

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

## 4.2.0

Generated by Doxygen 1.8.9.1

Fri Jul 24 2015 12:15:27



# 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
2.2	display.c File Reference	3
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	5
2.2.2.6	printGameOver	6
2.2.2.7	printHeader	6
2.2.2.8	printLicense	6
2.2.2.9	printLigne	6
2.2.2.10	printNames	7
2.2.2.11	printPoints	7
2.2.2.12	printRanking	8
2.2.2.13	printSpecial	9
2.2.2.14	printStringThreeTabs	9
2.2.2.15	printTotalPoints	9
2.3	display.h File Reference	10
2.3.1	Detailed Description	11
2.3.2	Enumeration Type Documentation	11
2.3.2.1	backgroundColor	11
2.3.2.2	foregroundColor	11

2.3.2.3	Writing	11
2.3.3	Function Documentation	12
2.3.3.1	color	12
2.3.3.2	printAllPoints	13
2.3.3.3	printCsuStruct	13
2.3.3.4	printDistributor	14
2.3.3.5	printGameConfig	15
2.3.3.6	printGameOver	15
2.3.3.7	printHeader	15
2.3.3.8	printLicense	16
2.3.3.9	printLigne	16
2.3.3.10	printNames	16
2.3.3.11	printPoints	16
2.3.3.12	printRanking	17
2.3.3.13	printSpecial	17
2.3.3.14	printStringThreeTabs	18
2.3.3.15	printTotalPoints	18
2.4	interface.c File Reference	18
2.4.1	Detailed Description	19
2.4.2	Function Documentation	19
2.4.2.1	changeFilePath	19
2.4.2.2	changePdfPreferences	20
2.4.2.3	deleteCsuFileNom	20
2.4.2.4	displayFile	20
2.4.2.5	displayFileLocale	21
2.4.2.6	exportCsu	21
2.4.2.7	exportListGameConfig	22
2.4.2.8	exportToCsvLocale	22
2.4.2.9	exportToGnuplotLocale	22
2.4.2.10	exportToMLocale	23
2.4.2.11	exportToPdfLocale	23
2.4.2.12	importListGameConfig	23
2.4.2.13	listCsuFiles	24
2.4.2.14	loadGame	24
2.4.2.15	loadGameLocale	25
2.4.2.16	mainMenu	25
2.4.2.17	newGame	26
2.4.2.18	newGameConfig	27
2.4.2.19	play	27
2.4.2.20	preferencesMenu	28

2.4.2.21	printGameConfigFile	29
2.4.2.22	printListGameConfig	29
2.4.2.23	readFilePath	29
2.4.2.24	removeGameConfig	29
2.5	interface.h File Reference	30
2.5.1	Detailed Description	31
2.5.2	Enumeration Type Documentation	31
2.5.2.1	FileType	31
2.5.2.2	MainMenu	31
2.5.2.3	PreferencesMenu	32
2.5.3	Function Documentation	32
2.5.3.1	changeFilePath	32
2.5.3.2	changePdfPreferences	32
2.5.3.3	deleteCsuFileNom	32
2.5.3.4	displayFile	33
2.5.3.5	displayFileLocale	33
2.5.3.6	exportCsu	34
2.5.3.7	exportListGameConfig	34
2.5.3.8	exportToCsvLocale	35
2.5.3.9	exportToGnuplotLocale	35
2.5.3.10	exportToMLocale	35
2.5.3.11	exportToPdfLocale	36
2.5.3.12	importListGameConfig	37
2.5.3.13	listCsuFiles	37
2.5.3.14	loadGame	37
2.5.3.15	loadGameLocale	38
2.5.3.16	mainMenu	39
2.5.3.17	newGame	40
2.5.3.18	newGameConfig	41
2.5.3.19	play	41
2.5.3.20	preferencesMenu	42
2.5.3.21	printGameConfigFile	43
2.5.3.22	printListGameConfig	43
2.5.3.23	readFilePath	43
2.5.3.24	removeGameConfig	43
2.6	keyboarding.c File Reference	44
2.6.1	Detailed Description	44
2.6.2	Function Documentation	44
2.6.2.1	charKey	44
2.6.2.2	clean_stdin	45

2.6.2.3	floatKey	45
2.6.2.4	floatKeyNoComma	45
2.6.2.5	intKey	46
2.6.2.6	stringKey	46
2.6.2.7	systemPause	47
2.7	keyboarding.h File Reference	47
2.7.1	Detailed Description	47
2.7.2	Macro Definition Documentation	48
2.7.2.1	NB_CARACT_DOUB	48
2.7.2.2	NB_CARACT_EXPRESSION	48
2.7.2.3	NB_CARACT_FLOAT	48
2.7.2.4	NB_CARACT_INT	48
2.7.3	Function Documentation	48
2.7.3.1	charKey	48
2.7.3.2	clean_stdin	48
2.7.3.3	floatKey	49
2.7.3.4	floatKeyNoComma	50
2.7.3.5	intKey	50
2.7.3.6	stringKey	51
2.7.3.7	systemPause	52
2.8	main.c File Reference	52
2.8.1	Detailed Description	52
2.8.2	Function Documentation	53
2.8.2.1	main	53
2.9	main.h File Reference	54
2.9.1	Detailed Description	55
2.9.2	Macro Definition Documentation	55
2.9.2.1	PORTABLE	55
2.9.3	Function Documentation	55
2.9.3.1	main	55
2.10	menu.c File Reference	56
2.10.1	Detailed Description	57
2.10.2	Function Documentation	57
2.10.2.1	menuChangeDistributor	57
2.10.2.2	menuChooseExportFileType	57
2.10.2.3	menuContinueChangeDistributorOrQuit	58
2.10.2.4	menuDelete	58
2.10.2.5	menuDistributor	58
2.10.2.6	menuExportListGameConfig	59
2.10.2.7	menuFileName	59

2.10.2.8	<a href="#">menuGameConfig</a>	60
2.10.2.9	<a href="#">menuImportListGameConfig</a>	60
2.10.2.10	<a href="#">menuNewPath</a>	61
2.10.2.11	<a href="#">menuPdfPreferences</a>	61
2.10.2.12	<a href="#">menuPlayerIndex</a>	62
2.10.2.13	<a href="#">menuPlayersName</a>	63
2.10.2.14	<a href="#">menuPlayersPoints</a>	63
2.10.2.15	<a href="#">menuStartGame</a>	64
2.11	<a href="#">menu.h File Reference</a>	65
2.11.1	<a href="#">Detailed Description</a>	65
2.11.2	<a href="#">Enumeration Type Documentation</a>	66
2.11.2.1	<a href="#">ContinueChangeDistributorOrQuit</a>	66
2.11.3	<a href="#">Function Documentation</a>	66
2.11.3.1	<a href="#">menuChangeDistributor</a>	66
2.11.3.2	<a href="#">menuChooseExportFileType</a>	66
2.11.3.3	<a href="#">menuContinueChangeDistributorOrQuit</a>	67
2.11.3.4	<a href="#">menuDelete</a>	67
2.11.3.5	<a href="#">menuDistributor</a>	67
2.11.3.6	<a href="#">menuExportListGameConfig</a>	68
2.11.3.7	<a href="#">menuFileName</a>	68
2.11.3.8	<a href="#">menuGameConfig</a>	69
2.11.3.9	<a href="#">menuImportListGameConfig</a>	69
2.11.3.10	<a href="#">menuNewPath</a>	69
2.11.3.11	<a href="#">menuPdfPreferences</a>	70
2.11.3.12	<a href="#">menuPlayerIndex</a>	70
2.11.3.13	<a href="#">menuPlayersName</a>	71
2.11.3.14	<a href="#">menuPlayersPoints</a>	72
2.11.3.15	<a href="#">menuStartGame</a>	72
<b>Index</b>		<b>75</b>





# 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 . . . . .	3
<a href="#">display.h</a>	Display the games . . . . .	10
<a href="#">interface.c</a>	Graphical interface . . . . .	18
<a href="#">interface.h</a>	Graphical interface . . . . .	30
<a href="#">keyboarding.c</a>	Function of keyboarding . . . . .	44
<a href="#">keyboarding.h</a>	Function of keyboarding . . . . .	47
<a href="#">main.c</a>	Begin csuper . . . . .	52
<a href="#">main.h</a>	Begin csuper . . . . .	54
<a href="#">menu.c</a>	Menu functions . . . . .	56
<a href="#">menu.h</a>	Menu functions . . . . .	65



## Chapter 2

# File Documentation

### 2.1 csuper.h File Reference

Include of csuper.

```
#include "../libcsuper/libcsuper.h"
```

#### Macros

- #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.

### 2.2 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

in	<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:



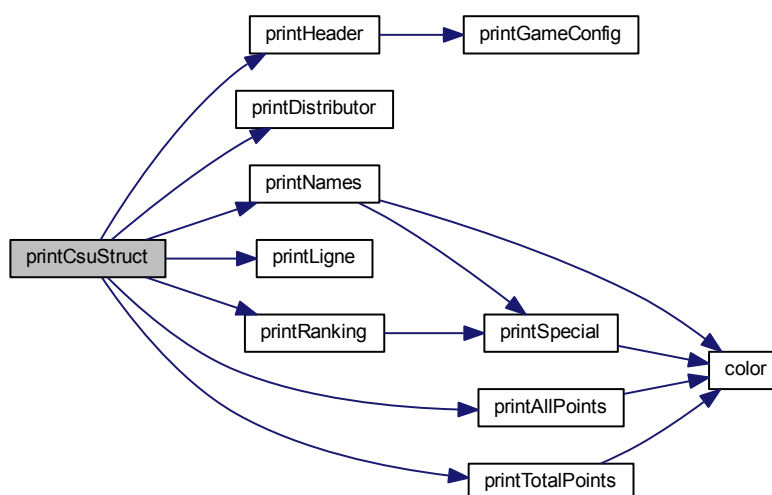
#### 2.2.2.3 void printCsuStruct ( csuStruct \* ptr\_csu\_struct )

Print all data of the csu structure.

Parameters

in	*ptr_csu_struct	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	*ptr_csu_struct	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
----	---------------	---------------

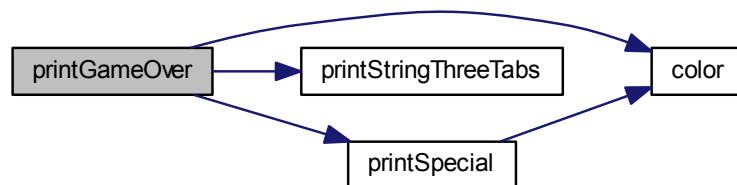
2.2.2.6 void printGameOver ( csuStruct \* *ptr\_csu\_struct* )

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
----	------------------	----------------------

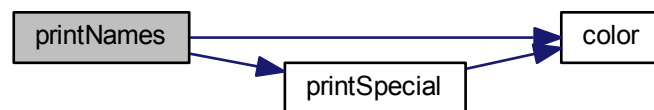
2.2.2.10 void printNames ( csuStruct \* *ptr\_csu\_struct*, int \* *ptr\_size\_line* )

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:

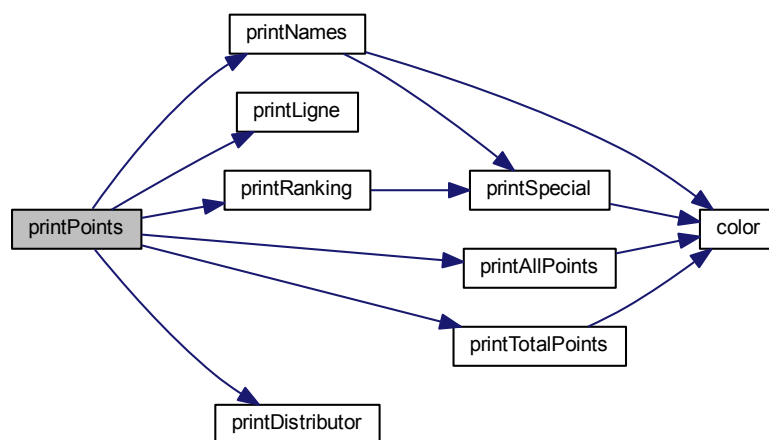
2.2.2.11 void printPoints ( csuStruct \* *ptr\_csu\_struct* )

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* )

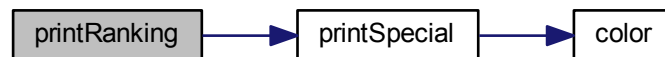
Print the rank of each player



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

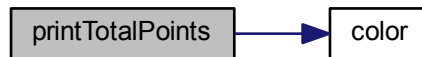
2.2.2.15 void printTotalPoints ( csuStruct \* *ptr\_csu\_struct* )

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:



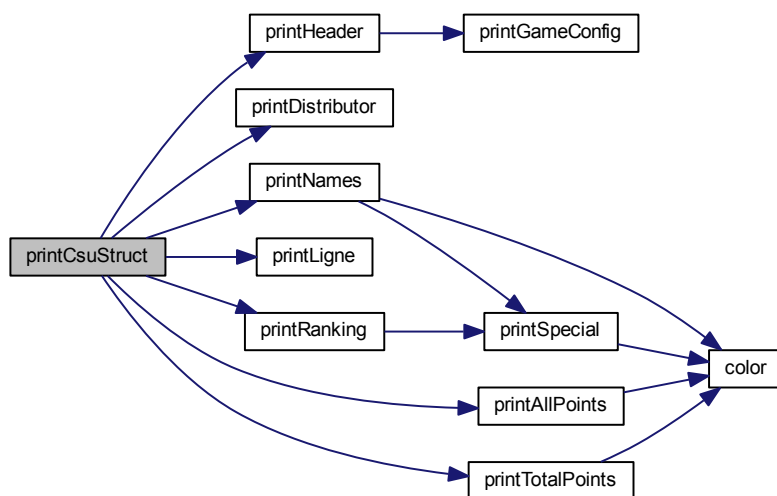
## 2.3.3.3 void printCsuStruct ( csuStruct \* ptr\_csu\_struct )

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
----	---------------	---------------

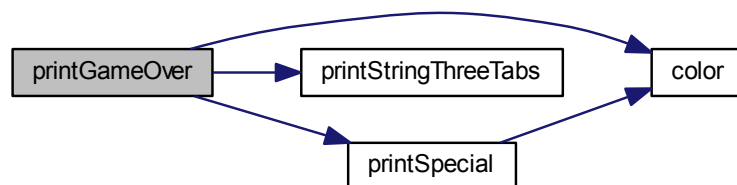
## 2.3.3.6 void printGameOver ( csuStruct \* ptr\_csu\_struct )

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
----	------------------	----------------------

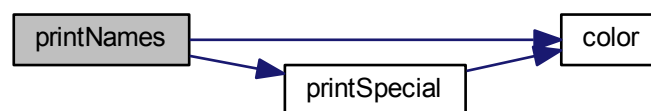
### 2.3.3.10 void printNames ( csuStruct \* *ptr\_csu\_struct*, int \* *ptr\_size\_line* )

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* )

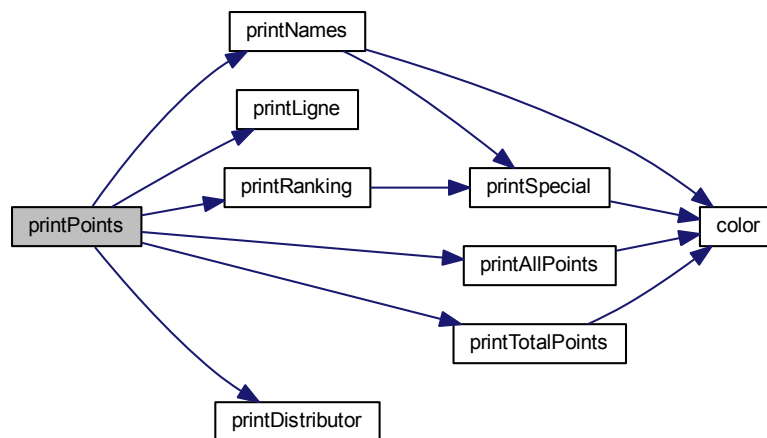
Print the names, the points, the total points and the rank of each player.

Parameters

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



Here is the call graph for this function:



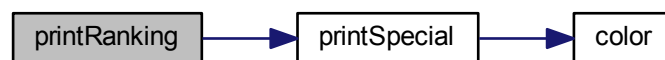
#### 2.3.3.12 void printRanking ( csuStruct \* ptr\_csu\_struct )

Print the rank of each player

##### Parameters

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

Here is the call graph for this function:



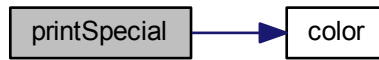
#### 2.3.3.13 void printSpecial ( char \* string, int nb\_arg, ... )

Print the string in stdin with special effect.

##### Parameters

in	<i>string</i>	a string to display
in	<i>nb_arg</i>	the number of argument
in	<i>...</i>	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	<i>*string</i>	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` ()
- void `exportListGameConfig` ()
- void `importListGameConfig` ()
- void `exportToPdfLocale` (char \*filename, char \*export\_filename)
- void `changePdfPreferences` ()
- void `exportToCsvLocale` (char \*filename, char \*export\_filename)
- void `exportToGnuplotLocale` (char \*filename, char \*export\_filename)
- void `exportToMLocale` (char \*filename, char \*export\_filename)
- void `exportCsu` ()

### 2.4.1 Detailed Description

Graphical interface.

Author

Remi BERTHO

Date

01/09/14

Version

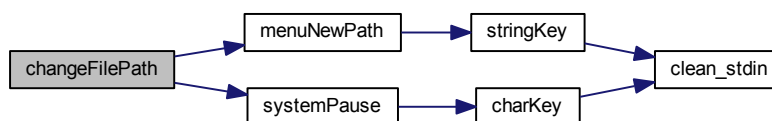
4.2.0

### 2.4.2 Function Documentation

#### 2.4.2.1 void `changeFilePath` ( )

Change the path which the file are saved.

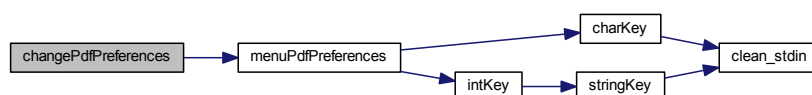
Here is the call graph for this function:



#### 2.4.2.2 void changePdfPreferences ( )

Change the export to pdf preferences

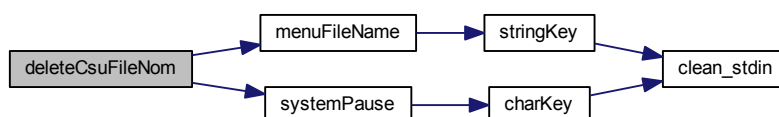
Here is the call graph for this function:



#### 2.4.2.3 void deleteCsuFileNom ( )

Ask a filename and deleted the file.

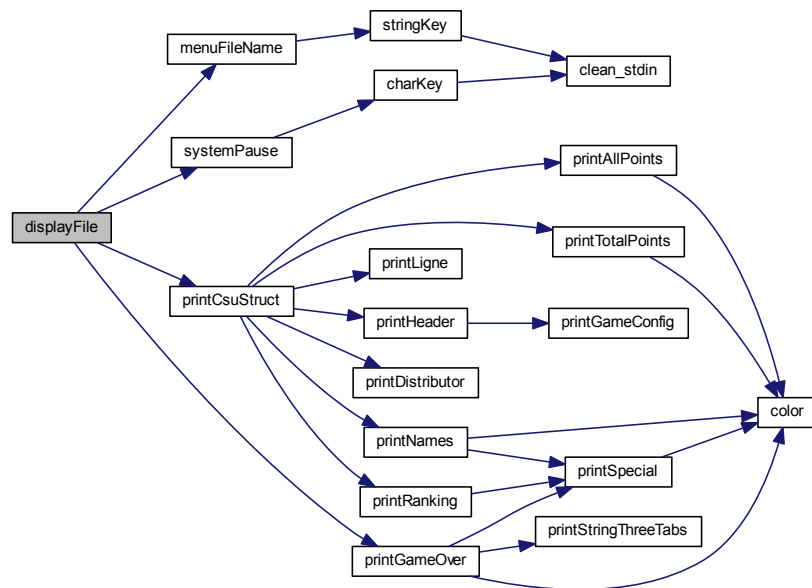
Here is the call graph for this function:



#### 2.4.2.4 void displayFile ( )

Ask a filename and display it.

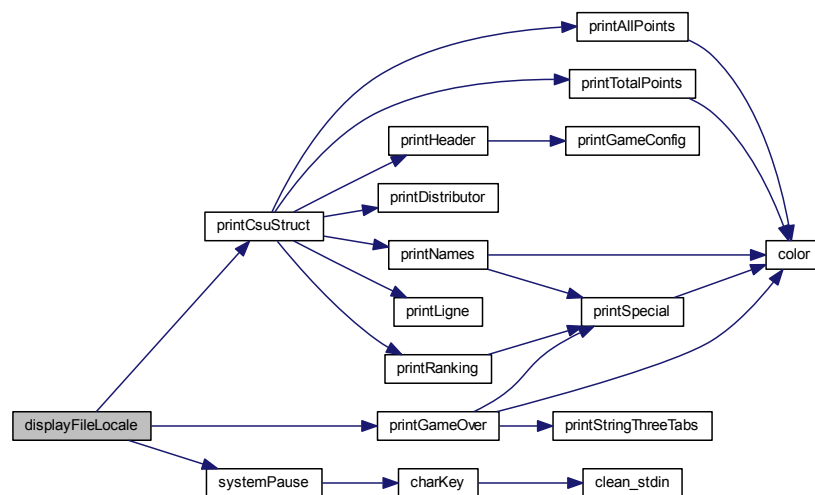
Here is the call graph for this function:



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

Display the file.

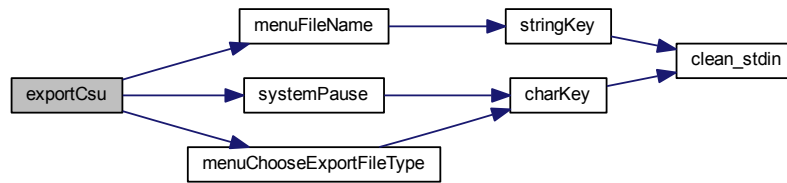
Here is the call graph for this function:



#### 2.4.2.6 void exportCsu ( )

Export a csu file into a csv or pdf file

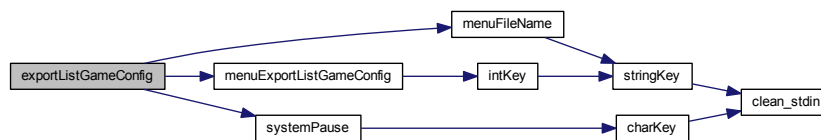
Here is the call graph for this function:



#### 2.4.2.7 void exportListGameConfig ( )

Export all game configuration in one file.

Here is the call graph for this function:



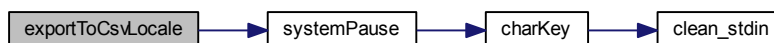
#### 2.4.2.8 void exportToCsvLocale ( char \* filename, char \* export\_filename )

Export the csu file named filename into a csv file named export\_filename

Parameters

in	<i>filename</i>	the csu filename
in	<i>export_filename</i>	the csv filename

Here is the call graph for this function:



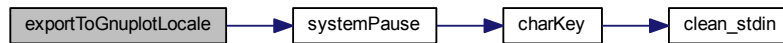
#### 2.4.2.9 void exportToGnuplotLocale ( char \* filename, char \* export\_filename )

Export the csu file named filename into gnuplot files named export\_filename

## Parameters

in	<i>filename</i>	the csu filename
in	<i>export_filename</i>	the filename

Here is the call graph for this function:



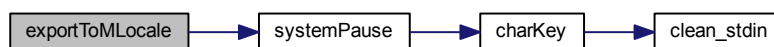
#### 2.4.2.10 void exportToMLocale ( char \* *filename*, char \* *export\_filename* )

Export the csu file named filename into a m file (octave/matlab)

## Parameters

in	<i>filename</i>	the csu filename
in	<i>export_filename</i>	

Here is the call graph for this function:



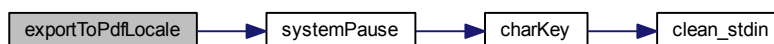
#### 2.4.2.11 void exportToPdfLocale ( char \* *filename*, char \* *export\_filename* )

Export the csu file named filename into a pdf file named export\_filename

## Parameters

in	<i>filename</i>	the csu filename
in	<i>export_filename</i>	the pdf filename

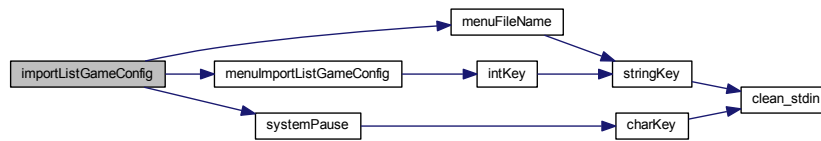
Here is the call graph for this function:



#### 2.4.2.12 void importListGameConfig ( )

Import all game configuration in one file.

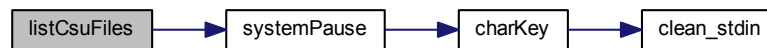
Here is the call graph for this function:



#### 2.4.2.13 void listCsuFiles ( )

List all the files with csu extension

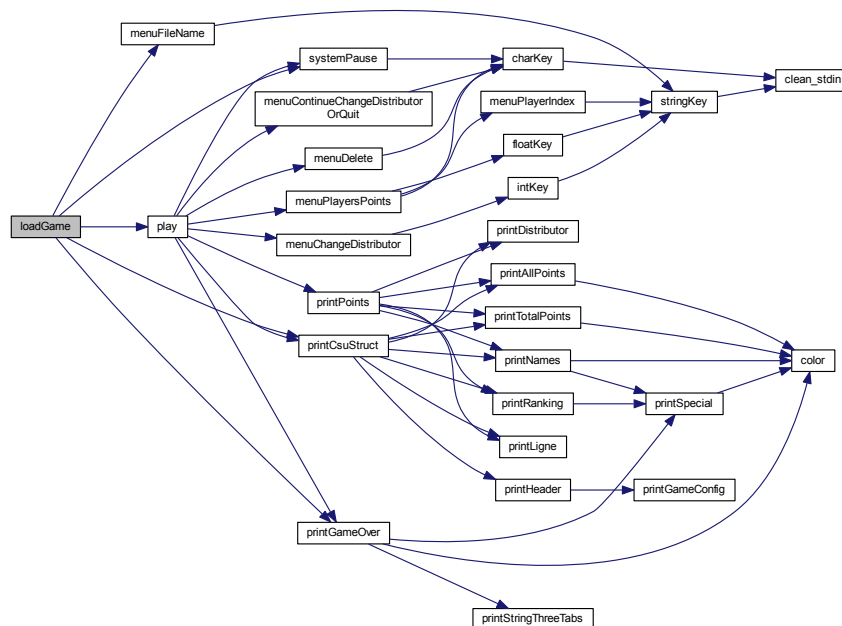
Here is the call graph for this function:



#### 2.4.2.14 void loadGame ( )

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

Here is the call graph for this function:





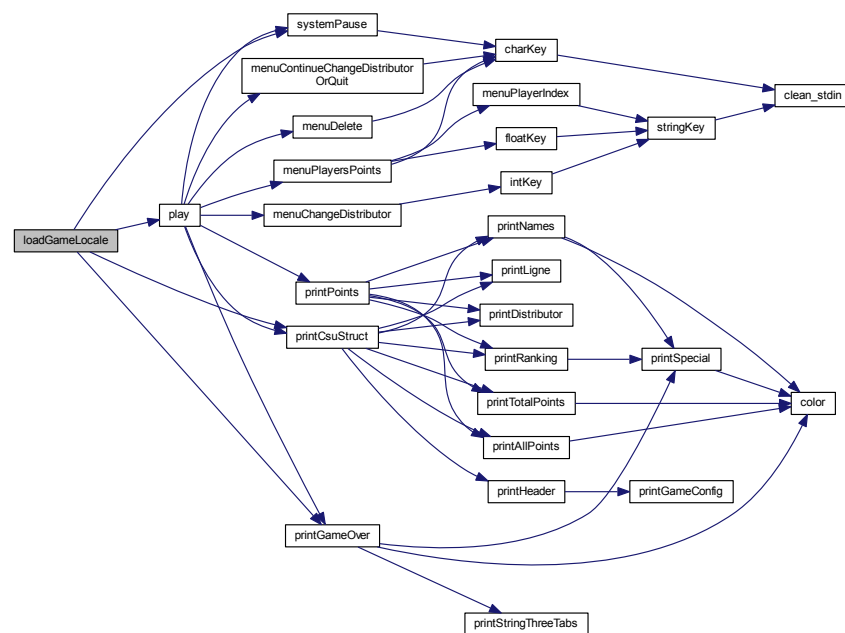
## 2.4.2.15 void loadGameLocale ( char \* file\_name )

Load the file and run the play function

Parameters

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

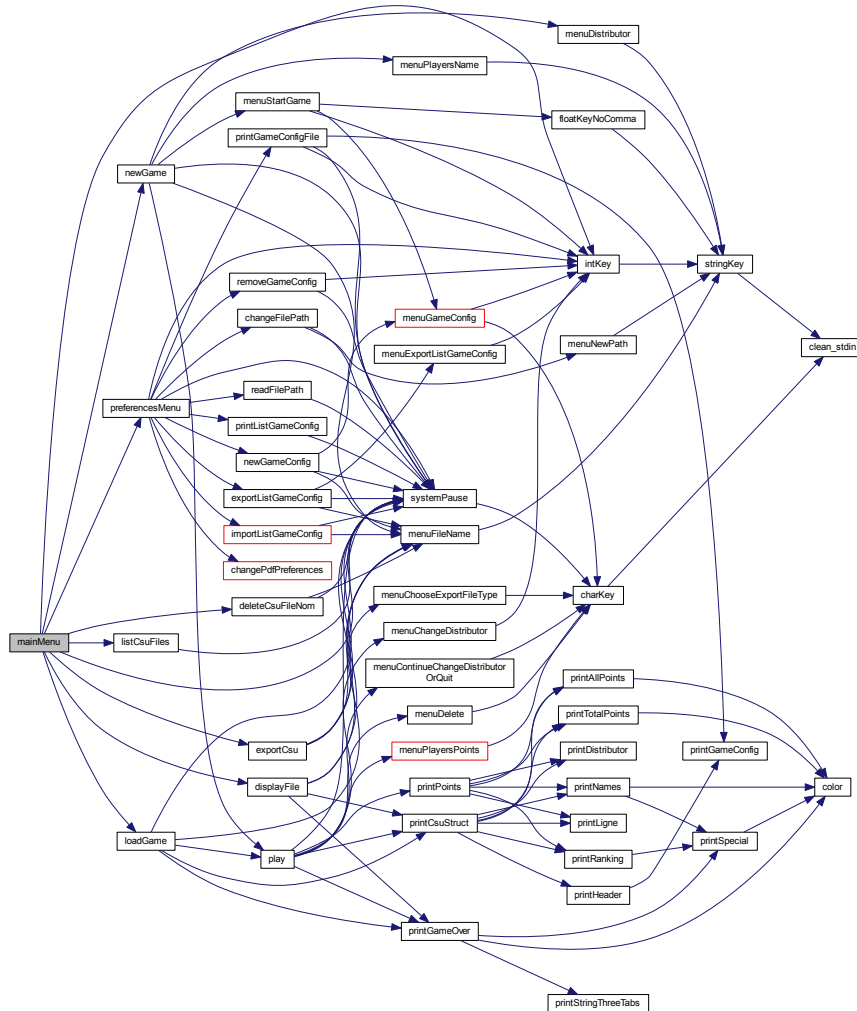
Here is the call graph for this function:



## 2.4.2.16 void mainMenu ( )

Main menu of csuper.

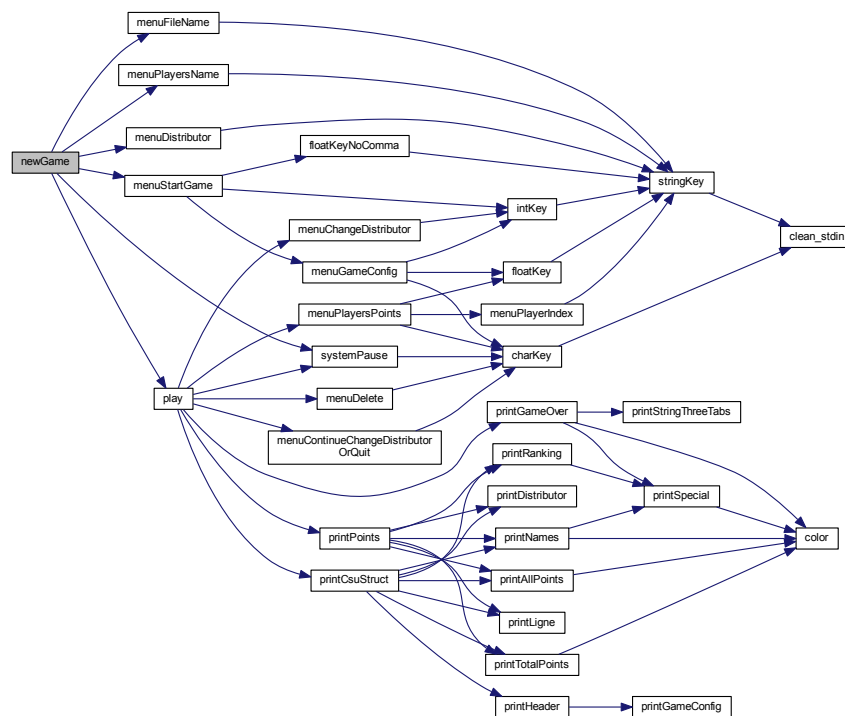
Here is the call graph for this function:



#### 2.4.2.17 void newGame ( )

Initialize a new game and run the play function.

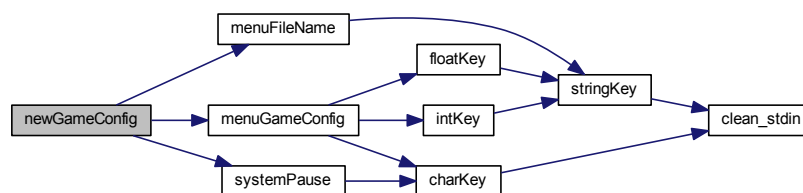
Here is the call graph for this function:



#### 2.4.2.18 void newGameConfig ( )

Add a new game configuration

Here is the call graph for this function:



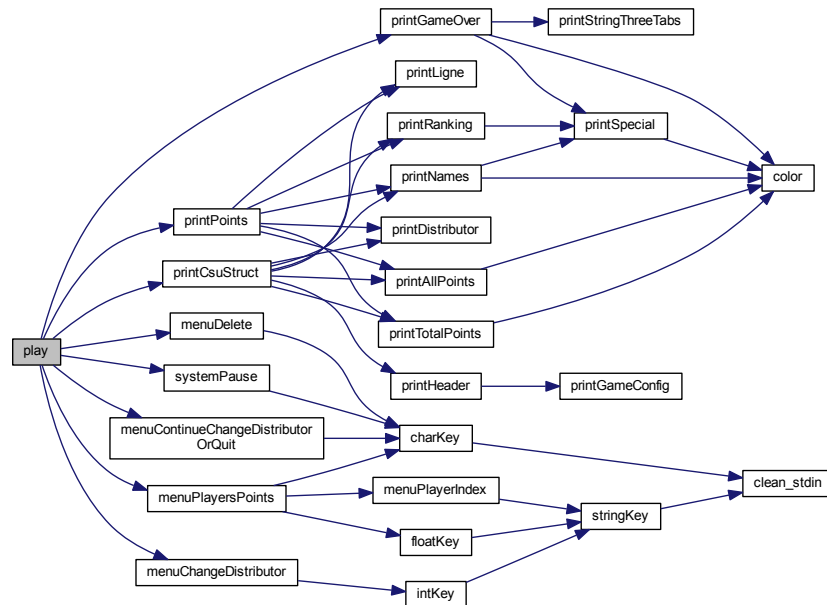
#### 2.4.2.19 void play ( csuStruct \* ptr\_csu\_struct, char \* file\_name )

Count the points

Parameters

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

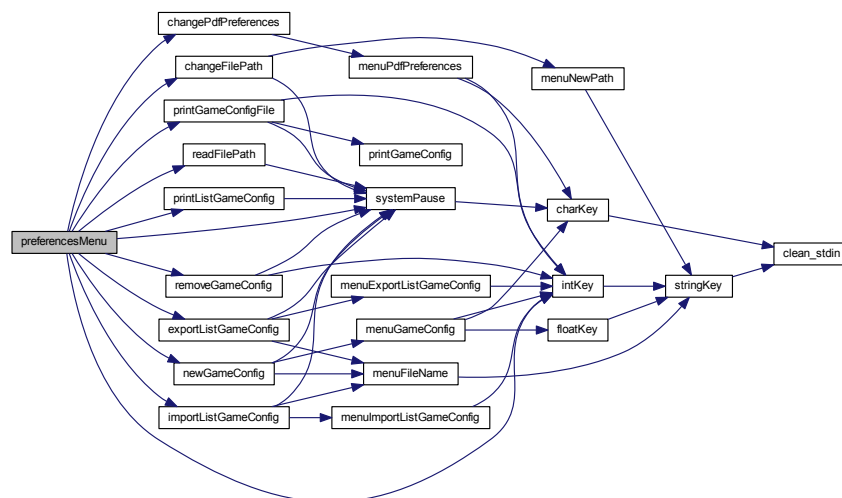
Here is the call graph for this function:



#### 2.4.2.20 void preferencesMenu ( )

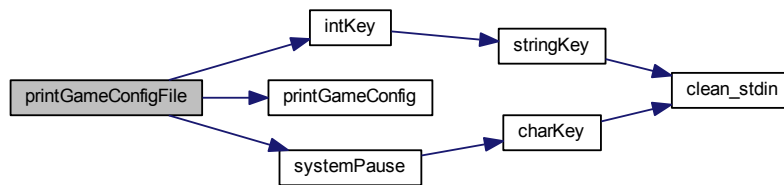
Preferences menu of csuper.

Here is the call graph for this function:



#### 2.4.2.21 void printGameConfigFile ( )

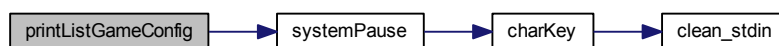
Here is the call graph for this function:



#### 2.4.2.22 void printListGameConfig ( )

Print the list of game configuration

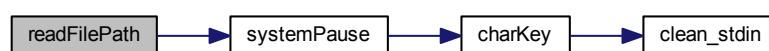
Here is the call graph for this function:



#### 2.4.2.23 void readFilePath ( )

Read the file path and display it.

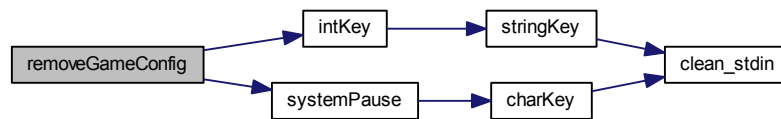
Here is the call graph for this function:



#### 2.4.2.24 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, `deleteFiles` =4,  
`listFile` =5, `export_file` =6, `pref` =7, `quit` =8,  
`easterEggs` = 42 }
- enum `PreferencesMenu` {  
`newPath` =1, `readPath` =2, `newGameConf` =3, `removeGameConf` =4,  
`printListGameConf` =5, `printGameConf` =6, `exportGameConf` =7, `importGameConf` =8,  
`pdfPreferences` =9, `backMainMenu` =10, `easterEggs2` = 42 }
- enum `FileType` {  
`csv_file`, `pdf_file`, `gnuplot_file`, `m_file`,  
`csu_file` }

### 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](#) ()
- void [exportListGameConfig](#) ()
- void [importListGameConfig](#) ()
- void [exportToPdfLocale](#) (char \*filename, char \*export\_filename)
- void [changePdfPreferences](#) ()
- void [exportToCsvLocale](#) (char \*filename, char \*export\_filename)
- void [exportToGnuplotLocale](#) (char \*filename, char \*export\_filename)
- void [exportToMLocale](#) (char \*filename, char \*export\_filename)
- void [exportCsu](#) ()

### 2.5.1 Detailed Description

Graphical interface.

Author

Remi BERTHO

Date

17/06/14

Version

4.0.0

### 2.5.2 Enumeration Type Documentation

#### 2.5.2.1 enum FileType

Enumerator

***csv\_file***  
***pdf\_file***  
***gnuplot\_file***  
***m\_file***  
***csu\_file***

#### 2.5.2.2 enum MainMenu

Enumerator

***newMatch***  
***loadMatch***  
***printFile***  
***deleteFiles***  
***listFile***  
***export\_file***  
***pref***  
***quit***  
***easterEggs***

### 2.5.2.3 enum PreferencesMenu

Enumerator

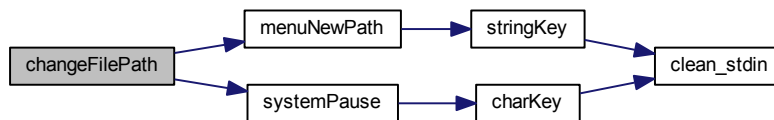
*newPath*  
*readPath*  
*newGameConf*  
*removeGameConf*  
*printListGameConf*  
*printGameConf*  
*exportGameConf*  
*importGameConf*  
*pdfPreferences*  
*backMainMenu*  
*easterEggs2*

## 2.5.3 Function Documentation

### 2.5.3.1 void changeFilePath ( )

Change the path which the file are saved.

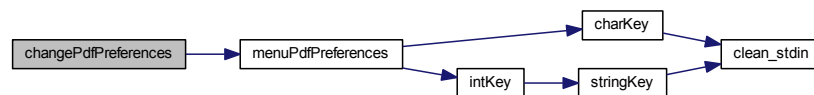
Here is the call graph for this function:



### 2.5.3.2 void changePdfPreferences ( )

Change the export to pdf preferences

Here is the call graph for this function:

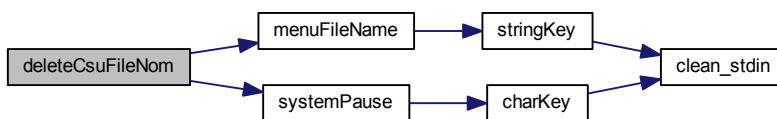


### 2.5.3.3 void deleteCsuFileNom ( )

Ask a filename and deleted the file.



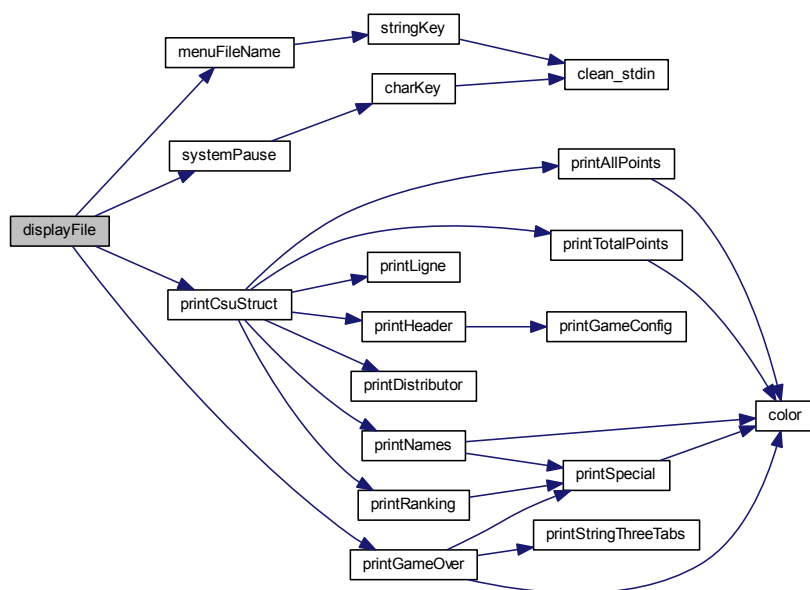
Here is the call graph for this function:



#### 2.5.3.4 void displayFile ( )

Ask a filename and display it.

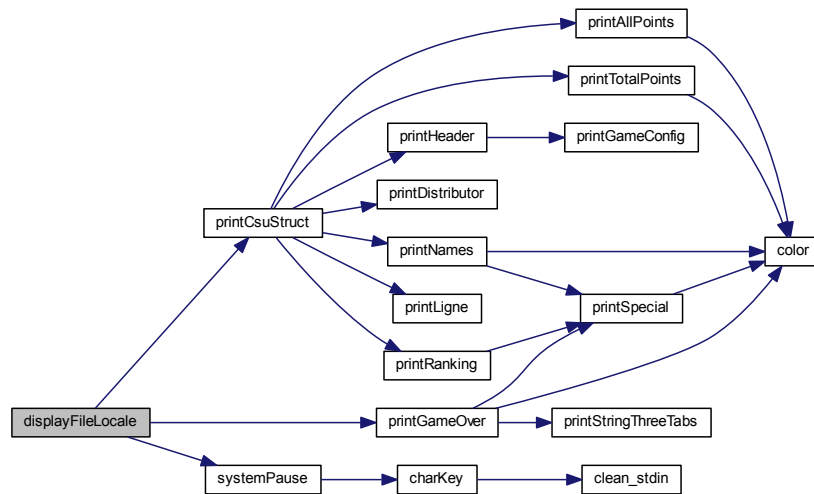
Here is the call graph for this function:



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

Display the file.

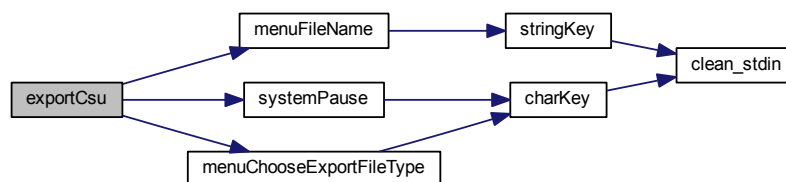
Here is the call graph for this function:



#### 2.5.3.6 void exportCsu ( )

Export a csu file into a csv or pdf file

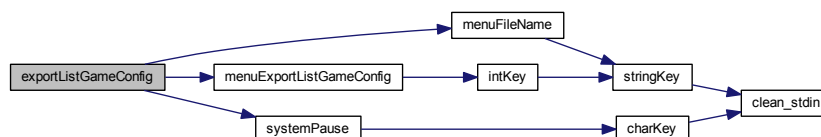
Here is the call graph for this function:



#### 2.5.3.7 void exportListGameConfig ( )

Export all game configuration in one file.

Here is the call graph for this function:



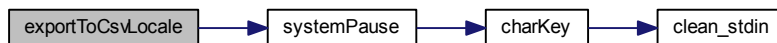
### 2.5.3.8 void exportToCsvLocale ( char \* filename, char \* export\_filename )

Export the csu file named filename into a csv file named export\_filename

#### Parameters

in	<i>filename</i>	the csu filename
in	<i>export_filename</i>	the csv filename

Here is the call graph for this function:



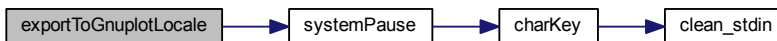
### 2.5.3.9 void exportToGnuplotLocale ( char \* filename, char \* export\_filename )

Export the csu file named filename into gnuplot files named export\_filename

#### Parameters

in	<i>filename</i>	the csu filename
in	<i>export_filename</i>	the filename

Here is the call graph for this function:



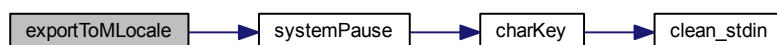
### 2.5.3.10 void exportToMLocale ( char \* filename, char \* export\_filename )

Export the csu file named filename into a m file (octave/matlab)

#### Parameters

in	<i>filename</i>	the csu filename
in	<i>export_filename</i>	

Here is the call graph for this function:



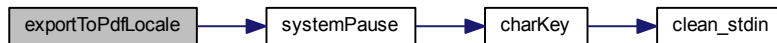
2.5.3.11 `void exportToPdfLocale ( char * filename, char * export_filename )`

Export the csu file named `filename` into a pdf file named `export_filename`

## Parameters

in	<i>filename</i>	the csu filename
in	<i>export_filename</i>	the pdf filename

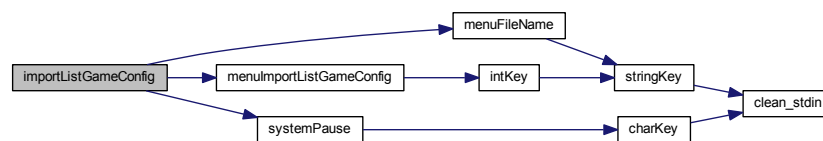
Here is the call graph for this function:



## 2.5.3.12 void importListGameConfig ( )

Import all game configuration in one file.

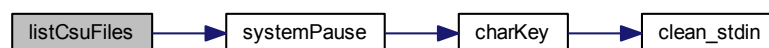
Here is the call graph for this function:



## 2.5.3.13 void listCsuFiles ( )

List all the files with csu extension

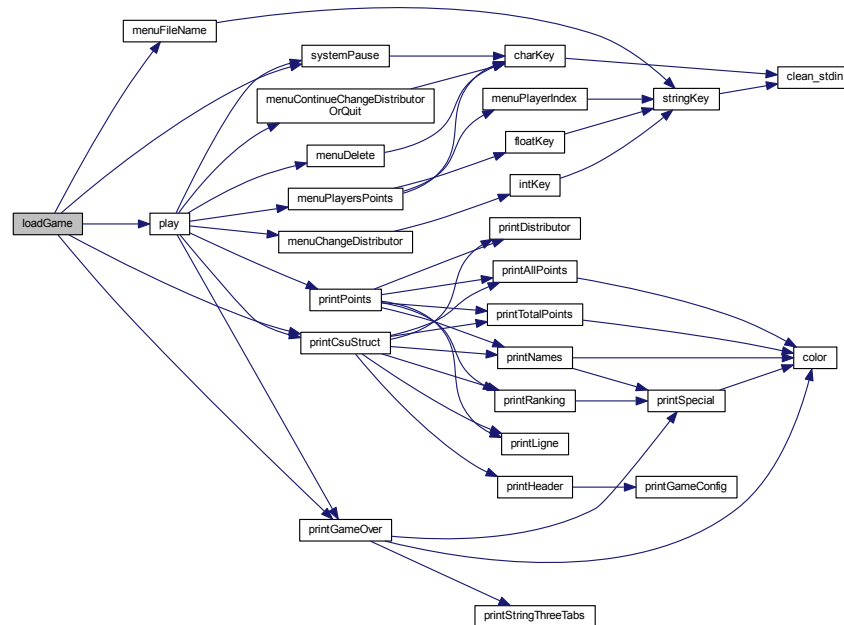
Here is the call graph for this function:



## 2.5.3.14 void loadGame ( )

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

Here is the call graph for this function:



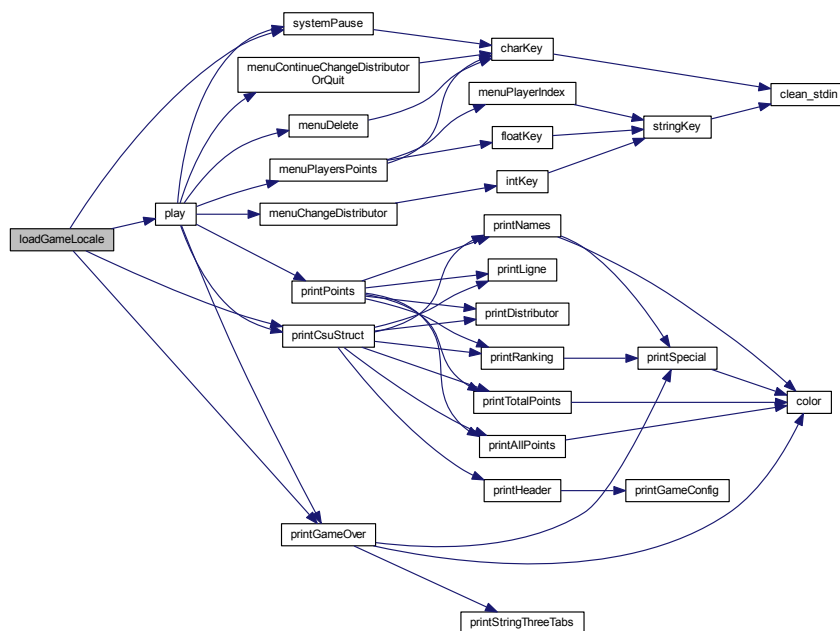
#### 2.5.3.15 void loadGameLocale ( char \* file\_name )

Load the file and run the play function

##### Parameters

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

Here is the call graph for this function:



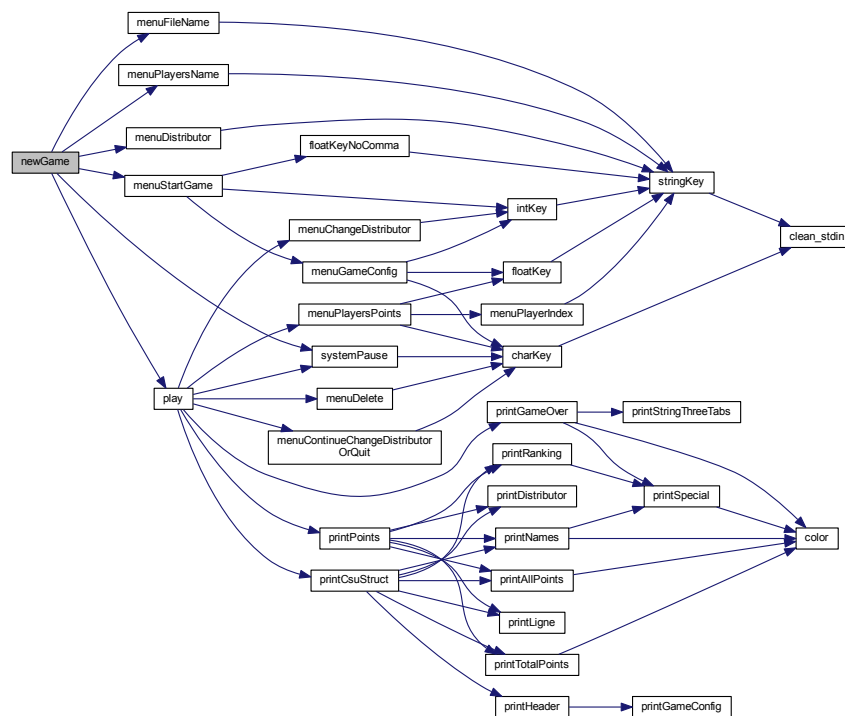
### 2.5.3.16 void mainMenu ( )

Main menu of csuper.





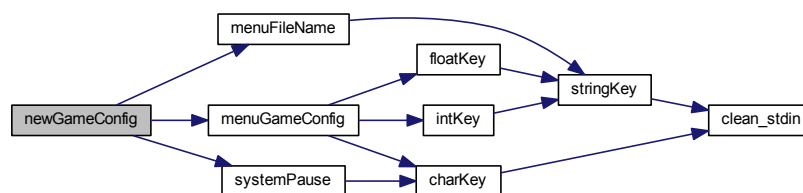
Here is the call graph for this function:



### 2.5.3.18 void newGameConfig ( )

Add a new game configuration

Here is the call graph for this function:



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

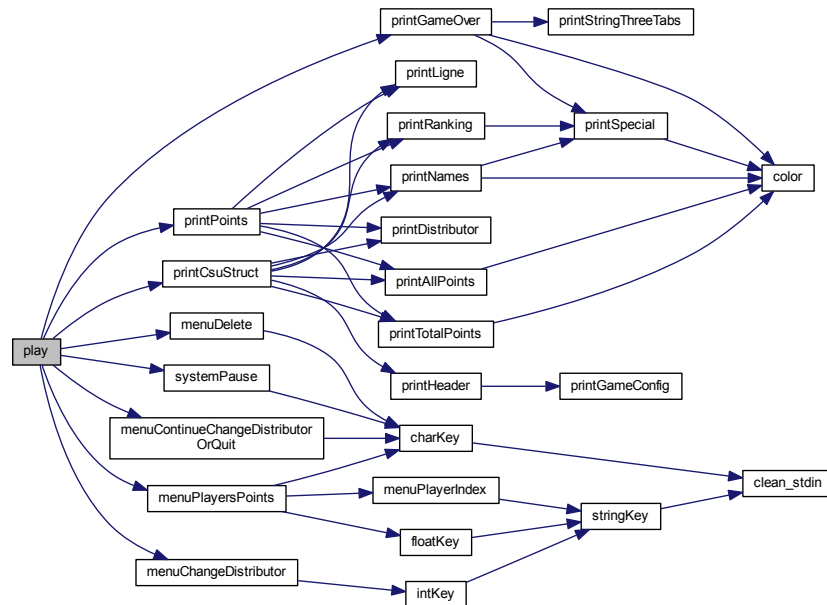
Count the points

Parameters

---

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

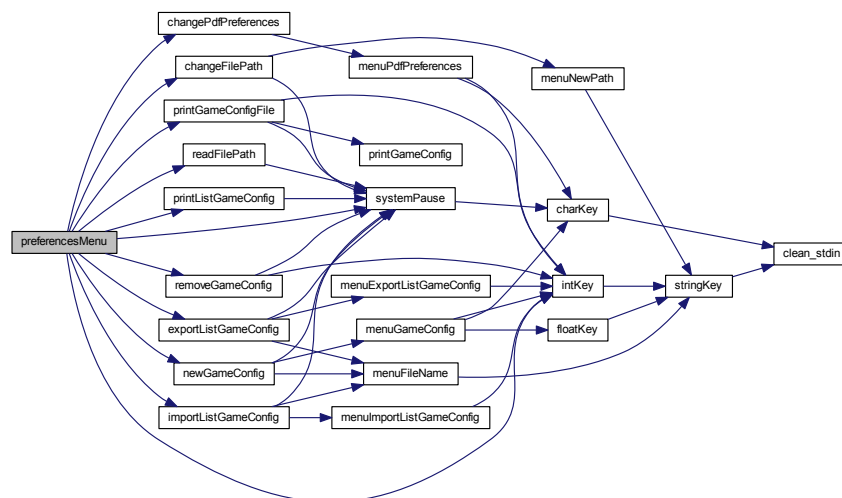
Here is the call graph for this function:



### 2.5.3.20 void preferencesMenu ( )

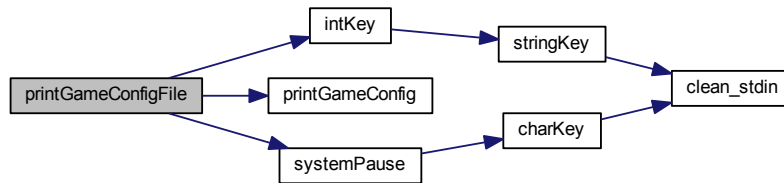
Preferences menu of csuper.

Here is the call graph for this function:



### 2.5.3.21 void printGameConfigFile ( )

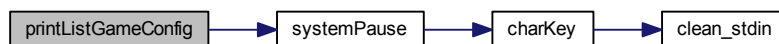
Here is the call graph for this function:



### 2.5.3.22 void printListGameConfig ( )

Print the list of game configuration

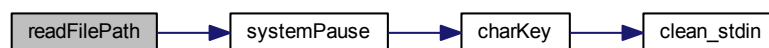
Here is the call graph for this function:



### 2.5.3.23 void readFilePath ( )

Read the file path and display it.

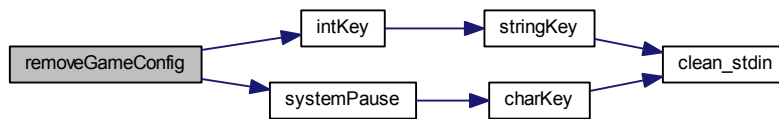
Here is the call graph for this function:



### 2.5.3.24 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)
- char \* `charKey` (char \*c)
- void `systemPause` ()

### 2.6.1 Detailed Description

Function of keyboarding.

#### Author

Remi BERTHO

#### Date

26/04/14

#### 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 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:



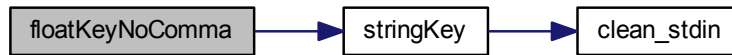
## 2.6.2.4 void floatKeyNoComma ( float \* nb )

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:



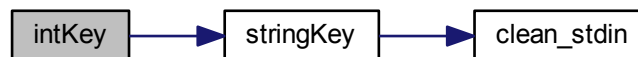
#### 2.6.2.5 void \* intKey ( int \* nb )

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:



#### 2.6.2.6 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	*string	a string
in	nb_char_plus_one ↔ one	the number of characters that the function will read plus one

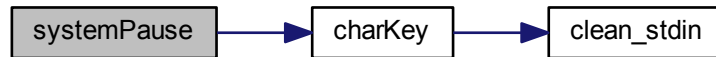
Here is the call graph for this function:



### 2.6.2.7 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_EXPRESSION` 256
- `#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)
- char \* `charKey` (char \*c)
- void `systemPause` ()

### 2.7.1 Detailed Description

Function of keyboarding.

Author

Remi BERTHO

## Date

26/04/14

## 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 EXPRESSION 256

Define the number of characters maximum that can contain a mathematical expression

### 2.7.2.3 #define NB\_CHARACTER FLOAT 39

Define the number of characters that need a float.

### 2.7.2.4 #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 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:

**2.7.3.4 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:

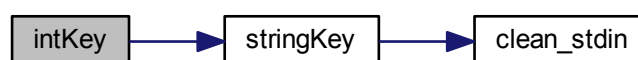
**2.7.3.5 void intKey ( int \* nb )**

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:



2.7.3.6 `char* stringKey ( char * string, int nb_char_plus_one )`

Do a keyboarding of a string with nb-char\_plus\_one minus 1 characters

**Parameters**

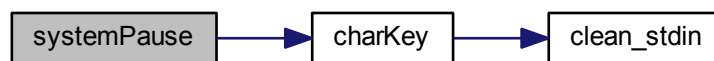
<code>in, out</code>	<i>*string</i>	a string
<code>in</code>	<i>nb_char_plus_one</i>	the number of characters that the function will read plus one

Here is the call graph for this function:

**2.7.3.7 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.

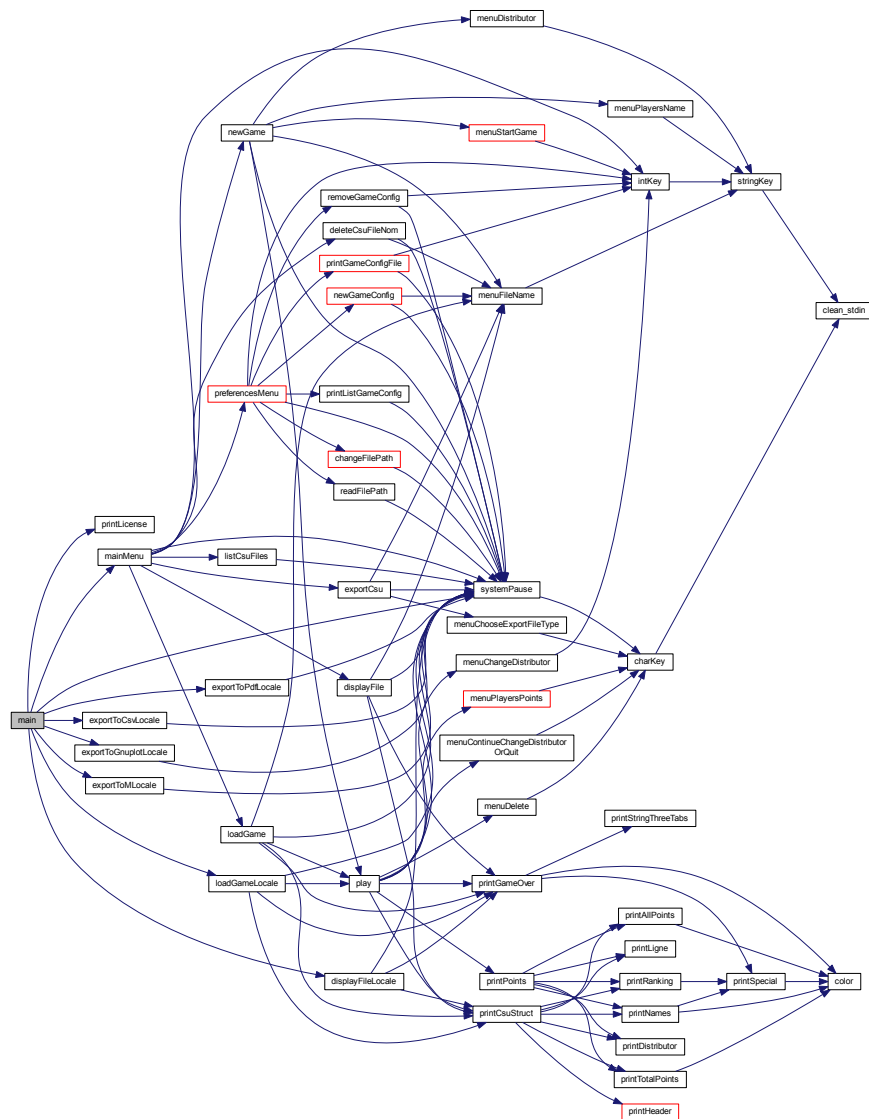
## Parameters

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"
```

### Macros

- `#define` [PORTABLE](#)

### 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 Macro Definition Documentation

### 2.9.2.1 #define PORTABLE

Define if csuper is portable.

## 2.9.3 Function Documentation

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

Begin csuper.

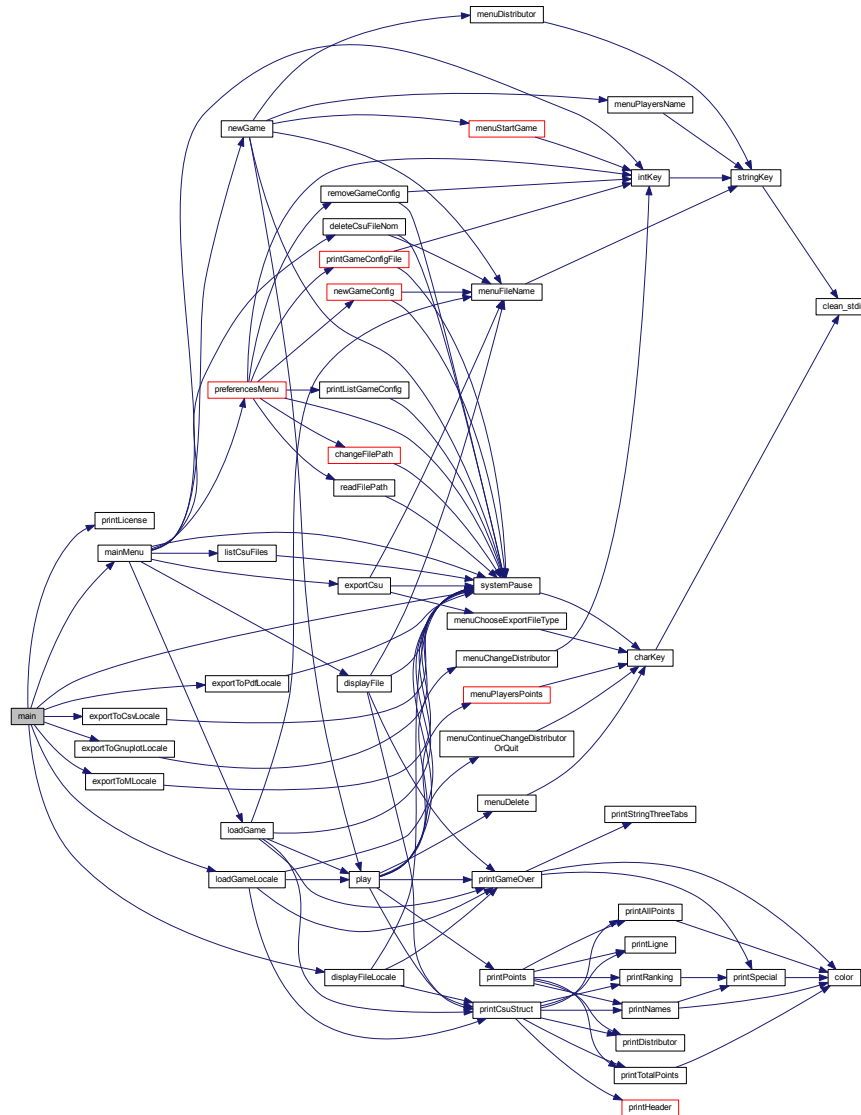
#### Parameters

<i>in</i>	<i>argc</i>	the number of argument.
<i>in</i>	<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)
- [ContinueChangeDistributorOrQuit](#) [menuContinueChangeDistributorOrQuit](#) ( )
- void [menuChangeDistributor](#) (csuStruct \*ptr\_csu\_struct)
- bool [menuDelete](#) ( )
- void [menuNewPath](#) (char \*new\_path)
- bool [menuExportListGameConfig](#) (int \*\*id, int \*nb\_id)
- bool [menuImportListGameConfig](#) (int \*\*id, int \*nb\_id, char \*filename)
- void [menuPdfPreferences](#) (export\_pdf\_preferences \*pref)
- [FileType](#) [menuChooseExportFileType](#) ( )

### 2.10.1 Detailed Description

Menu functions.

Author

Remi BERTHO

Date

01/09/14

Version

4.2.0

### 2.10.2 Function Documentation

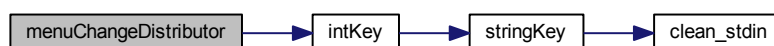
#### 2.10.2.1 void [menuChangeDistributor](#) ( csuStruct \* *ptr\_csu\_struct* )

Ask and validate the new distributor

Parameters

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

Here is the call graph for this function:



#### 2.10.2.2 [FileType](#) [menuChooseExportFileType](#) ( )

Choose the file type which will be exported

Here is the call graph for this function:



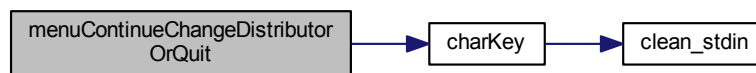
#### 2.10.2.3 `ContinueChangeDistributorOrQuit menuContinueChangeDistributorOrQuit ( )`

Ask if we want to continue

Returns

a `ContinueChangeDistributorOrQuit`

Here is the call graph for this function:



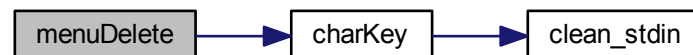
#### 2.10.2.4 `bool menuDelete ( )`

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:



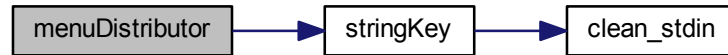
#### 2.10.2.5 `void menuDistributor ( char * distributor_name )`

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:

2.10.2.6 `bool menuExportListGameConfig ( int ** id, int * nb_id )`

Choose the game configuration which will be exported

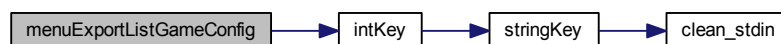
## Parameters

<code>in</code>	<code>id</code>	the id of the game configuration which will be exported
<code>in</code>	<code>nb_id</code>	the number of game configuration which will be exported

## Returns

true if there is no problem, false otherwise

Here is the call graph for this function:

2.10.2.7 `char * menuFileName ( char file_name[SIZE_MAX_FILE_NAME] )`

Ask and save the filename.

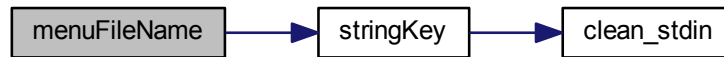
## Parameters

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

**Returns**

the filename

Here is the call graph for this function:



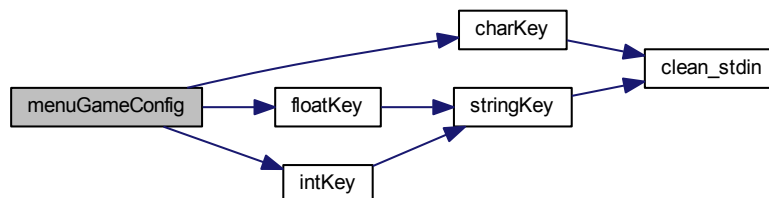
### 2.10.2.8 void menuGameConfig ( game\_config \* ptr\_config )

Ask and save a game configuration.

**Parameters**

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

Here is the call graph for this function:



### 2.10.2.9 bool menuImportListGameConfig ( int \*\* id, int \* nb\_id, char \* filename )

Choose the game configuration which will be imported

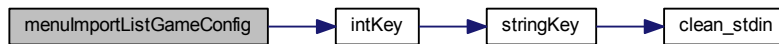
**Parameters**

in	<i>filename</i>	the filename of the imported file.
in	<i>id</i>	the id of the game configuration which will be imported
in	<i>nb_id</i>	the number of game configuration which will be imported

**Returns**

true if there is no problem, false otherwise

Here is the call graph for this function:

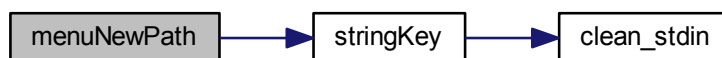
**2.10.2.10 void menuNewPath ( char \* new\_path )**

Ask and save the new path

**Parameters**

in, out	*new_path	the new path
---------	-----------	--------------

Here is the call graph for this function:

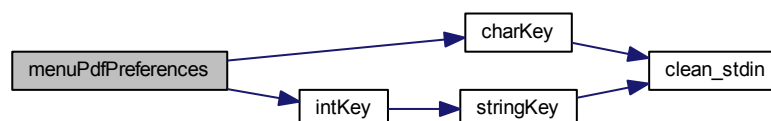
**2.10.2.11 void menuPdfPreferences ( export\_pdf\_preferences \* pref )**

Fill a export pdf preferences structure

**Parameters**

in	pref	a pointer on a export_pdf_preferences
----	------	---------------------------------------

Here is the call graph for this function:



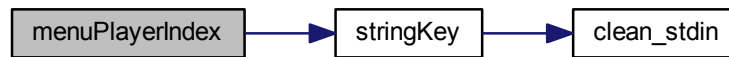
2.10.2.12 `int menuPlayerIndex ( csuStruct * ptr_csu_struct )`

Ask a player name and save his index

## Parameters

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

Here is the call graph for this function:



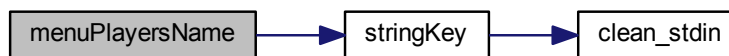
### 2.10.2.13 void menuPlayersName ( csuStruct \* ptr\_csu\_struct )

Ask and save the name of the players.

## Parameters

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

Here is the call graph for this function:



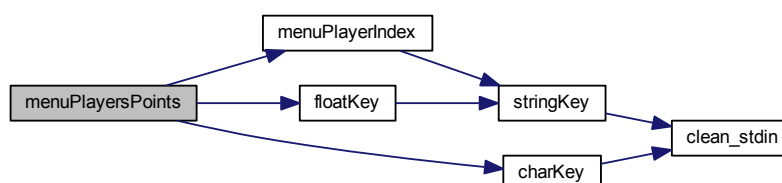
### 2.10.2.14 void menuPlayersPoints ( csuStruct \* ptr\_csu\_struct )

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

## Parameters

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

Here is the call graph for this function:



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

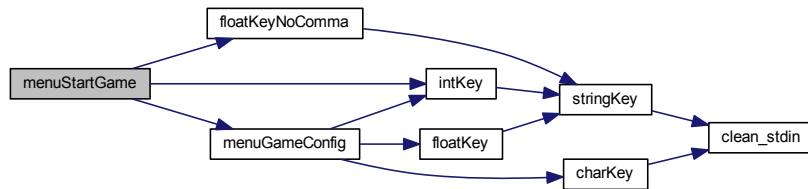
Ask and save the number of player and all the informations of a game configuration.



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

### Enumerations

- enum [ContinueChangeDistributorOrQuit](#) { [Continue](#), [ChangeDistributor](#), [Quit](#) }

### 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)
- [ContinueChangeDistributorOrQuit](#) [menuContinueChangeDistributorOrQuit](#) ()
- void [menuChangeDistributor](#) (csuStruct \*ptr\_csu\_struct)
- bool [menuDelete](#) ()
- void [menuNewPath](#) (char \*new\_path)
- bool [menuExportListGameConfig](#) (int \*\*id, int \*nb\_id)
- bool [menuImportListGameConfig](#) (int \*\*id, int \*nb\_id, char \*filename)
- void [menuPdfPreferences](#) (export\_pdf\_preferences \*pref)
- [FileType](#) [menuChooseExportFileType](#) ()

#### 2.11.1 Detailed Description

Menu functions.

**Author**

Remi BERTHO

**Date**

01/09/14

**Version**

4.2.0

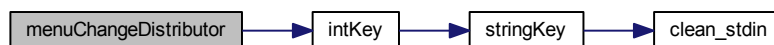
**2.11.2 Enumeration Type Documentation****2.11.2.1 enum ContinueChangeDistributorOrQuit****Enumerator*****Continue******ChangeDistributor******Quit*****2.11.3 Function Documentation****2.11.3.1 void menuChangeDistributor ( csuStruct \* ptr\_csu\_struct )**

Ask and validate the new distributor

**Parameters**

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

Here is the call graph for this function:

**2.11.3.2 FileType menuChooseExportFileType ( )**

Choose the file type which will be exported

Here is the call graph for this function:



### 2.11.3.3 ContinueChangeDistributorOrQuit menuContinueChangeDistributorOrQuit ( )

Ask if we want to continue

#### Returns

a ContinueChangeDistributorOrQuit

Here is the call graph for this function:



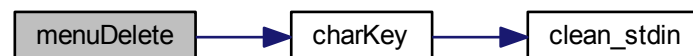
### 2.11.3.4 bool menuDelete ( )

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:



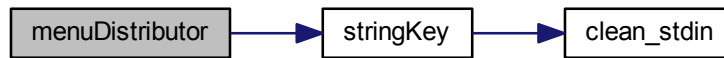
### 2.11.3.5 void menuDistributor ( char \* distributor\_name )

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:



#### 2.11.3.6 `bool menuExportListGameConfig ( int ** id, int * nb_id )`

Choose the game configuration which will be exported

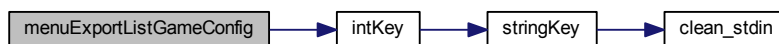
##### Parameters

in	<i>id</i>	the id of the game configuration which will be exported
in	<i>nb_id</i>	the number of game configuration which will be exported

##### Returns

true if there is no problem, false otherwise

Here is the call graph for this function:



#### 2.11.3.7 `char* menuFileName ( char file_name[SIZE_MAX_FILE_NAME] )`

Ask and save the filename.

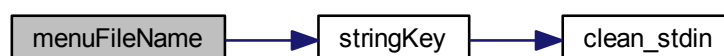
##### Parameters

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

##### Returns

the filename

Here is the call graph for this function:



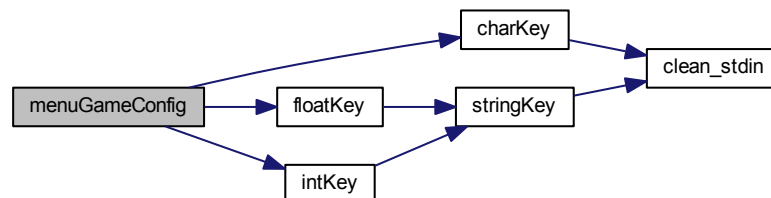
### 2.11.3.8 void menuGameConfig ( game\_config \* ptr\_config )

Ask and save a game configuration.

#### Parameters

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

Here is the call graph for this function:



### 2.11.3.9 bool menuImportListGameConfig ( int \*\* id, int \* nb\_id, char \* filename )

Choose the game configuration which will be imported

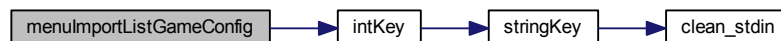
#### Parameters

in	<i>filename</i>	the filename of the imported file.
in	<i>id</i>	the id of the game configuration which will be imported
in	<i>nb_id</i>	the number of game configuration which will be imported

#### Returns

true if there is no problem, false otherwise

Here is the call graph for this function:



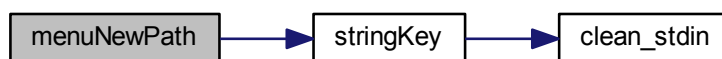
### 2.11.3.10 void menuNewPath ( char \* new\_path )

Ask and save the new path

#### Parameters

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

Here is the call graph for this function:



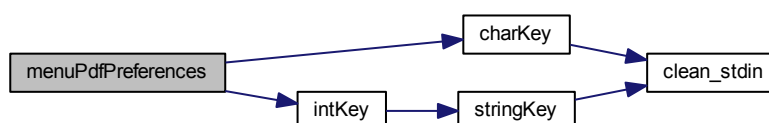
#### 2.11.3.11 void menuPdfPreferences ( export\_pdf\_preferences \* *pref* )

Fill a export pdf preferences structure

##### Parameters

<i>in</i>	<i>pref</i>	a pointer on a export_pdf_preferences
-----------	-------------	---------------------------------------

Here is the call graph for this function:



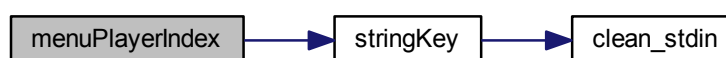
#### 2.11.3.12 int menuPlayerIndex ( csuStruct \* *ptr\_csu\_struct* )

Ask a player name and save his index

##### Parameters

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

Here is the call graph for this function:



2.11.3.13 void menuPlayersName ( csuStruct \* *ptr\_csu\_struct* )

Ask and save the name of the players.

## Parameters

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

Here is the call graph for this function:



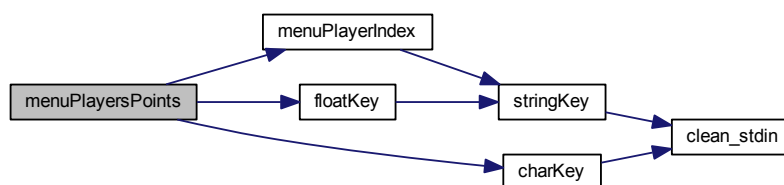
#### 2.11.3.14 void menuPlayersPoints ( csuStruct \* ptr\_csu\_struct )

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:



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

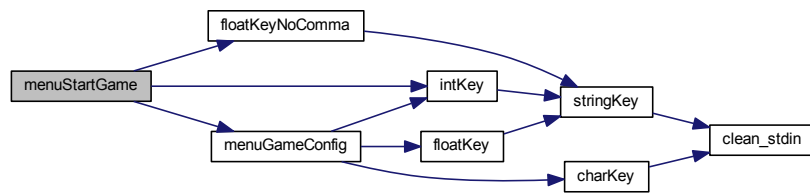
Ask and save the number of player and all the informations of a game configuration.

## 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:





# Index

—  
    csuper.h, 3

backMainMenu  
    interface.h, 32

backgroundBlack  
    display.h, 11

backgroundBlue  
    display.h, 11

backgroundBrown  
    display.h, 11

backgroundColor  
    display.h, 11

backgroundCyan  
    display.h, 11

backgroundDefault  
    display.h, 11

backgroundGreen  
    display.h, 11

backgroundMagenta  
    display.h, 11

backgroundRed  
    display.h, 11

backgroundWhite  
    display.h, 11

ChangeDistributor  
    menu.h, 66

changeFilePath  
    interface.c, 19  
    interface.h, 32

changePdfPreferences  
    interface.c, 19  
    interface.h, 32

charKey  
    keyboarding.c, 44  
    keyboarding.h, 48

clean\_stdin  
    keyboarding.c, 45  
    keyboarding.h, 48

color  
    display.c, 4  
    display.h, 12

Continue  
    menu.h, 66

ContinueChangeDistributorOrQuit  
    menu.h, 66

csu\_file  
    interface.h, 31

csuper.h, 3

csv\_file  
    interface.h, 31

deleteCsuFileNom  
    interface.c, 20  
    interface.h, 32

deleteFiles  
    interface.h, 31

display.c, 3  
    color, 4  
    printAllPoints, 4  
    printCsuStruct, 5  
    printDistributor, 5  
    printGameConfig, 5  
    printGameOver, 6  
    printHeader, 6  
    printLicense, 6  
    printLigne, 6  
    printNames, 7  
    printPoints, 7  
    printRanking, 7  
    printSpecial, 9  
    printStringThreeTabs, 9  
    printTotalPoints, 9

display.h, 10  
    backgroundBlack, 11  
    backgroundBlue, 11  
    backgroundBrown, 11  
    backgroundColor, 11  
    backgroundCyan, 11  
    backgroundDefault, 11  
    backgroundGreen, 11  
    backgroundMagenta, 11  
    backgroundRed, 11  
    backgroundWhite, 11  
    color, 12  
    foregroundBlack, 11  
    foregroundBlue, 11  
    foregroundBrown, 11  
    foregroundColor, 11  
    foregroundCyan, 11  
    foregroundGreen, 11  
    foregroundMagenta, 11  
    foregroundRed, 11  
    foregroundWhite, 11  
    printAllPoints, 13  
    printCsuStruct, 13  
    printDistributor, 13  
    printGameConfig, 15

- printGameOver, 15
- printHeader, 15
- printLicense, 15
- printLigne, 16
- printNames, 16
- printPoints, 16
- printRanking, 17
- printSpecial, 17
- printStringThreeTabs, 18
- printTotalPoints, 18
- Writing, 11
- writingBold, 11
- writingReset, 11
- writingUnderline, 11
- displayFile
  - interface.c, 20
  - interface.h, 33
- displayFileLocale
  - interface.c, 21
  - interface.h, 33
- easterEggs
  - interface.h, 31
- easterEggs2
  - interface.h, 32
- export\_file
  - interface.h, 31
- exportCsu
  - interface.c, 21
  - interface.h, 34
- exportGameConf
  - interface.h, 32
- exportListGameConfig
  - interface.c, 22
  - interface.h, 34
- exportToCsvLocale
  - interface.c, 22
  - interface.h, 34
- exportToGnuplotLocale
  - interface.c, 22
  - interface.h, 35
- exportToMLocale
  - interface.c, 23
  - interface.h, 35
- exportToPdfLocale
  - interface.c, 23
  - interface.h, 35
- FileType
  - interface.h, 31
- floatKey
  - keyboarding.c, 45
  - keyboarding.h, 48
- floatKeyNoComma
  - keyboarding.c, 45
  - keyboarding.h, 50
- foregroundBlack
  - display.h, 11
- foregroundBlue
  - display.h, 11
- foregroundBrown
  - display.h, 11
- foregroundColor
  - display.h, 11
- foregroundCyan
  - display.h, 11
- foregroundGreen
  - display.h, 11
- foregroundMagenta
  - display.h, 11
- foregroundRed
  - display.h, 11
- foregroundWhite
  - display.h, 11
- gnuplot\_file
  - interface.h, 31
- importGameConf
  - interface.h, 32
- importListGameConfig
  - interface.c, 23
  - interface.h, 37
- intKey
  - keyboarding.c, 46
  - keyboarding.h, 50
- interface.c, 18
  - changeFilePath, 19
  - changePdfPreferences, 19
  - deleteCsuFileNom, 20
  - displayFile, 20
  - displayFileLocale, 21
  - exportCsu, 21
  - exportListGameConfig, 22
  - exportToCsvLocale, 22
  - exportToGnuplotLocale, 22
  - exportToMLocale, 23
  - exportToPdfLocale, 23
  - importListGameConfig, 23
  - listCsuFiles, 24
  - loadGame, 24
  - loadGameLocale, 24
  - mainMenu, 25
  - newGame, 26
  - newGameConfig, 27
  - play, 27
  - preferencesMenu, 28
  - printGameConfigFile, 28
  - printListGameConfig, 29
  - readFilePath, 29
  - removeGameConfig, 29
- interface.h, 30
  - backMainMenu, 32
  - changeFilePath, 32
  - changePdfPreferences, 32
  - csu\_file, 31
  - csv\_file, 31
  - deleteCsuFileNom, 32

- deleteFiles, 31
  - displayFile, 33
  - displayFileLocale, 33
  - easterEggs, 31
  - easterEggs2, 32
  - export\_file, 31
  - exportCsu, 34
  - exportGameConf, 32
  - exportListGameConfig, 34
  - exportToCsvLocale, 34
  - exportToGnuplotLocale, 35
  - exportToMLocale, 35
  - exportToPdfLocale, 35
  - FileType, 31
  - gnuplot\_file, 31
  - importGameConf, 32
  - importListGameConfig, 37
  - listCsuFiles, 37
  - listFile, 31
  - loadGame, 37
  - loadGameLocale, 38
  - loadMatch, 31
  - m\_file, 31
  - MainMenu, 31
  - mainMenu, 39
  - newGame, 40
  - newGameConf, 32
  - newGameConfig, 41
  - newMatch, 31
  - newPath, 32
  - pdf\_file, 31
  - pdfPreferences, 32
  - play, 41
  - pref, 31
  - PreferencesMenu, 31
  - preferencesMenu, 42
  - printFile, 31
  - printGameConf, 32
  - printGameConfigFile, 42
  - printListGameConf, 32
  - printListGameConfig, 43
  - quit, 31
  - readFilePath, 43
  - readPath, 32
  - removeGameConf, 32
  - removeGameConfig, 43
- keyboarding.c, 44
- charKey, 44
  - clean\_stdin, 45
  - floatKey, 45
  - floatKeyNoComma, 45
  - intKey, 46
  - stringKey, 46
  - systemPause, 46
- keyboarding.h, 47
- charKey, 48
  - clean\_stdin, 48
  - floatKey, 48
  - floatKeyNoComma, 50
  - intKey, 50
  - NB\_CARACT\_DOUB, 48
  - NB\_CARACT\_EXPRESSION, 48
  - NB\_CARACT\_FLOAT, 48
  - NB\_CARACT\_INT, 48
  - stringKey, 50
  - systemPause, 52
- listCsuFiles
- interface.c, 24
  - interface.h, 37
- listFile
- interface.h, 31
- loadGame
- interface.c, 24
  - interface.h, 37
- loadGameLocale
- interface.c, 24
  - interface.h, 38
- loadMatch
- interface.h, 31
- m\_file
- interface.h, 31
- main
- main.c, 53
  - main.h, 55
- main.c, 52
- main, 53
- main.h, 54
- main, 55
  - PORTABLE, 55
- MainMenu
- interface.h, 31
- mainMenu
- interface.c, 25
  - interface.h, 39
- menu.c, 56
- menuChangeDistributor, 57
  - menuChooseExportFileType, 57
  - menuContinueChangeDistributorOrQuit, 58
  - menuDelete, 58
  - menuDistributor, 58
  - menuExportListGameConfig, 59
  - menuFileName, 59
  - menuGameConfig, 60
  - menuImportListGameConfig, 60
  - menuNewPath, 61
  - menuPdfPreferences, 61
  - menuPlayerIndex, 61
  - menuPlayersName, 63
  - menuPlayersPoints, 63
  - menuStartGame, 63
- menu.h, 65
- ChangeDistributor, 66
  - Continue, 66
  - ContinueChangeDistributorOrQuit, 66
  - menuChangeDistributor, 66

- menuChooseExportFileType, 66
- menuContinueChangeDistributorOrQuit, 66
- menuDelete, 67
- menuDistributor, 67
- menuExportListGameConfig, 68
- menuFileName, 68
- menuGameConfig, 69
- menuImportListGameConfig, 69
- menuNewPath, 69
- menuPdfPreferences, 70
- menuPlayerIndex, 70
- menuPlayersName, 70
- menuPlayersPoints, 72
- menuStartGame, 72
- Quit, 66
- menuChangeDistributor
  - menu.c, 57
  - menu.h, 66
- menuChooseExportFileType
  - menu.c, 57
  - menu.h, 66
- menuContinueChangeDistributorOrQuit
  - menu.c, 58
  - menu.h, 66
- menuDelete
  - menu.c, 58
  - menu.h, 67
- menuDistributor
  - menu.c, 58
  - menu.h, 67
- menuExportListGameConfig
  - menu.c, 59
  - menu.h, 68
- menuFileName
  - menu.c, 59
  - menu.h, 68
- menuGameConfig
  - menu.c, 60
  - menu.h, 69
- menuImportListGameConfig
  - menu.c, 60
  - menu.h, 69
- menuNewPath
  - menu.c, 61
  - menu.h, 69
- menuPdfPreferences
  - menu.c, 61
  - menu.h, 70
- menuPlayerIndex
  - menu.c, 61
  - menu.h, 70
- menuPlayersName
  - menu.c, 63
  - menu.h, 70
- menuPlayersPoints
  - menu.c, 63
  - menu.h, 72
- menuStartGame
  - menu.c, 63
  - menu.h, 72
- NB\_CARACT\_DOUB
  - keyboarding.h, 48
- NB\_CARACT\_EXPRESSION
  - keyboarding.h, 48
- NB\_CARACT\_FLOAT
  - keyboarding.h, 48
- NB\_CARACT\_INT
  - keyboarding.h, 48
- newGame
  - interface.c, 26
  - interface.h, 40
- newGameConf
  - interface.h, 32
- newGameConfig
  - interface.c, 27
  - interface.h, 41
- newMatch
  - interface.h, 31
- newPath
  - interface.h, 32
- PORTABLE
  - main.h, 55
- pdf\_file
  - interface.h, 31
- pdfPreferences
  - interface.h, 32
- play
  - interface.c, 27
  - interface.h, 41
- pref
  - interface.h, 31
- PreferencesMenu
  - interface.h, 31
- preferencesMenu
  - interface.c, 28
  - interface.h, 42
- printAllPoints
  - display.c, 4
  - display.h, 13
- printCsuStruct
  - display.c, 5
  - display.h, 13
- printDistributor
  - display.c, 5
  - display.h, 13
- printFile
  - interface.h, 31
- printGameConf
  - interface.h, 32
- printGameConfig
  - display.c, 5
  - display.h, 15
- printGameConfigFile
  - interface.c, 28
  - interface.h, 42

- printGameOver
  - display.c, [6](#)
  - display.h, [15](#)
- printHeader
  - display.c, [6](#)
  - display.h, [15](#)
- printLicense
  - display.c, [6](#)
  - display.h, [15](#)
- printLigne
  - display.c, [6](#)
  - display.h, [16](#)
- printListGameConf
  - interface.h, [32](#)
- printListGameConfig
  - interface.c, [29](#)
  - interface.h, [43](#)
- printNames
  - display.c, [7](#)
  - display.h, [16](#)
- printPoints
  - display.c, [7](#)
  - display.h, [16](#)
- printRanking
  - display.c, [7](#)
  - display.h, [17](#)
- printSpecial
  - display.c, [9](#)
  - display.h, [17](#)
- printStringThreeTabs
  - display.c, [9](#)
  - display.h, [18](#)
- printTotalPoints
  - display.c, [9](#)
  - display.h, [18](#)
- Quit
  - menu.h, [66](#)
- quit
  - interface.h, [31](#)
- readFilePath
  - interface.c, [29](#)
  - interface.h, [43](#)
- readPath
  - interface.h, [32](#)
- removeGameConf
  - interface.h, [32](#)
- removeGameConfig
  - interface.c, [29](#)
  - interface.h, [43](#)
- stringKey
  - keyboarding.c, [46](#)
  - keyboarding.h, [50](#)
- systemPause
  - keyboarding.c, [46](#)
  - keyboarding.h, [52](#)
- Writing
  - display.h, [11](#)
- writingBold
  - display.h, [11](#)
- writingReset
  - display.h, [11](#)
- writingUnderline
  - display.h, [11](#)