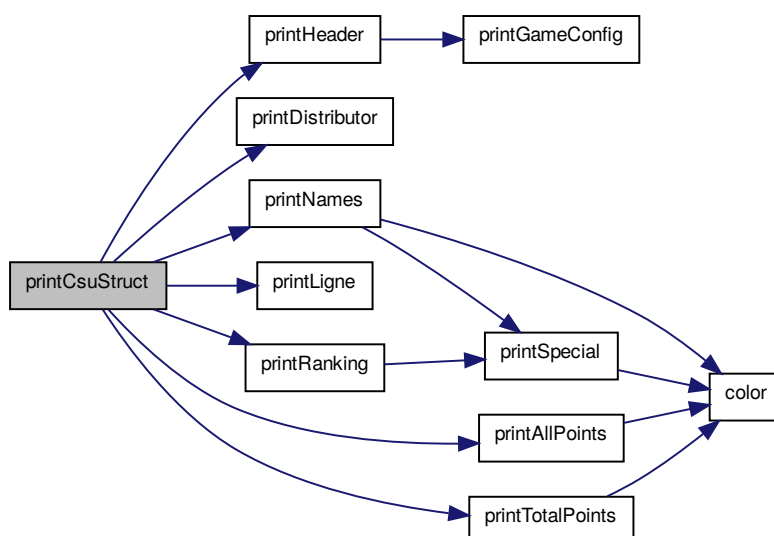
### 2.2.2.3 void printCsuStruct ( csuStruct \* ptr\_csu\_struct )

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:



### 2.2.2.4 void printDistributor ( csuStruct \* ptr\_csu\_struct )

Display the distributor.

## Parameters

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

## 2.2.2.5 void printGameConfig ( game\_config config )

Print the game config

## Parameters

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

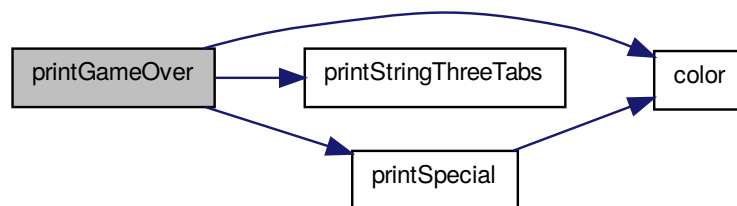
## 2.2.2.6 void printGameOver ( csuStruct \* ptr\_csu\_struct )

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:



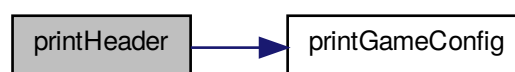
## 2.2.2.7 void printHeader ( csuStruct \* ptr\_csu\_struct )

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:



### 2.2.2.8 void printLicense ( )

Print the license.

### 2.2.2.9 void printLigne ( int *size\_line* )

Print a line of - after a tabulation.

Parameters

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

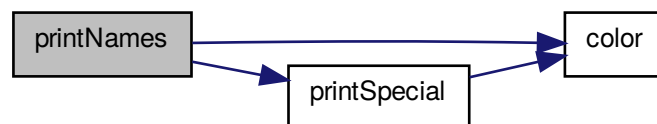
### 2.2.2.10 void printNames ( csuStruct \* *ptr\_csu\_struct*, int \* *ptr\_size\_line* )

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

Here is the call graph for this function:



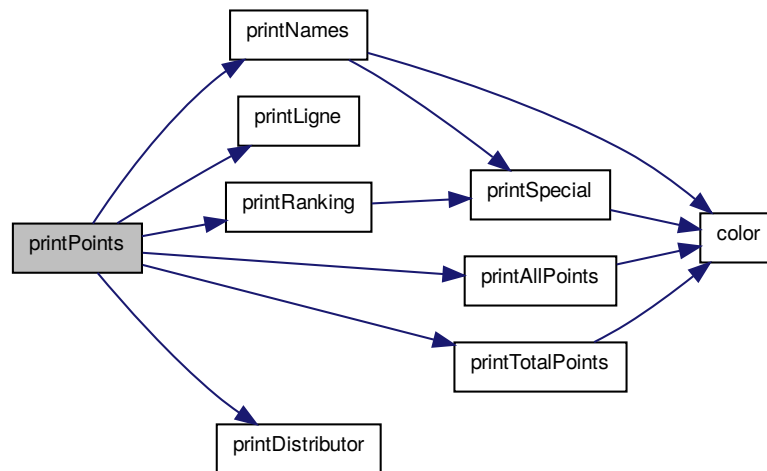
### 2.2.2.11 void printPoints ( csuStruct \* *ptr\_csu\_struct* )

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:



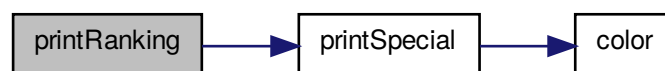
#### 2.2.2.12 void printRanking ( csuStruct \* ptr\_csu\_struct )

Print the rank of each player

##### Parameters

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

Here is the call graph for this function:



#### 2.2.2.13 void printSpecial ( char \* string, int nb\_arg, ... )

Print the string in stdin with special effect.

##### Parameters

in	<i>string</i>	a string to display
in	<i>nb_arg</i>	the number of argument



in	...	the argument
----	-----	--------------

Here is the call graph for this function:



#### 2.2.2.14 void printStringThreeTabs ( char \* *string* )

Print a string center into a space of three tabulations.

Parameters

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

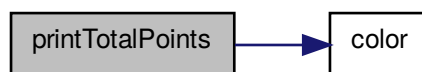
#### 2.2.2.15 void printTotalPoints ( csuStruct \* *ptr\_csu\_struct* )

Print the total score of the players.

Parameters

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

Here is the call graph for this function:



## 2.3 display.h File Reference

Display the games.

```

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

## Enumerations

- enum `foregroundColor` {  
    `foregroundBlack` =30, `foregroundRed` =31, `foregroundGreen` =32, `foregroundBrown` =33,  
    `foregroundBlue` =34, `foregroundMagenta` =35, `foregroundCyan` =36, `foregroundWhite` =37 }
- enum `backgroundColor` {  
    `backgroundBlack` =40, `backgroundRed` =41, `backgroundGreen` =42, `backgroundBrown` =43,  
    `backgroundBlue` =44, `backgroundMagenta` =45, `backgroundCyan` =46, `backgroundWhite` =47,  
    `backgroundDefault` =49 }
- enum `Writing` { `writingReset` =0, `writingBold` =1, `writingUnderline` =4 }

## 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)
- void `printSpecial` (char \*string, int nb\_arg,...)
- void `color` (int color)

### 2.3.1 Detailed Description

Display the games.

#### Author

Remi BERTHO

#### Date

22/04/14

#### Version

2.2.0

### 2.3.2 Enumeration Type Documentation

#### 2.3.2.1 enum `backgroundColor`

##### Enumerator

***backgroundBlack***

***backgroundRed***

***backgroundGreen***

***backgroundBrown***  
***backgroundBlue***  
***backgroundMagenta***  
***backgroundCyan***  
***backgroundWhite***  
***backgroundDefault***

#### 2.3.2.2 enum foregroundColor

Enumerator

***foregroundBlack***  
***foregroundRed***  
***foregroundGreen***  
***foregroundBrown***  
***foregroundBlue***  
***foregroundMagenta***  
***foregroundCyan***  
***foregroundWhite***

#### 2.3.2.3 enum Writing

Enumerator

***writingReset***  
***writingBold***  
***writingUnderline***

### 2.3.3 Function Documentation

#### 2.3.3.1 void color ( int *color* )

Chose a color for printing. Work only under Unix, under Windows do nothing.

Parameters

in	<i>color</i>	the code of the color (see foregroundColor, backgroundColor, Writing)
----	--------------	---

#### 2.3.3.2 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
----	-------------------------	------------------------------

Here is the call graph for this function:



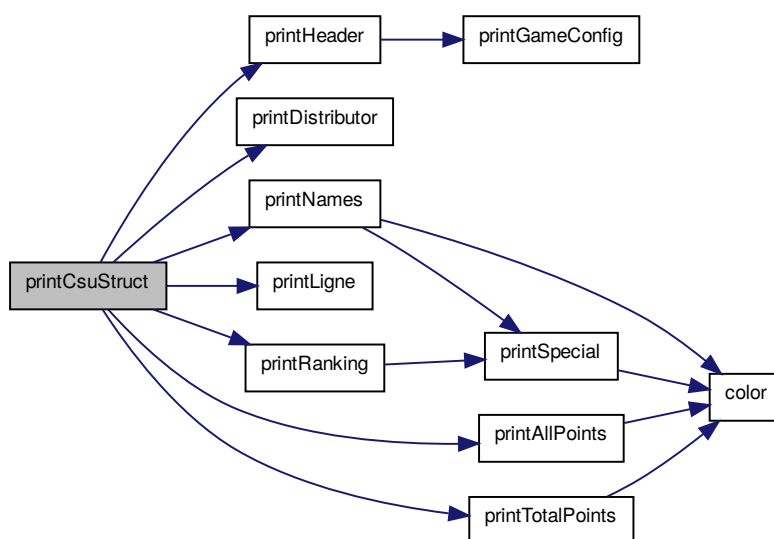
### 2.3.3.3 void printCsuStruct ( csuStruct \* ptr\_csu\_struct )

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:



### 2.3.3.4 void printDistributor ( csuStruct \* ptr\_csu\_struct )

Display the distributor.

#### Parameters

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

2.3.3.5 void printGameConfig ( game\_config *config* )

Print the game config

## Parameters

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

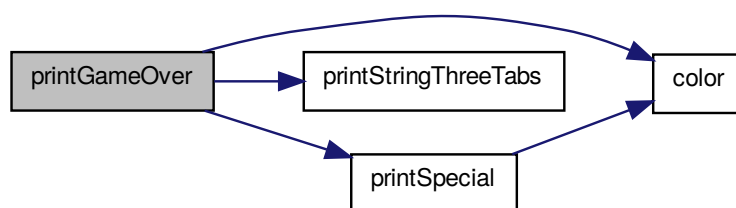
2.3.3.6 void printGameOver ( csuStruct \* *ptr\_csu\_struct* )

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:

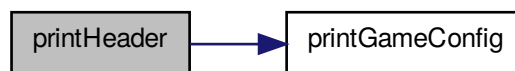
2.3.3.7 void printHeader ( csuStruct \* *ptr\_csu\_struct* )

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:



## 2.3.3.8 void printLicense ( )

Print the license.

2.3.3.9 void printLigne ( int *size\_line* )

Print a line of - after a tabulation.

## Parameters

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

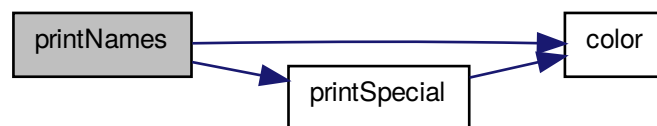
2.3.3.10 void printNames ( csuStruct \* *ptr\_csu\_struct*, int \* *ptr\_size\_line* )

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

Here is the call graph for this function:

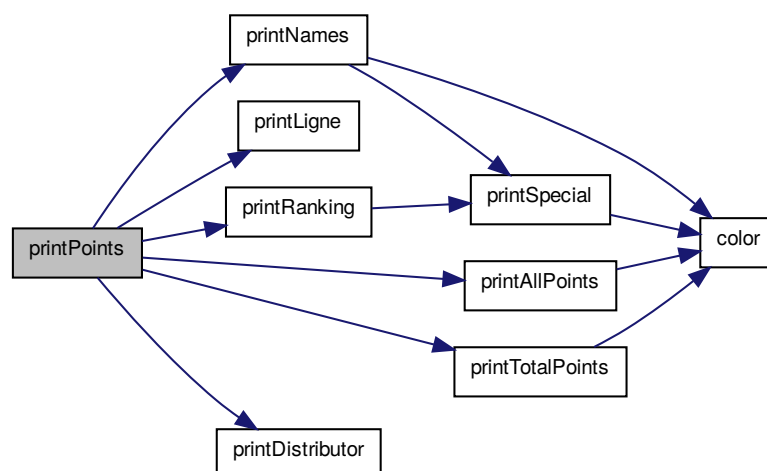
2.3.3.11 void printPoints ( csuStruct \* *ptr\_csu\_struct* )

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:



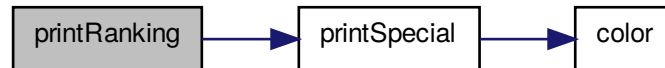
### 2.3.3.12 void printRanking ( csuStruct \* ptr\_csu\_struct )

Print the rank of each player

#### Parameters

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

Here is the call graph for this function:



### 2.3.3.13 void printSpecial ( char \* string, int nb\_arg, ... )

Print the string in stdin with special effect.

#### Parameters

in	string	a string to display
in	nb_arg	the number of argument
in	...	the argument

Here is the call graph for this function:



### 2.3.3.14 void printStringThreeTabs ( char \* string )

Print a string center into a space of three tabulations.

#### Parameters

in	*string	a string
----	---------	----------

### 2.3.3.15 void printTotalPoints ( csuStruct \* ptr\_csu\_struct )

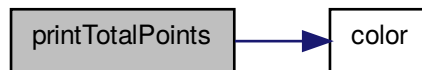
Print the total score of the players.



## Parameters

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

Here is the call graph for this function:



## 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

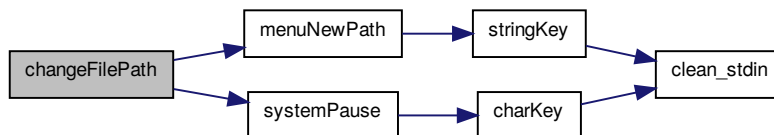
2.2.0

## 2.4.2 Function Documentation

### 2.4.2.1 void changeFilePath ( )

Change the path which the file are saved.

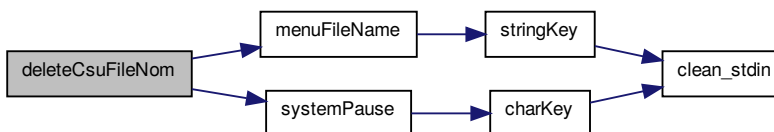
Here is the call graph for this function:



### 2.4.2.2 void deleteCsuFileNom ( )

Ask a filename and deleted the file.

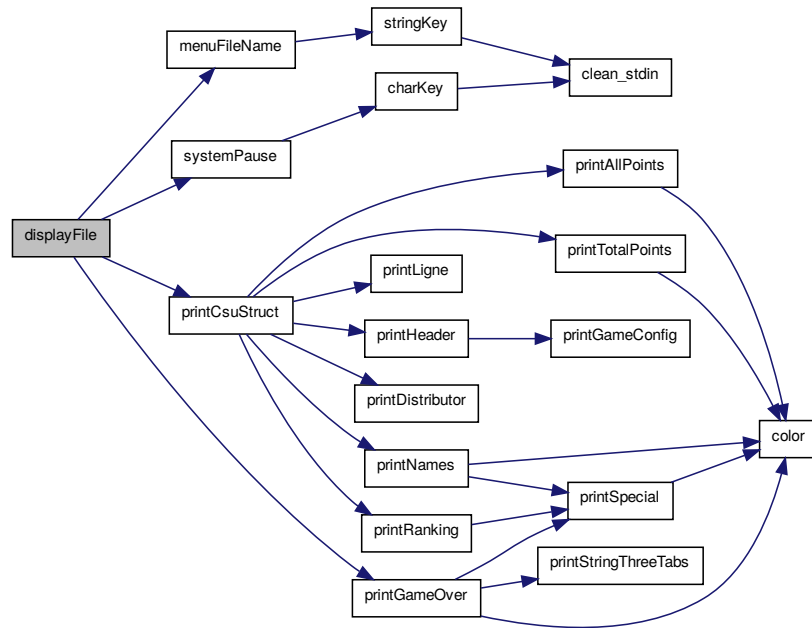
Here is the call graph for this function:



### 2.4.2.3 void displayFile ( )

Ask a filename and display it.

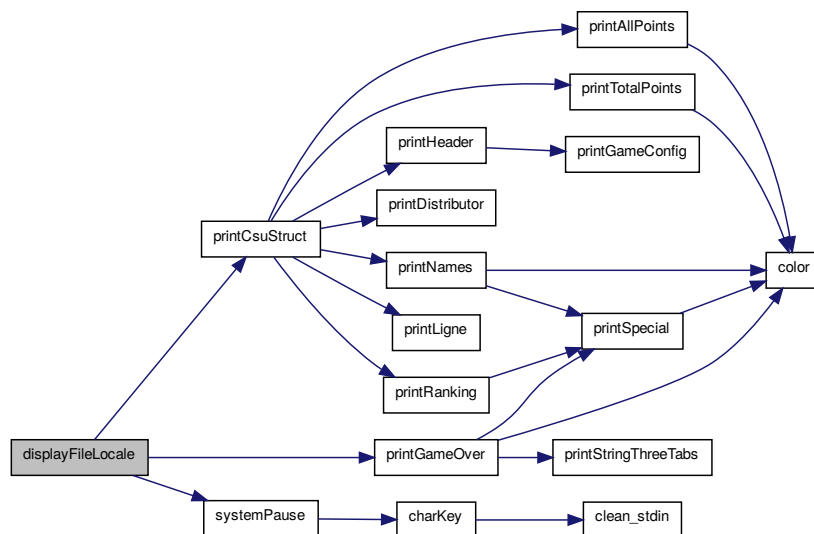
Here is the call graph for this function:



#### 2.4.2.4 void displayFileLocale ( char \* file\_name )

Display the file.

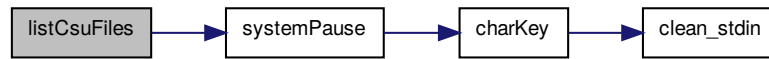
Here is the call graph for this function:



#### 2.4.2.5 void listCsuFiles ( )

List all the files with csu extension

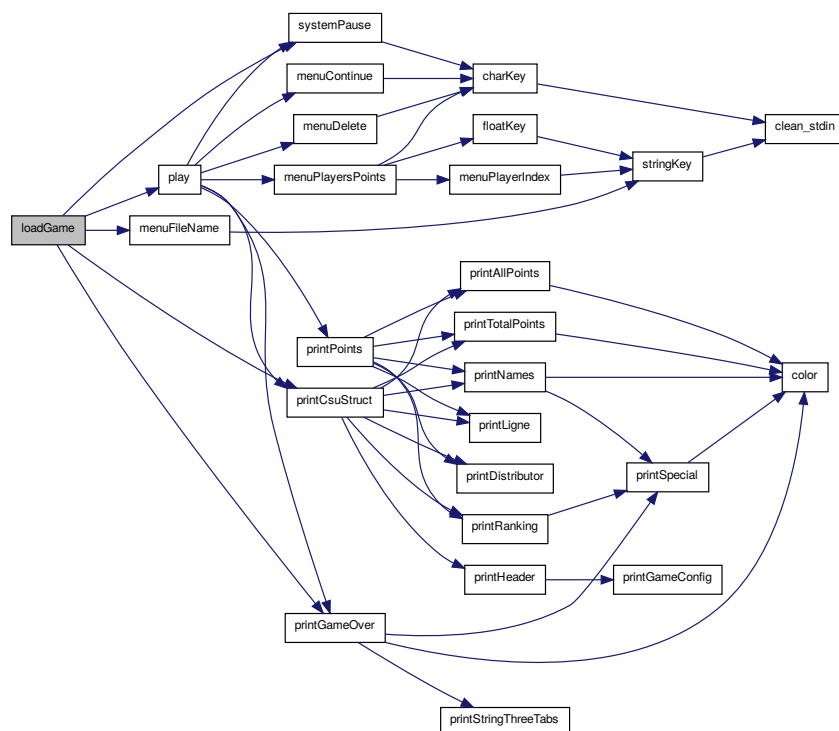
Here is the call graph for this function:



#### 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:



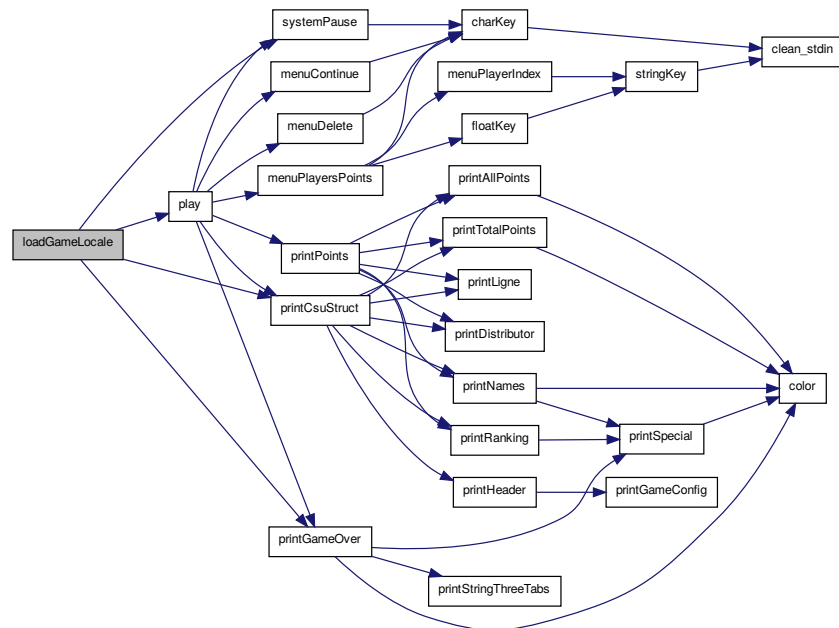
#### 2.4.2.7 void loadGameLocale ( char \* file\_name )

Load the file and run the play function

## Parameters

in	<i>file_name,the</i>	filename
----	----------------------	----------

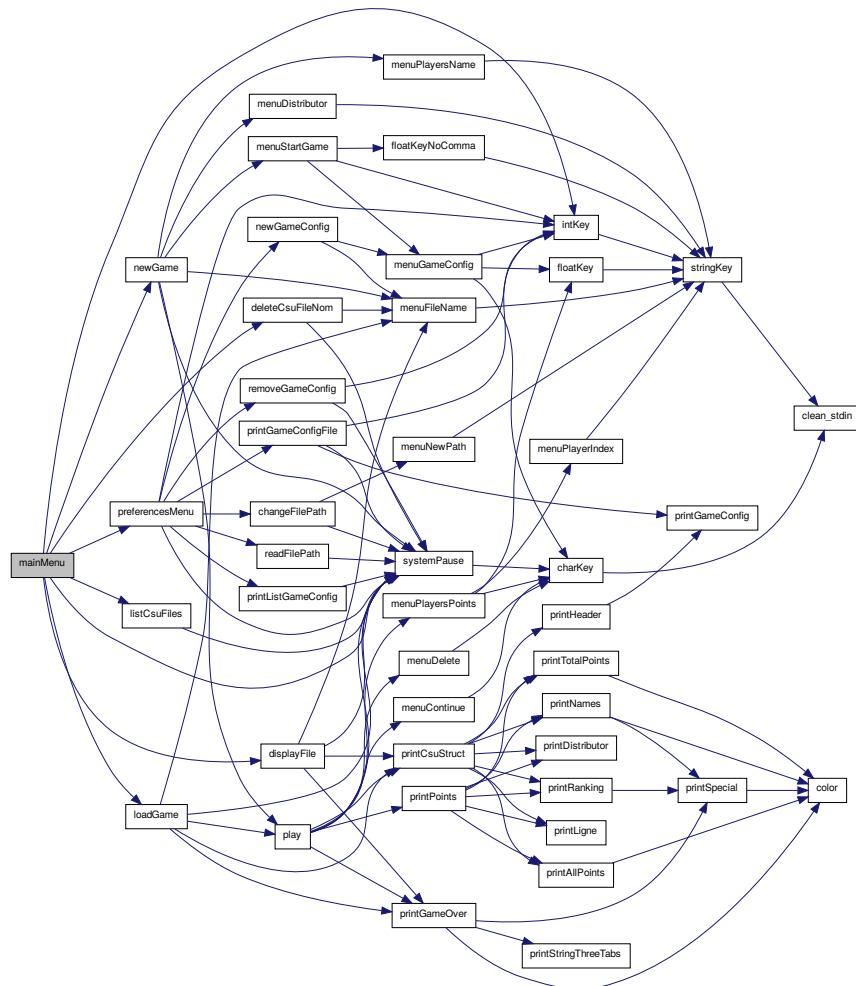
Here is the call graph for this function:



## 2.4.2.8 void mainMenu ( )

Main menu of csuper.

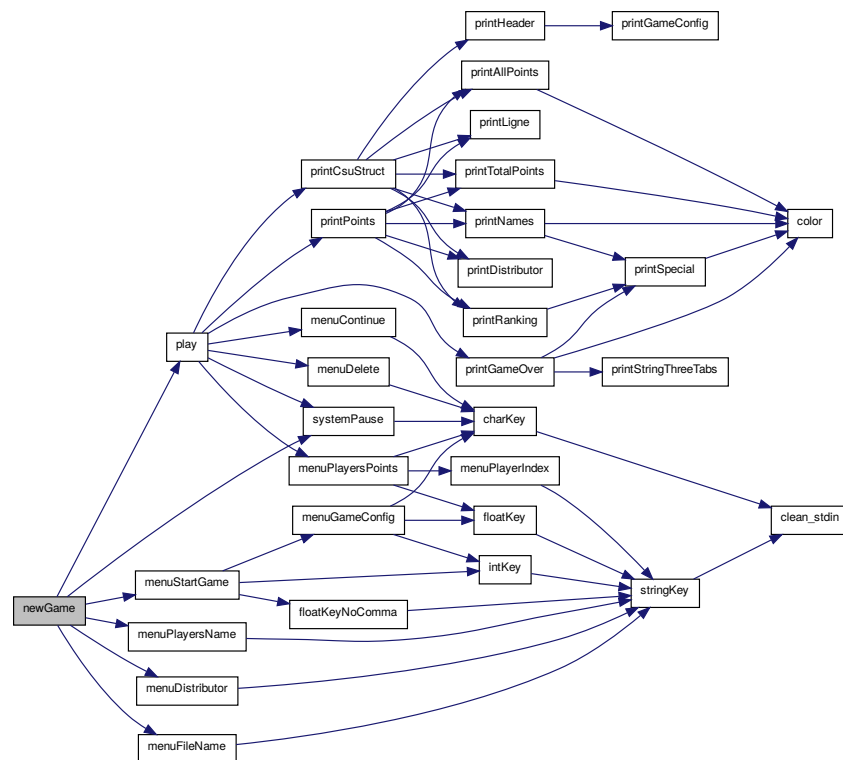
Here is the call graph for this function:



#### 2.4.2.9 void newGame ( )

Initialize a new game and run the play function.

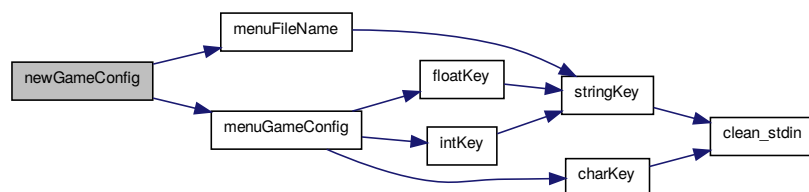
Here is the call graph for this function:



#### 2.4.2.10 void newGameConfig ( )

Add a new game configuration

Here is the call graph for this function:



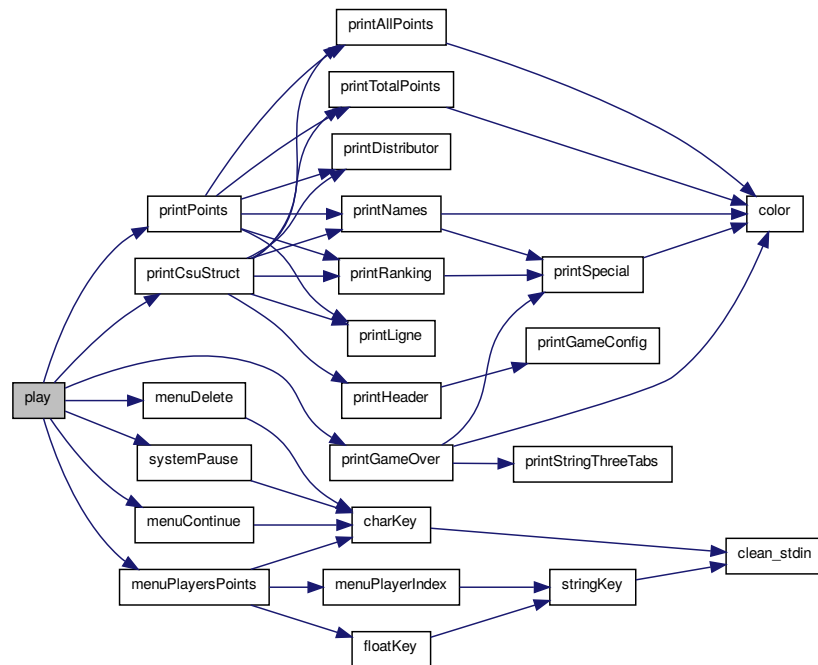
#### 2.4.2.11 void play ( csuStruct \* ptr\_csu\_struct, char \* file\_name )

Count the points

## Parameters

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

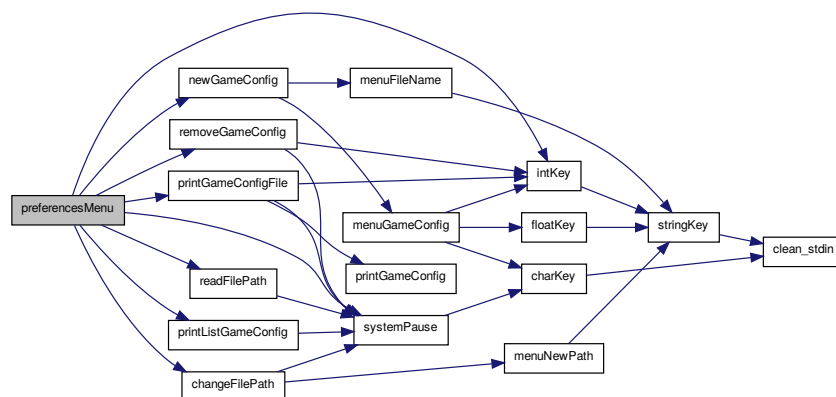
Here is the call graph for this function:



## 2.4.2.12 void preferencesMenu ( )

Preferences menu of csuper.

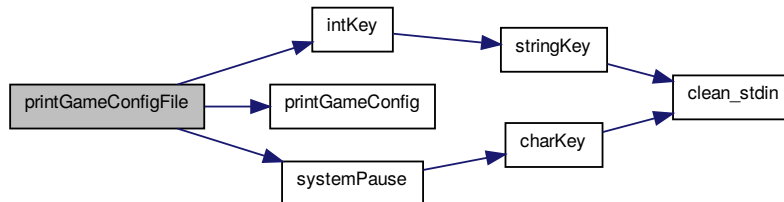
Here is the call graph for this function:





#### 2.4.2.13 void printGameConfigFile ( )

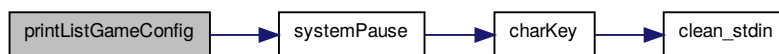
Here is the call graph for this function:



#### 2.4.2.14 void printListGameConfig ( )

Print the list of game configuration

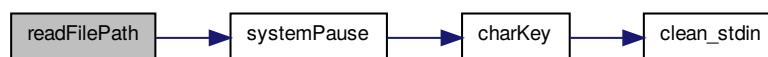
Here is the call graph for this function:



#### 2.4.2.15 void readFilePath ( )

Read the file path and display it.

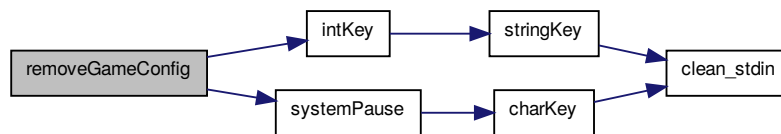
Here is the call graph for this function:



#### 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

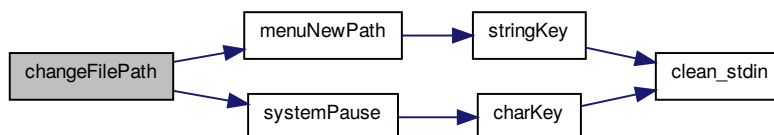
***newPath***  
***readPath***  
***newGameConf***  
***removeGameConf***  
***printListGameConf***  
***printGameConf***  
***backMainMenu***  
***easterEggs2***

### 2.5.3 Function Documentation

#### 2.5.3.1 void changeFilePath ( )

Change the path which the file are saved.

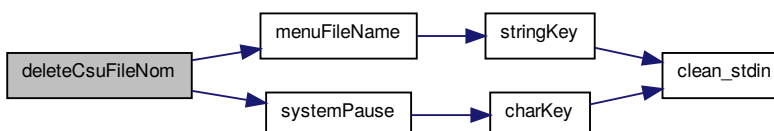
Here is the call graph for this function:



#### 2.5.3.2 void deleteCsuFileNom ( )

Ask a filename and deleted the file.

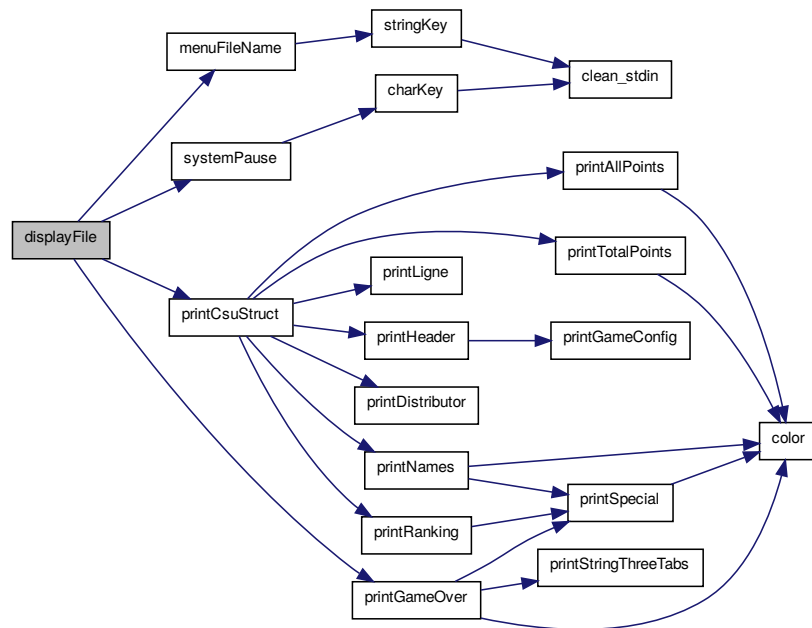
Here is the call graph for this function:



#### 2.5.3.3 void displayFile ( )

Ask a filename and display it.

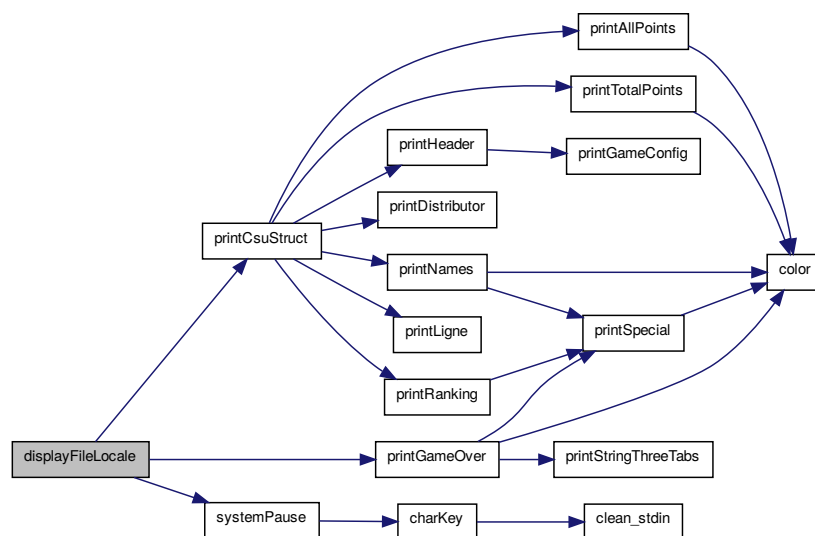
Here is the call graph for this function:



#### 2.5.3.4 void displayFileLocale ( char \* file\_name )

Display the file.

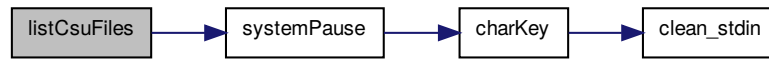
Here is the call graph for this function:



### 2.5.3.5 void listCsuFiles ( )

List all the files with csu extension

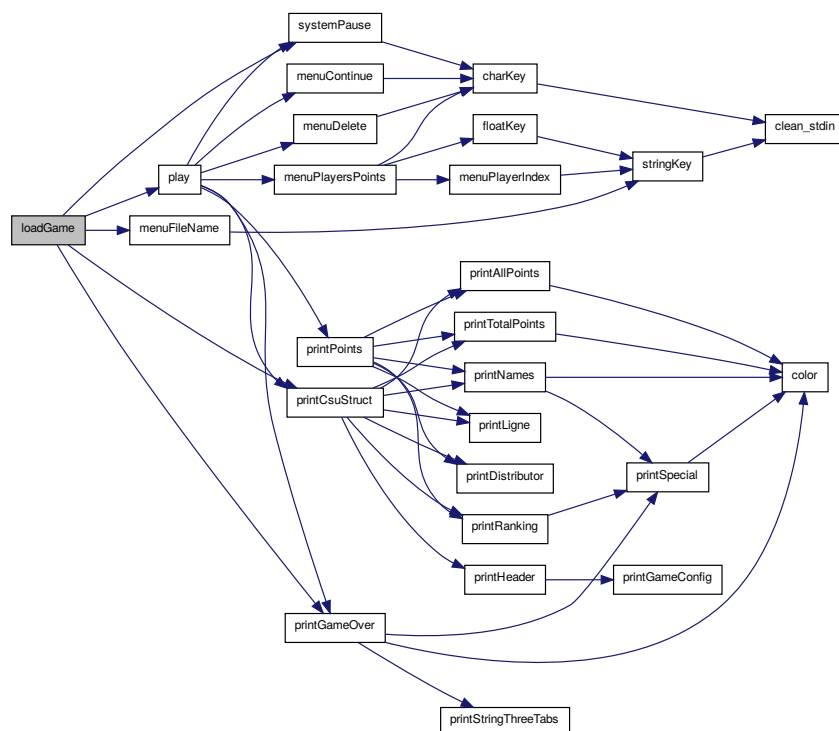
Here is the call graph for this function:



### 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:



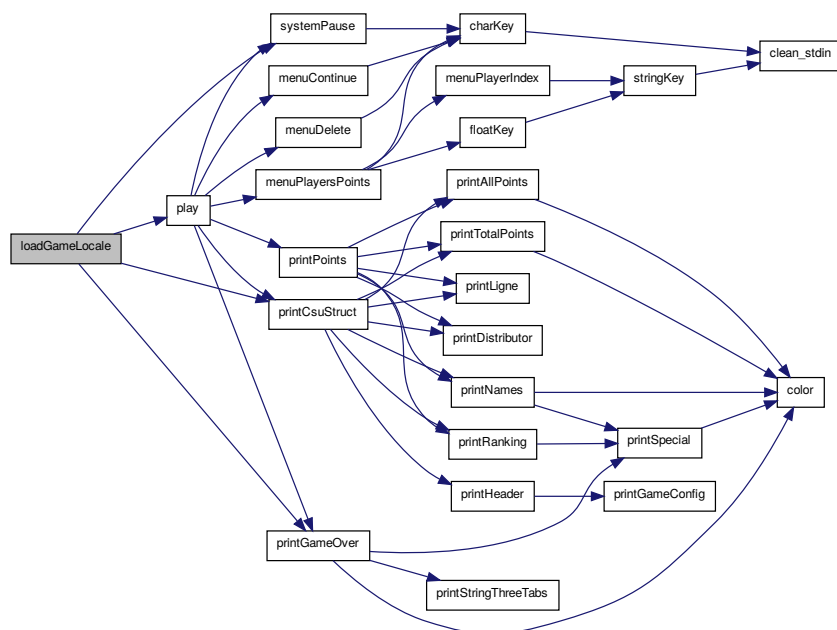
### 2.5.3.7 void loadGameLocale ( char \* file\_name )

Load the file and run the play function

## Parameters

in	<i>file_name,the</i>	filename
----	----------------------	----------

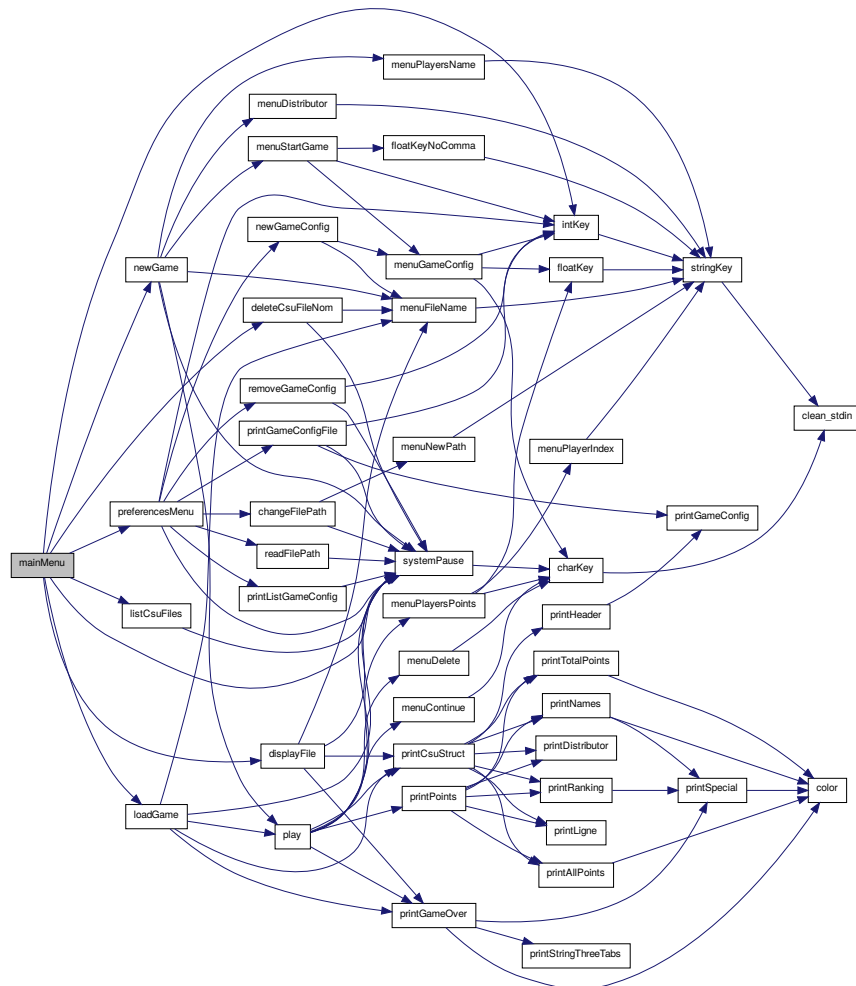
Here is the call graph for this function:



## 2.5.3.8 void mainMenu ( )

Main menu of csuper.

Here is the call graph for this function:

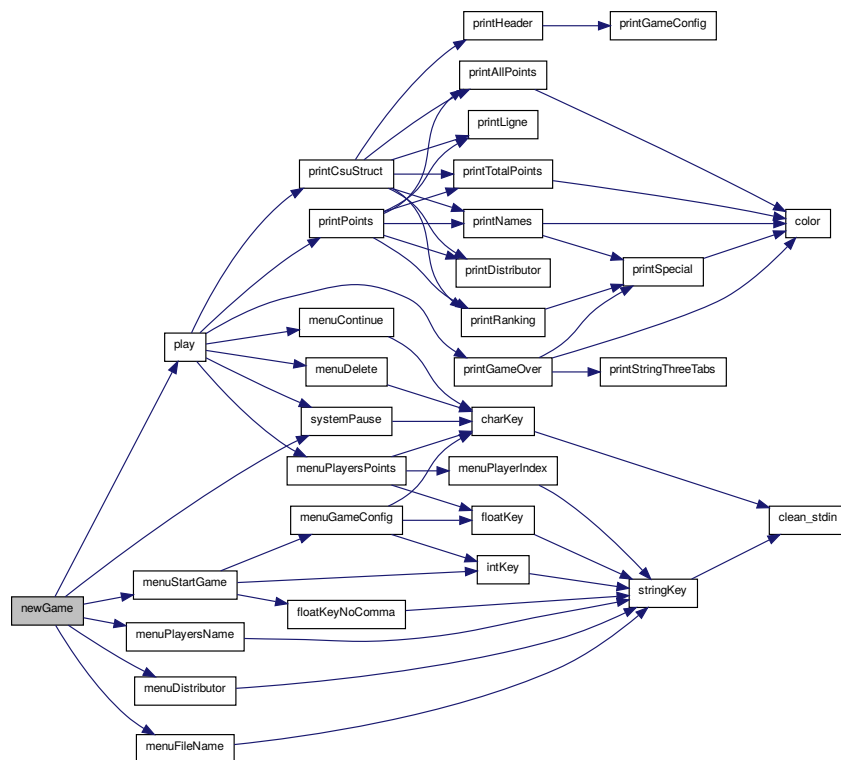


### 2.5.3.9 void newGame ( )

Initialize a new game and run the play function.



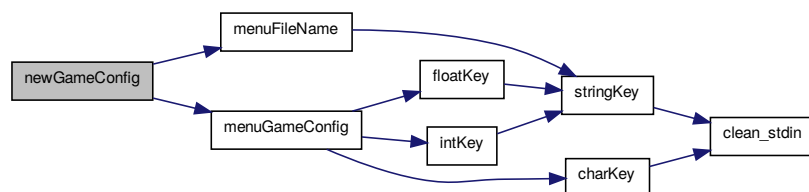
Here is the call graph for this function:



#### 2.5.3.10 void newGameConfig ( )

Add a new game configuration

Here is the call graph for this function:



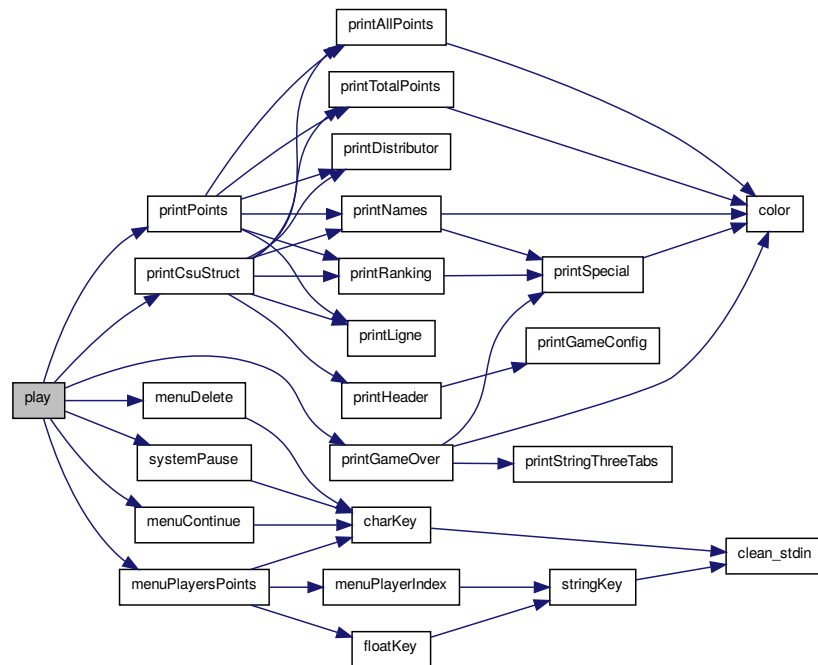
#### 2.5.3.11 void play ( csuStruct \* ptr\_csu\_struct, char \* file\_name )

Count the points

## Parameters

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

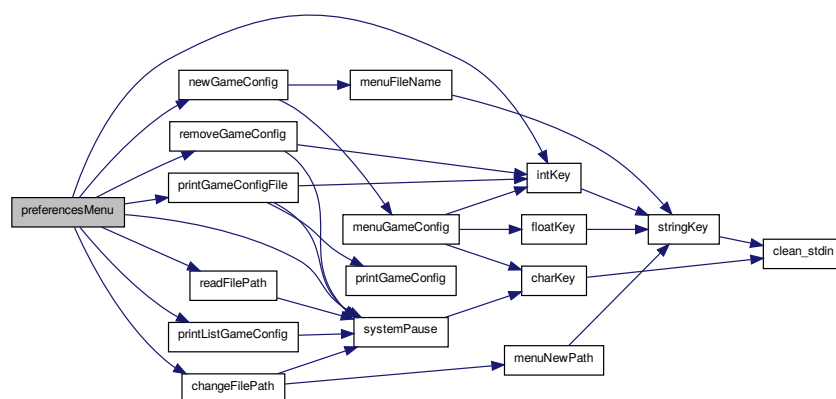
Here is the call graph for this function:



## 2.5.3.12 void preferencesMenu ( )

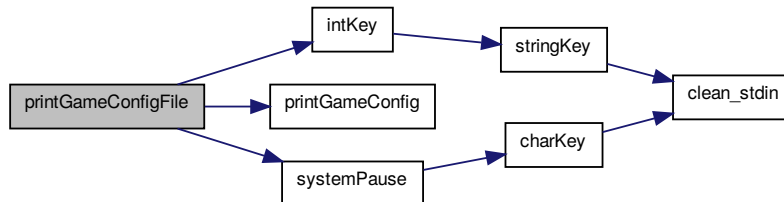
Preferences menu of csuper.

Here is the call graph for this function:



### 2.5.3.13 void printGameConfigFile ( )

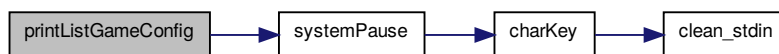
Here is the call graph for this function:



### 2.5.3.14 void printListGameConfig ( )

Print the list of game configuration

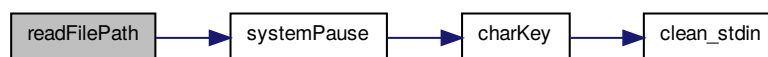
Here is the call graph for this function:



### 2.5.3.15 void readFilePath ( )

Read the file path and display it.

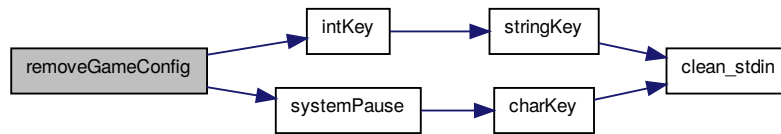
Here is the call graph for this function:



### 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.

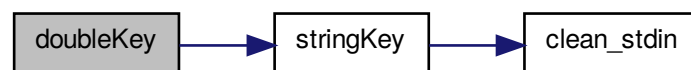
## 2.6.2.3 void doubleKey ( double \* nb )

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:



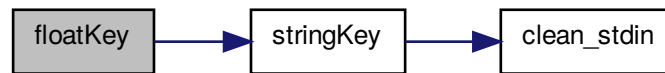
## 2.6.2.4 void floatKey ( float \* nb )

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

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

Here is the call graph for this function:



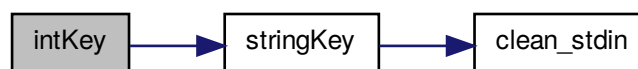
#### 2.6.2.6 void \* intKey ( int \* nb )

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

Parameters

in, out	*nb	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

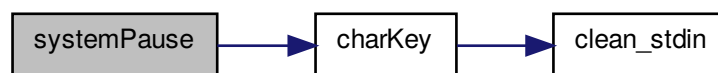
Here is the call graph for this function:



## 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

<code>in, out</code>	<code>*c</code>	the character
----------------------	-----------------	---------------



Here is the call graph for this function:



### 2.7.3.2 void clean\_stdin ( void )

Clean the buffer of stdin.

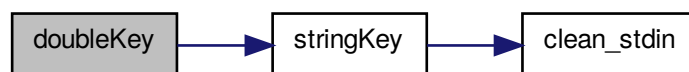
### 2.7.3.3 void doubleKey ( double \* nb )

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:



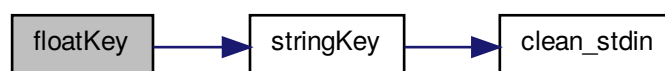
### 2.7.3.4 void floatKey ( float \* nb )

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:



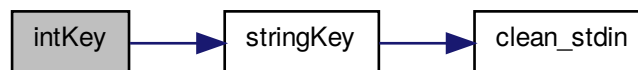
### 2.7.3.6 void intKey ( int \* *nb* )

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

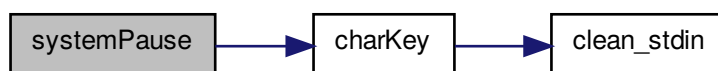
Here is the call graph for this function:



#### 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

## 2.8.2 Function Documentation

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

Begin csuper.

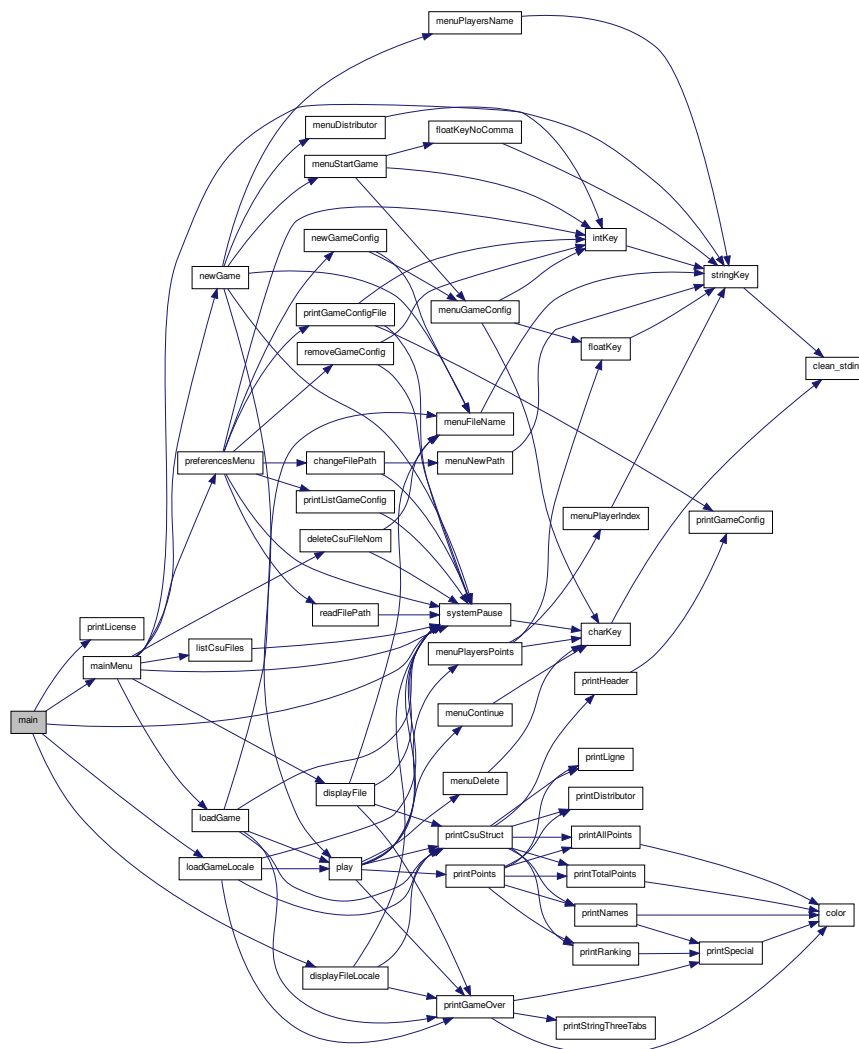
#### Parameters

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:



## 2.9 main.h File Reference

Begin csuper.

```
#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.



- 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

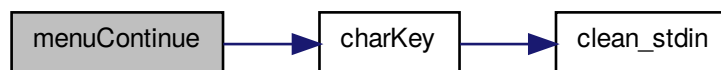
#### 2.10.2.1 int menuContinue ( )

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:



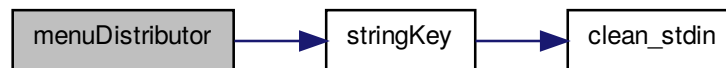
### 2.10.2.3 void menuDistributor ( char \* distributor\_name )

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**

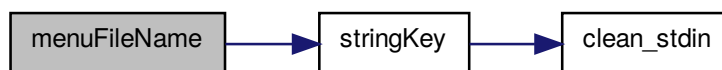
<code>in, out</code>	<code>file_name</code>	the filename
----------------------	------------------------	--------------



## Returns

filename

Here is the call graph for this function:



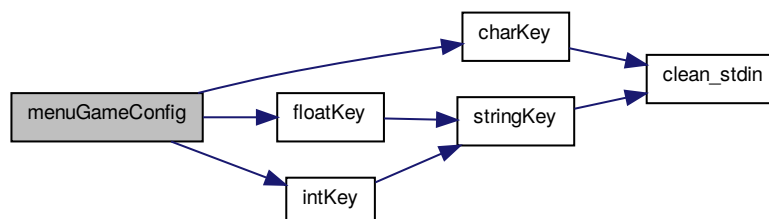
#### 2.10.2.5 void menuGameConfig ( game\_config \* ptr\_config )

Ask and save a game configuration.

## Parameters

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

Here is the call graph for this function:



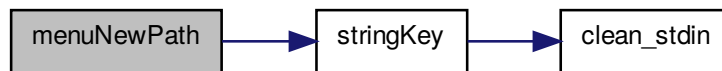
#### 2.10.2.6 void menuNewPath ( char \* new\_path )

Ask and save the new path

## Parameters

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

Here is the call graph for this function:



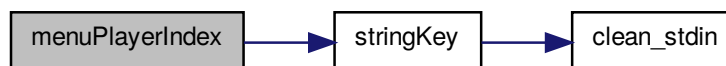
#### 2.10.2.7 `int menuPlayerIndex ( csuStruct * ptr_csu_struct )`

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:



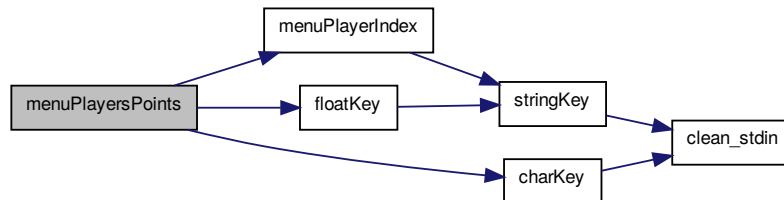
#### 2.10.2.9 `void menuPlayersPoints ( csuStruct * ptr_csu_struct )`

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:



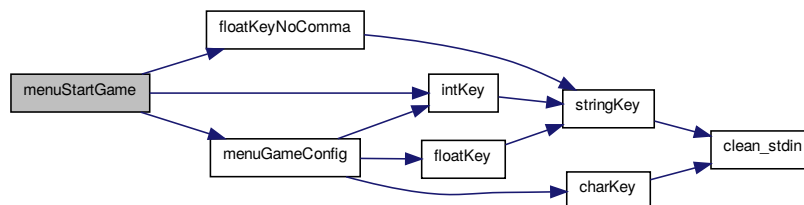
#### 2.10.2.10 void menuStartGame ( float \* ptr\_nb\_player, game\_config \* ptr\_config )

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

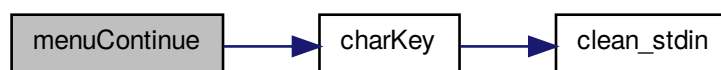
#### 2.11.2.1 int menuContinue ( )

Ask if we want to continue3

#### Returns

TRUE if we want to continue, FALSE otherwise

Here is the call graph for this function:



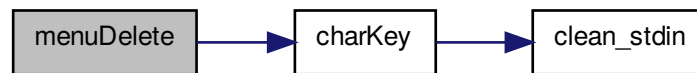
#### 2.11.2.2 int menuDelete ( )

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:

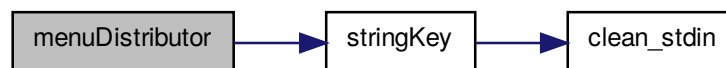
**2.11.2.3 void menuDistributor ( char \* distributor\_name )**

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.

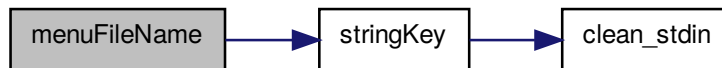
**Parameters**

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

## Returns

filename

Here is the call graph for this function:



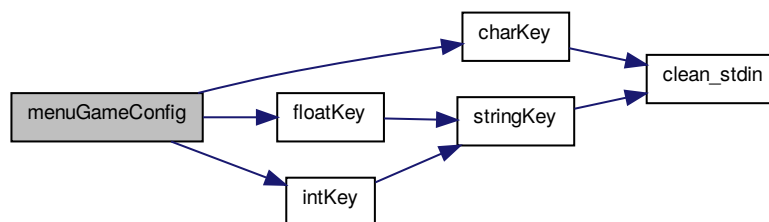
#### 2.11.2.5 void menuGameConfig ( game\_config \* ptr\_config )

Ask and save a game configuration.

## Parameters

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

Here is the call graph for this function:



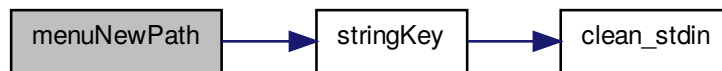
#### 2.11.2.6 void menuNewPath ( char \* new\_path )

Ask and save the new path

## Parameters

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

Here is the call graph for this function:



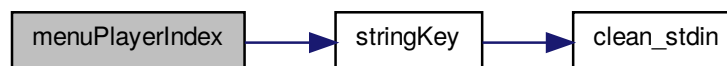
#### 2.11.2.7 `int menuPlayerIndex ( csuStruct * ptr_csu_struct )`

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:



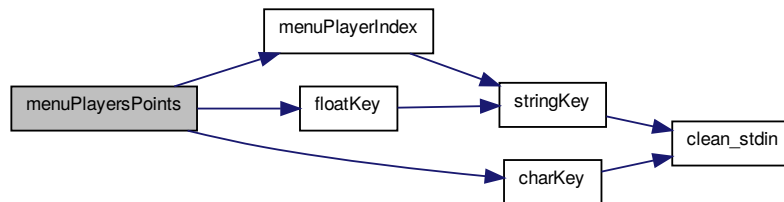
#### 2.11.2.9 `void menuPlayersPoints ( csuStruct * ptr_csu_struct )`

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:



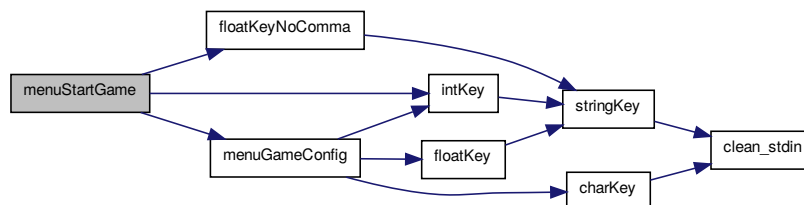
#### 2.11.2.10 void menuStartGame ( float \* *ptr\_nb\_player*, game\_config \* *ptr\_config* )

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, 27

backgroundBlack  
    display.h, 10

backgroundBlue  
    display.h, 11

backgroundBrown  
    display.h, 10

backgroundCyan  
    display.h, 11

backgroundDefault  
    display.h, 11

backgroundGreen  
    display.h, 10

backgroundMagenta  
    display.h, 11

backgroundRed  
    display.h, 10

backgroundWhite  
    display.h, 11

backgroundColor  
    display.h, 10

CSUPER  
    csuper.h, 3

changeFilePath  
    interface.c, 18  
    interface.h, 27

charKey  
    keyboarding.c, 36  
    keyboarding.h, 40

clean\_stdin  
    keyboarding.c, 37  
    keyboarding.h, 41

color  
    display.c, 4  
    display.h, 11

csuper.h, 3  
    \_, 3  
    CSUPER, 3

deleteFile  
    interface.h, 27

deleteCsuFileNom  
    interface.c, 18  
    interface.h, 28

display.h

backgroundBlack, 10  
backgroundBlue, 11  
backgroundBrown, 10  
backgroundCyan, 11  
backgroundDefault, 11  
backgroundGreen, 10  
backgroundMagenta, 11  
backgroundRed, 10  
backgroundWhite, 11  
foregroundBlack, 11  
foregroundBlue, 11  
foregroundBrown, 11  
foregroundCyan, 11  
foregroundGreen, 11  
foregroundMagenta, 11  
foregroundRed, 11  
foregroundWhite, 11  
writingBold, 11  
writingReset, 11  
writingUnderline, 11

display.c, 4  
    color, 4  
    printAllPoints, 4  
    printCsuStruct, 5  
    printDistributor, 5  
    printGameConfig, 6  
    printGameOver, 6  
    printHeader, 6  
    printLicense, 6  
    printLigne, 7  
    printNames, 7  
    printPoints, 7  
    printRanking, 8  
    printSpecial, 8  
    printStringThreeTabs, 9  
    printTotalPoints, 9

display.h, 9  
    backgroundColor, 10  
    color, 11  
    foregroundColor, 11  
    printAllPoints, 11  
    printCsuStruct, 12  
    printDistributor, 12  
    printGameConfig, 12  
    printGameOver, 14  
    printHeader, 14  
    printLicense, 14  
    printLigne, 14  
    printNames, 15

- printPoints, 15
- printRanking, 16
- printSpecial, 16
- printStringThreeTabs, 16
- printTotalPoints, 16
- Writing, 11
- displayFile
  - interface.c, 18
  - interface.h, 28
- displayFileLocale
  - interface.c, 19
  - interface.h, 29
- doubleKey
  - keyboarding.c, 37
  - keyboarding.h, 41
- easterEggs
  - interface.h, 27
- easterEggs2
  - interface.h, 27
- floatKey
  - keyboarding.c, 37
  - keyboarding.h, 41
- floatKeyNoComma
  - keyboarding.c, 38
  - keyboarding.h, 42
- foregroundBlack
  - display.h, 11
- foregroundBlue
  - display.h, 11
- foregroundBrown
  - display.h, 11
- foregroundCyan
  - display.h, 11
- foregroundGreen
  - display.h, 11
- foregroundMagenta
  - display.h, 11
- foregroundRed
  - display.h, 11
- foregroundWhite
  - display.h, 11
- foregroundColor
  - display.h, 11
- intKey
  - keyboarding.c, 38
  - keyboarding.h, 42
- interface.h
  - backMainMenu, 27
  - deleteFile, 27
  - easterEggs, 27
  - easterEggs2, 27
  - listFile, 27
  - loadMatch, 27
  - newGameConf, 27
  - newMatch, 27
  - newPath, 27
  - pref, 27
  - printFile, 27
  - printGameConf, 27
  - printListGameConf, 27
  - quit, 27
  - readPath, 27
  - removeGameConf, 27
- interface.c, 17
  - changeFilePath, 18
  - deleteCsuFileNom, 18
  - displayFile, 18
  - displayFileLocale, 19
  - listCsuFiles, 19
  - loadGame, 20
  - loadGameLocale, 20
  - mainMenu, 21
  - newGame, 22
  - newGameConfig, 23
  - play, 23
  - preferencesMenu, 24
  - printGameConfigFile, 24
  - printListGameConfig, 25
  - readFilePath, 25
  - removeGameConfig, 25
- interface.h, 26
  - changeFilePath, 27
  - deleteCsuFileNom, 28
  - displayFile, 28
  - displayFileLocale, 29
  - listCsuFiles, 29
  - loadGame, 30
  - loadGameLocale, 30
  - MainMenu, 27
  - mainMenu, 31
  - newGame, 32
  - newGameConfig, 33
  - play, 33
  - PreferencesMenu, 27
  - preferencesMenu, 34
  - printGameConfigFile, 34
  - printListGameConfig, 35
  - readFilePath, 35
  - removeGameConfig, 35
- keyboarding.c, 36
  - charKey, 36
  - clean\_stdin, 37
  - doubleKey, 37
  - floatKey, 37
  - floatKeyNoComma, 38
  - intKey, 38
  - stringKey, 38
  - systemPause, 39
- keyboarding.h, 39
  - charKey, 40
  - clean\_stdin, 41
  - doubleKey, 41
  - floatKey, 41
  - floatKeyNoComma, 42

- intKey, [42](#)
- NB\_CARACT\_DOUB, [40](#)
- NB\_CARACT\_FLOAT, [40](#)
- NB\_CARACT\_INT, [40](#)
- stringKey, [42](#)
- systemPause, [43](#)
- listFile
  - interface.h, [27](#)
- listCsuFiles
  - interface.c, [19](#)
  - interface.h, [29](#)
- loadMatch
  - interface.h, [27](#)
- loadGame
  - interface.c, [20](#)
  - interface.h, [30](#)
- loadGameLocale
  - interface.c, [20](#)
  - interface.h, [30](#)
- main
  - main.c, [44](#)
  - main.h, [45](#)
- main.c, [43](#)
  - main, [44](#)
- main.h, [44](#)
  - main, [45](#)
- MainMenu
  - interface.h, [27](#)
- mainMenu
  - interface.c, [21](#)
  - interface.h, [31](#)
- menu.c, [46](#)
  - menuContinue, [47](#)
  - menuDelete, [47](#)
  - menuDistributor, [48](#)
  - menuFileName, [48](#)
  - menuGameConfig, [49](#)
  - menuNewPath, [49](#)
  - menuPlayerIndex, [50](#)
  - menuPlayersName, [50](#)
  - menuPlayersPoints, [50](#)
  - menuStartGame, [51](#)
- menu.h, [51](#)
  - menuContinue, [52](#)
  - menuDelete, [52](#)
  - menuDistributor, [53](#)
  - menuFileName, [53](#)
  - menuGameConfig, [54](#)
  - menuNewPath, [54](#)
  - menuPlayerIndex, [55](#)
  - menuPlayersName, [55](#)
  - menuPlayersPoints, [55](#)
  - menuStartGame, [56](#)
- menuContinue
  - menu.c, [47](#)
  - menu.h, [52](#)
- menuDelete
  - menu.c, [47](#)
  - menu.h, [52](#)
- menuDistributor
  - menu.c, [48](#)
  - menu.h, [53](#)
- menuFileName
  - menu.c, [48](#)
  - menu.h, [53](#)
- menuGameConfig
  - menu.c, [49](#)
  - menu.h, [54](#)
- menuNewPath
  - menu.c, [49](#)
  - menu.h, [54](#)
- menuPlayerIndex
  - menu.c, [50](#)
  - menu.h, [55](#)
- menuPlayersName
  - menu.c, [50](#)
  - menu.h, [55](#)
- menuPlayersPoints
  - menu.c, [50](#)
  - menu.h, [55](#)
- menuStartGame
  - menu.c, [51](#)
  - menu.h, [56](#)
- NB\_CARACT\_DOUB
  - keyboarding.h, [40](#)
- NB\_CARACT\_FLOAT
  - keyboarding.h, [40](#)
- NB\_CARACT\_INT
  - keyboarding.h, [40](#)
- newGameConf
  - interface.h, [27](#)
- newMatch
  - interface.h, [27](#)
- newPath
  - interface.h, [27](#)
- newGame
  - interface.c, [22](#)
  - interface.h, [32](#)
- newGameConfig
  - interface.c, [23](#)
  - interface.h, [33](#)
- play
  - interface.c, [23](#)
  - interface.h, [33](#)
- pref
  - interface.h, [27](#)
- PreferencesMenu
  - interface.h, [27](#)
- preferencesMenu
  - interface.c, [24](#)
  - interface.h, [34](#)
- printFile
  - interface.h, [27](#)
- printGameConf

- interface.h, 27
- printListGameConf
  - interface.h, 27
- printAllPoints
  - display.c, 4
  - display.h, 11
- printCsuStruct
  - display.c, 5
  - display.h, 12
- printDistributor
  - display.c, 5
  - display.h, 12
- printGameConfig
  - display.c, 6
  - display.h, 12
- printGameConfigFile
  - interface.c, 24
  - interface.h, 34
- printGameOver
  - display.c, 6
  - display.h, 14
- printHeader
  - display.c, 6
  - display.h, 14
- printLicense
  - display.c, 6
  - display.h, 14
- printLigne
  - display.c, 7
  - display.h, 14
- printListGameConfig
  - interface.c, 25
  - interface.h, 35
- printNames
  - display.c, 7
  - display.h, 15
- printPoints
  - display.c, 7
  - display.h, 15
- printRanking
  - display.c, 8
  - display.h, 16
- printSpecial
  - display.c, 8
  - display.h, 16
- printStringThreeTabs
  - display.c, 9
  - display.h, 16
- printTotalPoints
  - display.c, 9
  - display.h, 16
- quit
  - interface.h, 27
- readPath
  - interface.h, 27
- readFilePath
  - interface.c, 25
- interface.h, 35
- removeGameConf
  - interface.h, 27
- removeGameConfig
  - interface.c, 25
  - interface.h, 35
- stringKey
  - keyboarding.c, 38
  - keyboarding.h, 42
- systemPause
  - keyboarding.c, 39
  - keyboarding.h, 43
- Writing
  - display.h, 11
- writingBold
  - display.h, 11
- writingReset
  - display.h, 11
- writingUnderline
  - display.h, 11