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

2.1.9

Generated by Doxygen 1.8.6

Thu Apr 17 2014 16:55:42

Contents

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

Chapter 2

File Documentation

2.1 csuper.h File Reference

Include of csuper.

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

Macros

- #define CSUPER
- #define _(STRING) gettext(STRING)

2.1.1 Detailed Description

Include of csuper.

Author

Remi BERTHO

Date

17/04/14

Version

2.2.0

2.1.2 Macro Definition Documentation

2.1.2.1 #define _(*STRING*) gettext(STRING)

Define the _ for gettext.

2.1.2.2 #define CSUPER

Define that we compile csuper.

2.2 display.c File Reference

Display the games.

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

Functions

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

2.2.1 Detailed Description

Display the games.

Author

Remi BERTHO

Date

17/04/14

Version

2.2.0

2.2.2 Function Documentation

2.2.2.1 void printAllPoints (csuStruct ∗ *ptr_csu_struct*)

Print the scores of all player in each turn.

Parameters

in	*∗ptr_csu_struct*	a pointer on a csu structure

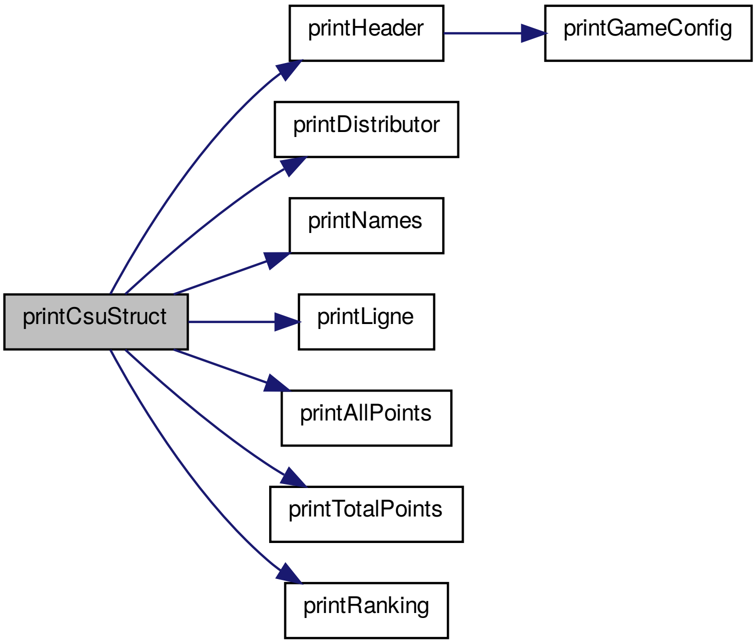
2.2.2.2 void printCsuStruct (csuStruct ∗ *ptr_csu_struct*)

Print all data of the csu structure.

Parameters

in	**ptr_csu_struct*	a pointer on a csu structure

Here is the call graph for this function:



2.2.2.3 void printDistributor (csuStruct * *ptr_csu_struct*)

Display the distributor.

Parameters

in	**ptr_csu_struct*	a pointer on a csu structure

2.2.2.4 void printGameConfig (game_config *config*)

Print the game config

Parameters

in	*config*	a game config

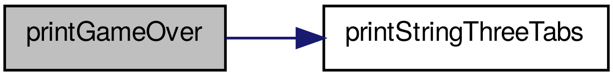
2.2.2.5 void printGameOver (csuStruct * *ptr_csu_struct*)

Display that the game is over and a podium.

Parameters

in	*∗ptr_csu_struct*	a pointer on a csu structure

Here is the call graph for this function:



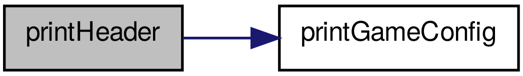
2.2.2.6 void printHeader (csuStruct ∗ *ptr_csu_struct*)

Print the header of the structure

Parameters

in	*∗ptr_csu_struct*	a pointer on a csu structure

Here is the call graph for this function:



2.2.2.7 void printLicense ( )

Print the license.

2.2.2.8 void printLigne (int *size_line*)

Print a line of - after a tabulation.

Parameters

in	*size_line*	the size of the line

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

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

Parameters

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

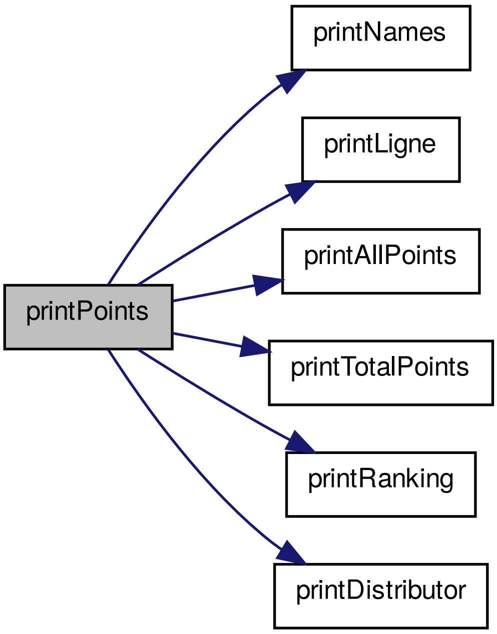
2.2.2.10 void printPoints (csuStruct ∗ *ptr_csu_struct*)

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

Parameters

`in`	*∗ptr_csu_struct*	a pointer on a csu structure

Here is the call graph for this function:



2.2.2.11 void printRanking (csuStruct ∗ *ptr_csu_struct*)

Print the rank of each player

Parameters

`in`	*∗ptr_csu_struct*	a pointer on a csu structure

2.2.2.12 void printStringThreeTabs (char ∗ *string*)

Print a string center into a space of three tabulations.

Parameters

in	∗*string*	a string

2.2.2.13 void printTotalPoints (csuStruct ∗ *ptr_csu_struct*)

Print the total score of the players.

Parameters

in	∗*ptr_csu_struct*	a pointer on a csu structure

2.3 display.h File Reference

Display the games.

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

Functions

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

2.3.1 Detailed Description

Display the games.

Author

Remi BERTHO

Date

17/04/14

Version

2.2.0

2.3.2 Function Documentation

2.3.2.1 void printAllPoints (csuStruct ∗ *ptr_csu_struct*)

Print the scores of all player in each turn.

Parameters

in	*∗ptr_csu_struct*	a pointer on a csu structure

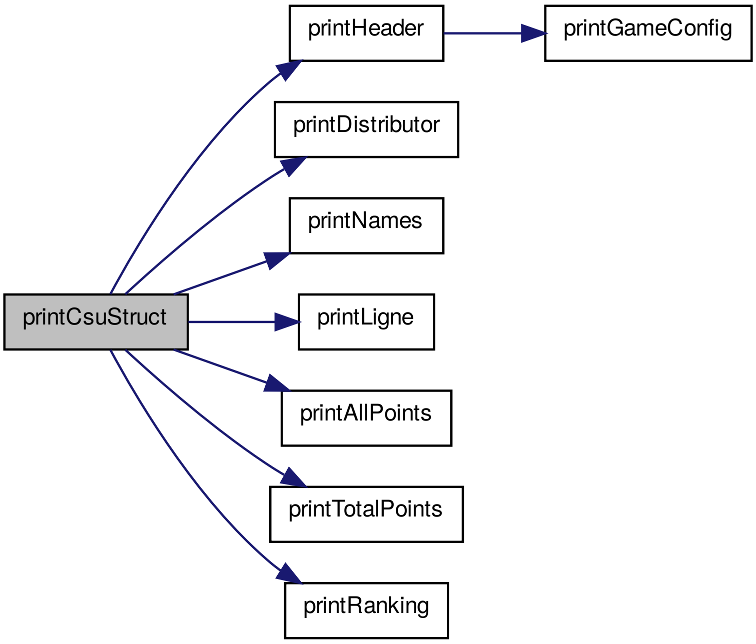
2.3.2.2 void printCsuStruct (csuStruct ∗ *ptr_csu_struct*)

Print all data of the csu structure.

Parameters

in	*∗ptr_csu_struct*	a pointer on a csu structure

Here is the call graph for this function:



2.3.2.3 void printDistributor (csuStruct ∗ *ptr_csu_struct*)

Display the distributor.

Parameters

in	*∗ptr_csu_struct*	a pointer on a csu structure

2.3.2.4 void printGameConfig (game_config *config*)

Print the game config

Parameters

in	*config*	a game config

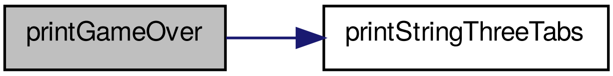
2.3.2.5 void printGameOver (csuStruct ∗ *ptr_csu_struct*)

Display that the game is over and a podium.

Parameters

in	∗*ptr_csu_struct*	a pointer on a csu structure

Here is the call graph for this function:



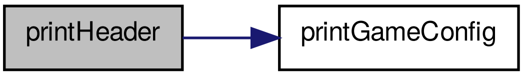
2.3.2.6 void printHeader (csuStruct ∗ *ptr_csu_struct*)

Print the header of the structure

Parameters

in	∗*ptr_csu_struct*	a pointer on a csu structure

Here is the call graph for this function:



2.3.2.7 void printLicense ( )

Print the license.

2.3.2.8 void printLigne (int *size_line*)

Print a line of - after a tabulation.

Parameters

in	*size_line*	the size of the line

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

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

Parameters

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

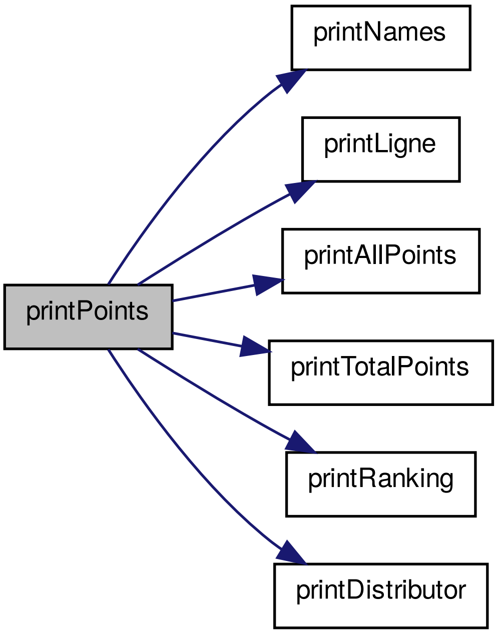
2.3.2.10 void printPoints (csuStruct ∗ *ptr_csu_struct*)

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

Parameters

in	∗*ptr_csu_struct*	a pointer on a csu structure

Here is the call graph for this function:



2.3.2.11 void printRanking (csuStruct ∗ *ptr_csu_struct*)

Print the rank of each player

Parameters

in	∗*ptr_csu_struct*	a pointer on a csu structure

2.3.2.12 void printStringThreeTabs (char ∗ *string*)

Print a string center into a space of three tabulations.

Parameters

in	∗*string*	a string

2.3.2.13 void printTotalPoints (csuStruct ∗ *ptr_csu_struct*)

Print the total score of the players.

Parameters

in	∗*ptr_csu_struct*	a pointer on a csu structure

2.4 interface.c File Reference

Graphical interface.

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

Functions

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

2.4.1 Detailed Description

Graphical interface.

Author

Remi BERTHO

Date

17/04/14

Version

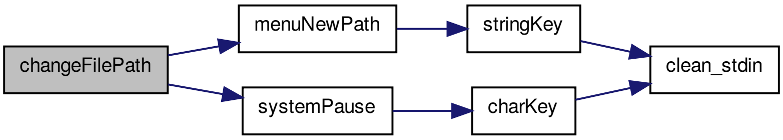
2.2.0

2.4.2 Function Documentation

2.4.2.1 void changeFilePath ( )

Change the path which the file are saved.

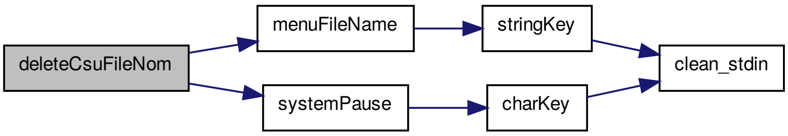
Here is the call graph for this function:



2.4.2.2 void deleteCsuFileNom ( )

Ask a filename and deleted the file.

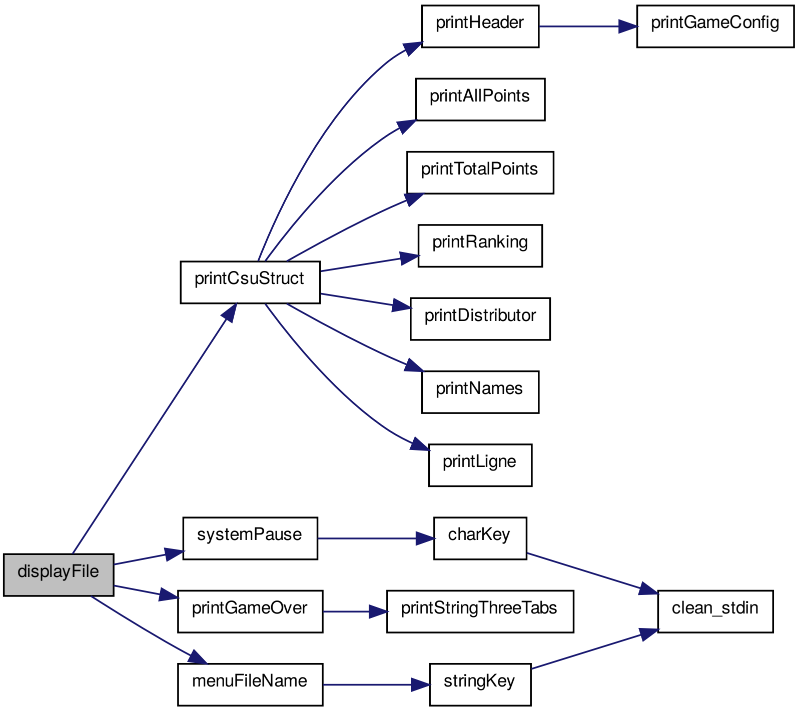
Here is the call graph for this function:



2.4.2.3 void displayFile ( )

Ask a filename and display it.

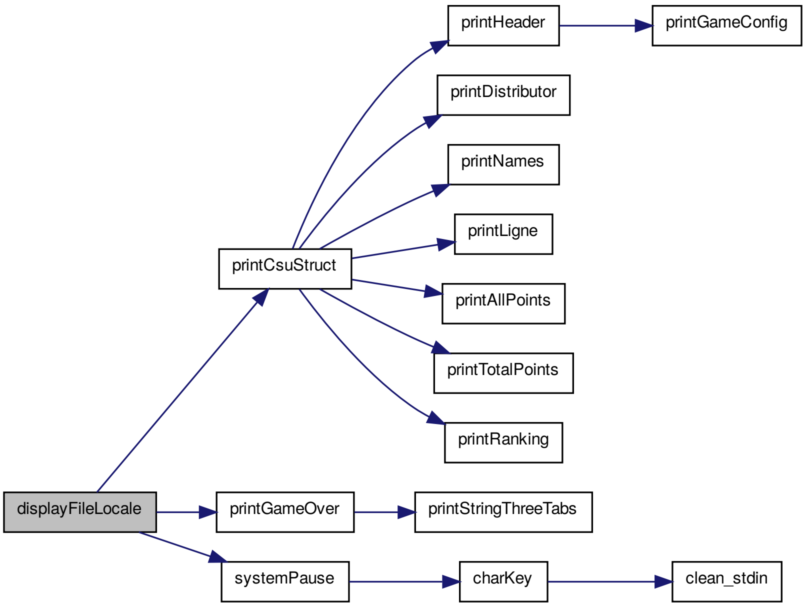
Here is the call graph for this function:



2.4.2.4 void displayFileLocale (char ∗ *file_name*)

Display the file.

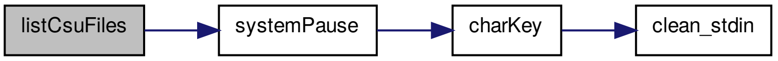
Here is the call graph for this function:



2.4.2.5 void listCsuFiles ( )

List all the files with csu extension

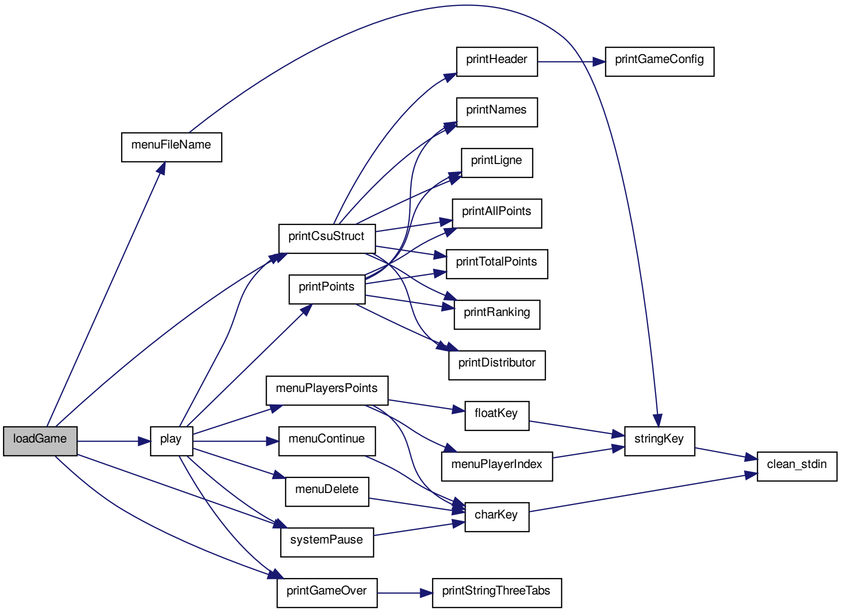
Here is the call graph for this function:



2.4.2.6 void loadGame ( )

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

Here is the call graph for this function:



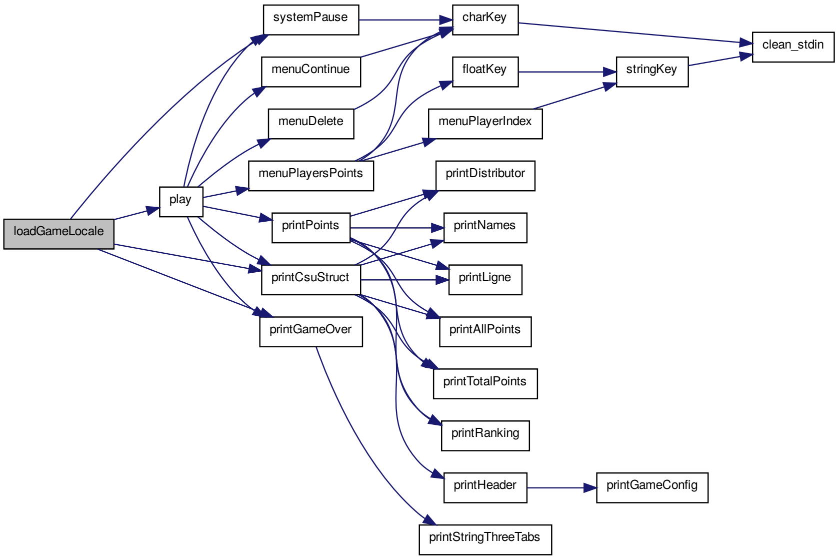
2.4.2.7 void loadGameLocale (char ∗ *file_name*)

Load the file and run the play function

Parameters

in	*file_name,the*	filename

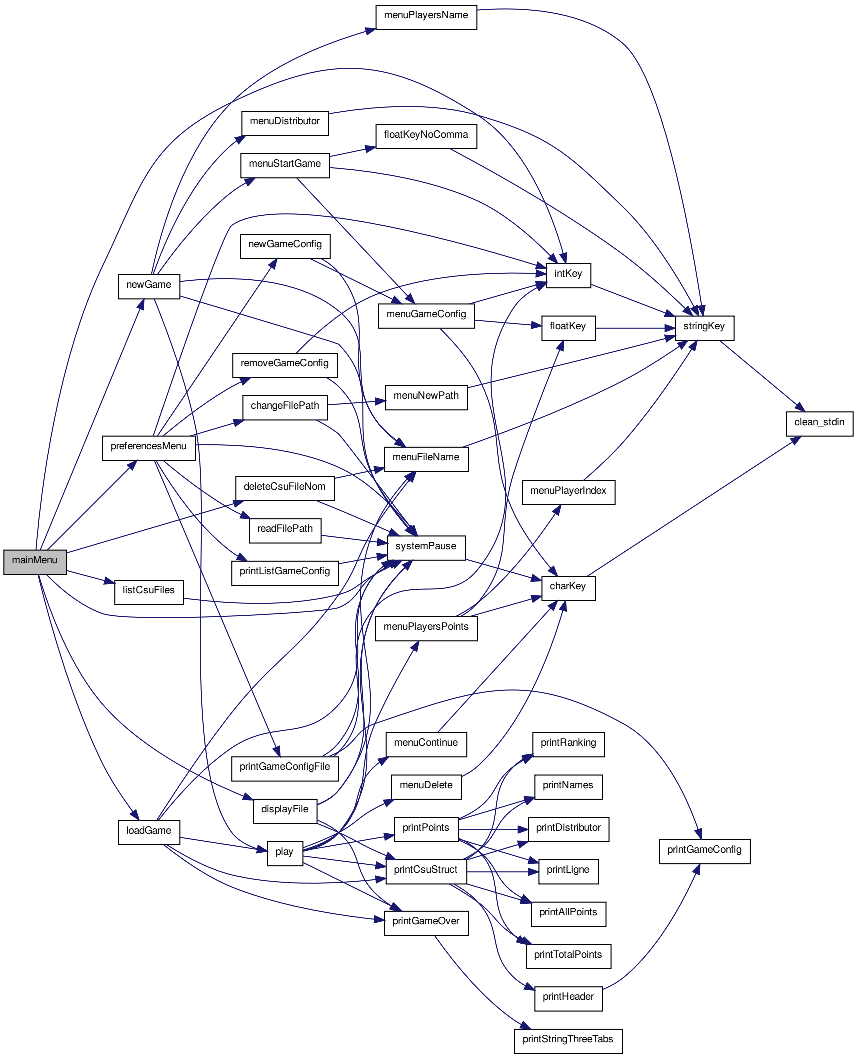
Here is the call graph for this function:



2.4.2.8 void mainMenu ( )

Main menu of csuper.

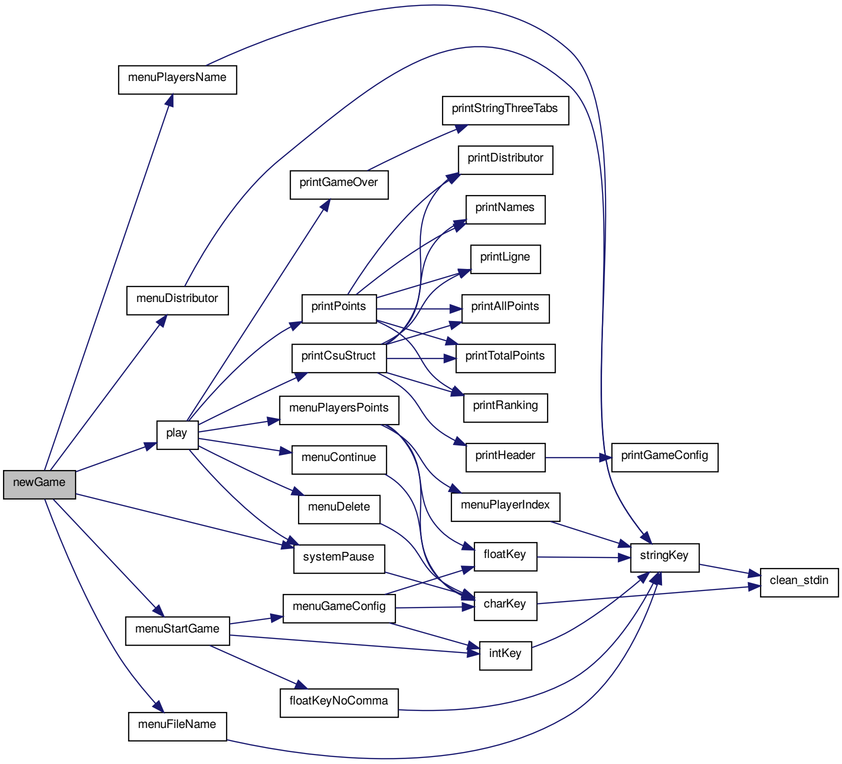
Here is the call graph for this function:



2.4.2.9 void newGame ( )

Initialize a new game and run the play function.

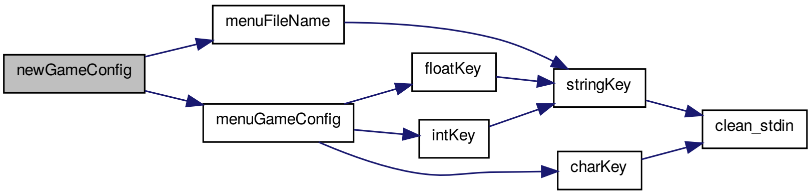
Here is the call graph for this function:



2.4.2.10 void newGameConfig ( )

Add a new game configuration

Here is the call graph for this function:



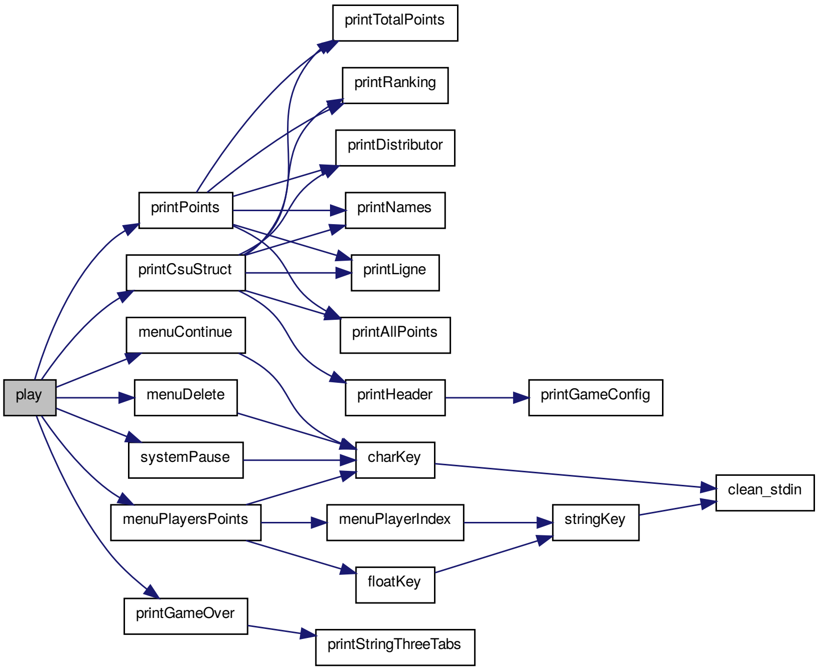
2.4.2.11 void play (csuStruct ∗ *ptr_csu_struct,* char ∗ *file_name*)

Count the points

Parameters

in,out	**ptr_csu_struct*	a pointer on a csu structure
in	**file_name*	the filename

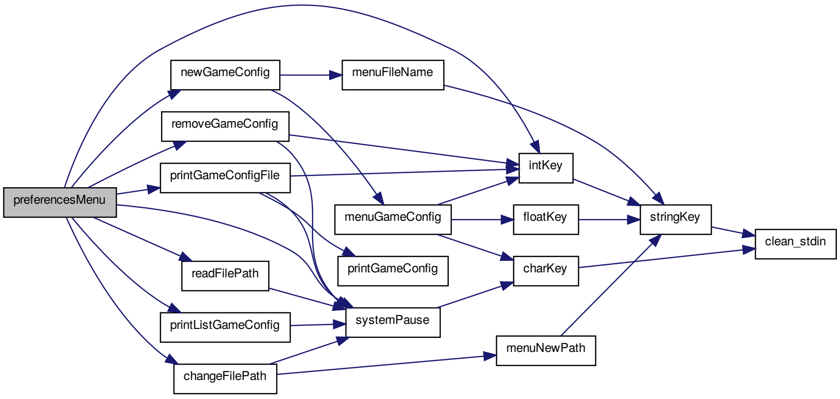
Here is the call graph for this function:



2.4.2.12 void preferencesMenu ( )

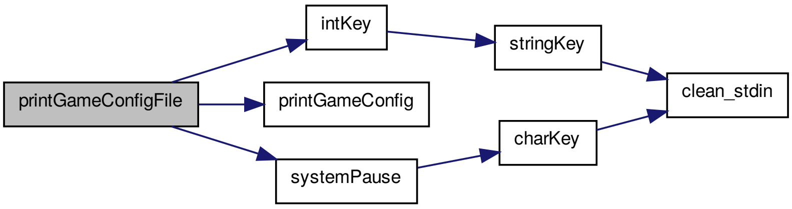
Preferences menu of csuper.

Here is the call graph for this function:



2.4.2.13 void printGameConfigFile ( )

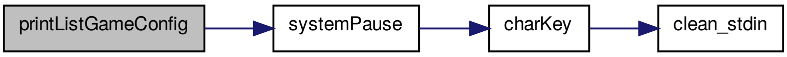
Here is the call graph for this function:



2.4.2.14 void printListGameConfig ( )

Print the list of game configuration

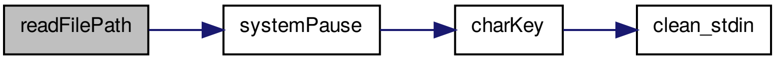
Here is the call graph for this function:



2.4.2.15 void readFilePath ( )

Read the file path and display it.

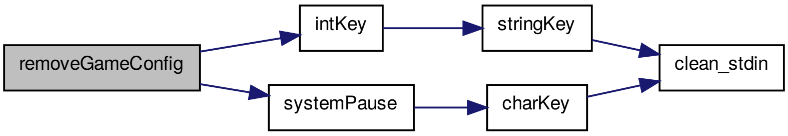
Here is the call graph for this function:



2.4.2.16 void removeGameConfig ( )

Ask and remove a game configuration

Here is the call graph for this function:



2.5 interface.h File Reference

Graphical interface.

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

Enumerations

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

Functions

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

2.5.1 Detailed Description

Graphical interface.

Author

Remi BERTHO

Date

17/04/14

Version

2.2.0

2.5.2 Enumeration Type Documentation

2.5.2.1 enum MainMenu

Enumerator

newMatch

loadMatch

printFile

deleteFile

listFile

pref

quit

easterEggs

2.5.2.2 enum PreferencesMenu

Enumerator

newPath

readPath

newGameConf

removeGameConf

printListGameConf

printGameConf

backMainMenu

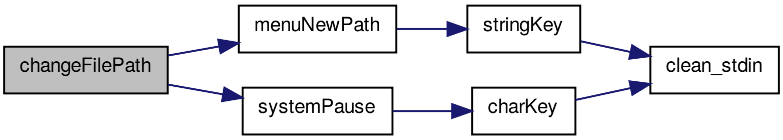
easterEggs2

2.5.3 Function Documentation

2.5.3.1 void changeFilePath ( )

Change the path which the file are saved.

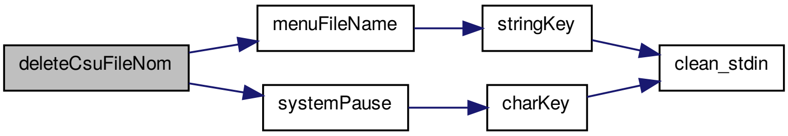
Here is the call graph for this function:



2.5.3.2 void deleteCsuFileNom ( )

Ask a filename and deleted the file.

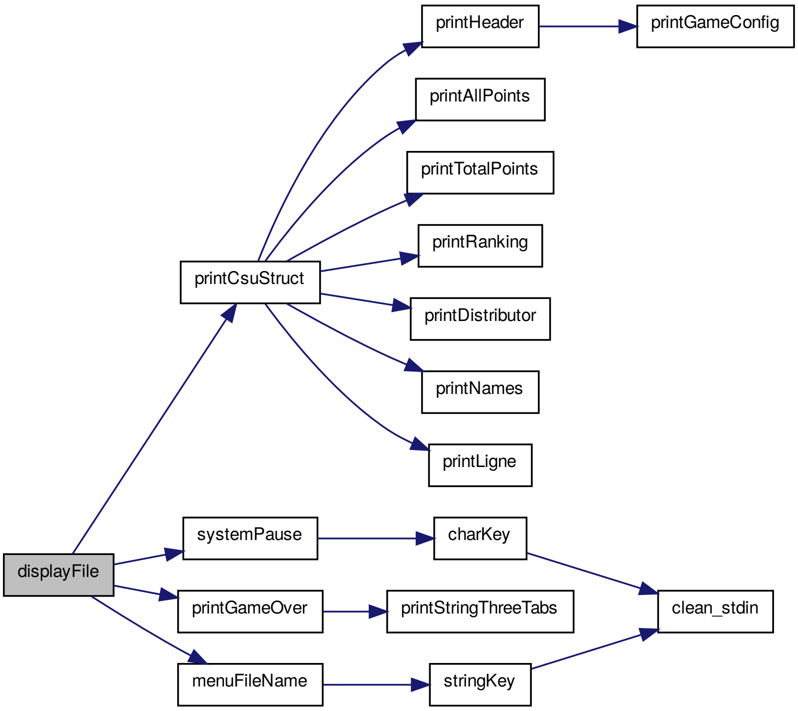
Here is the call graph for this function:



2.5.3.3 void displayFile ( )

Ask a filename and display it.

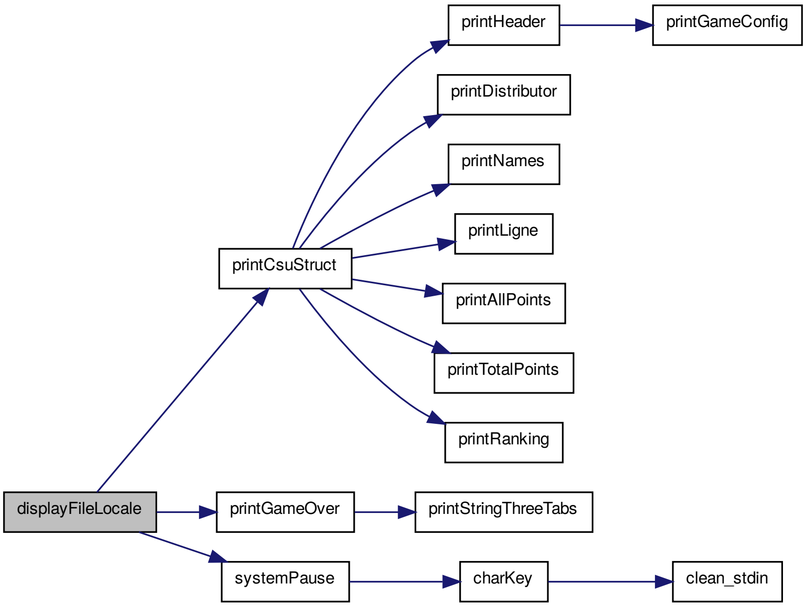
Here is the call graph for this function:



2.5.3.4 void displayFileLocale (char ∗ *file_name*)

Display the file.

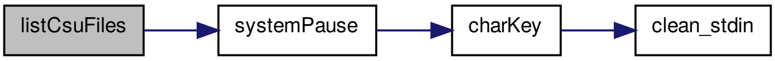
Here is the call graph for this function:



2.5.3.5 void listCsuFiles ( )

List all the files with csu extension

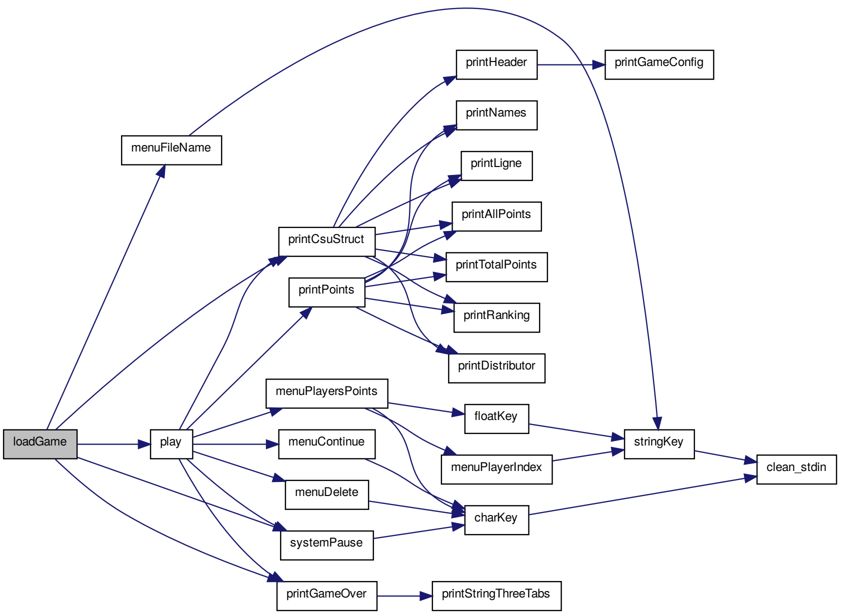
Here is the call graph for this function:



2.5.3.6 void loadGame ( )

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

Here is the call graph for this function:



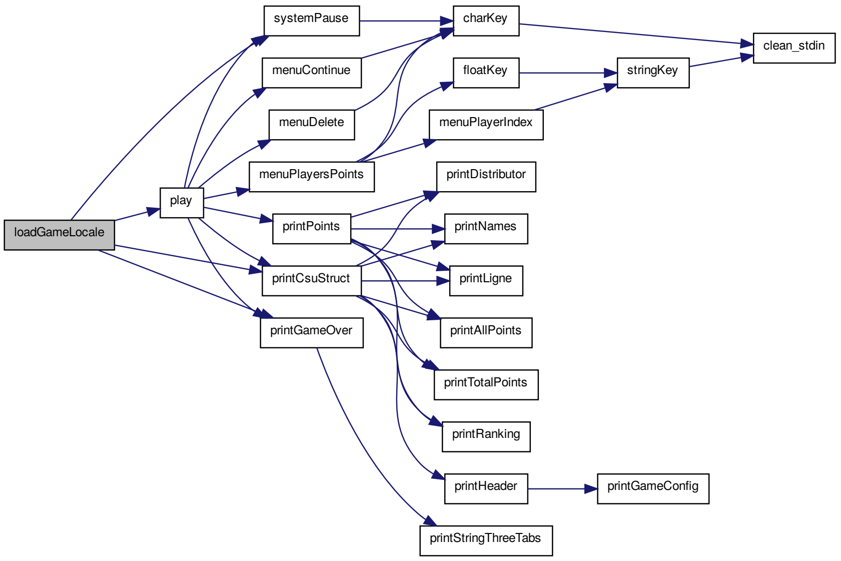
2.5.3.7 void loadGameLocale (char ∗ *file_name*)

Load the file and run the play function

Parameters

in	*file_name,the*	filename

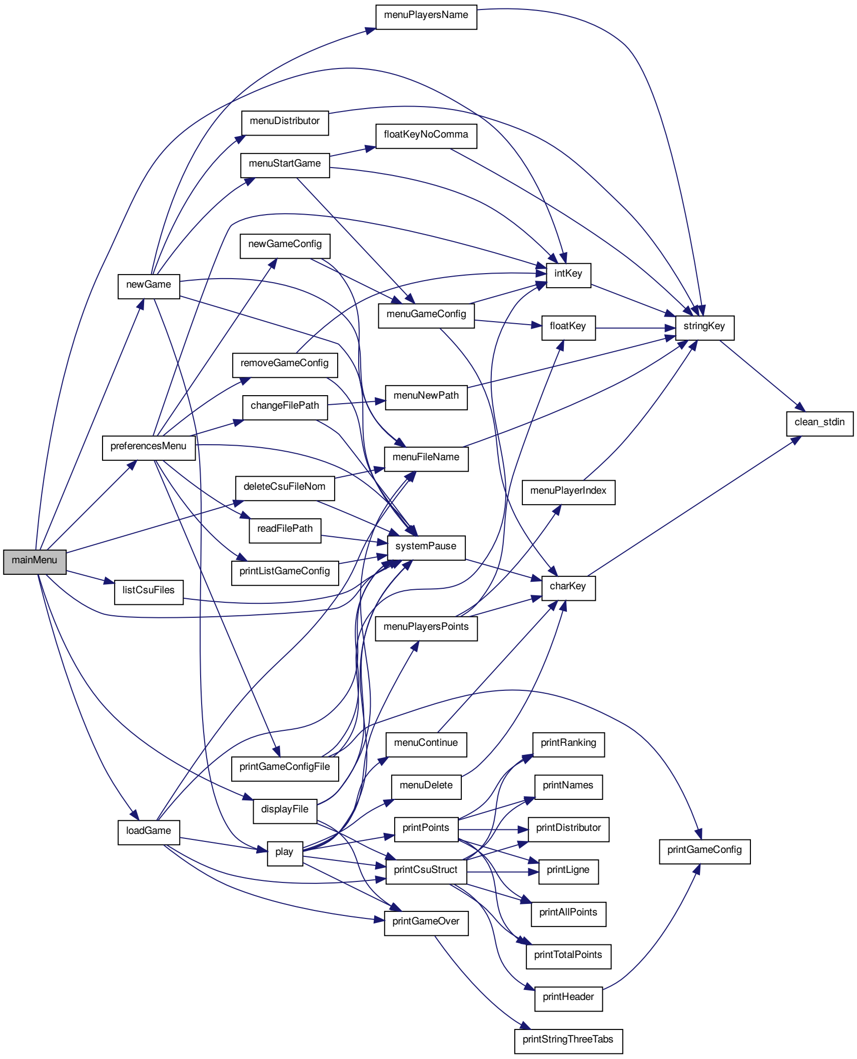
Here is the call graph for this function:



2.5.3.8 void mainMenu ( )

Main menu of csuper.

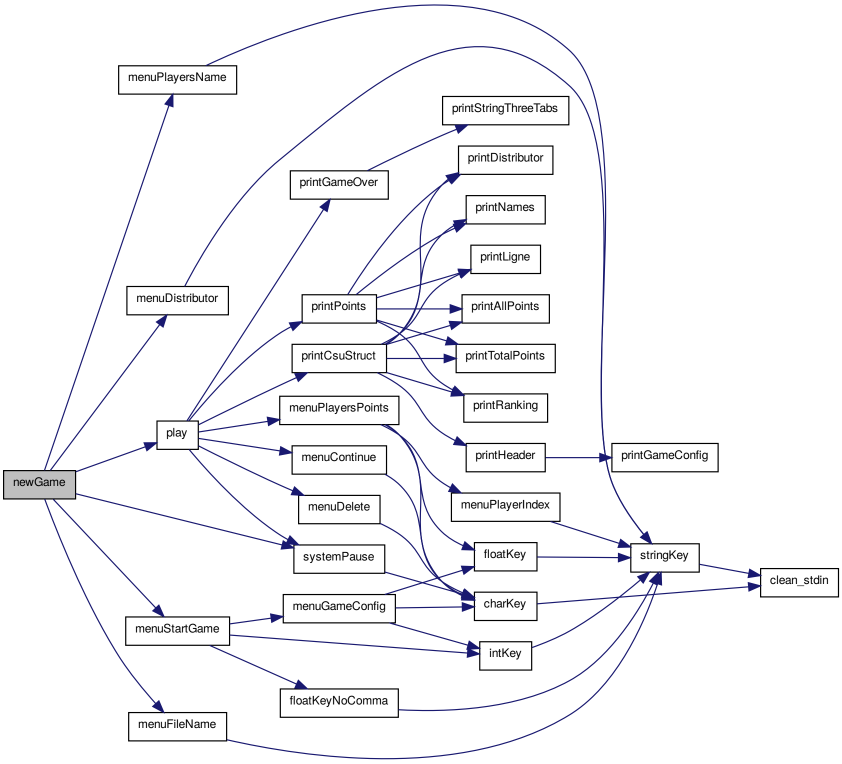
Here is the call graph for this function:



2.5.3.9 void newGame ( )

Initialize a new game and run the play function.

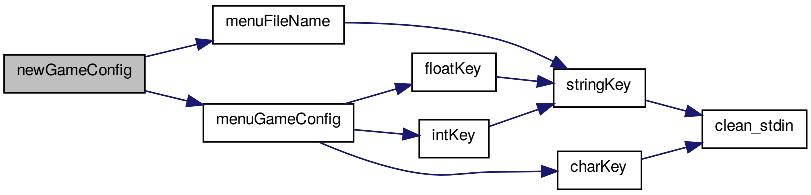
Here is the call graph for this function:



2.5.3.10 void newGameConfig ( )

Add a new game configuration

Here is the call graph for this function:



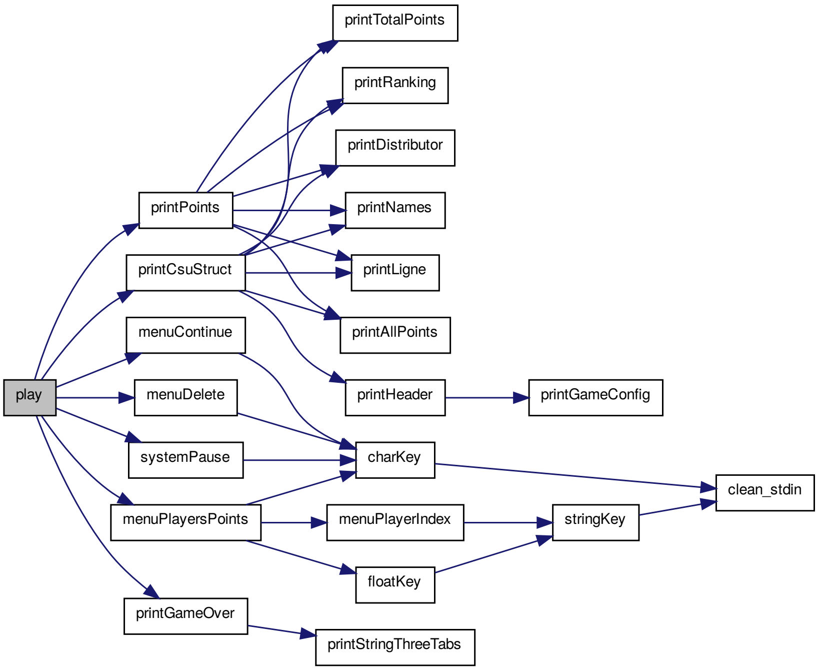
2.5.3.11 void play (csuStruct ∗ *ptr_csu_struct,* char ∗ *file_name*)

Count the points

Parameters

in,out	**ptr_csu_struct*	a pointer on a csu structure
in	**file_name*	the filename

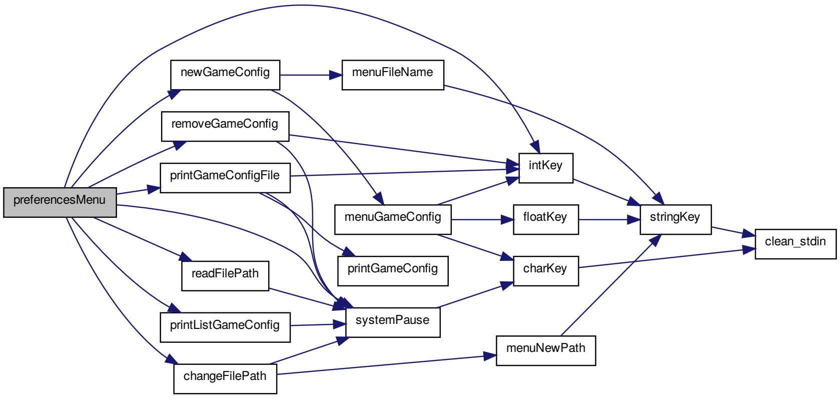
Here is the call graph for this function:



2.5.3.12 void preferencesMenu ( )

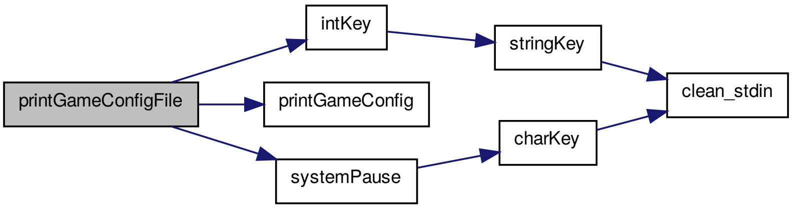
Preferences menu of csuper.

Here is the call graph for this function:



2.5.3.13 void printGameConfigFile ( )

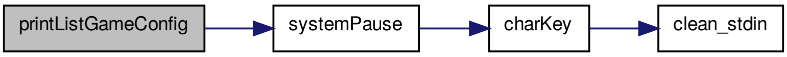
Here is the call graph for this function:



2.5.3.14 void printListGameConfig ( )

Print the list of game configuration

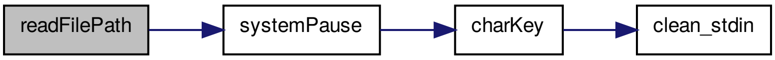
Here is the call graph for this function:



2.5.3.15 void readFilePath ( )

Read the file path and display it.

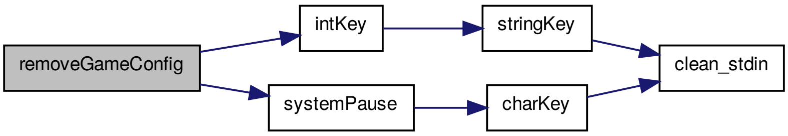
Here is the call graph for this function:



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

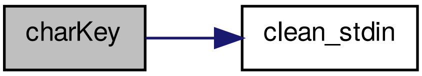
2.6.2.1 char ∗ charKey (char ∗ *c*)

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.

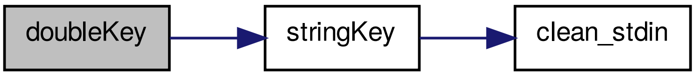
2.6.2.3 void doubleKey (double * *nb*)

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:



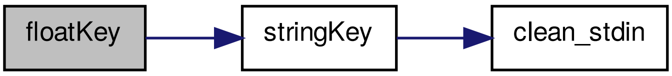
2.6.2.4 void floatKey (float * *nb*)

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:



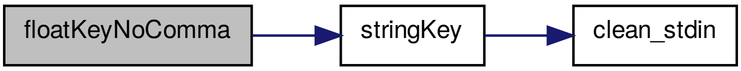
2.6.2.5 void floatKeyNoComma (float ∗ *nb*)

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:



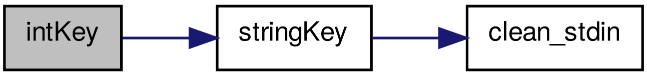
2.6.2.6 void ∗ intKey (int ∗ *nb*)

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:



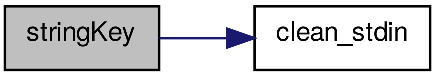
2.6.2.7 char ∗ stringKey (char ∗ *string,* int *nb_char_plus_one*)

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

Parameters

in,out	*∗string*	a string
in	*nb_char_plus_one*	the number of characters that the function will read plus one

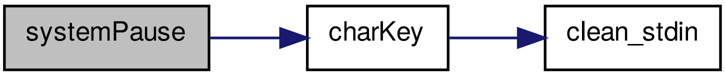
Here is the call graph for this function:



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_CARACT_DOUB 309

Define the number of characters that need a double

2.7.2.2 #define NB_CARACT_FLOAT 39

Define the number of characters that need a float.

2.7.2.3 #define NB_CARACT_INT 12

Define the number of characters that need an int.

2.7.3 Function Documentation

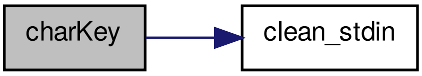
2.7.3.1 char∗ charKey (char ∗ *c*)

Do a keyboarding of a character.

Parameters

`in,out`	∗*c*	the character

Here is the call graph for this function:



2.7.3.2 void clean_stdin (void)

Clean the buffer of stdin.

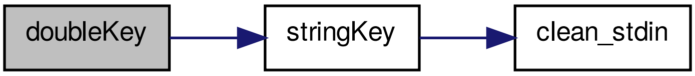
2.7.3.3 void doubleKey (double ∗ *nb*)

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

Parameters

`in,out`	∗*nb*	the number

Here is the call graph for this function:



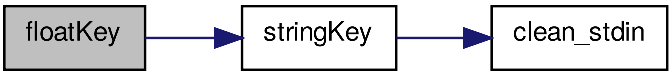
2.7.3.4 void floatKey (float ∗ *nb*)

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

Parameters

`in,out`	∗*nb*	the number

Here is the call graph for this function:



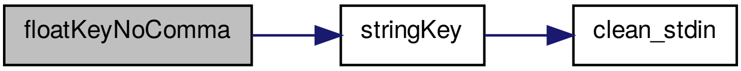
2.7.3.5 void floatKeyNoComma (float ∗ *nb*)

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

Parameters

in,out	∗*nb*	the number

Here is the call graph for this function:



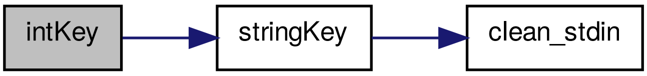
2.7.3.6 void intKey (int ∗ *nb*)

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

Parameters

in,out	∗*nb*	the number

Here is the call graph for this function:



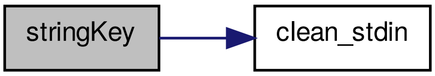
2.7.3.7 char∗ stringKey (char ∗ *string,* int *nb_char_plus_one*)

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

Parameters

in,out	∗*string*	a string
in	*nb_char_plus_one*	the number of characters that the function will read plus one

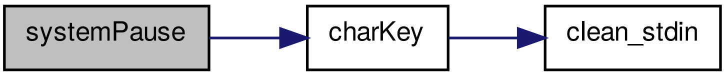
Here is the call graph for this function:



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.

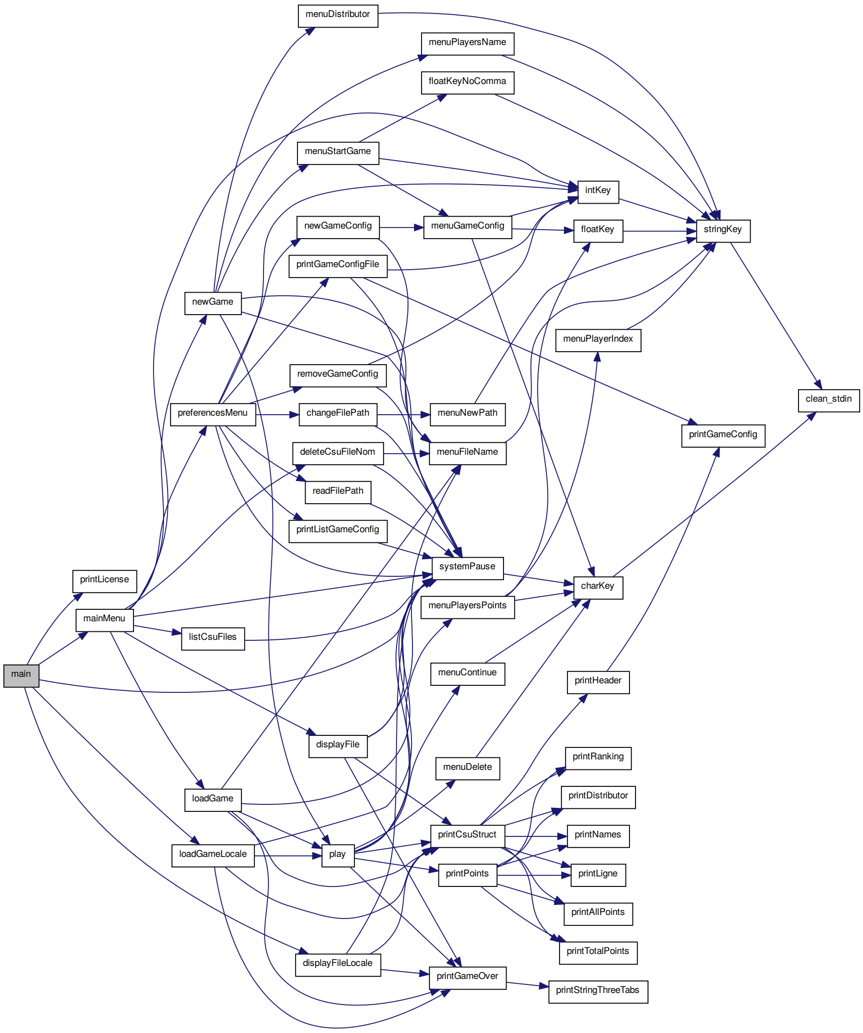
Parameters

in	*argc*	the number of argument.
in	*argv*	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.

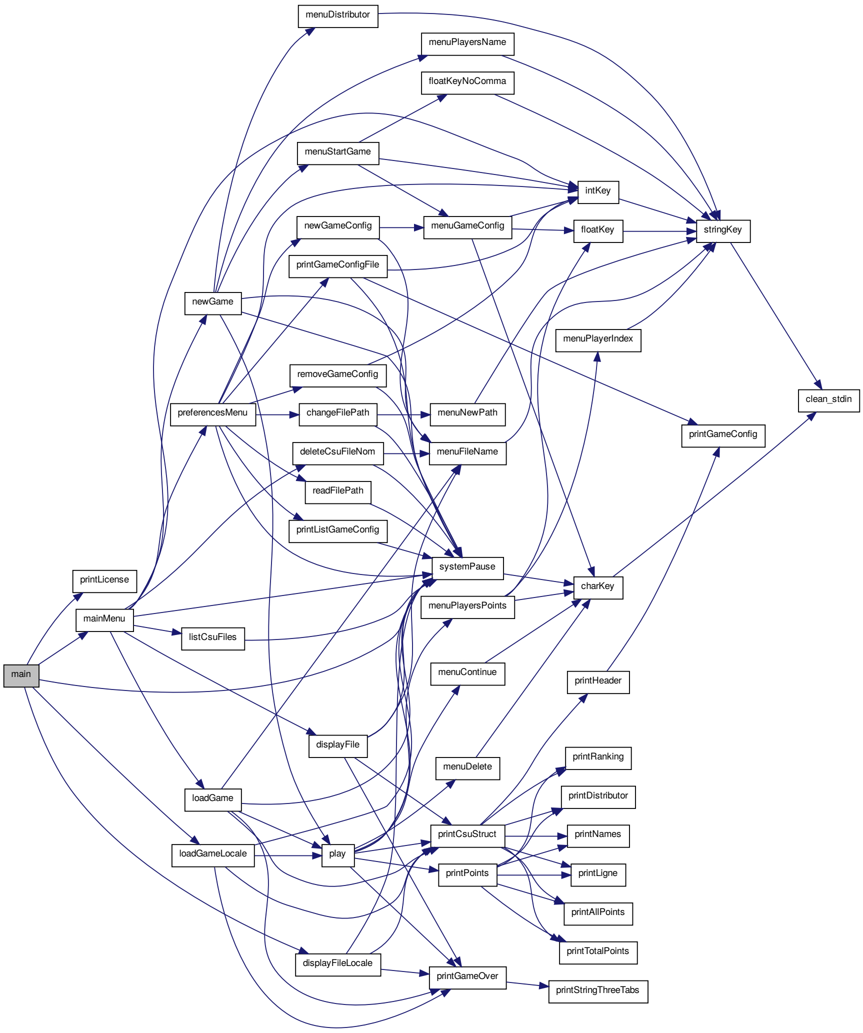
Parameters

in	*argc*	the number of argument.
in	*argv*	the array of argument.

Returns

EXIT_SUCCESS if everything is OK

Here is the call graph for this function:



2.10 menu.c File Reference

Menu functions.

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

Functions

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

- void menuPlayersPoints (csuStruct ∗ptr_csu_struct)
- int menuPlayerIndex (csuStruct ∗ptr_csu_struct)
- int menuContinue ()
- int menuDelete ()
- void menuNewPath (char ∗new_path)

2.10.1 Detailed Description

Menu functions.

Author

Remi BERTHO

Date

17/04/14

Version

2.2.0

2.10.2 Function Documentation

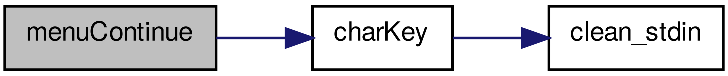
2.10.2.1 int menuContinue ( )

Ask if we want to continue3

Returns

TRUE if we want to continue, FALSE otherwise

Here is the call graph for this function:



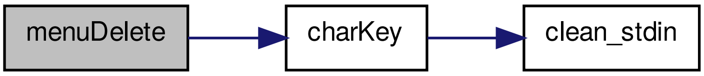
2.10.2.2 int menuDelete ( )

Ask if we want to delete a file.

Returns

TRUE if we want to deleted the file, FALSE otherwise

Here is the call graph for this function:



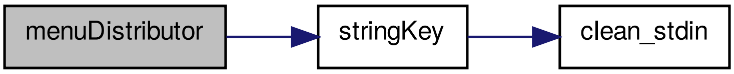
2.10.2.3 void menuDistributor (char ∗ *distributor_name*)

Ask and save the distributor.

Parameters

`in,out`	∗*distributor_name*	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.

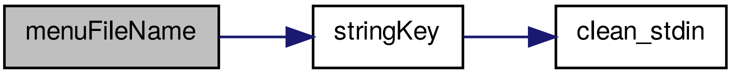
Parameters

`in,out`	*file_name*	the filename

Returns

filename

Here is the call graph for this function:



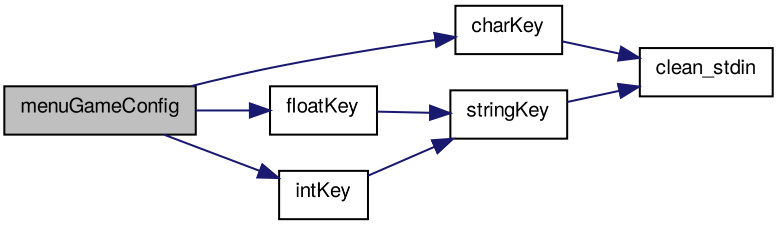
2.10.2.5 void menuGameConfig (game_config ∗ *ptr_config*)

Ask and save a game configuration.

Parameters

in,out	*ptr_config*	a game configuration

Here is the call graph for this function:



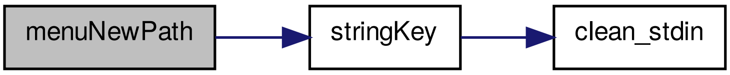
2.10.2.6 void menuNewPath (char ∗ *new_path*)

Ask and save the new path

Parameters

in,out	∗*new_path*	the new path

Here is the call graph for this function:



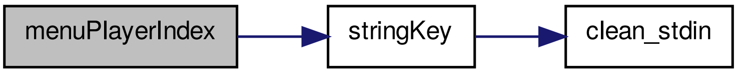
2.10.2.7 int menuPlayerIndex (csuStruct ∗ *ptr_csu_struct*)

Ask a player name and save his index

Parameters

`in,out`	∗*ptr_csu_struct*	a pointer on a csu structure

Here is the call graph for this function:



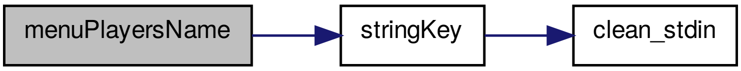
2.10.2.8 void menuPlayersName (csuStruct ∗ *ptr_csu_struct*)

Ask and save the name of the players.

Parameters

`in,out`	∗*ptr_csu_struct*	a pointer on a csu structure

Here is the call graph for this function:



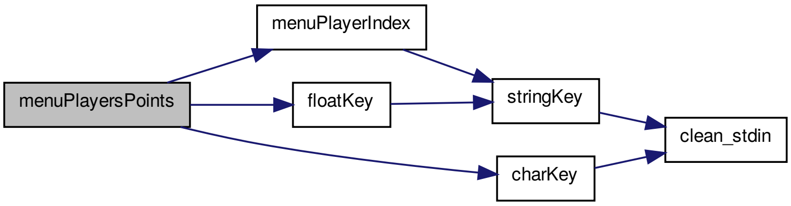
2.10.2.9 void menuPlayersPoints (csuStruct ∗ *ptr_csu_struct*)

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

Parameters

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

Here is the call graph for this function:



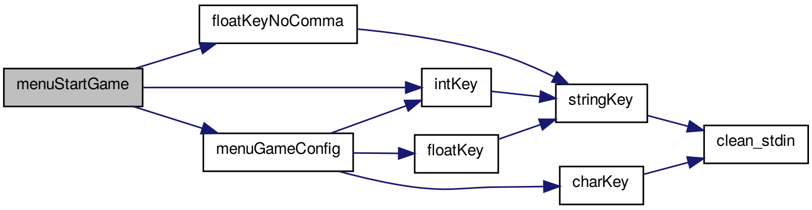
2.10.2.10 void menuStartGame (float ∗ *ptr_nb_player,* game_config ∗ *ptr_config*)

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

Parameters

in,out	*ptr_nb_player*	the number of player
in,out	*ptr_config*	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

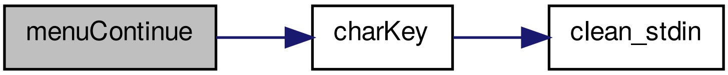
2.11.2.1 int menuContinue ( )

Ask if we want to continue3

Returns

TRUE if we want to continue, FALSE otherwise

Here is the call graph for this function:



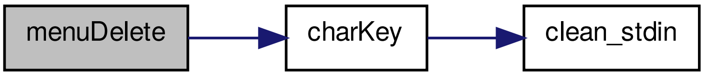
2.11.2.2 int menuDelete ( )

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:



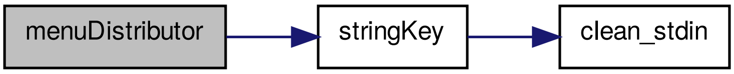
2.11.2.3 void menuDistributor (char ∗ *distributor_name*)

Ask and save the distributor.

Parameters

in,out	*∗distributor_name*	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.

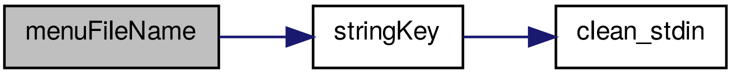
Parameters

in,out	*file_name*	the filename

Returns

filename

Here is the call graph for this function:



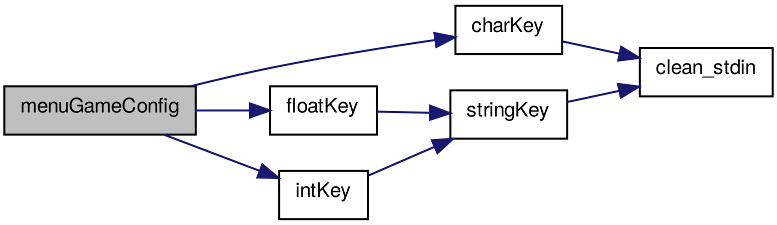
2.11.2.5 void menuGameConfig (game_config ∗ *ptr_config*)

Ask and save a game configuration.

Parameters

`in,out`	*ptr_config*	a game configuration

Here is the call graph for this function:



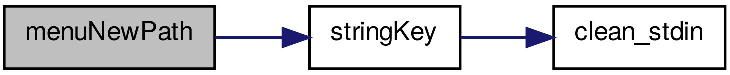
2.11.2.6 void menuNewPath (char ∗ *new_path*)

Ask and save the new path

Parameters

`in,out`	∗*new_path*	the new path

Here is the call graph for this function:



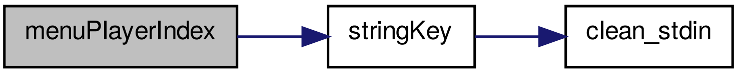
2.11.2.7 int menuPlayerIndex (csuStruct ∗ *ptr_csu_struct*)

Ask a player name and save his index

Parameters

`in,out`	∗*ptr_csu_struct*	a pointer on a csu structure

Here is the call graph for this function:



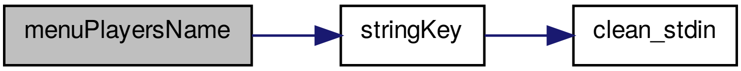
2.11.2.8 void menuPlayersName (csuStruct ∗ *ptr_csu_struct*)

Ask and save the name of the players.

Parameters

`in,out`	∗*ptr_csu_struct*	a pointer on a csu structure

Here is the call graph for this function:



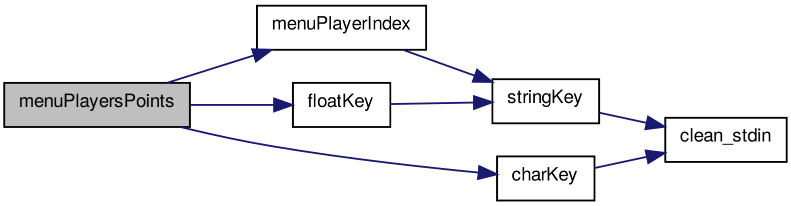
2.11.2.9 void menuPlayersPoints (csuStruct ∗ *ptr_csu_struct*)

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

Parameters

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

Here is the call graph for this function:



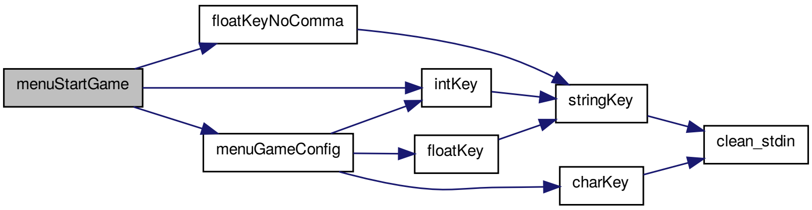
2.11.2.10 void menuStartGame (float ∗ *ptr_nb_player,* game_config ∗ *ptr_config*)

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

Parameters

in,out	*ptr_nb_player*	the number of player
in,out	*ptr_config*	the game configuration

Here is the call graph for this function:



Index