

Six qui prend!

Generated by Doxygen 1.8.9.1

Wed Mar 30 2016 23:05:45

Contents

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

__attribute__	??
Button	élément graphique, cliquable	??
Card	Information de base pour les cartes du jeu	??
ChatBox	Regroupement de composants graphiques pour faire un chat	??
Data	Données de base Données pour toute la partie connection	??
DataGame	Données pour le jeu en lui même	??
DataList	Données spécifiques à la liste des salles sont comme une liste chaînée, dernier paquet contient end à true	??
Joueur	Objet positionnement et valeurs de données sur chaque joueur	??
MultiText	Objet pour l'affichage sur plusieurs lignes	??
Picture	Object contenant toutes les informations nssaires d'une image	??
Statistique	Donnée de base des statistiques	??
Text	Objet comprenant ce qu'il faut pour afficher du texte	??
TextBox	??
textBox	Regroupement de composants graphiques pour faire de la saisie de texte	??

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

src/main.c	??
src/Game/animCard.c	??
src/Game/animCard.h	
Composant de l'interface graphique utilisateur: la carte avec animation La carte se dace selon une ation prdnie	??
src/Game/client.hpp	??
src/Game/game.c	??
src/Game/game.h	
Interface graphique du jeu	??
src/Game/inc.c	??
src/Game/inc.h	
Macro et fonction utiles partout	??
src/Game/init.c	??
src/Game/init.h	
Comprend L'initialisation de la SDL, et la liberation des ressources	??
src/Game/login.c	??
src/Game/login.h	
Fichier correspondant à l'affichage et la gestion du login du jeu	??
src/Game/menu.c	??
src/Game/menu.h	
Fichier correspondant à l'affichage et la gestion du menu du jeu	??
src/Game/mode.c	??
src/Game/mode.h	
Fichier correspondant à l'affichage et la gestion de la page de parametre, avant l'acces au jeu	??
src/Game/stat.c	
Implémentation de l'affichage et gestions des statistiques	??
src/Game/stat.h	
Affichage et gestions des statistiques	??
src/GUI/button.c	??
src/GUI/button.h	
élément de l'interface graphique utilisateur: le bouton	??
src/GUI/card.c	??
src/GUI/card.h	
Composant de l'interface graphique utilisateur: la carte	??
src/GUI/chatBox.c	??
src/GUI/chatBox.h	
élément de l'interface graphique utilisateur: le chat	??
src/GUI/multiText.c	??

src/GUI/ multiText.h	
élément de l'interface graphique utilisateur: l'affichage sur plusieurs lignes	??
src/GUI/ picture.c	??
src/GUI/ picture.h	
Composant de l'interface graphique utilisateur: l'image	??
src/GUI/ text.c	??
src/GUI/ text.h	
élément de l'interface graphique utilisateur: le texte	??
src/GUI/ textBox.c	??
src/GUI/ textBox.h	
élément de l'interface graphique utilisateur: la saisie de texte	??
src/Network/ client.c	
Implémentation réseau	??
src/Network/ client.h	
Réseau	??
src/Network/ network.h	
Fichier correspondant au structures et types de donnée pour le reseau doit être le même sur le serveur et le client (note: sur le serveur c'est inc.hpp)	??
src/Network/ network.hpp	??

Chapter 3

Data Structure Documentation

3.1 `__attribute__` Struct Reference

```
#include <network.h>
```

Data Fields

- `DataType` `dataType`
- `char` `car`
- `char` `tab` [`BUF_SIZE`]
- `int32_t` `from`
- `_Bool` `end`
- `int32_t` `table` [4][5]
- `char` `users` [10][`BUF_SIZE`]
- `int32_t` `turn` [10]
- `int32_t` `place` [10]
- `int32_t` `scores` [10]
- `int32_t` `hand` [10]
- `int` `from`
- `bool` `end`
- `int` `table` [4][6]
- `int` `turn` [10]
- `int` `scores` [10]
- `int` `hand` [10]

3.1.1 Detailed Description

Definition at line 106 of file network.h.

3.1.2 Field Documentation

3.1.2.1 `char` `car`

information complémentaire sur la nature des données

Definition at line 109 of file network.h.

3.1.2.2 `DataType` `dataType`

type de donnée concerné

Definition at line 108 of file `network.h`.

3.1.2.3 `bool` `end`

Definition at line 101 of file `network.hpp`.

3.1.2.4 `_Bool` `end`

true si le client n'a pas à attendre d'autre données de ce type

Definition at line 122 of file `network.h`.

3.1.2.5 `int` `from`

Definition at line 89 of file `network.hpp`.

3.1.2.6 `int32_t` `from`

source d'un message ou mot de passe pour rejoindre une salle

Definition at line 111 of file `network.h`.

3.1.2.7 `int` `hand[10]`

Definition at line 113 of file `network.hpp`.

3.1.2.8 `int32_t` `hand[10]`

main du joueur

Definition at line 136 of file `network.h`.

3.1.2.9 `int32_t` `place[10]`

où chaque joueur place sa carte

Definition at line 134 of file `network.h`.

3.1.2.10 `int` `scores[10]`

Definition at line 112 of file `network.hpp`.

3.1.2.11 `int32_t` `scores[10]`

scores de chaque joueur

Definition at line 135 of file `network.h`.

3.1.2.12 char tab

buffer pour des chaines de caractère

nom des salles espacé par '
,

Definition at line 110 of file network.h.

3.1.2.13 int table[4][6]

Definition at line 109 of file network.hpp.

3.1.2.14 int32_t table[4][5]

les cartes sur la table, 4 ranges de 10 cartes maximum, vaut 0 si derniere carte de la rangée

Definition at line 131 of file network.h.

3.1.2.15 int turn[10]

Definition at line 111 of file network.hpp.

3.1.2.16 int32_t turn[10]

choix de chaque joueurs

Definition at line 133 of file network.h.

3.1.2.17 char users

nom de chaques joueurs

Definition at line 132 of file network.h.

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

- src/Network/[network.h](#)
- src/Network/[network.hpp](#)

3.2 Button Struct Reference

élément graphique, cliquable.

```
#include <button.h>
```

Data Fields

- SDL_Texture * [texture](#)
- SDL_Texture * [selTexture](#)
- SDL_Rect [rect](#)
- bool [select](#)
- int [size](#)
- void(* [callback](#))(void)

3.2.1 Detailed Description

élément graphique, cliquable.

Definition at line 16 of file button.h.

3.2.2 Field Documentation

3.2.2.1 void(* callback) (void)

callback appelé lors d'un clic sur rect

Definition at line 24 of file button.h.

3.2.2.2 SDL_Rect rect

position et dimension du bouton

Definition at line 20 of file button.h.

3.2.2.3 bool select

si le bouton change de texture

Definition at line 22 of file button.h.

3.2.2.4 SDL_Texture* selTexture

texture pour le bouton sélectionné

Definition at line 19 of file button.h.

3.2.2.5 int size

rapport entre la taille de la surface (dépend de la police) et la texture (entre 10 et 5)

Definition at line 23 of file button.h.

3.2.2.6 SDL_Texture* texture

texture que l'on affiche à l'écran

Definition at line 18 of file button.h.

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

- [src/GUI/button.h](#)

3.3 Card Struct Reference

information de base pour les cartes du jeu

```
#include <card.h>
```

Data Fields

- SDL_Rect [rect](#)
- int [id](#)
- bool [selected](#)

3.3.1 Detailed Description

information de base pour les cartes du jeu

Definition at line 20 of file card.h.

3.3.2 Field Documentation

3.3.2.1 int id

numéro de la carte, compris entre 0 et 104, 0 si non affiché

Definition at line 23 of file card.h.

3.3.2.2 SDL_Rect rect

position et dimension de la carte

Definition at line 22 of file card.h.

3.3.2.3 bool selected

la carte est mise en évidence (si sélectionnée par le joueur)

Definition at line 24 of file card.h.

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

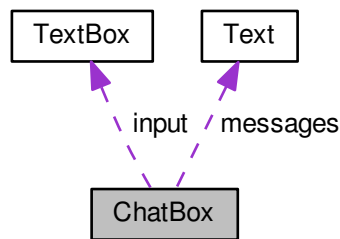
- src/GUI/[card.h](#)

3.4 ChatBox Struct Reference

regroupement de composants graphiques pour faire un chat

```
#include <chatBox.h>
```

Collaboration diagram for ChatBox:



Data Fields

- SDL_Rect [rect](#)
- [Text](#) [messages](#) [NB_MSG]
- char [text](#) [NB_MSG][100]
- [TextBox](#) [input](#)
- void(* [callback](#))(char *)
- bool [update](#)

3.4.1 Detailed Description

regroupement de composants graphiques pour faire un chat

Definition at line 20 of file chatBox.h.

3.4.2 Field Documentation

3.4.2.1 void(* callback) (char *)

optionel (appelé lors de la saisie d'un message par l'utilisateur)

Definition at line 26 of file chatBox.h.

3.4.2.2 TextBox input

entrée de l'utilisateur

Definition at line 25 of file chatBox.h.

3.4.2.3 Text messages[NB_MSG]

lignes, un par message

Definition at line 23 of file chatBox.h.

3.4.2.4 SDL_Rect rect

position et dimension pour l'ensemble des composants

Definition at line 22 of file chatBox.h.

3.4.2.5 char text[NB_MSG][100]

chaîne de caractères, correspondant au contenu de messages

Definition at line 24 of file chatBox.h.

3.4.2.6 bool update

true si avant rendu on doit mettre à jour message avec text

Definition at line 27 of file chatBox.h.

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

- src/GUI/[chatBox.h](#)

3.5 Data Struct Reference

données de base Données pour toute la partie connection

```
#include <network.h>
```

3.5.1 Detailed Description

données de base Données pour toute la partie connection

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

- src/Network/[network.h](#)

3.6 DataGame Struct Reference

Données pour le jeu en lui même.

```
#include <network.h>
```

3.6.1 Detailed Description

Données pour le jeu en lui même.

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

- src/Network/[network.h](#)

3.7 DataList Struct Reference

données spécifiques à la liste des salles sont comme une liste chaînée, dernier paquet contient end à true

```
#include <network.h>
```

3.7.1 Detailed Description

données spécifiques à la liste des salles sont comme une liste chaînée, dernier paquet contient end à true

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

- src/Network/[network.h](#)

3.8 Joueur Struct Reference

Objet positionnement et valeurs de données sur chaque joueur.

```
#include <game.h>
```

Data Fields

- char [nom](#) [[SIZESTR](#)]
- int [tete](#)
- SDL_Rect [rect](#)

3.8.1 Detailed Description

Objet positionnement et valeurs de données sur chaque joueur.

permet de savoir quoi afficher et ou pour les score et joueurs

Definition at line 43 of file game.h.

3.8.2 Field Documentation

3.8.2.1 char [nom](#)[[SIZESTR](#)]

chaîne de caractère nom du joueur

Definition at line 44 of file game.h.

3.8.2.2 SDL_Rect [rect](#)

positionnement à l'écran de ces informations

Definition at line 46 of file game.h.

3.8.2.3 int [tete](#)

nombre de tête de boeuf (score) du joueur

Definition at line 45 of file game.h.

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

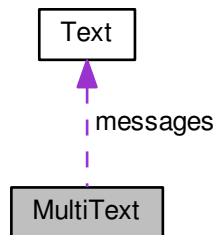
- src/Game/[game.h](#)

3.9 MultiText Struct Reference

Objet pour l'affichage sur plusieurs lignes.

```
#include <multiText.h>
```

Collaboration diagram for MultiText:



Data Fields

- [Text messages](#) [[NB_LINE](#)]
- char [text](#) [[NB_LINE](#)][100]
- bool [update](#)
- void(* [callback](#))(char *)
- bool [initialised](#)

3.9.1 Detailed Description

Objet pour l'affichage sur plusieurs lignes.

Definition at line 17 of file multiText.h.

3.9.2 Field Documentation

3.9.2.1 void(* callback)(char *)

appelle le callback lors d'un click, texte sélectionné donné en parametre

Definition at line 22 of file multiText.h.

3.9.2.2 bool initialised

true si on peut exploiter les données lors de clic false sinon

Definition at line 23 of file multiText.h.

3.9.2.3 Text messages[NB_LINE]

chaque ligne

Definition at line 19 of file multiText.h.

3.9.2.4 char text[NB_LINE][100]

chaines de caractères correspondant aux messages

Definition at line 20 of file multiText.h.

3.9.2.5 bool update

pour une mise à jour de messages avec text dans renderMultiText

Definition at line 21 of file multiText.h.

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

- [src/GUI/multiText.h](#)

3.10 Picture Struct Reference

Object contenant toutes les informations nssaires d'une image.

```
#include <picture.h>
```

Data Fields

- SDL_Texture * [texture](#)
- SDL_Rect [rect](#)

3.10.1 Detailed Description

Object contenant toutes les informations nssaires d'une image.

Definition at line 15 of file picture.h.

3.10.2 Field Documentation

3.10.2.1 SDL_Rect rect

position et dimenssion de l'image

Definition at line 18 of file picture.h.

3.10.2.2 SDL_Texture* texture

donn affich

Definition at line 17 of file picture.h.

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

- [src/GUI/picture.h](#)

3.11 Statistique Struct Reference

donnée de base des statistiques

```
#include <stat.h>
```

Data Fields

- char [nom](#) [[SIZESTR](#)]
- int [nb_vic](#)
- int [nb_def](#)

3.11.1 Detailed Description

donnée de base des statistiques

Definition at line 26 of file stat.h.

3.11.2 Field Documentation

3.11.2.1 int nb_def

nombre de défaite

Definition at line 30 of file stat.h.

3.11.2.2 int nb_vic

nombre de victoire

Definition at line 29 of file stat.h.

3.11.2.3 char nom[SIZESTR]

nom du joueur

Definition at line 28 of file stat.h.

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

- [src/Game/stat.h](#)

3.12 Text Struct Reference

Objet comprenant ce qu'il faut pour afficher du texte.

```
#include <text.h>
```

Data Fields

- SDL_Texture * [texture](#)
- SDL_Rect [rect](#)
- int [size](#)

3.12.1 Detailed Description

Objet comprenant ce qu'il faut pour afficher du texte.

Definition at line 16 of file text.h.

3.12.2 Field Documentation

3.12.2.1 SDL_Rect rect

position et dimension

Definition at line 19 of file text.h.

3.12.2.2 int size

rapport entre les dimension de la surface (police) et de la texture(affiché à l'écran)

Definition at line 20 of file text.h.

3.12.2.3 SDL_Texture* texture

données à afficher

Definition at line 18 of file text.h.

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

- [src/GUI/text.h](#)

3.13 TextBox Struct Reference

```
#include <textBox.h>
```

Data Fields

- SDL_Texture * [texture](#)
- SDL_Rect [rect](#)
- SDL_Rect [maxRect](#)
- SDL_Rect [rBar](#)
- bool [select](#)
- char [text](#) [100]
- int [maxcar](#)
- int [wLettre](#)

3.13.1 Detailed Description

Definition at line 17 of file textBox.h.

3.13.2 Field Documentation

3.13.2.1 int maxcar

nombre maximum de caractère

Definition at line 28 of file textBox.h.

3.13.2.2 SDL_Rect maxRect

cadre noir pour délimiter la zone de saisie

Definition at line 21 of file textBox.h.

3.13.2.3 SDL_Rect rBar

rectangle de la barre clignotante

Definition at line 23 of file textBox.h.

3.13.2.4 SDL_Rect rect

Definition at line 20 of file textBox.h.

3.13.2.5 bool select

si la text box capture la saisie et affiche la bare clignotante

Definition at line 25 of file textBox.h.

3.13.2.6 char text[100]

Definition at line 27 of file textBox.h.

3.13.2.7 SDL_Texture* texture

donnée à afficher

Definition at line 19 of file textBox.h.

3.13.2.8 int wLettre

largeur d'une lettre (maximale)

Definition at line 29 of file textBox.h.

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

- [src/GUI/textBox.h](#)

3.14 textBox Struct Reference

regroupement de composants graphiques pour faire de la saisie de texte

```
#include <textBox.h>
```

3.14.1 Detailed Description

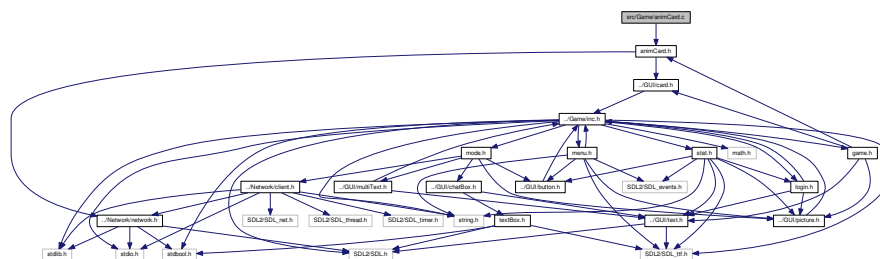
regroupement de composants graphiques pour faire de la saisie de texte

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

- [src/GUI/textBox.h](#)

File Documentation

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



- bool `isAnimating` ()
retourn si les animations sont en cours
- bool `compareTable` ()
comparer table et new table pour dg affiche dans la terminable les diffnces
- int `getNbBuffle` (int cardId)
retourne le nombre de t d'une carte
- void `afficherTable` (Card tab[RANGEE][CPRANGEE])
- void `positionnerTable` (Card tab[RANGEE][CPRANGEE])
- void `positionnerJoueur` (SDL_Rect *tab)
- void `initAnimCard` ()
initialisation des ressources graphiques pour les cartes anim
- void `updateAnimCard` (DataGame *data)
met ur l'affichage avec les donne paquet note: doit e apeller le thread principale
- void `renderAnimCard` ()
- void `destroyAnimCard` ()

- int xi

- int `yi`
- int `xf`
- int `yf`

4.1.1 Function Documentation

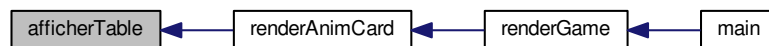
4.1.1.1 void `afficherTable` (`Card tab[RANGEE][CPRANGEE]`)

Definition at line 324 of file `animCard.c`.

Here is the call graph for this function:



Here is the caller graph for this function:



4.1.1.2 bool `compareTable` ()

comparer table et new table pour dg affiche dans la terminable les diffnces

Returns

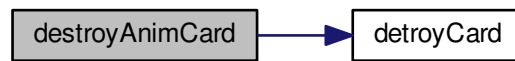
true si les deux tableau sont identiques

Definition at line 46 of file `animCard.c`.

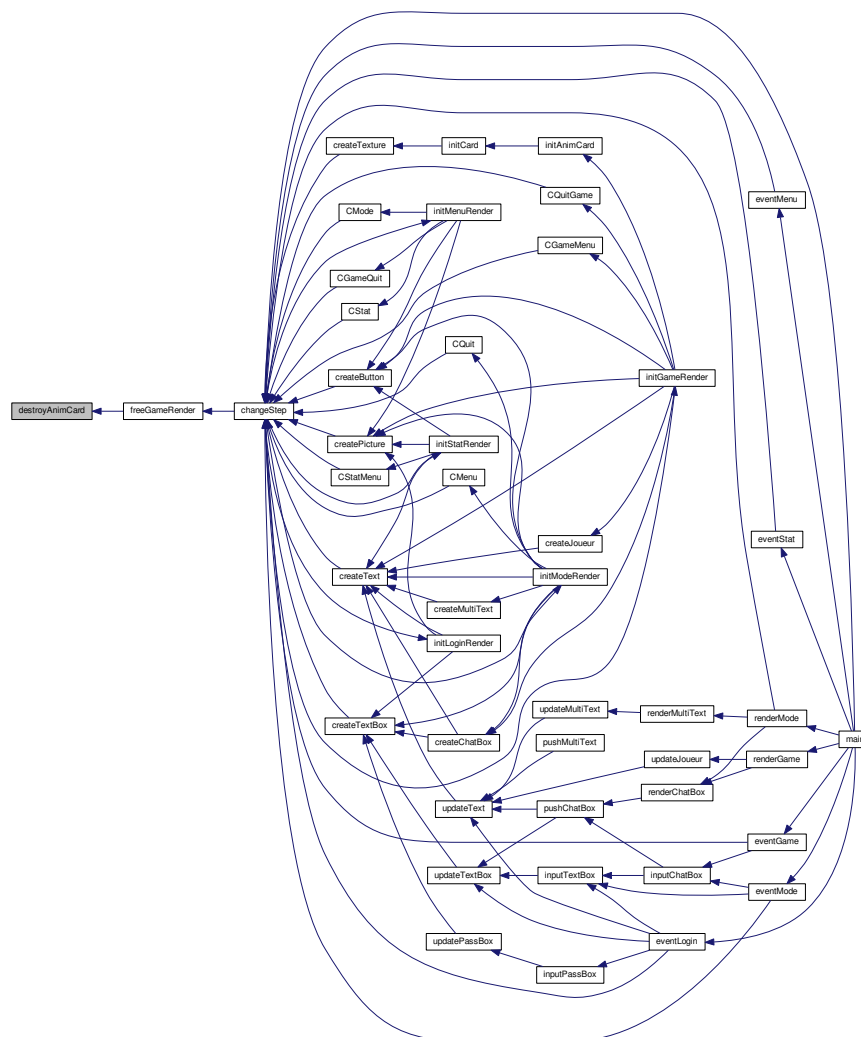
4.1.1.3 void `destroyAnimCard` ()

Definition at line 447 of file `animCard.c`.

Here is the call graph for this function:



Here is the caller graph for this function:



4.1.1.4 int getNbBuffle (int cardId)

retourne le nombre de t d'une carte

Parameters

<i>cardId</i>	le num sur la carte
---------------	---------------------

Returns

le nombre de carte

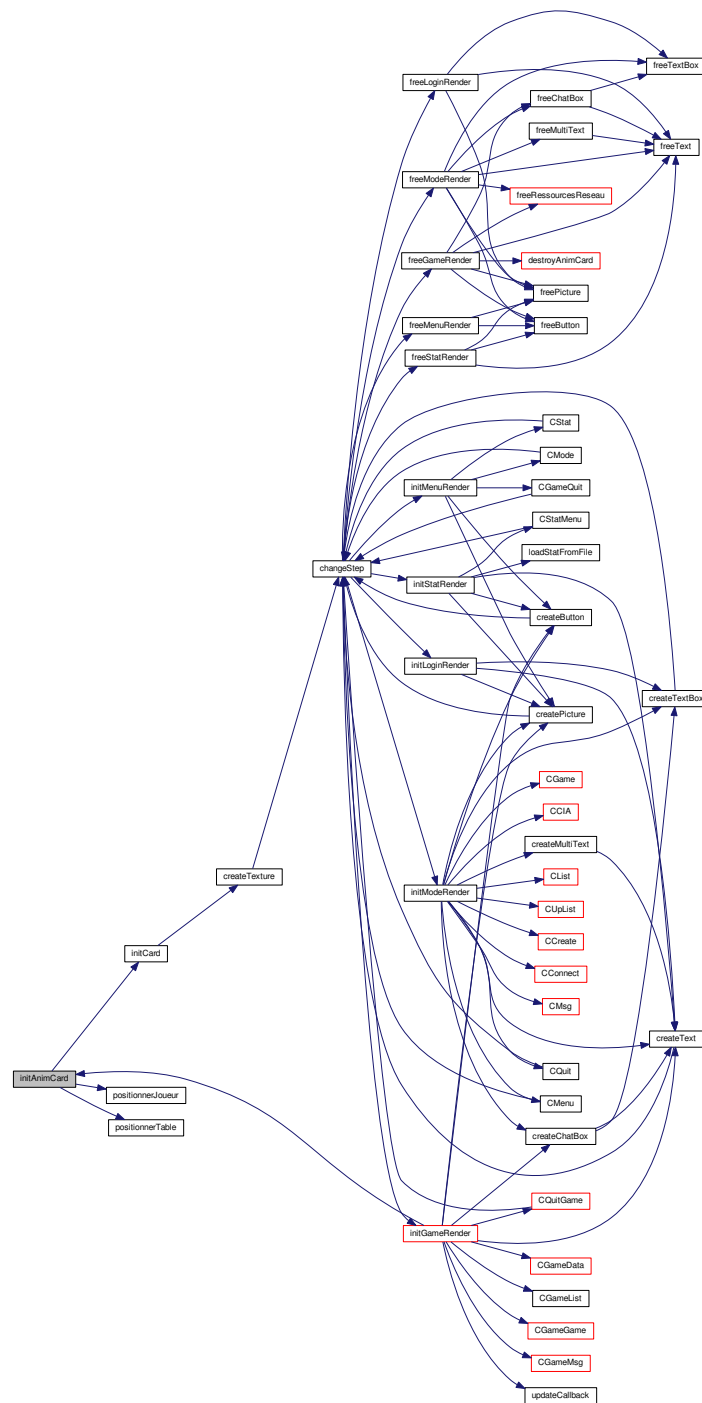
Definition at line 74 of file animCard.c.

4.1.1.5 void initAnimCard ()

initialisation des ressources graphiques pour les cartes anim

Definition at line 368 of file animCard.c.

Here is the call graph for this function:



4.2.2.2 `#define VITESSE 1000.0f`

Definition at line 13 of file animCard.h.

4.2.3 Function Documentation

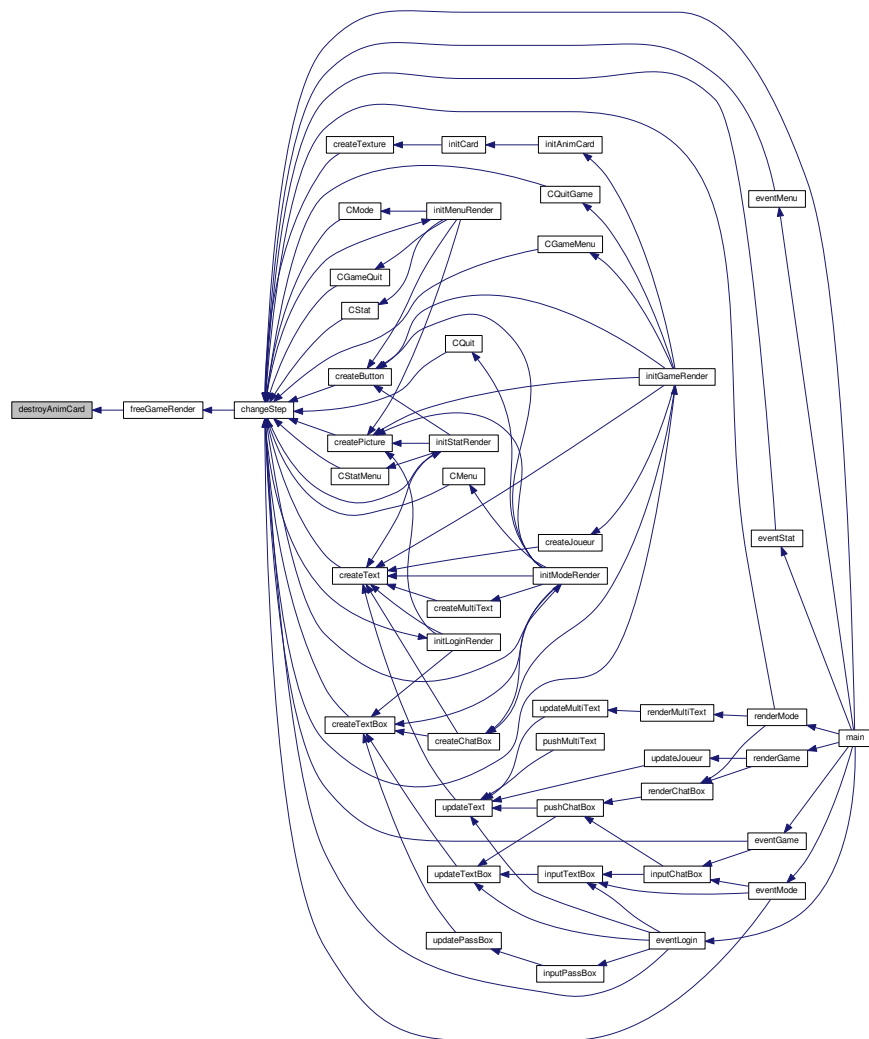
4.2.3.1 `void destroyAnimCard ()`

Definition at line 447 of file animCard.c.

Here is the call graph for this function:



Here is the caller graph for this function:



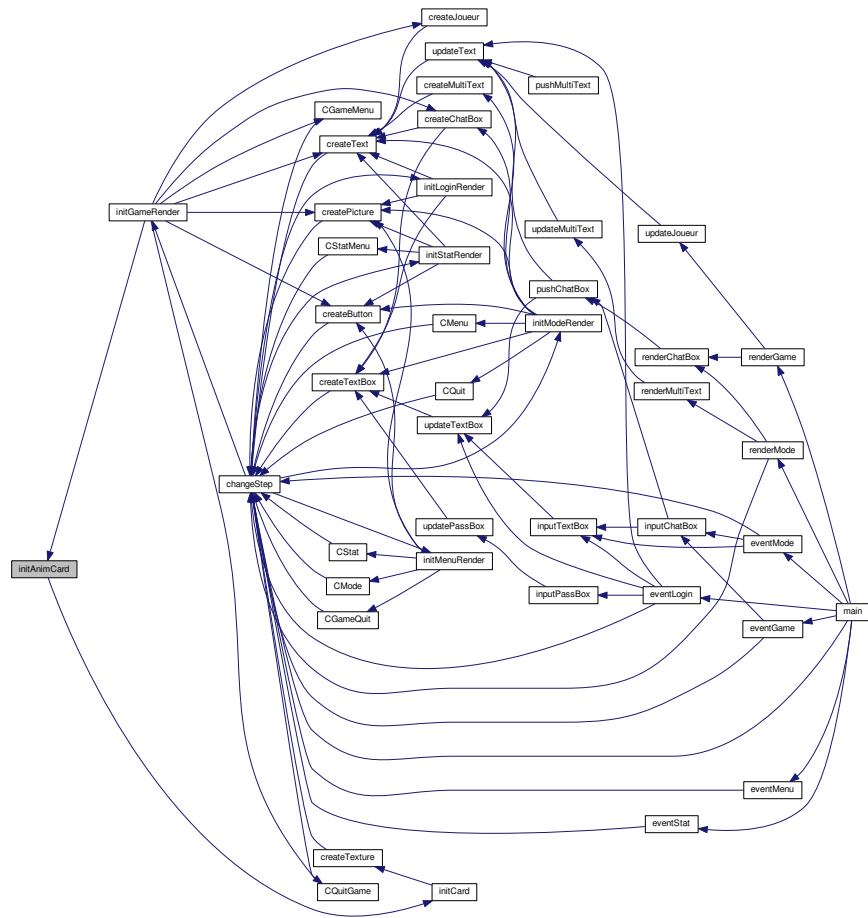
4.2.3.2 void initAnimCard ()

initialisation des ressources graphiques pour les cartes anim

Definition at line 368 of file animCard.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.2.3.3 bool isAnimating ()

retourn si les animations sont en cours

Returns

true si les animations sont en cours

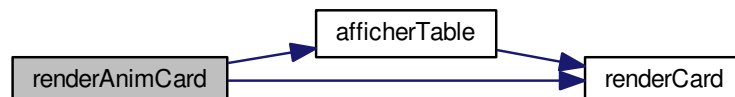
Definition at line 34 of file animCard.c.

Here is the caller graph for this function:

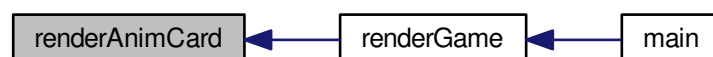
4.2.3.4 void renderAnimCard ()

Definition at line 415 of file animCard.c.

Here is the call graph for this function:



Here is the caller graph for this function:



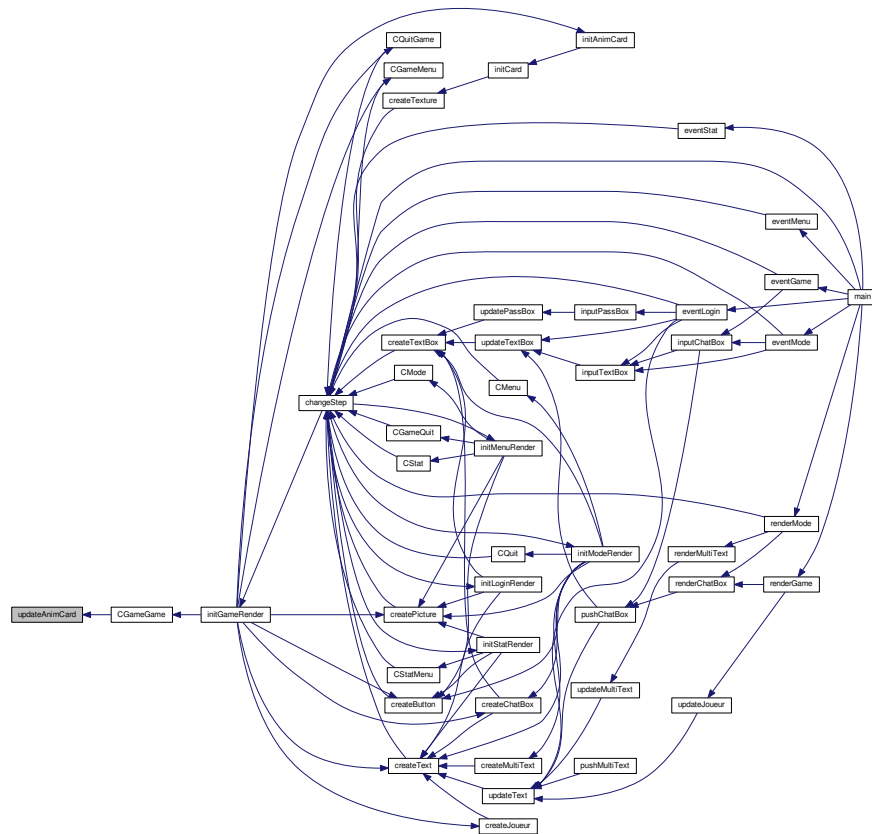
4.2.3.5 void updateAnimCard (**DataGame** * *data*)

met ur l'affichage avec les donne paquet note: doit e apellr le thread principale

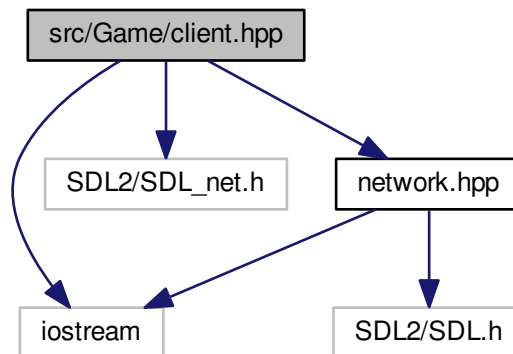
Parameters

<i>data</i>	paquet contenant les donn
-------------	---------------------------

Definition at line 382 of file animCard.c.



Include dependency graph for client.hpp:



Functions

- void `initialisation` (int argc, char **argv)
- void `freeRessources` ()
- void `wait` ()
- void `receptList` ()
- void `receptGame` ()
- void `reception` ()

lancer la réception dans un nouveau thread les données sont retournées par des callback définis dans initialisation↔ Réseau ou updateCallback

- void `sendMsg` ()
- void `askList` ()

demande au serveur de transmettre la liste des salles non bloquante et la réponse du serveur sera transmise au callback associé à son type (par le thread de réception)

- void `join` ()
- void `create` ()
- void `startGame` ()

demande à commencer une partie réponse du serveur via callback

- void `choice` ()

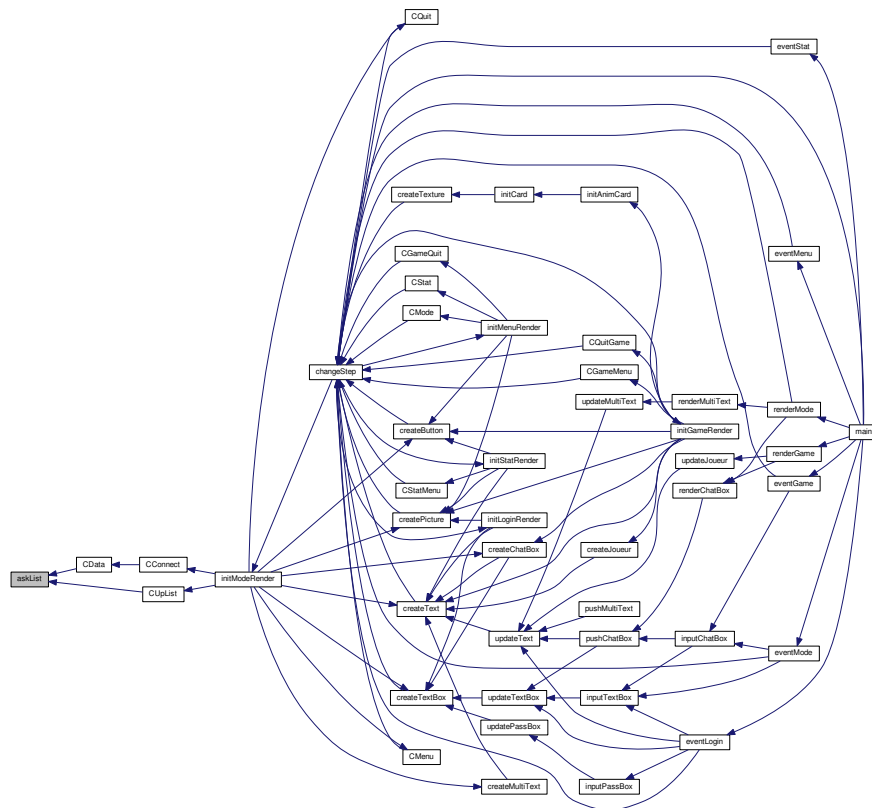
4.3.1 Function Documentation

4.3.1.1 void askList ()

demande au serveur de transmettre la liste des salles non bloquante et la réponse du serveur sera transmise au callback associé à son type (par le thread de réception)

Definition at line 174 of file client.c.

Here is the caller graph for this function:



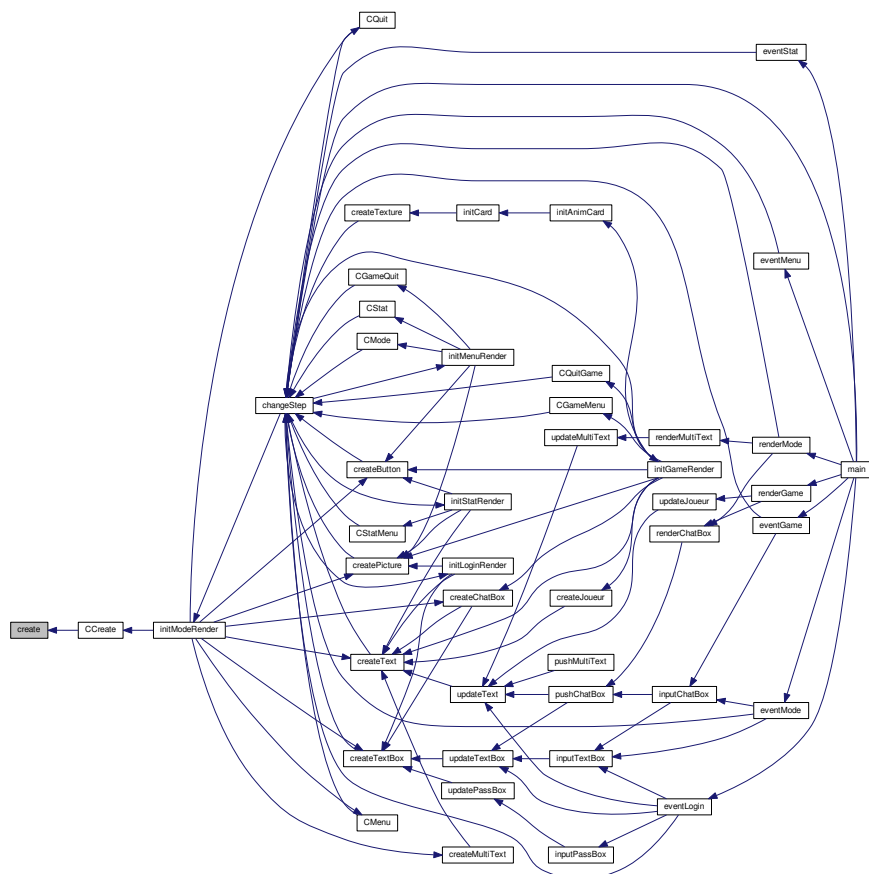
4.3.1.2 void choice ()

Here is the caller graph for this function:



4.3.1.3 void create ()

Here is the caller graph for this function:

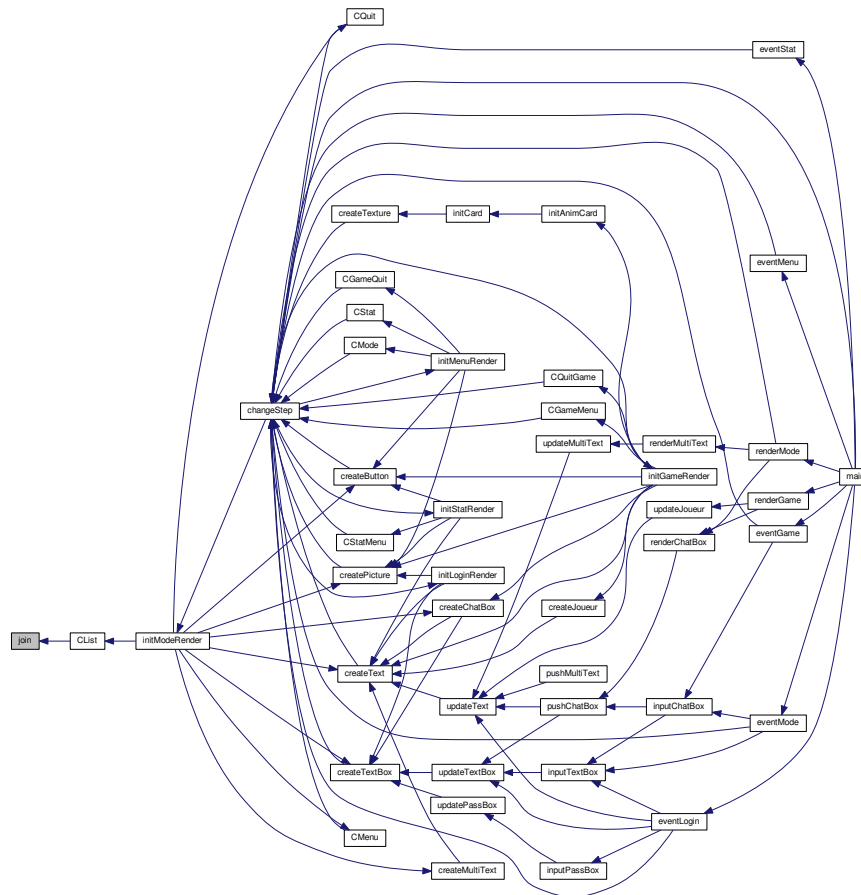


4.3.1.4 void freeResources ()

4.3.1.5 void initialisation (int *argc*, char ** *argv*)

4.3.1.6 void join ()

Here is the caller graph for this function:



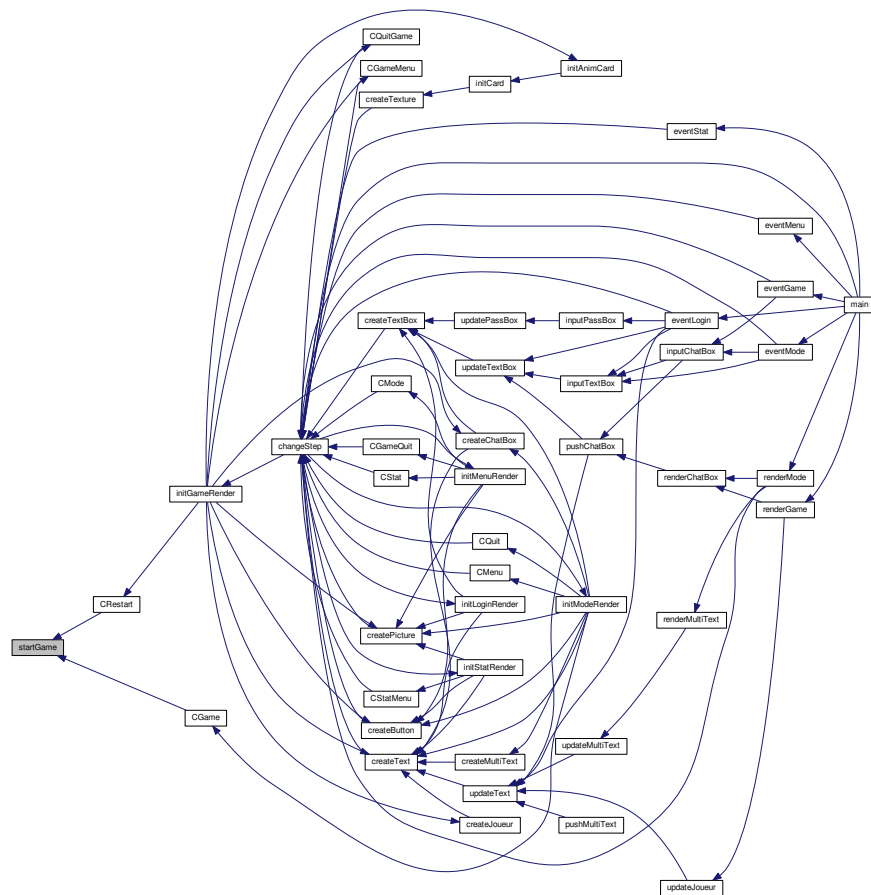
4.3.1.7 void receptGame ()

4.3.1.8 void reception ()

lancer la réception dans un nouveau thread les données sont retournées par des callback définis dans initialisationReseau ou updateCallback

Definition at line 149 of file client.c.

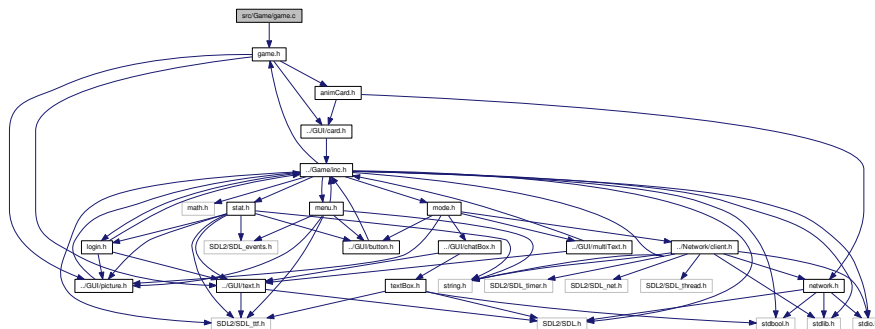
Here is the caller graph for this function:



4.3.1.12 void wait ()

4.4 src/Game/game.c File Reference

```
#include "game.h"
// Include dependency graph for game.c:
```



Functions

- void [CGameData](#) ([Data](#) *data)
- void [CGameList](#) ([DataList](#) *data)
- void [CGameGame](#) ([DataGame](#) *data)
- void [updateJoueur](#) ()
met à jour les donnees à afficher pour chaque joueur
- void [createJoueur](#) ()
creer les donnees à afficher pour chaque joueur se sert de données du tableau joueurs (cf [game.c](#))
- void [ordonner](#) ()
ranger la main le tableau représentant la main du joueur est 'poignee' (cf [game.c](#))
- void [CCard](#) (int id)
- void [eventGame](#) ()
Gestion des evenements spécifique au jeu.
- void [CGameMsg](#) (char *msg)
- void [CQuitGame](#) ()
- void [CGameMenu](#) ()
- void [CRestart](#) ()
- void [initGameRender](#) ()
Initialisation des ressources graphiques spécifique au jeu.
- void [renderGame](#) ()
dessine ressources graphiques spécifique au jeu
- void [freeGameRender](#) ()
Libere les ressources graphiques initialisees par initGameRender.

4.4.1 Function Documentation

4.4.1.1 void CCard (int id)

Definition at line 122 of file game.c.

Here is the call graph for this function:



Here is the caller graph for this function:



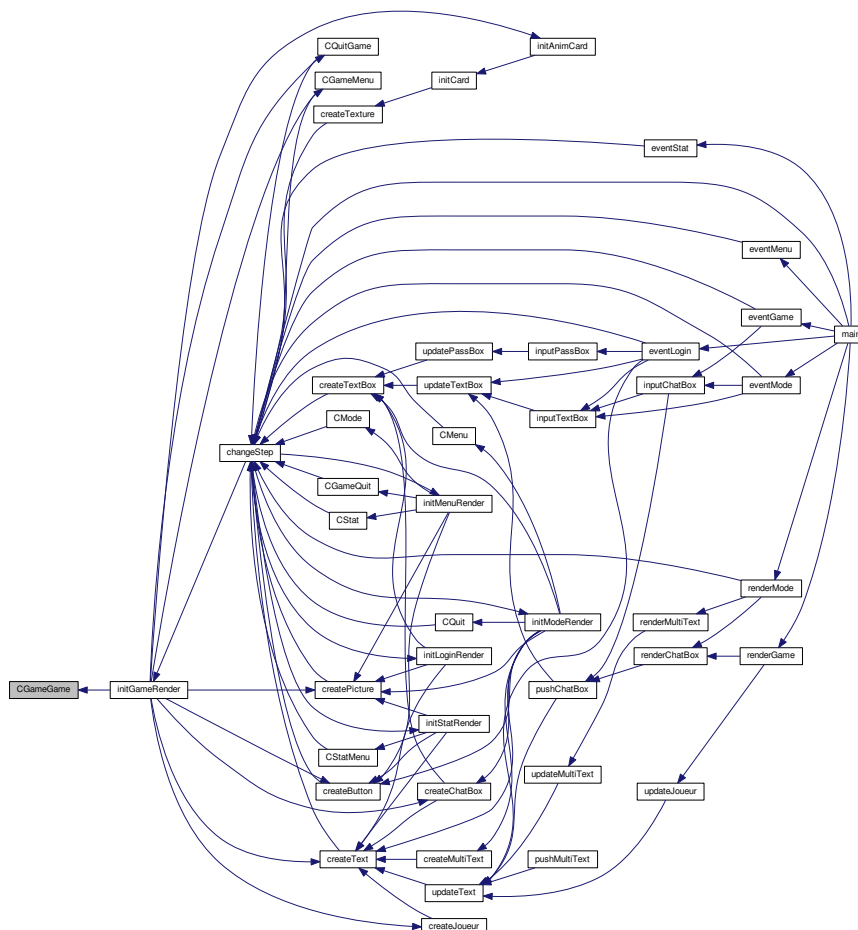
4.4.1.3 void CGameGame (DataGame * data)

Definition at line 63 of file game.c.

Here is the call graph for this function:



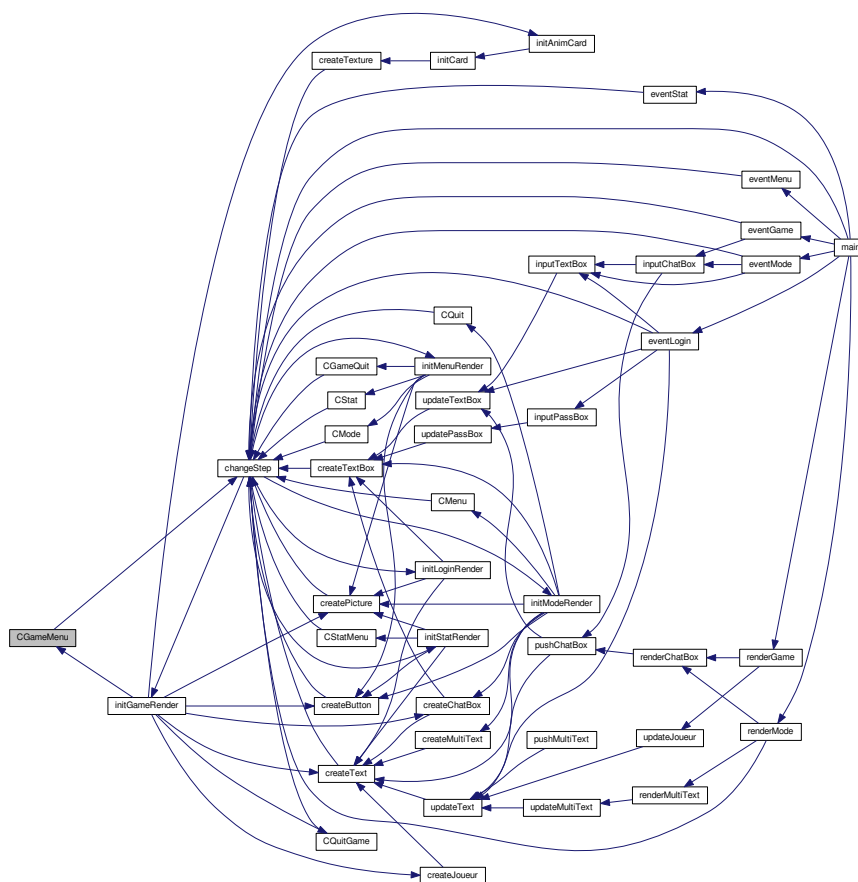
Here is the caller graph for this function:



4.4.1.4 void CGameList (DataList * data)

Definition at line 59 of file game.c.

Here is the caller graph for this function:



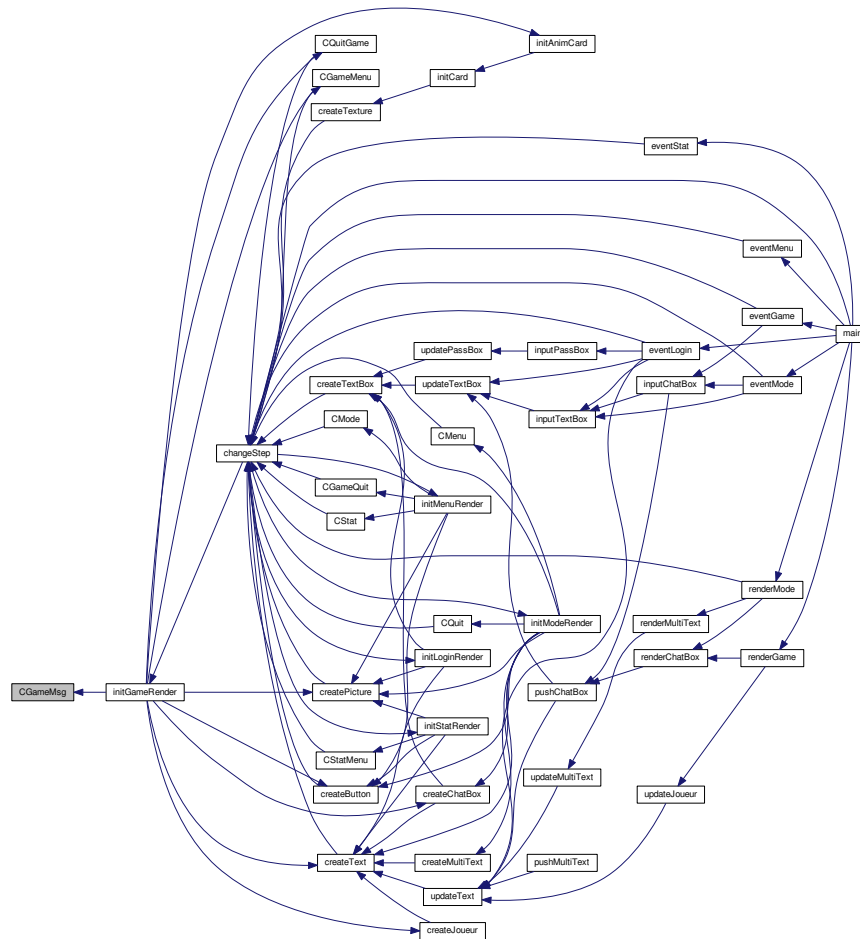
4.4.1.6 void CGameMsg (char * msg)

Definition at line 152 of file game.c.

Here is the call graph for this function:



Here is the caller graph for this function:

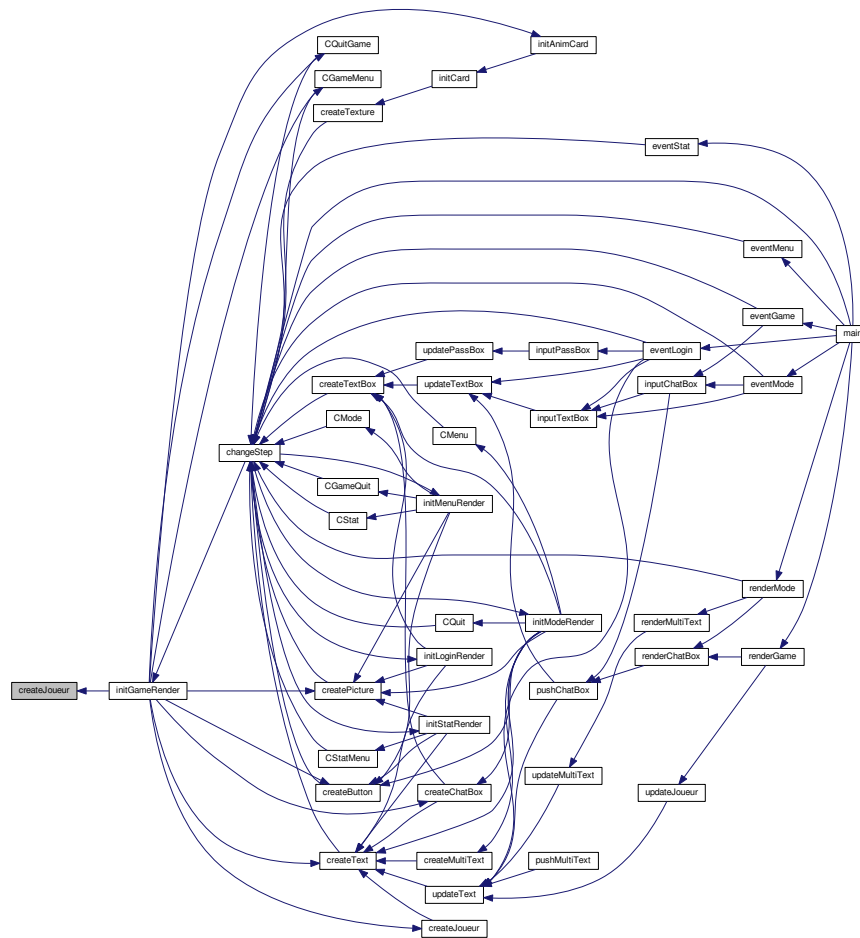


4.4.1.7 void CQuitGame ()

Definition at line 157 of file game.c.

Here is the call graph for this function:

Here is the caller graph for this function:



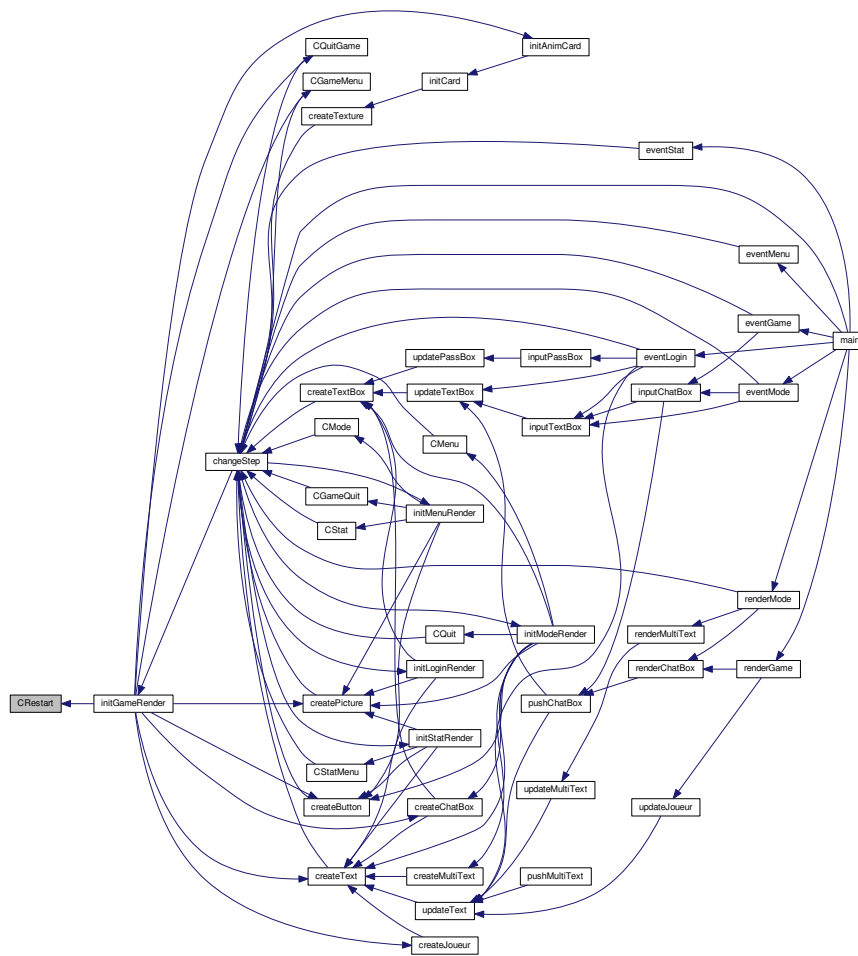
4.4.1.9 void CRestart ()

Definition at line 169 of file game.c.

Here is the call graph for this function:



Here is the caller graph for this function:



4.4.1.10 void eventGame ()

Gestion des evenements spécifique au jeu.

Definition at line 127 of file game.c.

Here is the call graph for this function:

Here is the caller graph for this function:

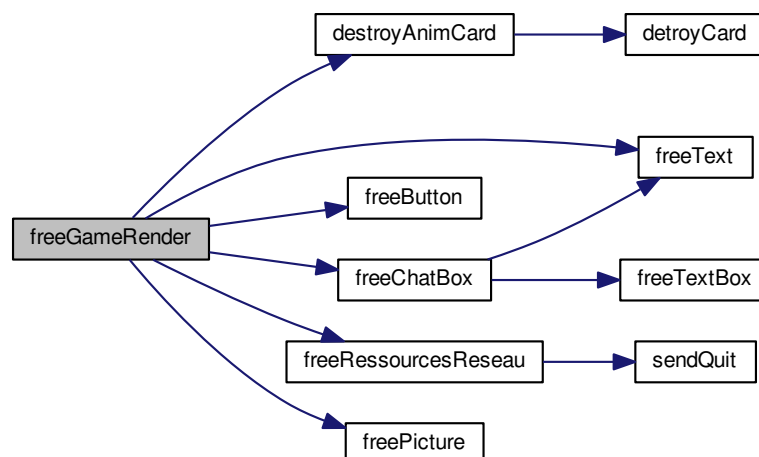


4.4.1.11 void freeGameRender ()

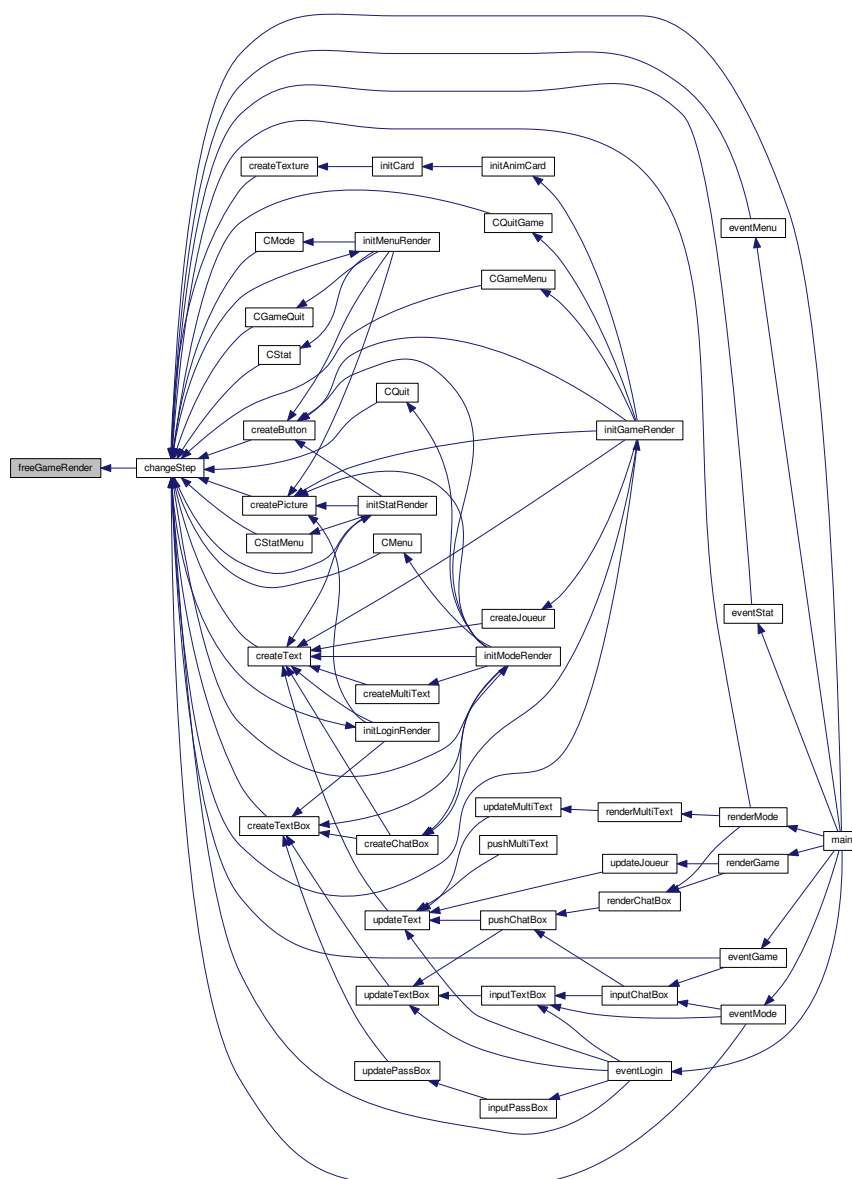
Libere les ressources graphiques initialisees par initGameRender.

Definition at line 250 of file game.c.

Here is the call graph for this function:



Here is the caller graph for this function:

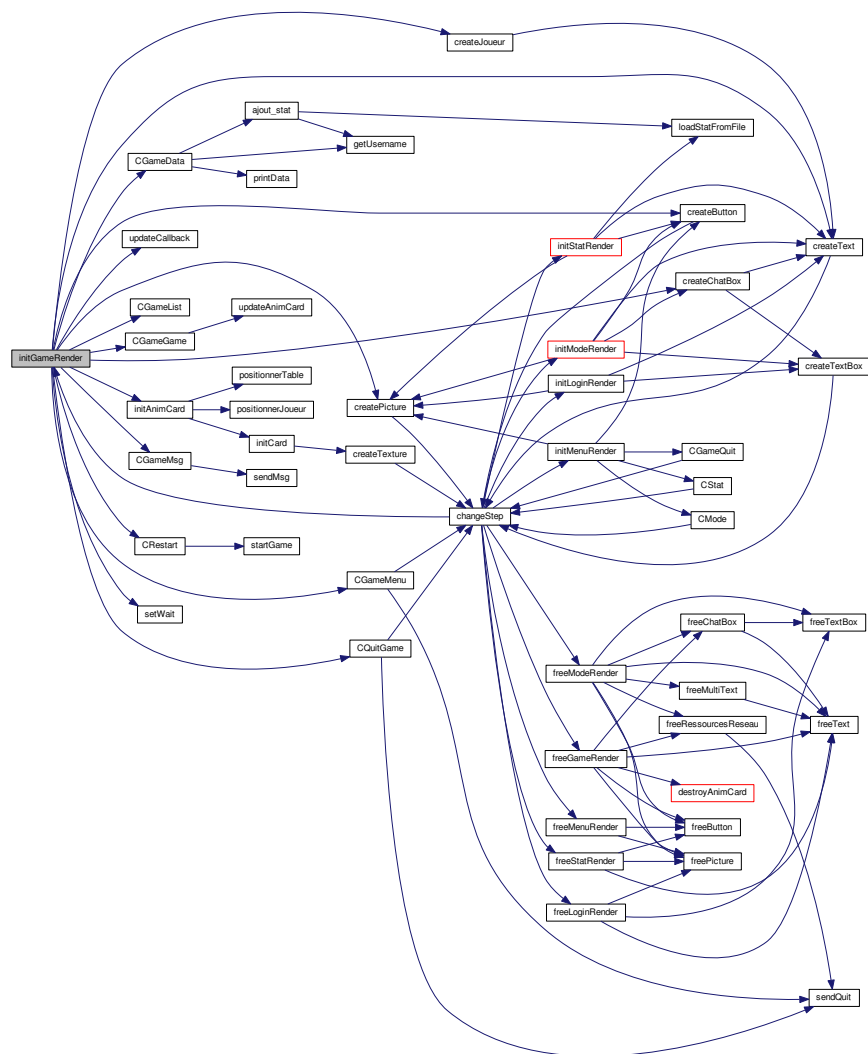


4.4.1.12 void initGameRender ()

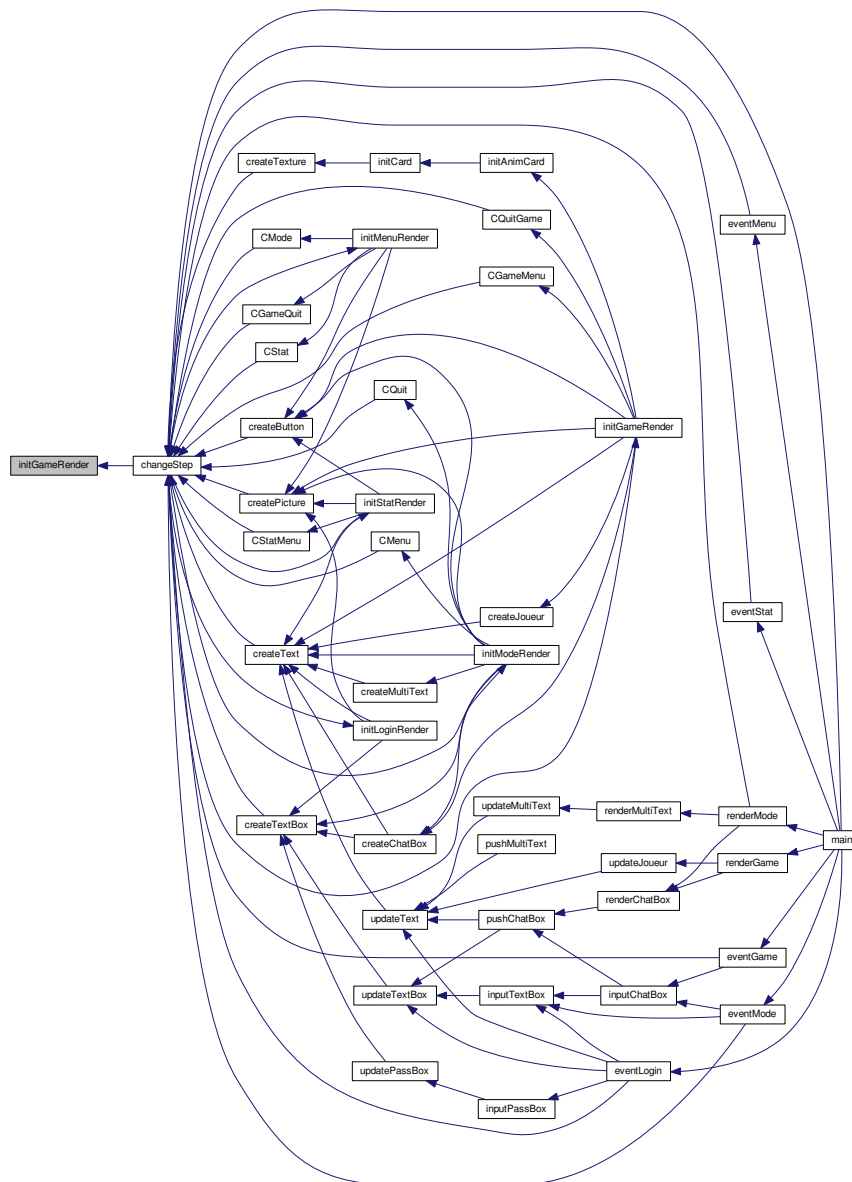
Initialisation des ressources graphiques spécifique au jeu.

Definition at line 176 of file game.c.

Here is the call graph for this function:



Here is the caller graph for this function:



4.4.1.13 void ordonner ()

ranger la main le tableau représentant la main du joueur est 'poignee' (cf [game.c](#))

Definition at line 106 of file game.c.

4.4.1.14 void renderGame ()

dessine ressources graphiques spécifique au jeu

Definition at line 221 of file game.c.

Here is the call graph for this function:

Here is the caller graph for this function:



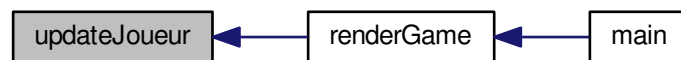
4.4.1.15 void updateJoueur ()

met à jour les donnees à afficher pour chaque joueur

Definition at line 81 of file game.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.5 src/Game/game.h File Reference

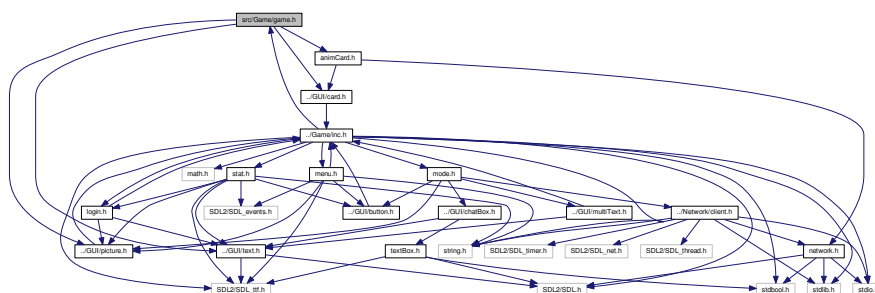
interface graphique du jeu

```

#include "../GUI/card.h"
#include "animCard.h"
#include "../GUI/text.h"
#include "../GUI/picture.h"

```

Include dependency graph for game.h:



4.5.1 Detailed Description

interface graphique du jeu

Author

Aubin Detrez

affichage des informations depuis les données du serveur

4.5.2 Macro Definition Documentation

4.5.2.1 `#define CPRANGEE 5`

Definition at line 28 of file game.h.

4.5.2.2 `#define DECHAND 5`

Definition at line 26 of file game.h.

4.5.2.3 `#define DECRANGEE 7`

Definition at line 31 of file game.h.

4.5.2.4 `#define HAND 10`

Definition at line 23 of file game.h.

4.5.2.5 `#define MAXJOUEUR 10`

Definition at line 21 of file game.h.

4.5.2.6 `#define NB_CARD 104`

Definition at line 22 of file game.h.

4.5.2.7 `#define POSHAND_X 20`

Definition at line 24 of file game.h.

4.5.2.8 `#define POSHAND_Y 400`

Definition at line 25 of file game.h.

4.5.2.9 `#define POSJOUEUR_X 800`

Definition at line 32 of file game.h.

4.5.2.10 `#define POSJOUEUR_Y 5`

Definition at line 33 of file game.h.

4.5.2.11 `#define POSTABLE_X 10`

Definition at line 29 of file game.h.

4.5.2.12 `#define POSTABLE_Y 20`

Definition at line 30 of file game.h.

4.5.2.13 `#define RANGE 4`

Definition at line 27 of file game.h.

4.5.2.14 `#define SIZEJOUEUR 80`

Definition at line 34 of file game.h.

4.5.2.15 `#define SIZESTR 21`

Definition at line 20 of file game.h.

4.5.3 Function Documentation

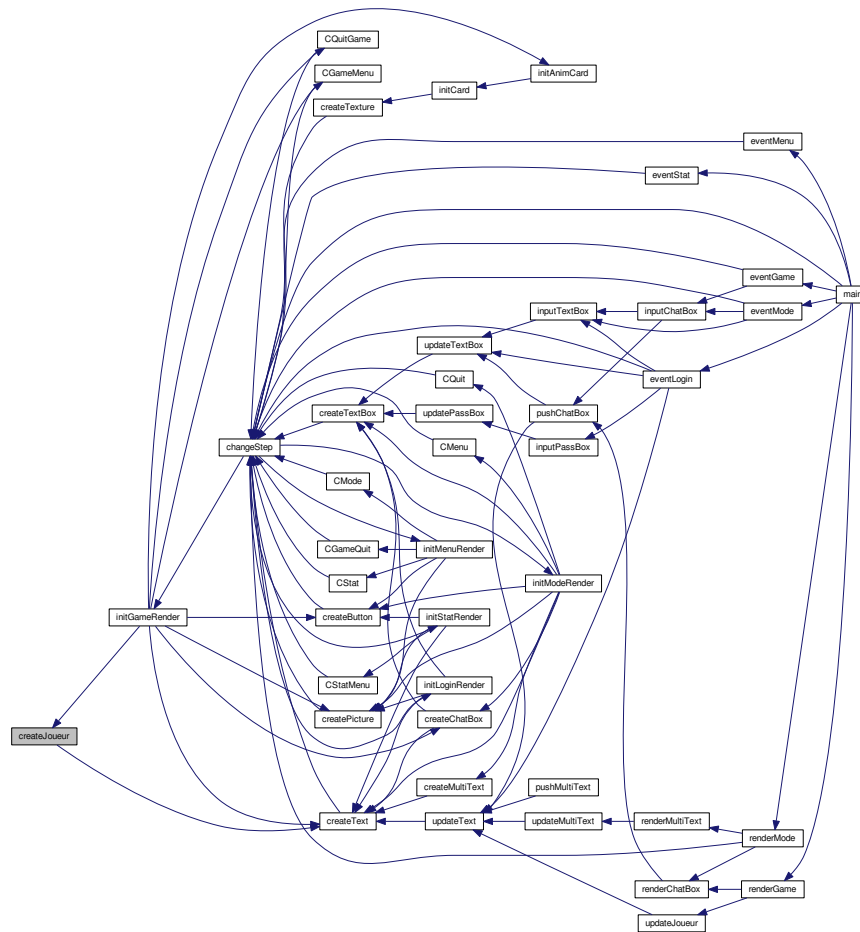
4.5.3.1 `void createJoueur ()`

créer les données à afficher pour chaque joueur se sert de données du tableau joueurs (cf [game.c](#))

Definition at line 95 of file game.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.5.3.2 void eventGame ()

Gestion des evenements spécifique au jeu.

Definition at line 127 of file game.c.

Here is the call graph for this function:

Here is the caller graph for this function:

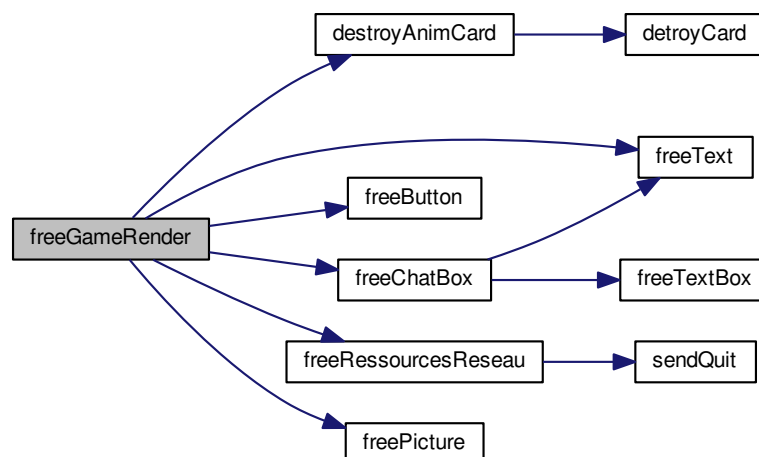


4.5.3.3 void freeGameRender ()

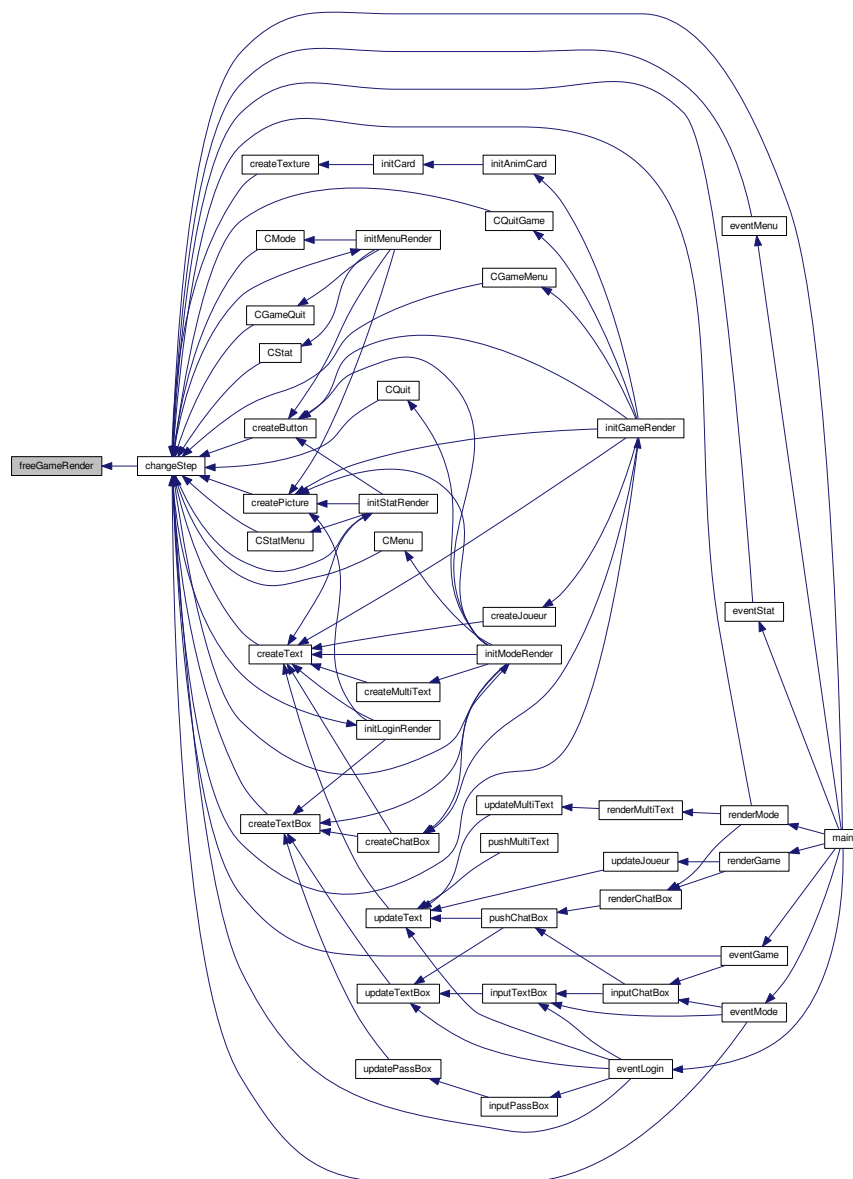
Libere les ressources graphiques initialisees par initGameRender.

Definition at line 250 of file game.c.

Here is the call graph for this function:



Here is the caller graph for this function:

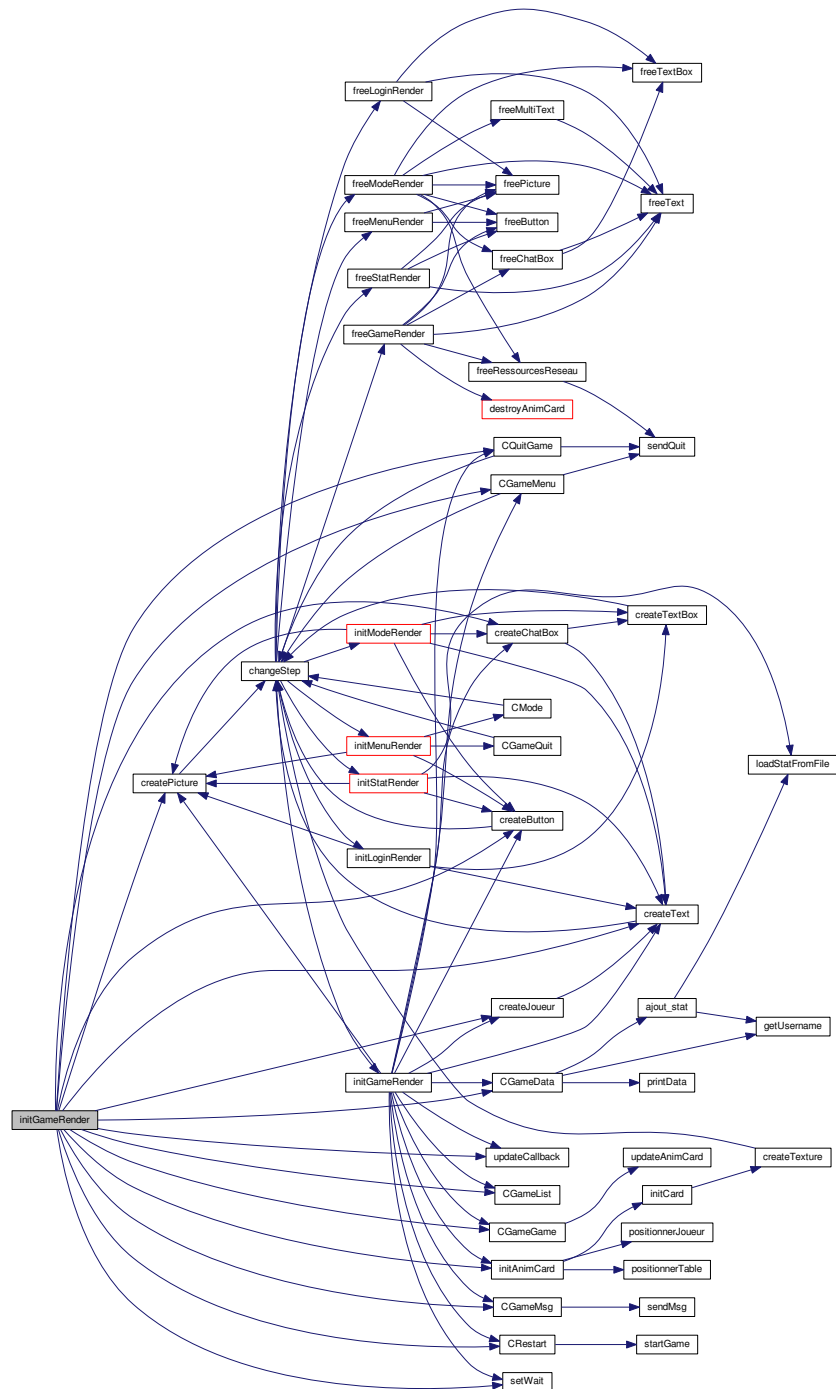


4.5.3.4 void initGameRender ()

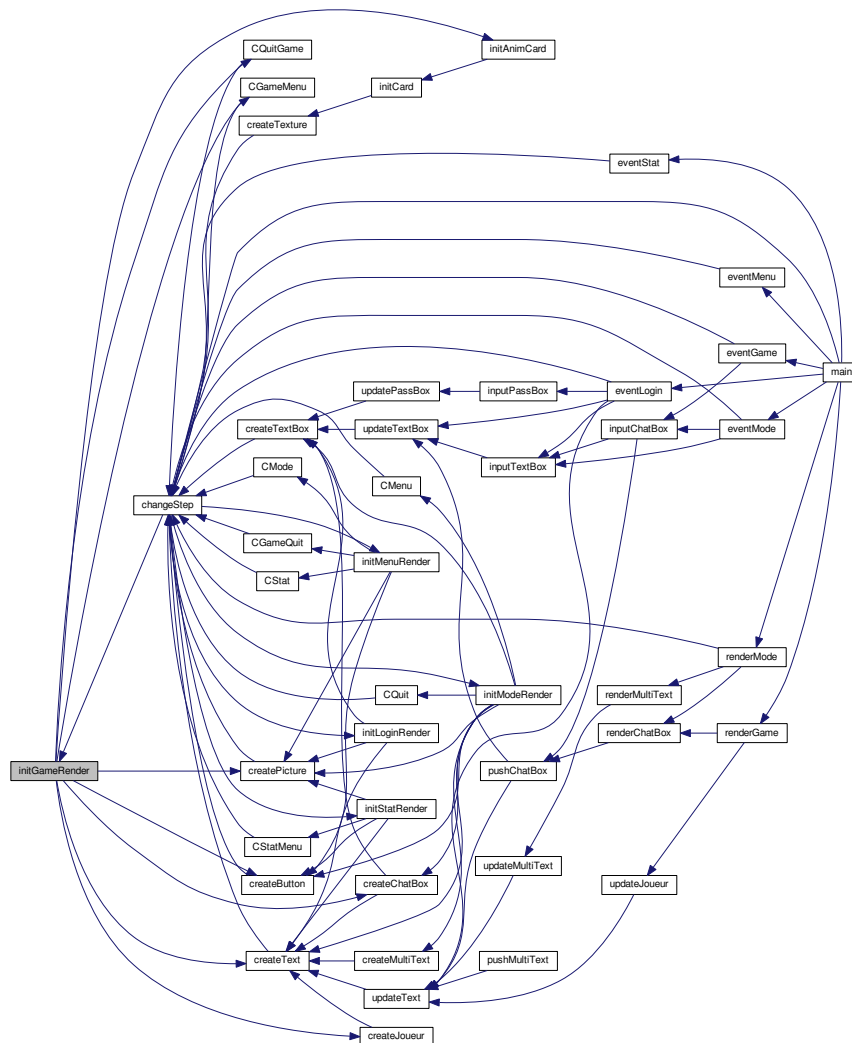
Initialisation des ressources graphiques spécifique au jeu.

Definition at line 176 of file game.c.

Here is the call graph for this function:



Here is the caller graph for this function:



4.5.3.5 void ordonner ()

ranger la main le tableau représentant la main du joueur est 'poignee' (cf [game.c](#))

Definition at line 106 of file game.c.

4.5.3.6 void renderGame ()

dessine ressources graphiques spécifique au jeu

Definition at line 221 of file game.c.

Here is the call graph for this function:

4.6.1 Function Documentation

4.6.1.1 void changeStep (MainStep nextStep)

permet de mettre à jour l'étape principale initialise et libere les ressources

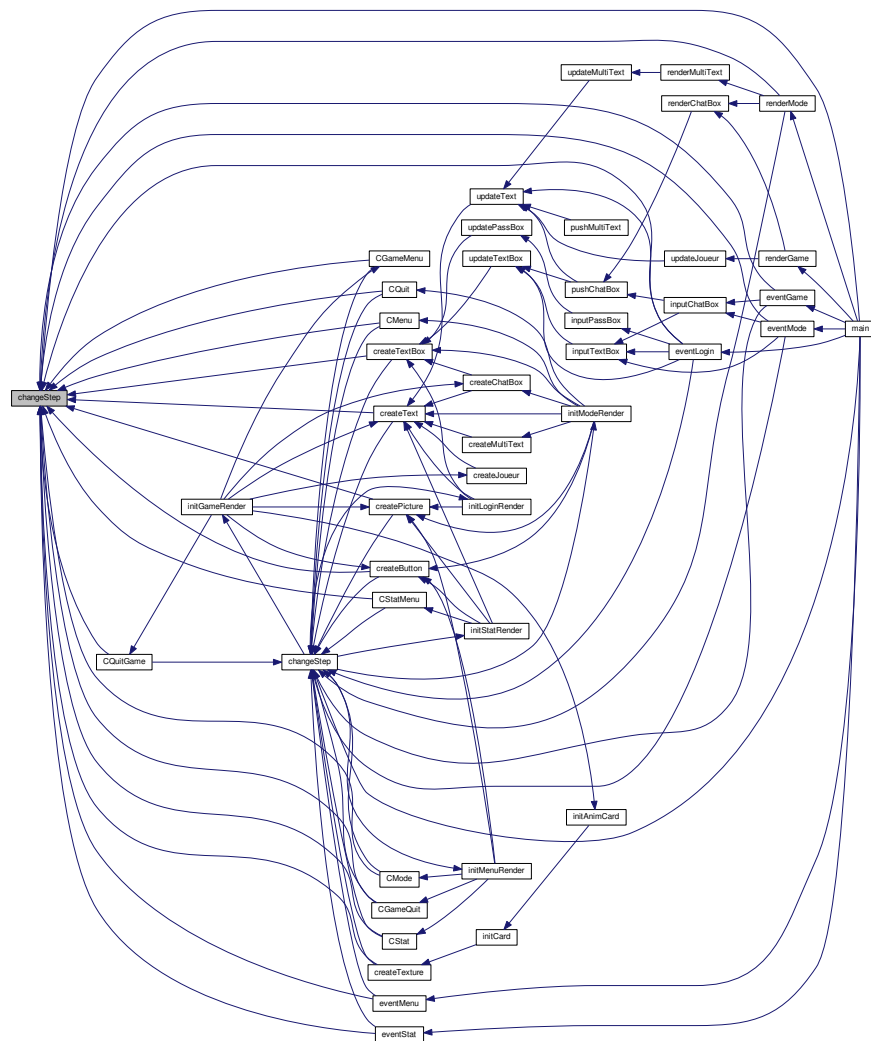
Parameters

<i>nextStep</i>	étape principale suivante
-----------------	---------------------------

Definition at line 6 of file inc.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.6.1.2 bool collisionWithMouse (SDL_Rect arg0, int mx, int my)

gère les collision de la souris avec un rectangle peut s'appliquer à tout point

Parameters

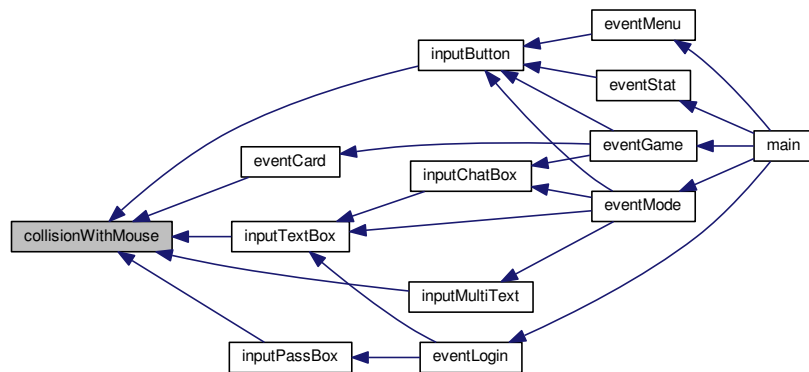
<i>arg0</i>	hitbox du rectangle
<i>mx</i>	position x de la souris (ou du point)
<i>my</i>	position y de la souris (ou du point)

Returns

true s'il y a collision, false sinon

Definition at line 56 of file inc.c.

Here is the caller graph for this function:



4.7 src/Game/inc.h File Reference

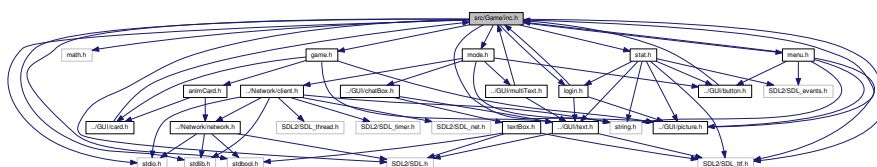
macro et function utiles partout

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include <stdbool.h>
#include <SDL2/SDL.h>
#include <SDL2/SDL_ttf.h>
#include "login.h"
#include "menu.h"
#include "game.h"
#include "mode.h"
#include "stat.h"

```

Include dependency graph for inc.h:



4.7.2 Macro Definition Documentation

4.7.2.1 `#define BACKGROUNDPATH "/media/background.bmp"`

Definition at line 26 of file inc.h.

4.7.2.2 `#define CLIGNE 500`

Definition at line 32 of file inc.h.

4.7.2.3 `#define FILELOGIN "/media/login.txt"`

Definition at line 25 of file inc.h.

4.7.2.4 `#define FONTPATH "/media/arial.ttf"`

Definition at line 30 of file inc.h.

4.7.2.5 `#define FONTSIZE 256`

Definition at line 31 of file inc.h.

4.7.2.6 `#define SCREEN_HEIGHT 480`

Definition at line 28 of file inc.h.

4.7.2.7 `#define SCREEN_WIDTH 800`

Definition at line 27 of file inc.h.

4.7.2.8 `#define SIZESTR 21`

Definition at line 29 of file inc.h.

4.7.3 Enumeration Type Documentation

4.7.3.1 `enum MainStep`

étapes principales du jeu elles correspondent à des fonctions de rendus ... particulières (cf [inc.c](#), [main.c](#))

Enumerator

- none*** état de base, pas de ressources propres
- login*** l'utilisateur est invité à s'identifier
- menu*** choix entre différentes étapes
- mode*** choix des paramètre essentiellement réseau
- stat*** affichage des statistiques du joueur
- game*** jeu
- end*** libération des ressources et fin de l'application

Definition at line 39 of file inc.h.

4.7.4 Function Documentation

4.7.4.1 void changeStep (MainStep *nextStep*)

permet de mettre à jour l'étape principale initialise et libere les ressources

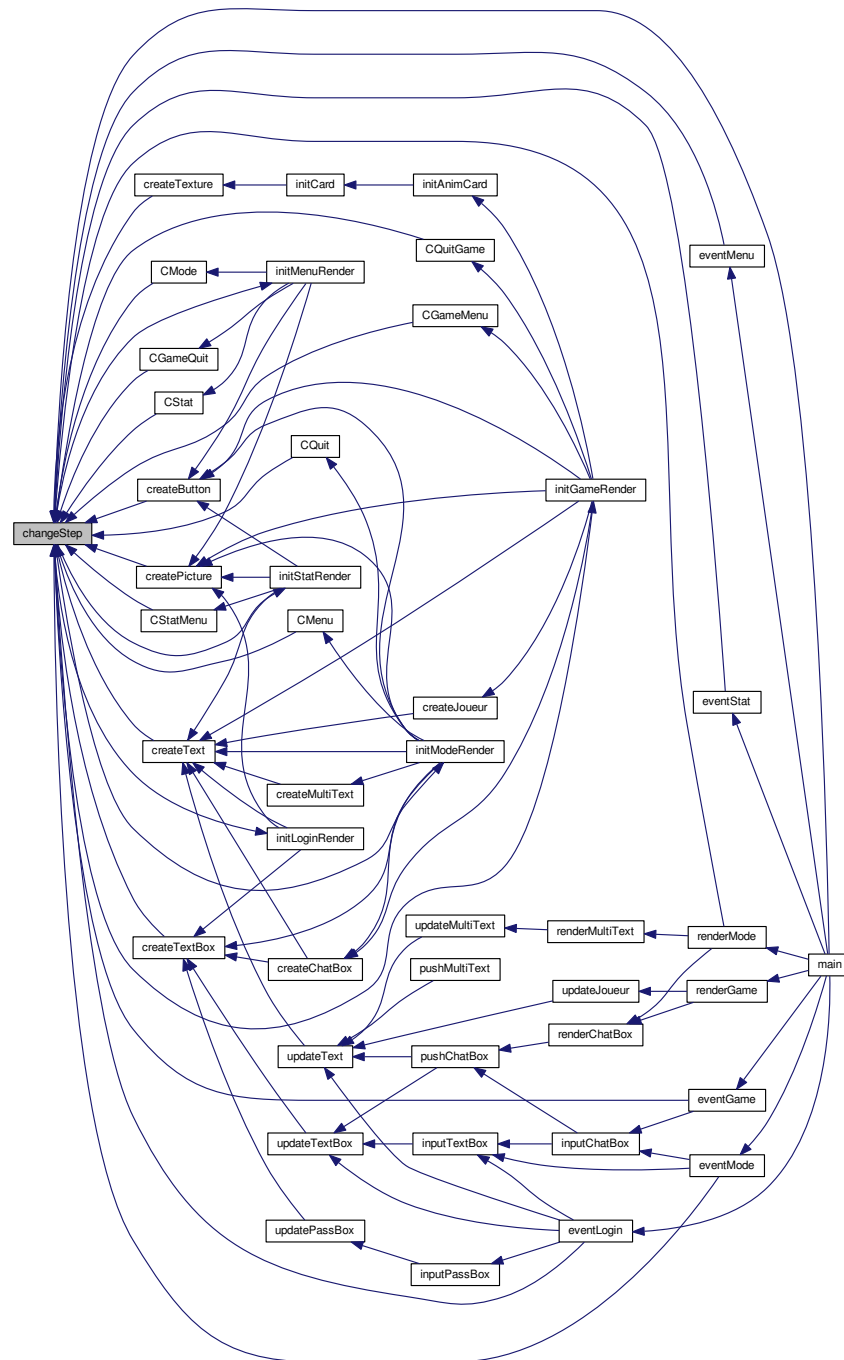
Parameters

<i>nextStep</i>	étape principale suivante
-----------------	---------------------------

Definition at line 6 of file inc.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.7.4.2 bool collisionWithMouse (SDL_Rect *arg0*, int *mx*, int *my*)

gère les collision de la souris avec un rectangle peut s'appliqué à tout point

Parameters

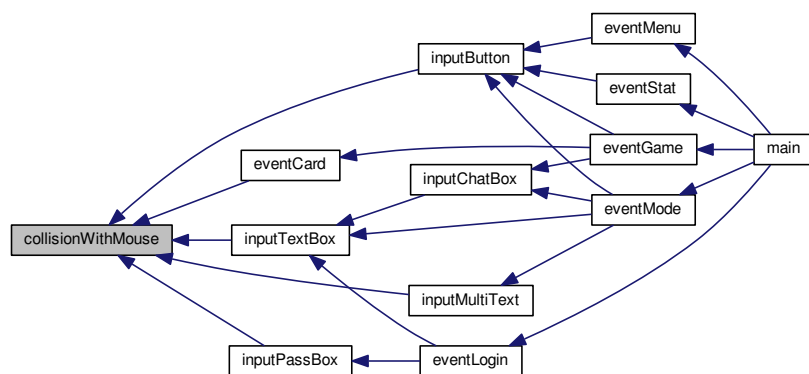
<i>arg0</i>	hitbox du rectangle
<i>mx</i>	position x de la souris (ou du point)
<i>my</i>	position y de la souris (ou du point)

Returns

true s'il y a collision, false sinon

Definition at line 56 of file inc.c.

Here is the caller graph for this function:



4.7.5 Variable Documentation

4.7.5.1 TTF_Font* font

Definition at line 54 of file inc.h.

4.7.5.2 MainStep globalStep

Definition at line 57 of file inc.h.

4.7.5.3 SDL_Renderer* renderer

Definition at line 56 of file inc.h.

4.7.5.4 SDL_Window* window

Definition at line 55 of file inc.h.

Here is the caller graph for this function:



4.8.2.2 void quit ()

libere les ressources 'globales'

Definition at line 48 of file init.c.

Here is the caller graph for this function:



4.8.3 Variable Documentation

4.8.3.1 TTF_Font* font

Definition at line 54 of file inc.h.

4.8.3.2 SDL_Renderer* renderer

Definition at line 56 of file inc.h.

4.8.3.3 SDL_Window* window

Definition at line 55 of file inc.h.

4.9 src/Game/init.h File Reference

Comprend L'initialisation de la SDL, et la liberation des ressources.

```
#include <SDL2/SDL_ttf.h>
#include "inc.h"
```


- void `renderLogin` ()
dessine ressources graphiques spécifique au rendu du login
- void `freeLoginRender` ()
Libere les ressources graphiques initialisees par initLoginRender.
- int `identifier` (char *username, char *password)
- bool `exist` (char *username)

Variables

- SDL_Renderer * `renderer`

4.10.1 Function Documentation

4.10.1.1 void eventLogin ()

Gestion des evenements spécifique au login.

Definition at line 21 of file login.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.10.1.2 bool exist (char * username)

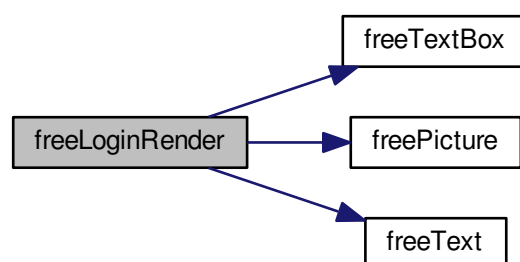
Definition at line 177 of file login.c.

4.10.1.3 void freeLoginRender ()

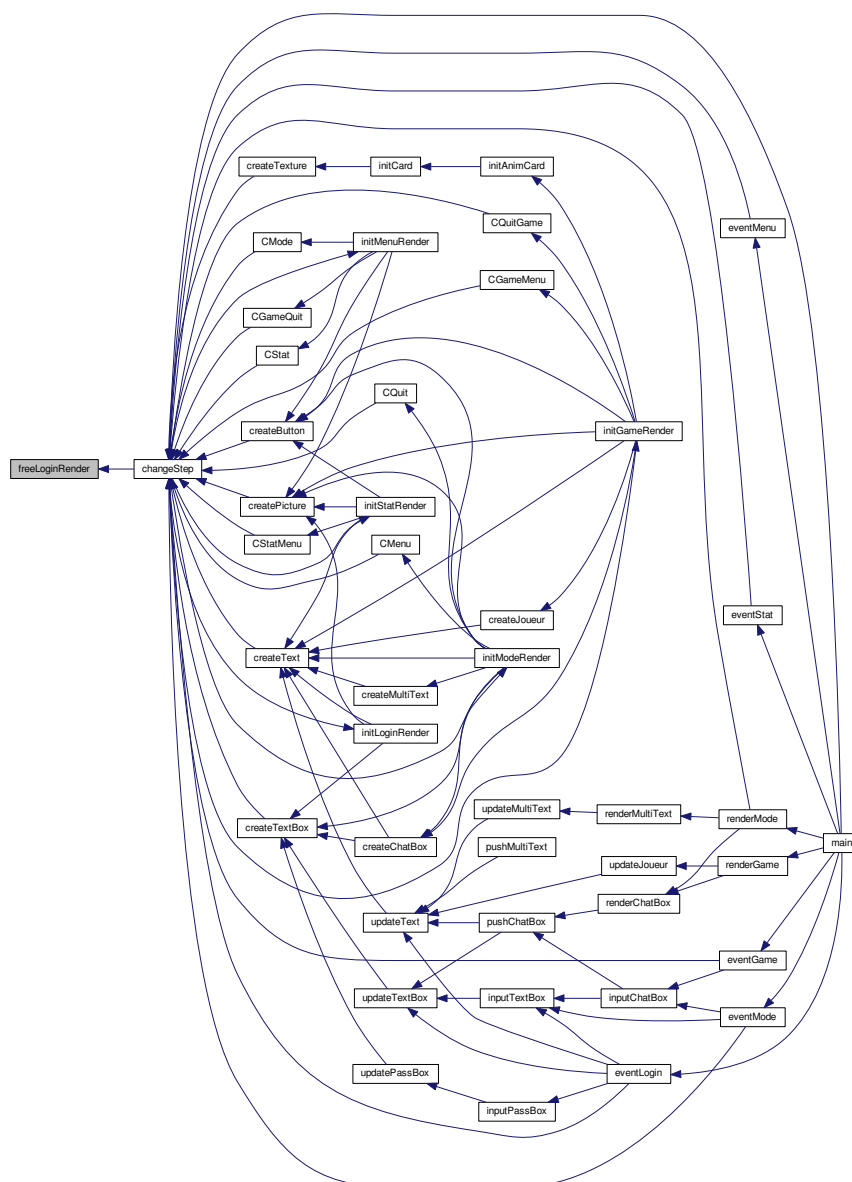
Libere les ressources graphiques initialisees par initLoginRender.

Definition at line 124 of file login.c.

Here is the call graph for this function:



Here is the caller graph for this function:



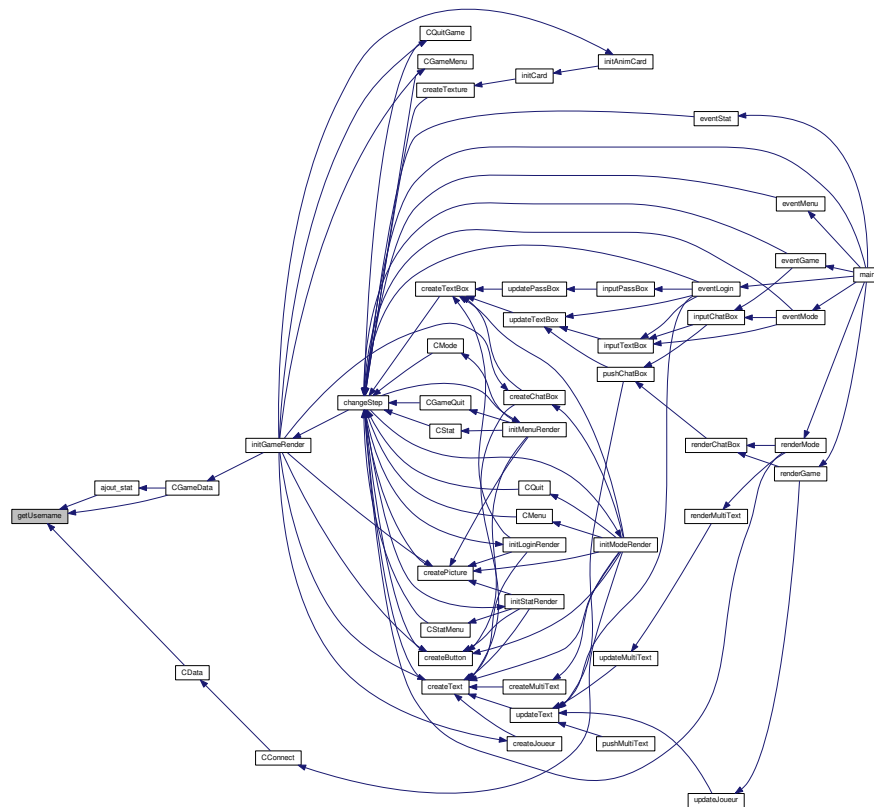
4.10.1.4 char * getUsername ()

Returns

nom de l'utilisateur, NULL si non identifié

Definition at line 10 of file login.c.

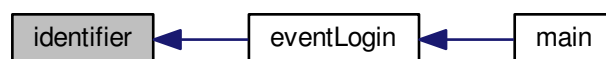
Here is the caller graph for this function:



4.10.1.5 int identifier (char * username, char * password)

Definition at line 141 of file login.c.

Here is the caller graph for this function:



4.10.1.6 void initLoginRender ()

Initialisation des ressources graphiques spécifique au rendu du login.

4.11.3 Function Documentation

4.11.3.1 void eventLogin ()

Gestion des evenements spécifique au login.

Definition at line 21 of file login.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.11.3.2 bool exist ()

Cherche dans le fichier si l'utilisateur à un compte ou pas.

Returns

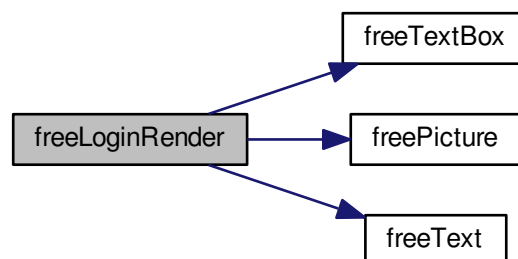
true si il est trouvé, false sinon

4.11.3.3 void freeLoginRender ()

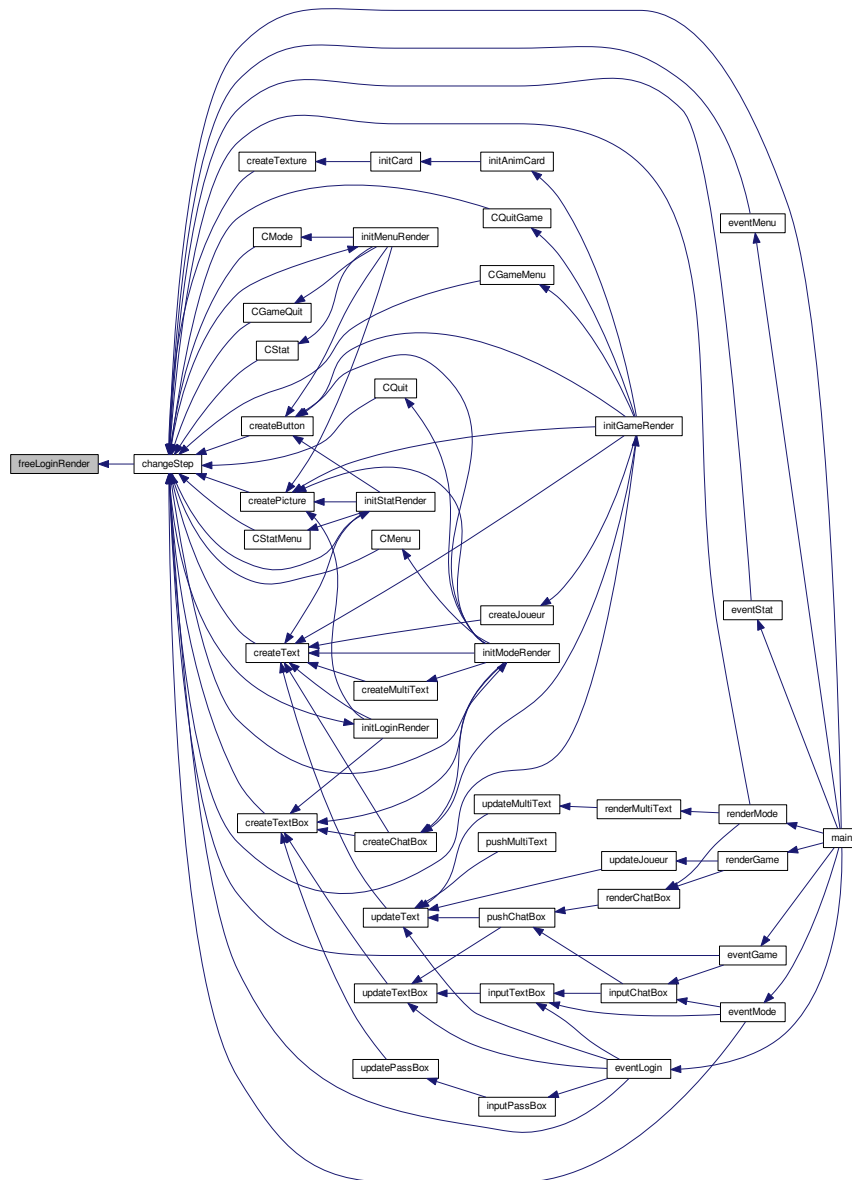
Libere les ressources graphiques initialisees par initLoginRender.

Definition at line 124 of file login.c.

Here is the call graph for this function:



Here is the caller graph for this function:



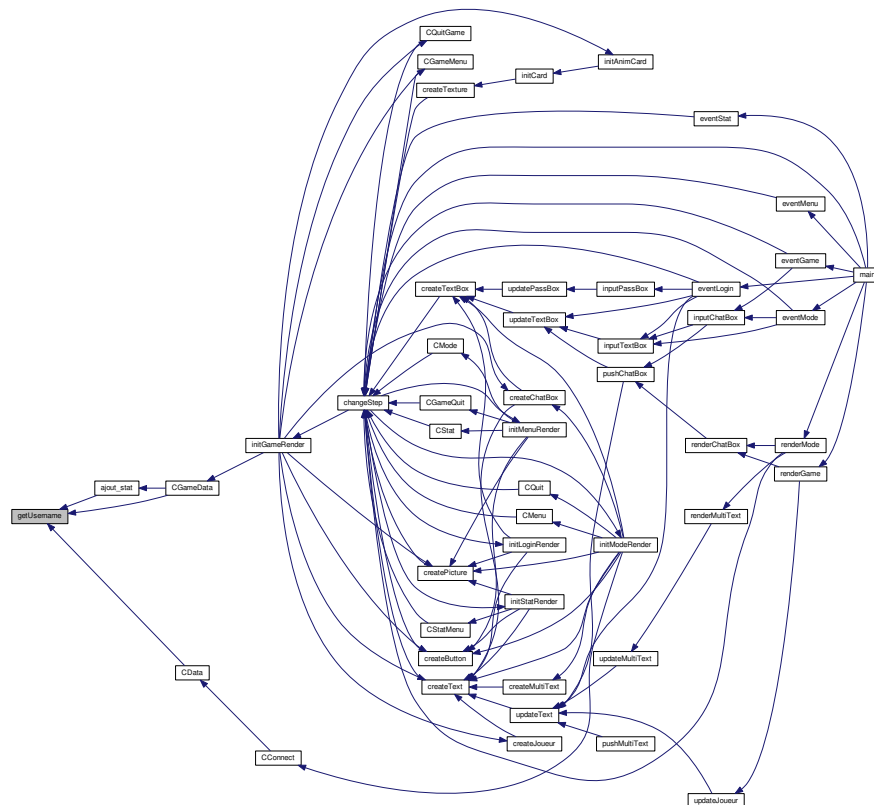
4.11.3.4 char* getUsername ()

Returns

nom de l'utilisateur, NULL si non identifié

Definition at line 10 of file login.c.

Here is the caller graph for this function:

**4.11.3.5 int identifier ()**

Cherche dans le fichier si l'utilisateur à un compte et si le mot de passe est juste.

Returns

0 si le mot de passe est faux, 1 si l'utilisateur est bien loguer, 2 si un nouveaux compte est créé

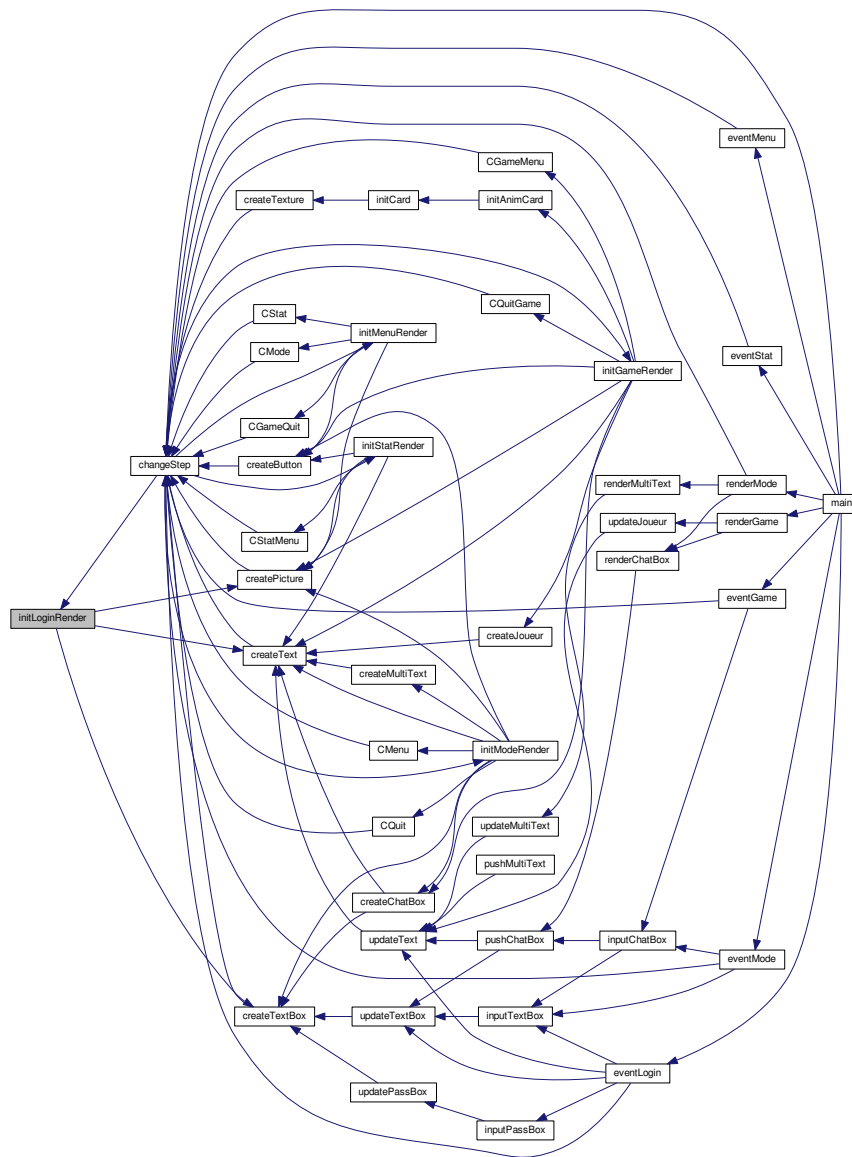
4.11.3.6 void initLoginRender ()

Initialisation des ressources graphiques spécifique au rendu du login.

Definition at line 98 of file login.c.

Here is the call graph for this function:

Here is the caller graph for this function:

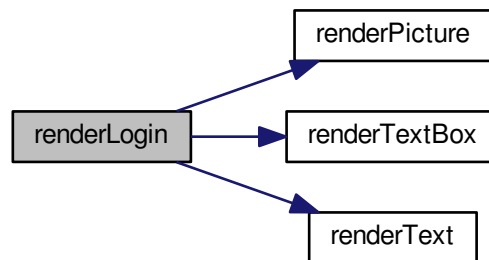


4.11.3.7 void renderLogin ()

dessine ressources graphiques spécifique au rendu du login

Definition at line 114 of file login.c.

Here is the call graph for this function:



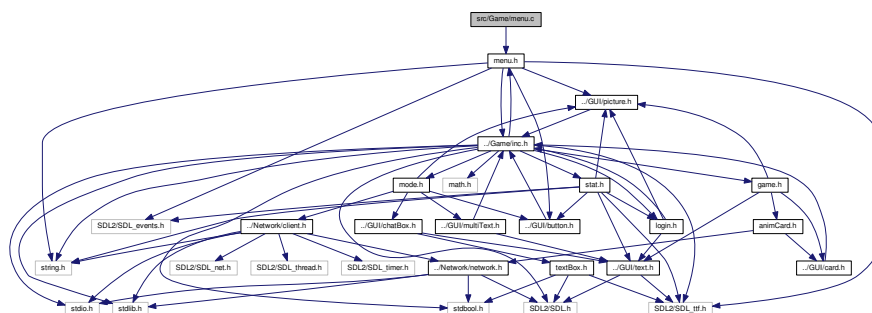
Here is the caller graph for this function:



4.12 src/Game/menu.c File Reference

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

Include dependency graph for menu.c:



Functions

- void `eventMenu` ()
Gestion des evenements spécifique au menu.
- void `CGameQuit` ()
- void `CStat` ()

- void **CMode** ()
- void **initMenuRender** ()
Initialisation des ressources graphiques spécifique au rendu du menu.
- void **renderMenu** ()
dessine ressources graphiques spécifique au rendu du menu
- void **freeMenuRender** ()
Libere les ressources graphiques initialisees par initMenuRender.

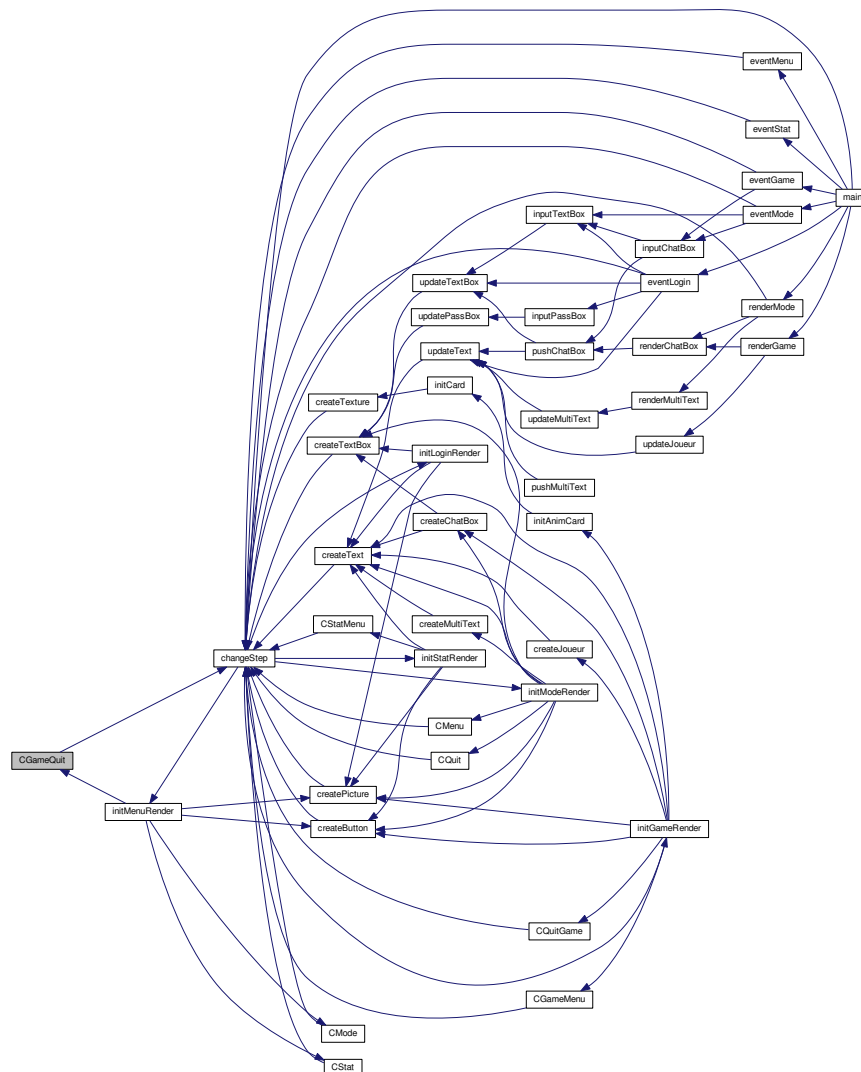
4.12.1 Function Documentation

4.12.1.1 void CGameQuit ()

Definition at line 33 of file menu.c.

Here is the call graph for this function:

Here is the caller graph for this function:

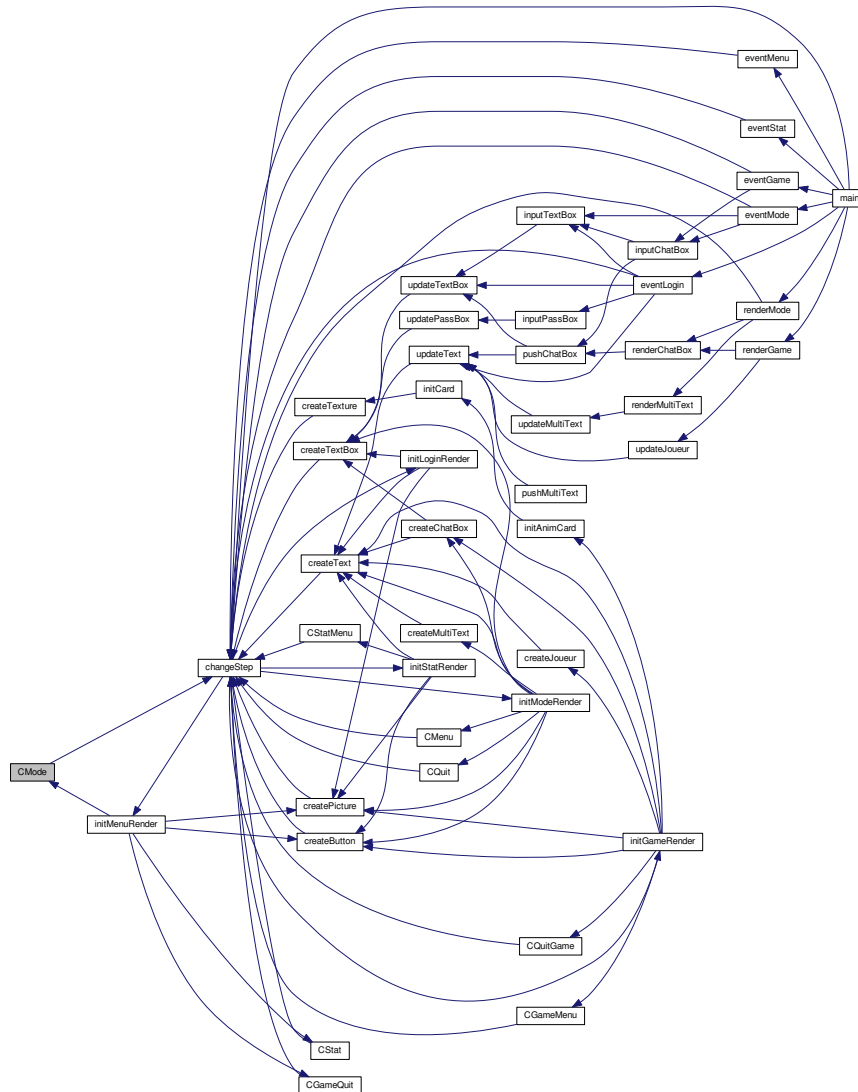


4.12.1.2 void CMode ()

Definition at line 41 of file menu.c.

Here is the call graph for this function:

Here is the caller graph for this function:

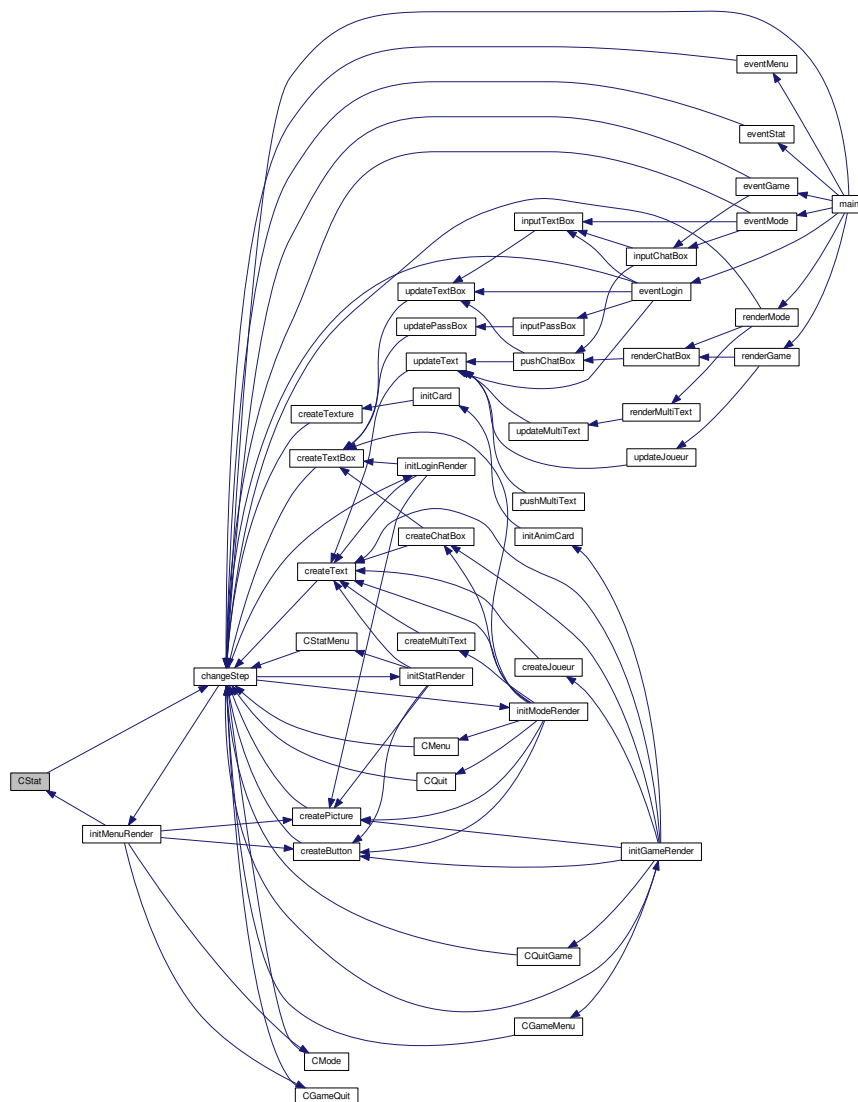


4.12.1.3 void CStat ()

Definition at line 37 of file menu.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.12.1.4 void eventMenu ()

Gestion des evenements spécifique au menu.

Definition at line 8 of file menu.c.

Here is the call graph for this function:

Here is the caller graph for this function:

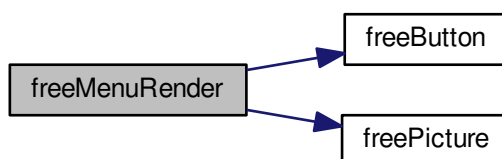


4.12.1.5 void freeMenuRender ()

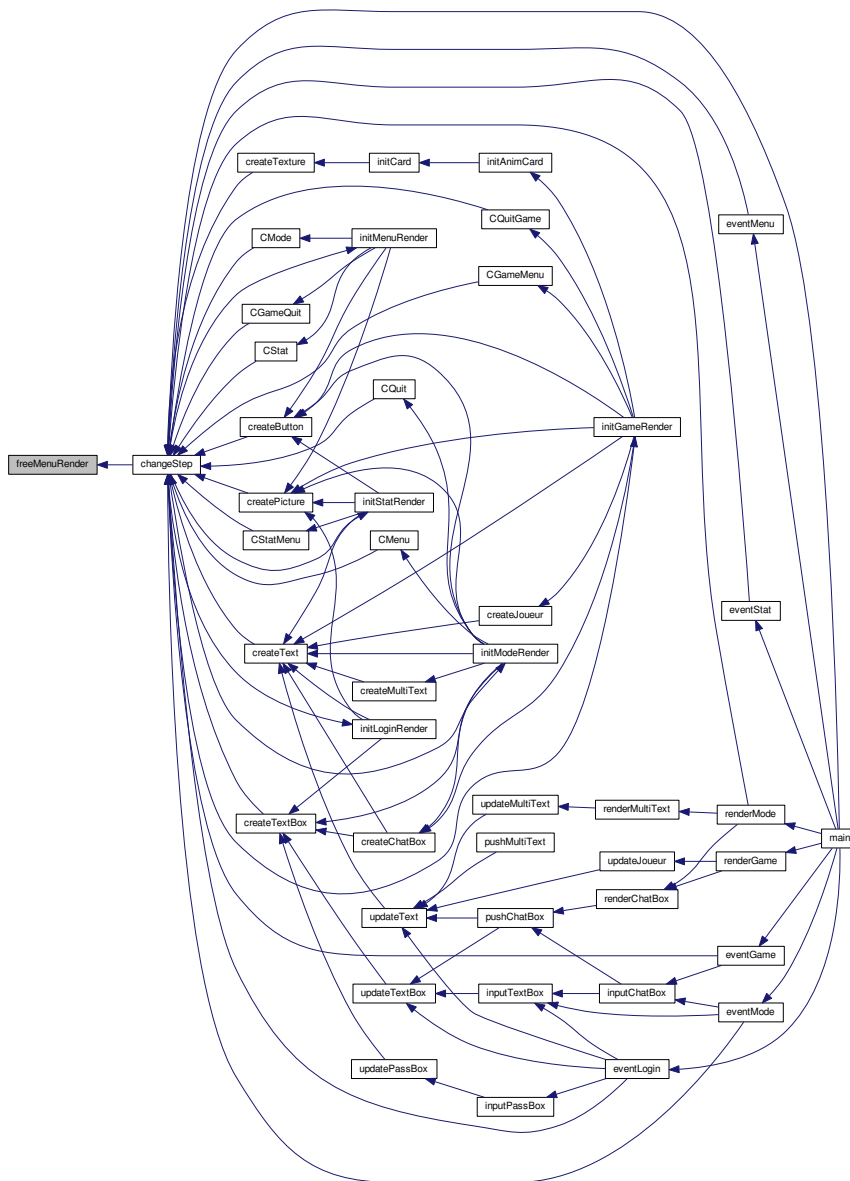
Libere les ressources graphiques initialisees par initMenuRender.

Definition at line 71 of file menu.c.

Here is the call graph for this function:



Here is the caller graph for this function:



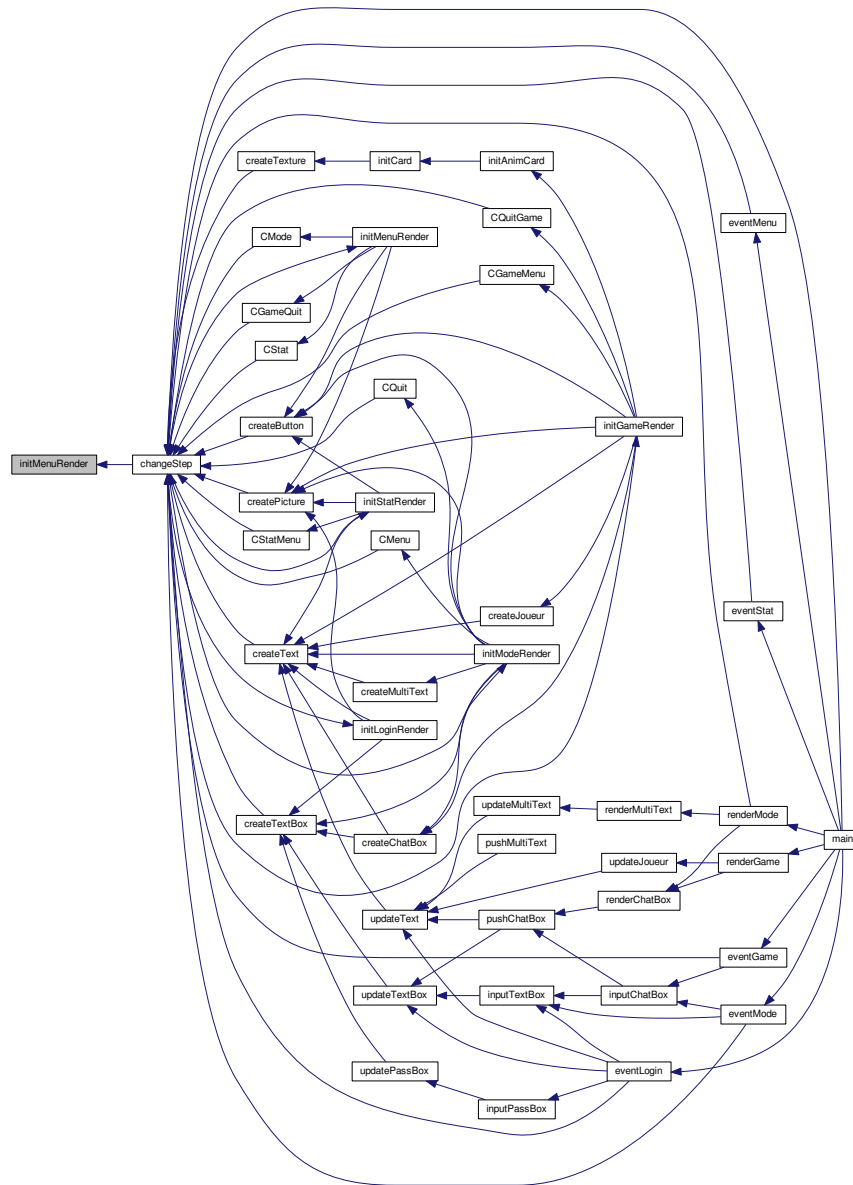
4.12.1.6 void initMenuRender ()

Initialisation des ressources graphiques spécifique au rendu du menu.

Definition at line 45 of file menu.c.

Here is the call graph for this function:

Here is the caller graph for this function:

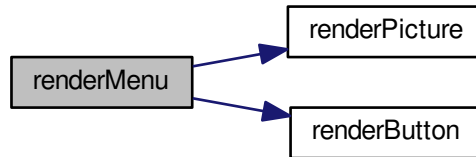


4.12.1.7 void renderMenu ()

dessine ressources graphiques spécifique au rendu du menu

Definition at line 61 of file menu.c.

Here is the call graph for this function:



Here is the caller graph for this function:

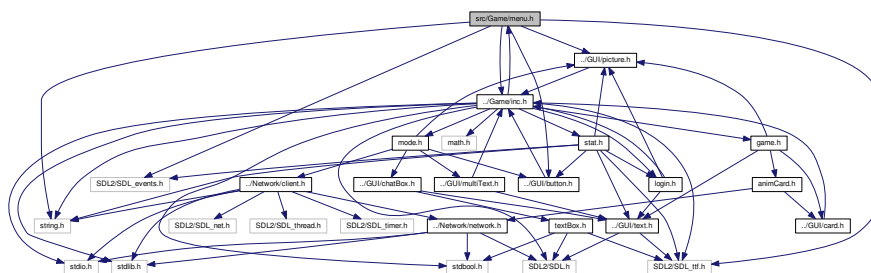


4.13 src/Game/menu.h File Reference

Fichier correspondant à l'affichage et la gestion du menu du jeu.

```
#include <SDL2/SDL_ttf.h>
#include <SDL2/SDL_events.h>
#include <string.h>
#include "../GUI/picture.h"
#include "../GUI/button.h"
#include "inc.h"
```

Include dependency graph for menu.h:

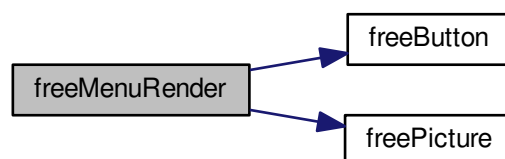


4.13.2.2 void freeMenuRender ()

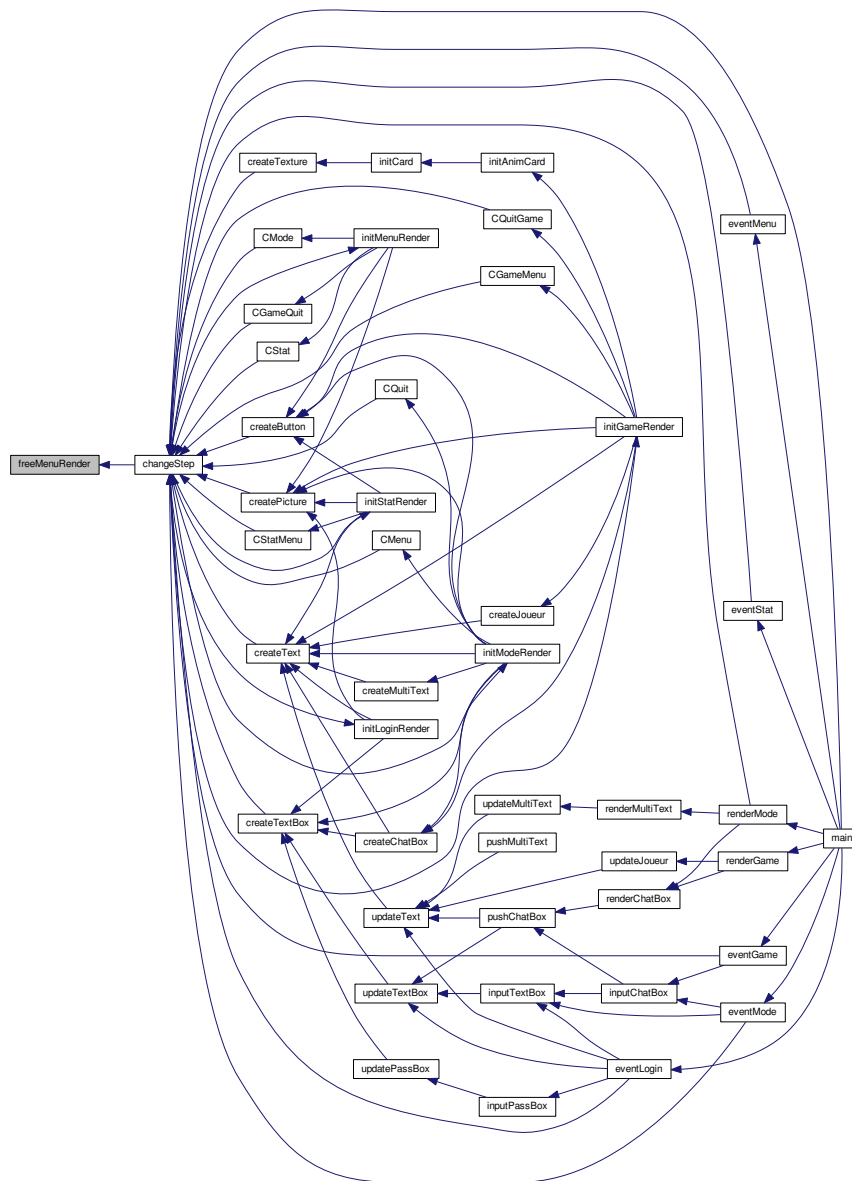
Libere les ressources graphiques initialisees par initMenuRender.

Definition at line 71 of file menu.c.

Here is the call graph for this function:



Here is the caller graph for this function:



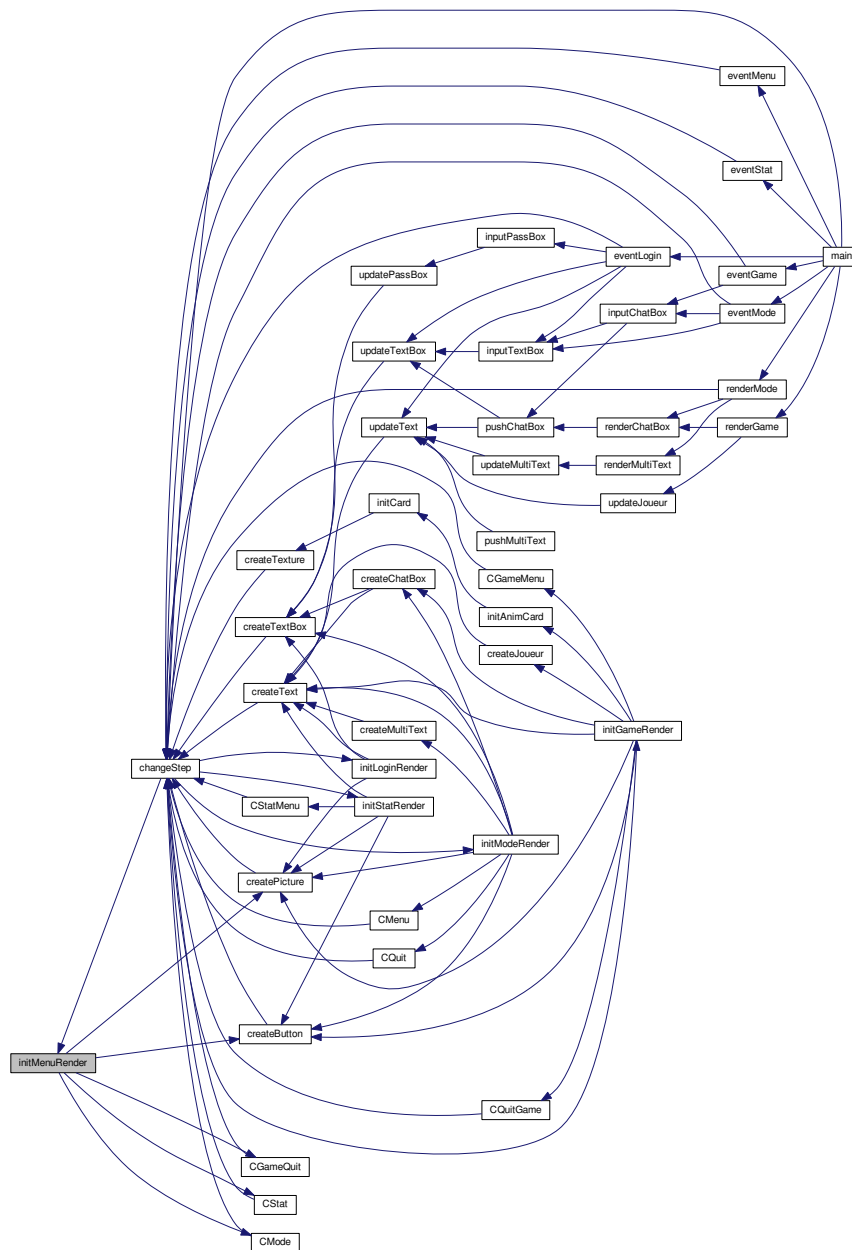
4.13.2.3 void initMenuRender ()

Initialisation des ressources graphiques spécifique au rendu du menu.

Definition at line 45 of file menu.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.13.2.4 void renderMenu ()

dessine ressources graphiques spécifique au rendu du menu

Definition at line 61 of file menu.c.

- void `CData` (`Data *data`)
- void `CDtList` (`DataList *data`)
- void `CDtGame` (`DataGame *data`)
- void `CQuit` ()
- void `CMenu` ()
- void `CGame` ()
- void `CUpList` ()
- void `CConnect` ()
- void `CMsg` (`char *msg`)
- void `CList` (`char *choix`)
- void `CCreate` ()
- void `CCIA` ()
- void `CHalt` ()
- void `initModeRender` ()
Initialisation des ressources graphiques spécifique à la page de parametre.
- void `renderMode` ()
dessine ressources graphiques spécifique à la page de parametre
- void `freeModeRender` ()
Libere les ressources graphiques initialisees par initModeRender.

Variables

- enum { ... } `modeStep`

4.14.1 Enumeration Type Documentation

4.14.1.1 anonymous enum

Enumerator

GETIP
CONNECT
SALLE
READY
GO

Definition at line 34 of file mode.c.

4.14.2 Function Documentation

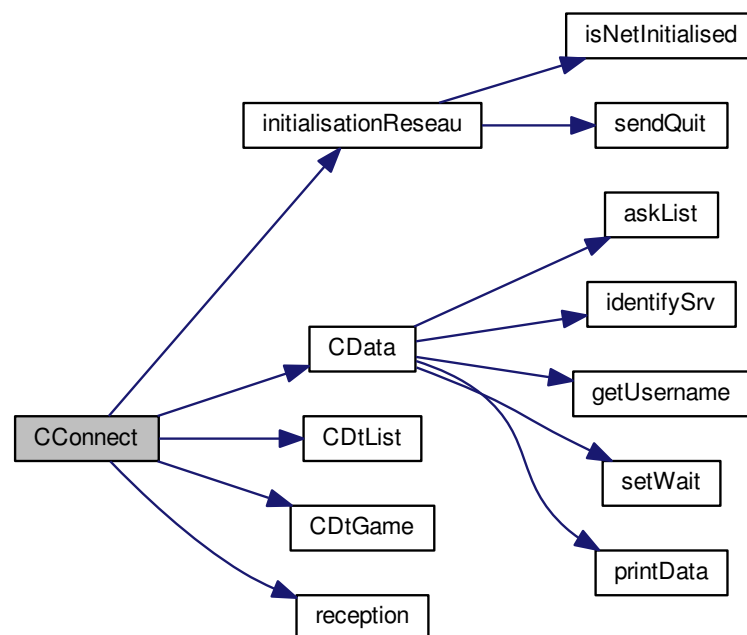
4.14.2.1 void CCIA ()

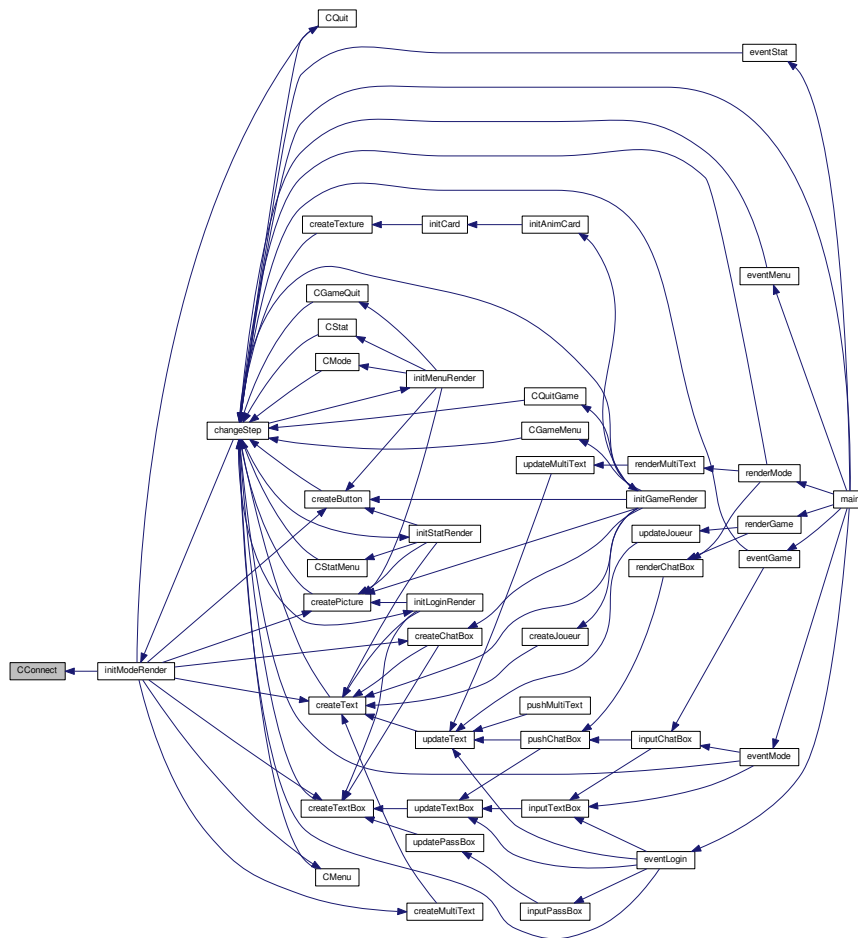
Definition at line 174 of file mode.c.

Here is the call graph for this function:



Here is the call graph for this function:

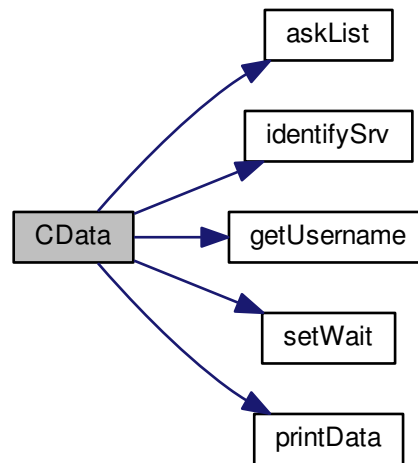




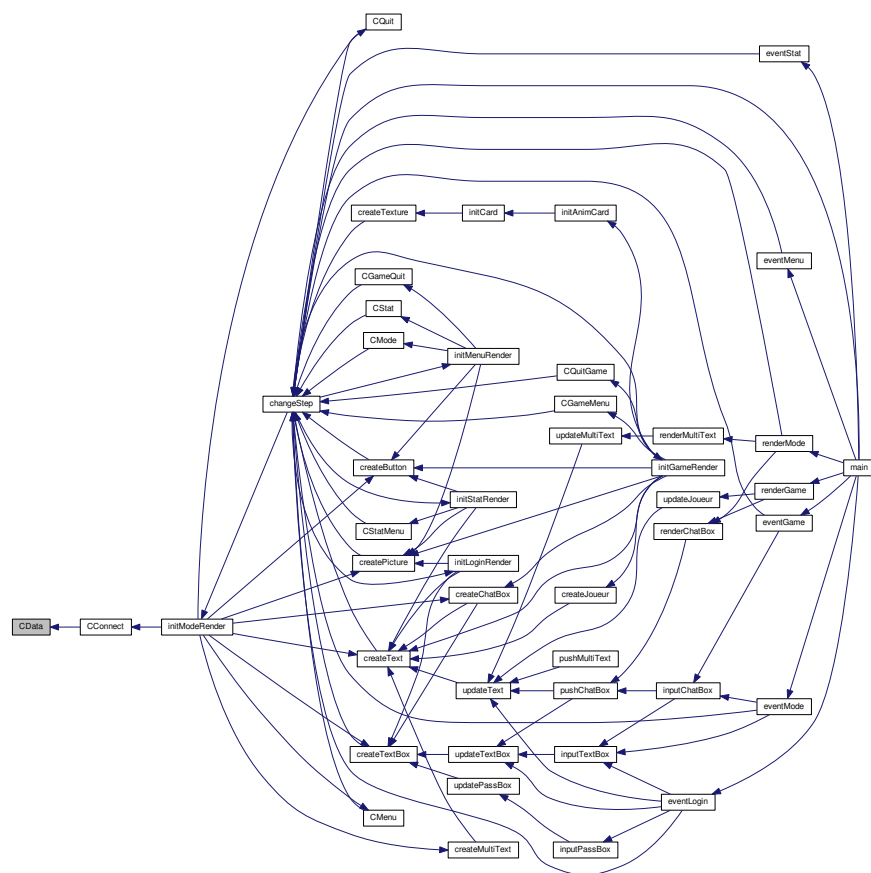
Here is the call graph for this function



Here is the call graph for this function:

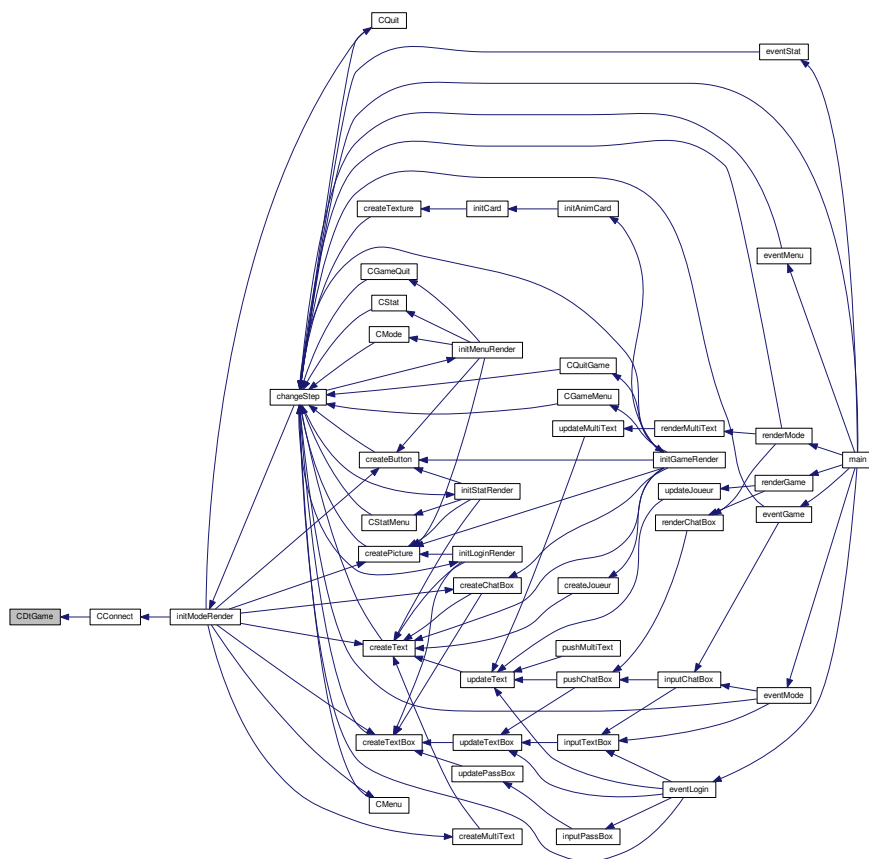


Here is the caller graph for this function:



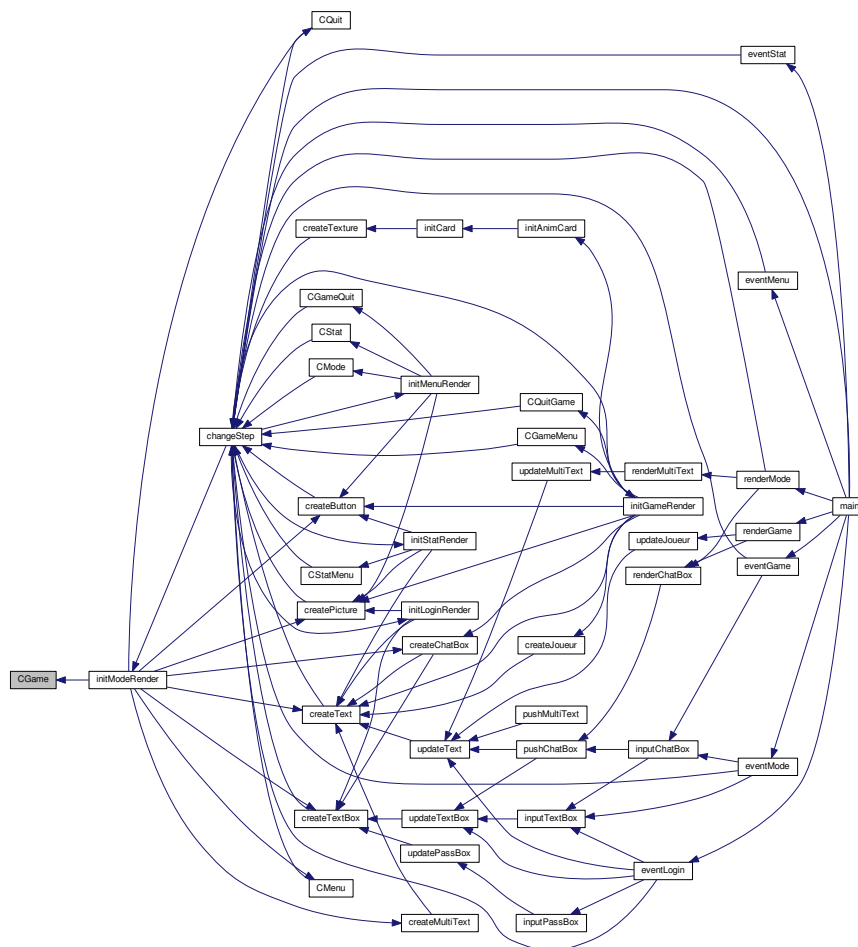
Definition at line 126 of file mode.c.

Here is the caller graph for this function:



Definition at line 102 of file mode.c.

Here is the caller graph for this function:



4.14.2.8 void CHalt ()

Definition at line 180 of file mode.c.

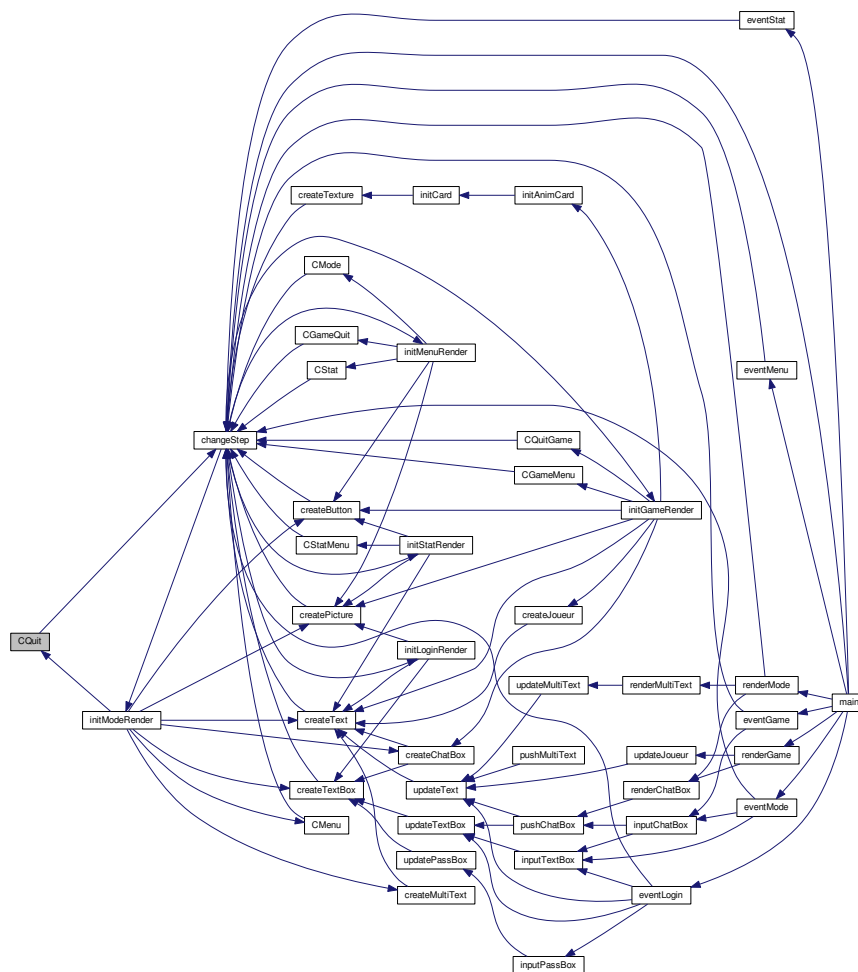
Here is the call graph for this function:

Here is the caller graph for this function:

4.14.2.9 void CList (char * *choix*)

Definition at line 161 of file mode.c.

Here is the caller graph for this function:



4.14.2.13 void CUpList ()

Definition at line 144 of file mode.c.

Here is the call graph for this function:

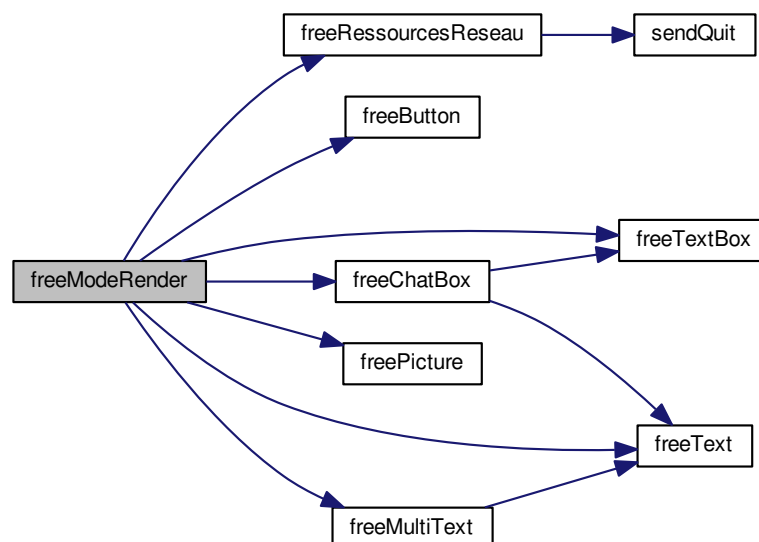


4.14.2.15 void freeModeRender ()

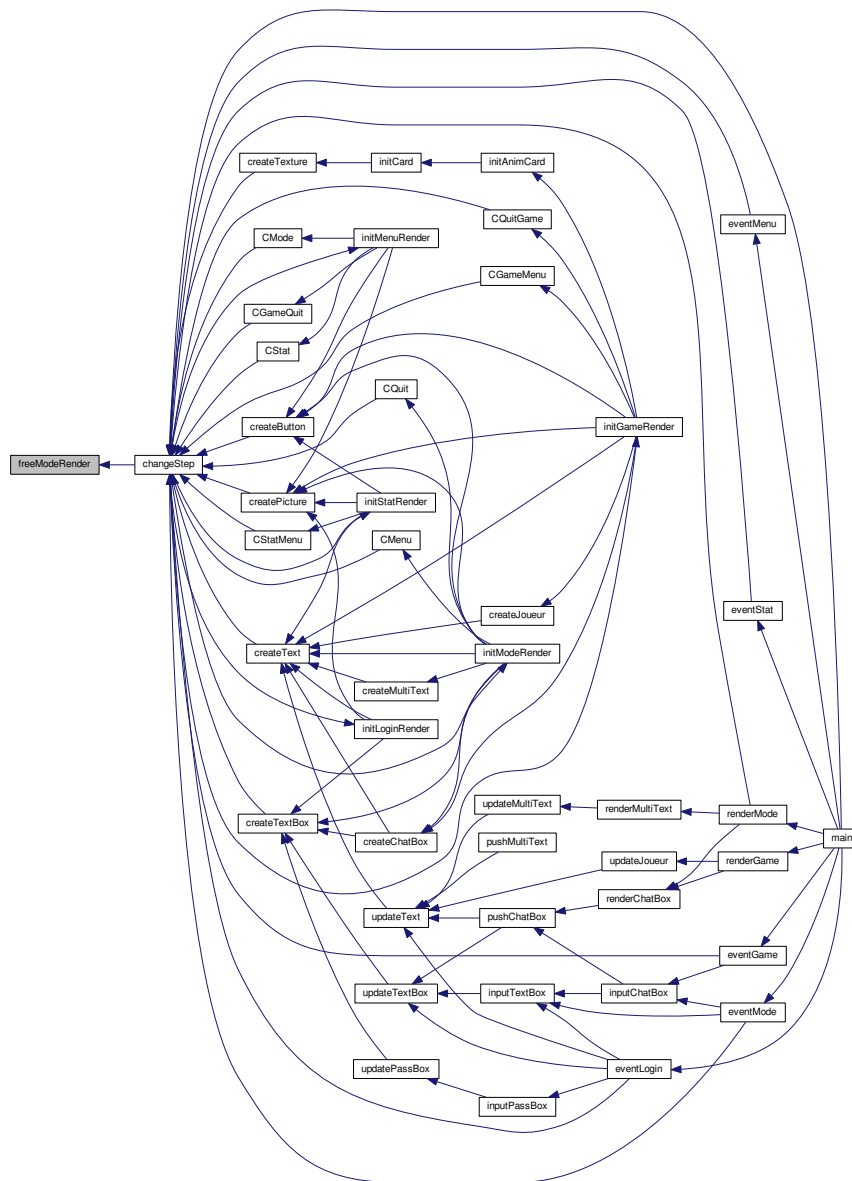
Libere les ressources graphiques initialisees par initModeRender.

Definition at line 261 of file mode.c.

Here is the call graph for this function:



Here is the caller graph for this function:



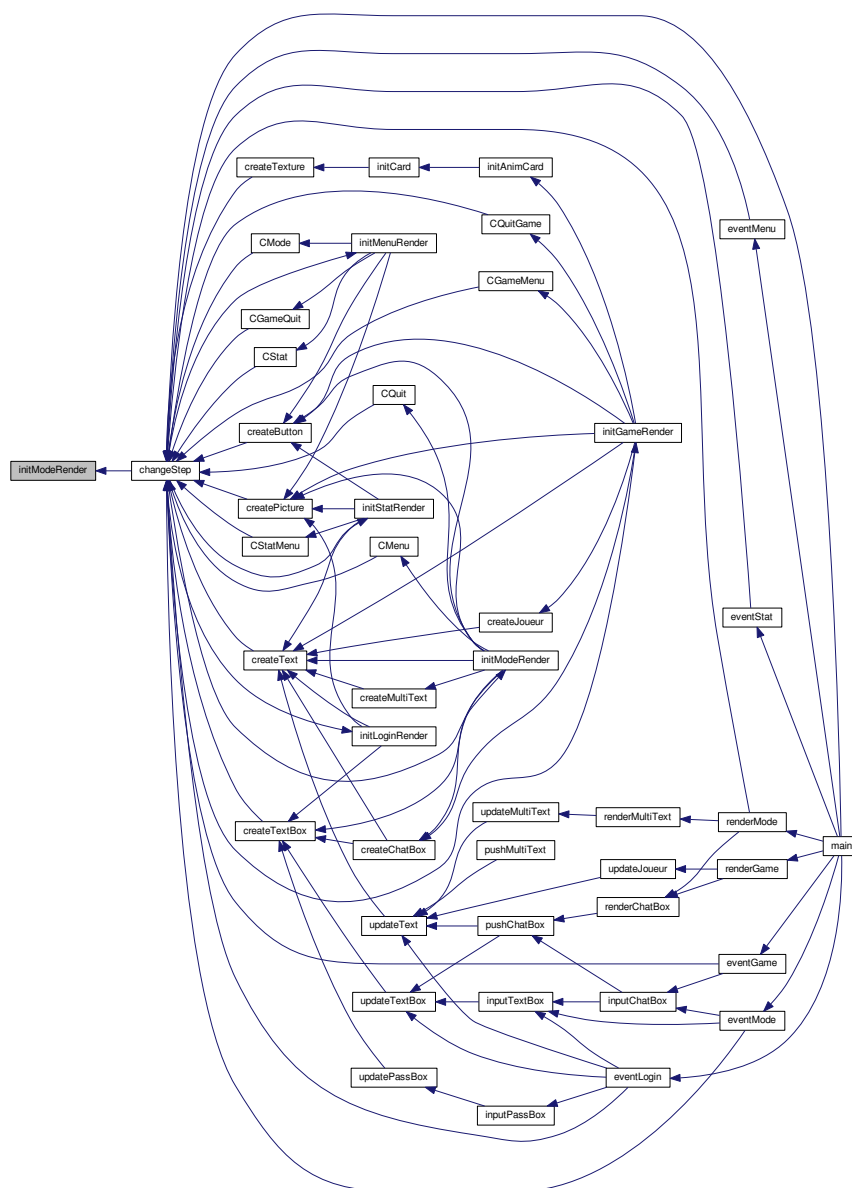
4.14.2.16 void initModeRender ()

Initialisation des ressources graphiques spécifique à la page de parametre.

Definition at line 185 of file mode.c.

Here is the call graph for this function:

Here is the caller graph for this function:



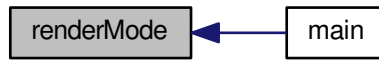
4.14.2.17 void renderMode ()

dessine ressources graphiques spécifique à la page de parametre

Definition at line 226 of file mode.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.14.3 Variable Documentation

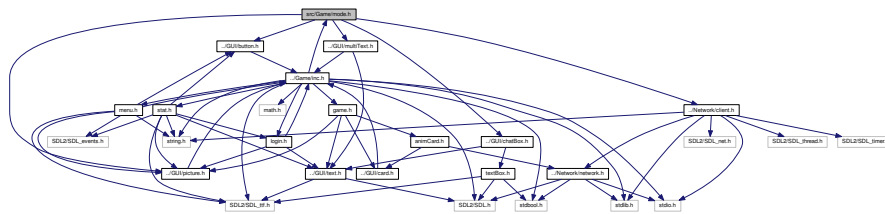
4.14.3.1 enum { ... } modeStep

4.15 src/Game/mode.h File Reference

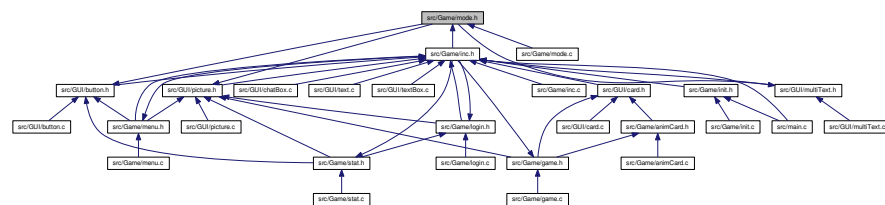
Fichier correspondant à l'affichage et la gestion de la page de parametre, avant l'accès au jeu.

```
#include "../GUI/button.h"
#include "../GUI/picture.h"
#include "../GUI/chatBox.h"
#include "../Network/client.h"
#include "../GUI/multiText.h"
```

Include dependency graph for mode.h:



This graph shows which files directly or indirectly include this file:



Functions

- void [eventMode](#) ()
Gestion des evenements spécifique à la page de parametre.
- void [initModeRender](#) ()
Initialisation des ressources graphiques spécifique à la page de parametre.

- void `renderMode` ()

dessine ressources graphiques spécifique à la page de parametre

- void `freeModeRender` ()

Libere les ressources graphiques initialisees par initModeRender.

4.15.1 Detailed Description

Fichier correspondant à l'affichage et la gestion de la page de parametre, avant l'accès au jeu.

Author

Aubin Detrez

4.15.2 Function Documentation

4.15.2.1 void `eventMode` ()

Gestion des evenements spécifique à la page de parametre.

Definition at line 36 of file mode.c.

Here is the call graph for this function:

Here is the caller graph for this function:

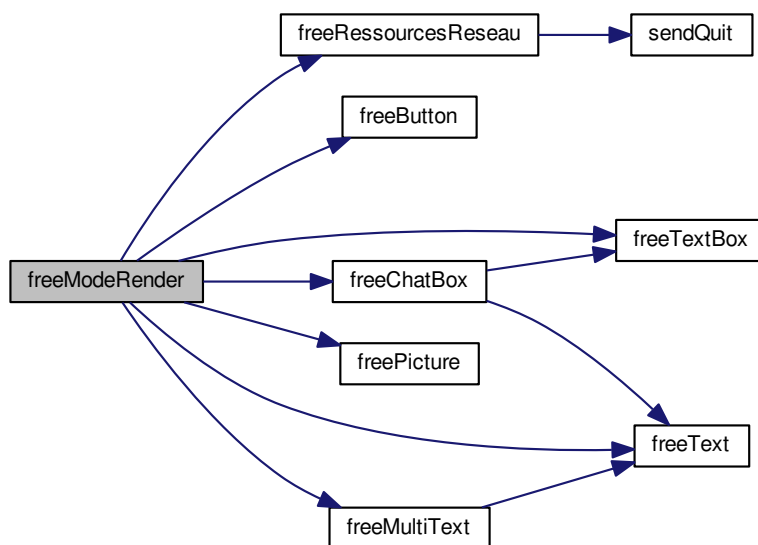


4.15.2.2 void `freeModeRender` ()

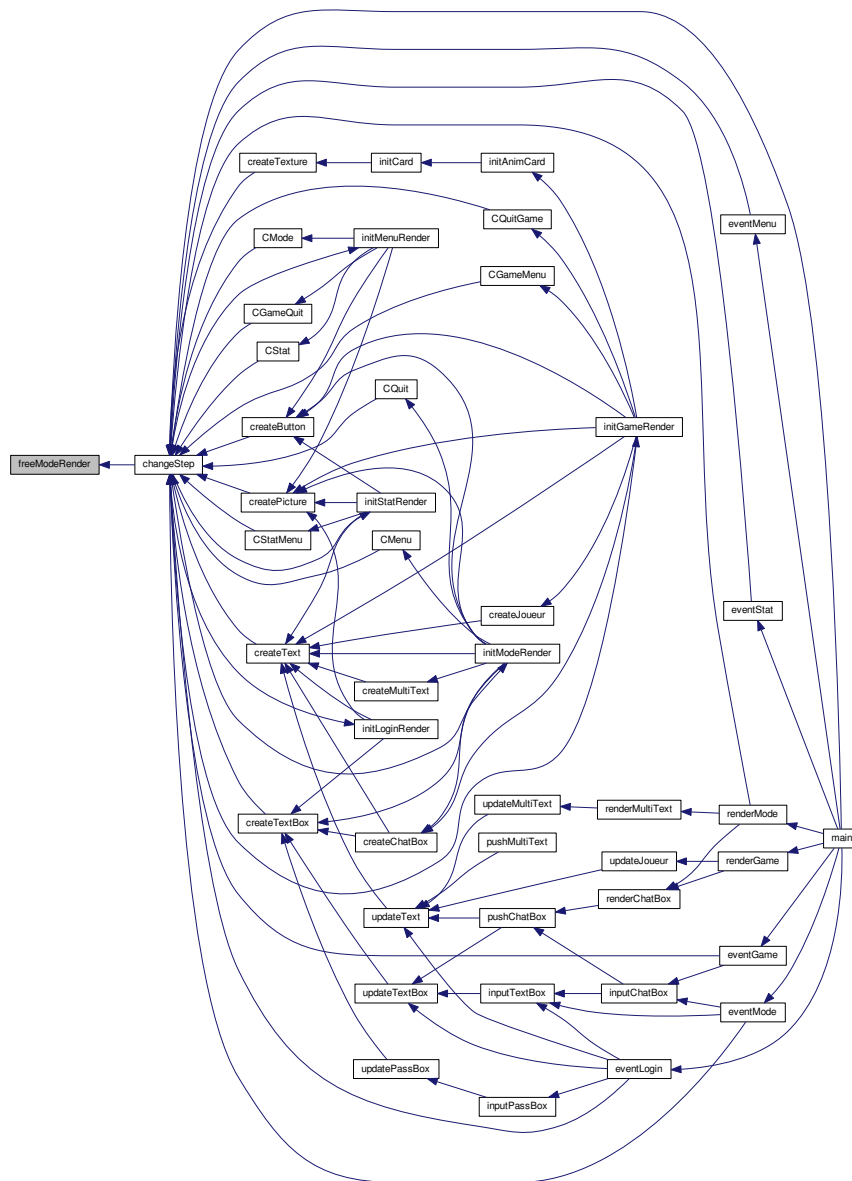
Libere les ressources graphiques initialisees par `initModeRender`.

Definition at line 261 of file mode.c.

Here is the call graph for this function:



Here is the caller graph for this function:



4.15.2.3 void initModeRender ()

Initialisation des ressources graphiques spécifique à la page de parametre.

Definition at line 185 of file mode.c.

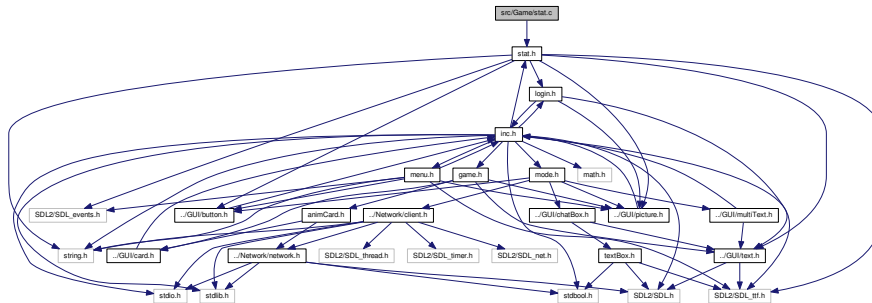
Here is the call graph for this function:

4.16 src/Game/stat.c File Reference

implémentation de l'affichage et gestions des statistiques

```
#include "stat.h"
```

Include dependency graph for stat.c:



Functions

- void [eventStat](#) ()
Gestion des evenements spécifique au statistiques.
- void [CStatMenu](#) ()
callback pour se diriger vers le menu appelé par le bouton par le bouton lors d'un clic
- void [initStatRender](#) ()
Initialisation des ressources graphiques spécifique au statistiques.
- void [renderStat](#) ()
dessine ressources graphiques spécifique au statistiques
- void [freeStatRender](#) ()
Libere les ressources graphiques initialisees par initStatRender.
- void [loadStatFromFile](#) ()
charge le tableau de statistique
- void [ajout_stat](#) (bool victoire)
ajouter une victoire ou une défaite selon la valeur du booléen

4.16.1 Detailed Description

implémentation de l'affichage et gestions des statistiques

Author

Aubin Detrez et Zongwei WU

4.16.2 Function Documentation

4.16.2.1 void ajout_stat (bool victoire)

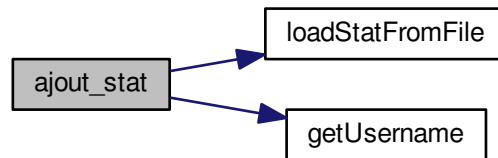
ajouter une victoire ou une défaite selon la valeur du booléen

Parameters

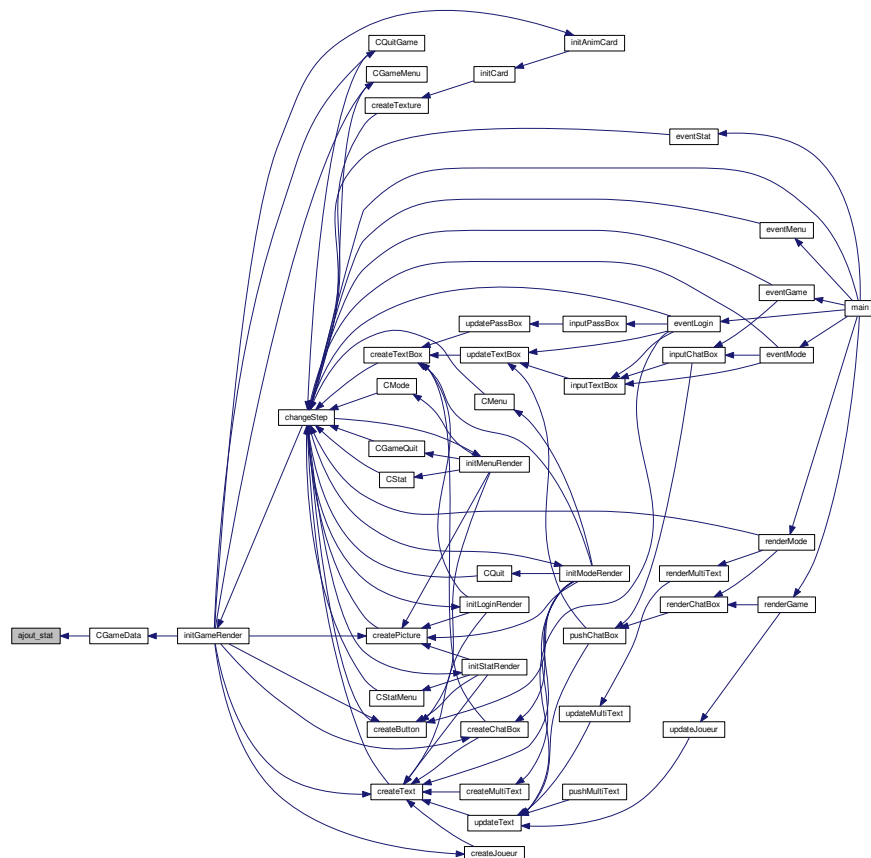
<i>victoire</i>	true, ajouter une victoire, false une défaite
-----------------	---

Definition at line 160 of file stat.c.

Here is the call graph for this function:



Here is the caller graph for this function:



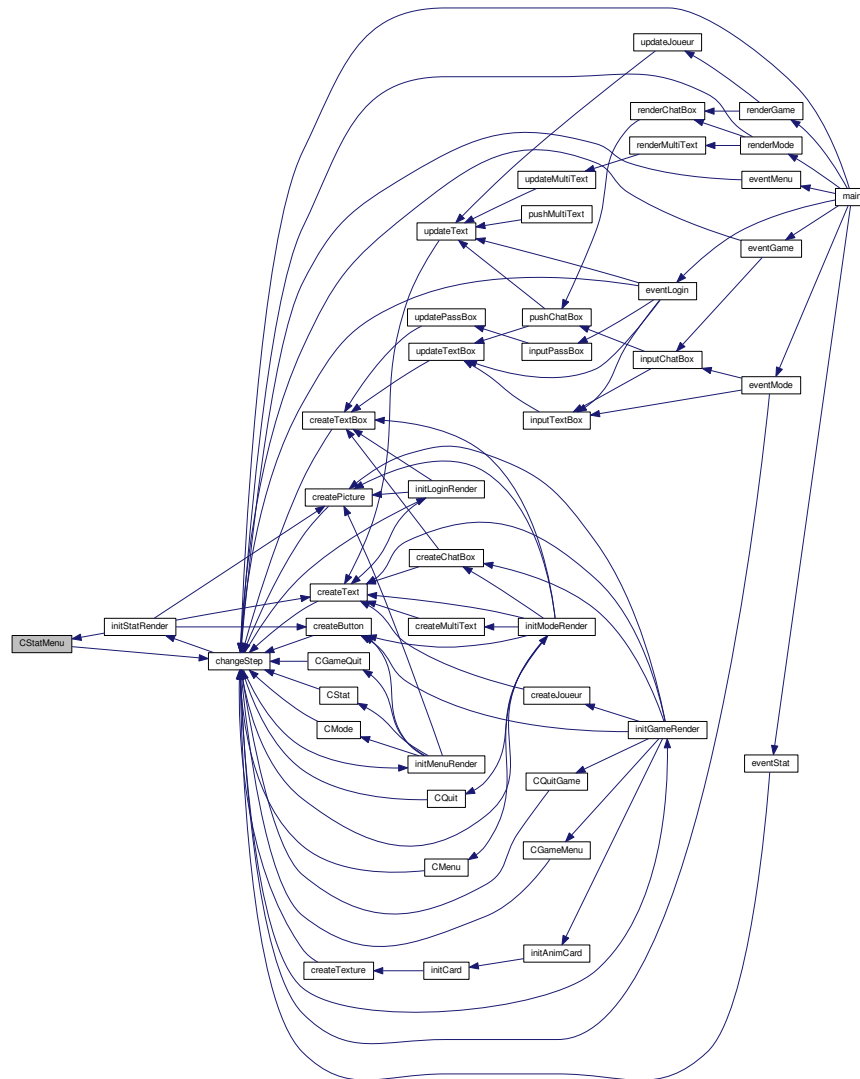
4.16.2.2 void CStatMenu ()

callback pour se diriger vers le menu appelé par le bouton lors d'un clic

Definition at line 51 of file stat.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.16.2.3 void eventStat ()

Gestion des evenements spécifique au statistiques.

Definition at line 28 of file stat.c.

Here is the call graph for this function:

Here is the caller graph for this function:

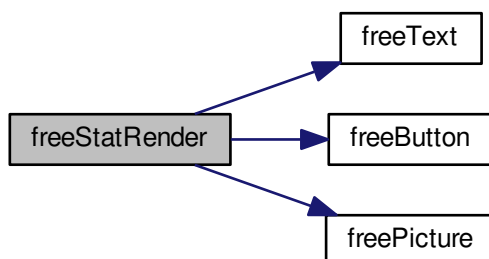


4.16.2.4 void freeStatRender ()

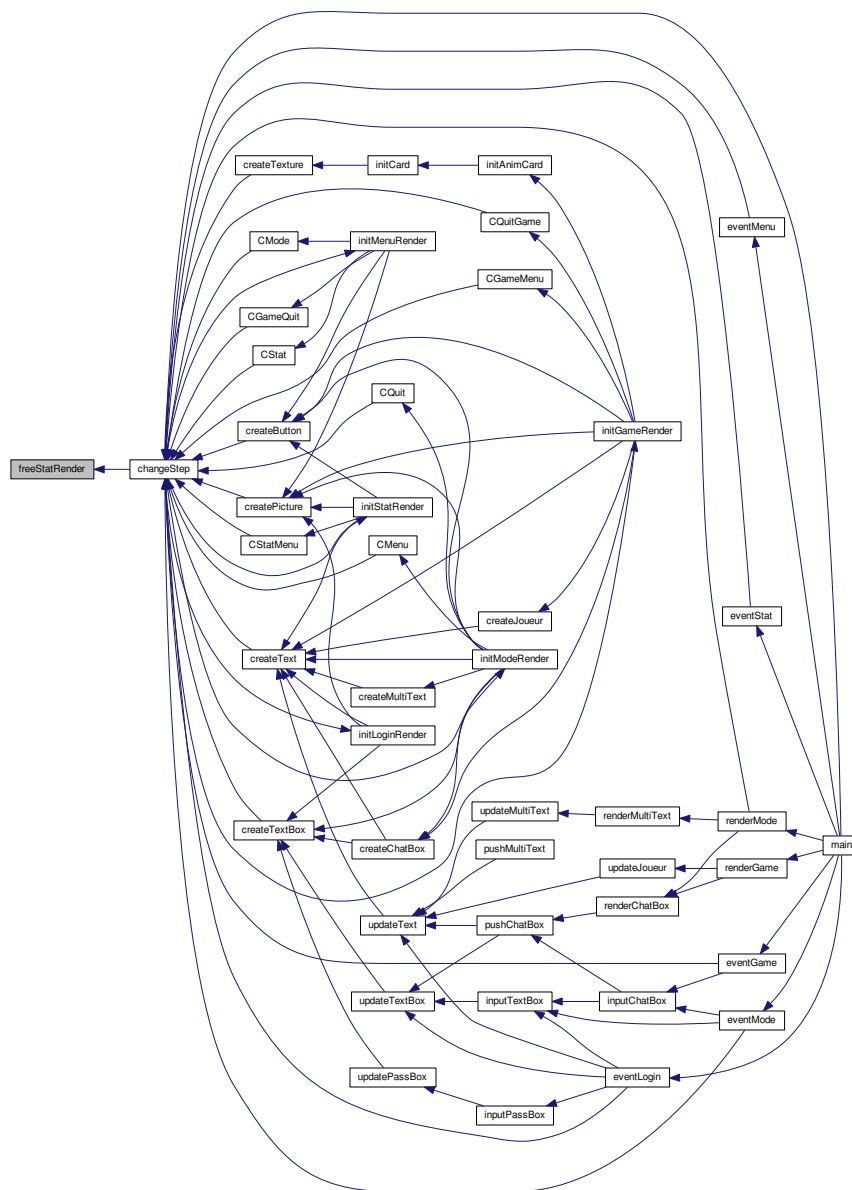
Libere les ressources graphiques initialisees par initStatRender.

Definition at line 114 of file stat.c.

Here is the call graph for this function:



Here is the caller graph for this function:



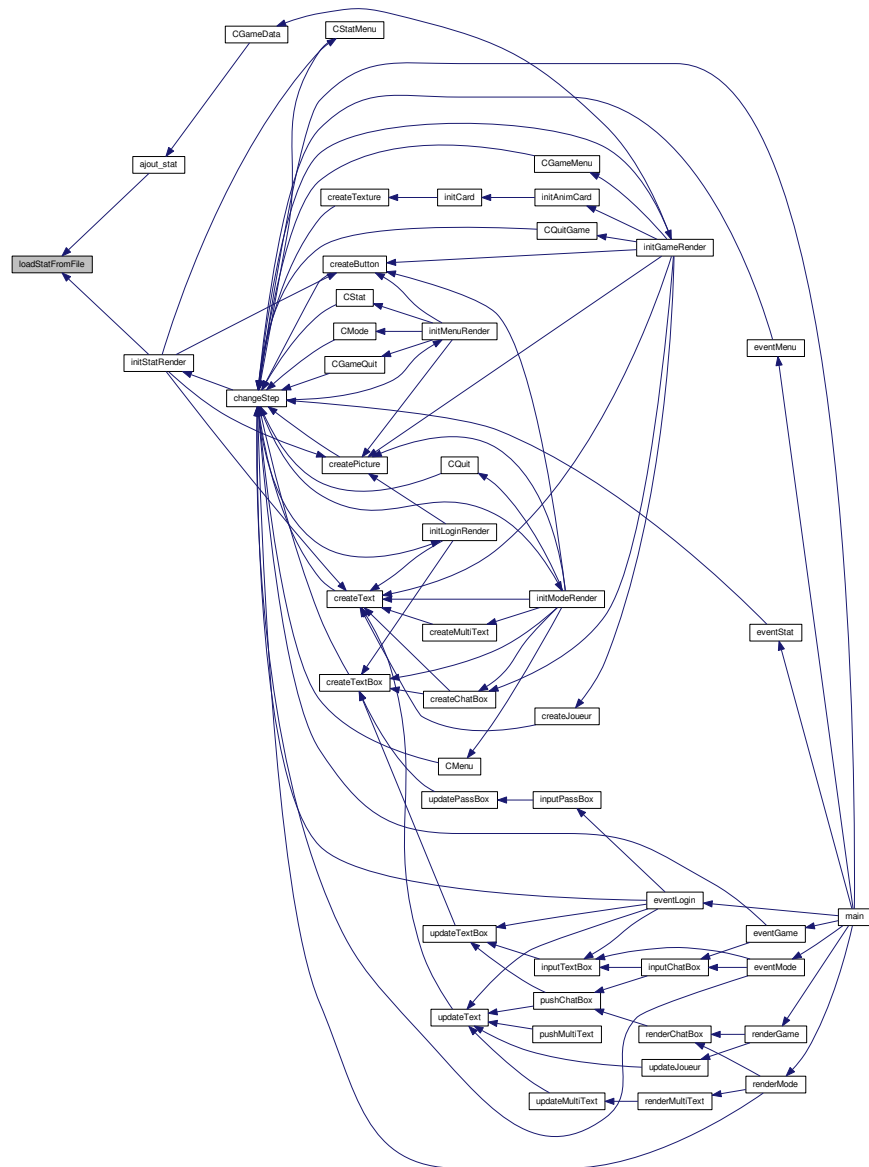
4.16.2.5 void initStatRender ()

Initialisation des ressources graphiques spécifique au statistiques.

Definition at line 55 of file stat.c.

Here is the call graph for this function:

Here is the caller graph for this function:

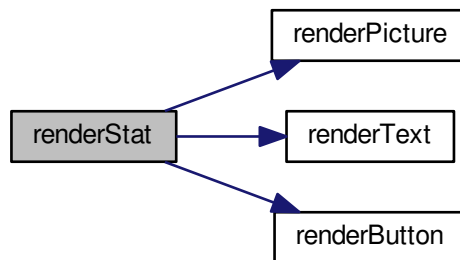


4.16.2.7 void renderStat ()

dessine ressources graphiques spécifique au statistiques

Definition at line 90 of file stat.c.

Here is the call graph for this function:



Here is the caller graph for this function:

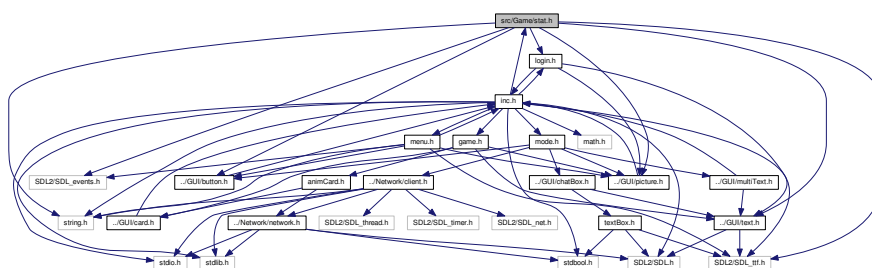


4.17 src/Game/stat.h File Reference

affichage et gestions des statistiques

```
#include <SDL2/SDL_ttf.h>
#include <SDL2/SDL_events.h>
#include <string.h>
#include "login.h"
#include "../GUI/button.h"
#include "../GUI/text.h"
#include "../GUI/picture.h"
```

Include dependency graph for stat.h:



4.17.2.2 #define MAXSTAT 100

Definition at line 19 of file stat.h.

4.17.3 Function Documentation

4.17.3.1 void ajout_stat (bool victoire)

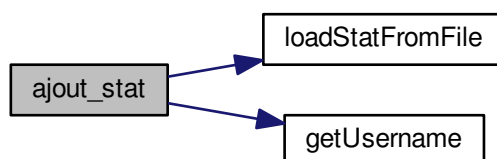
ajouter une victoire ou une défaite selon la valeur du booléen

Parameters

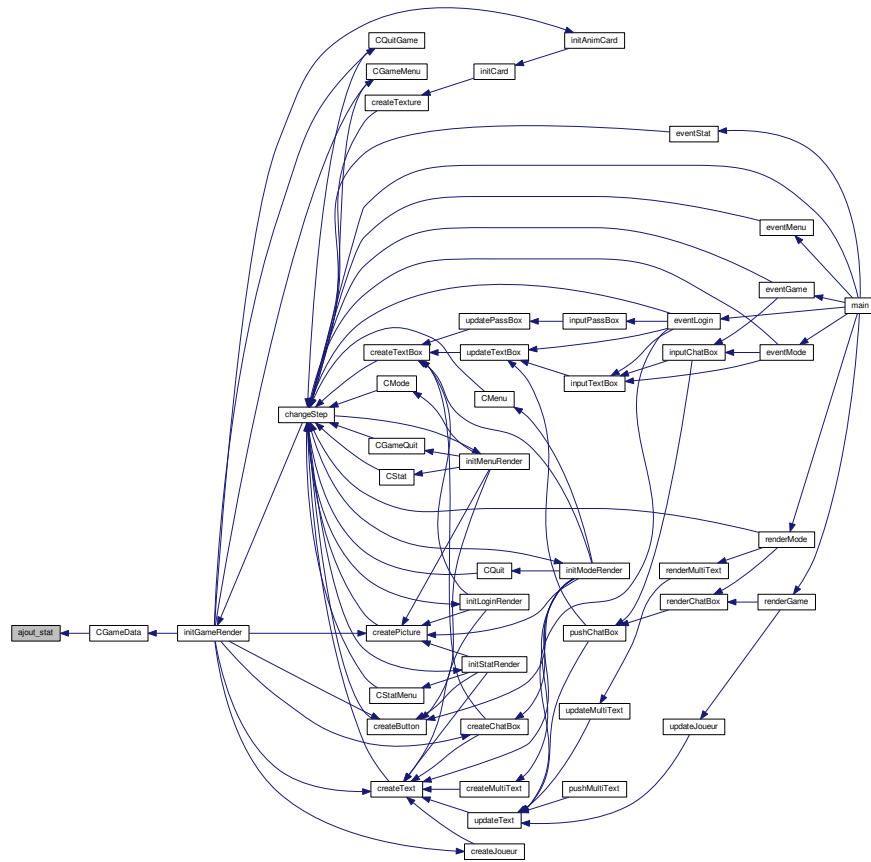
<i>victoire</i>	true, ajouter une victoire, false une défaite
-----------------	---

Definition at line 160 of file stat.c.

Here is the call graph for this function:



Here is the caller graph for this function:



4.17.3.2 void eventStat ()

Gestion des evenements spécifique au statistiques.

Definition at line 28 of file stat.c.

Here is the call graph for this function:

Here is the caller graph for this function:

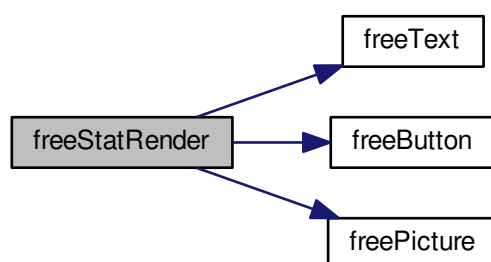


4.17.3.3 void freeStatRender ()

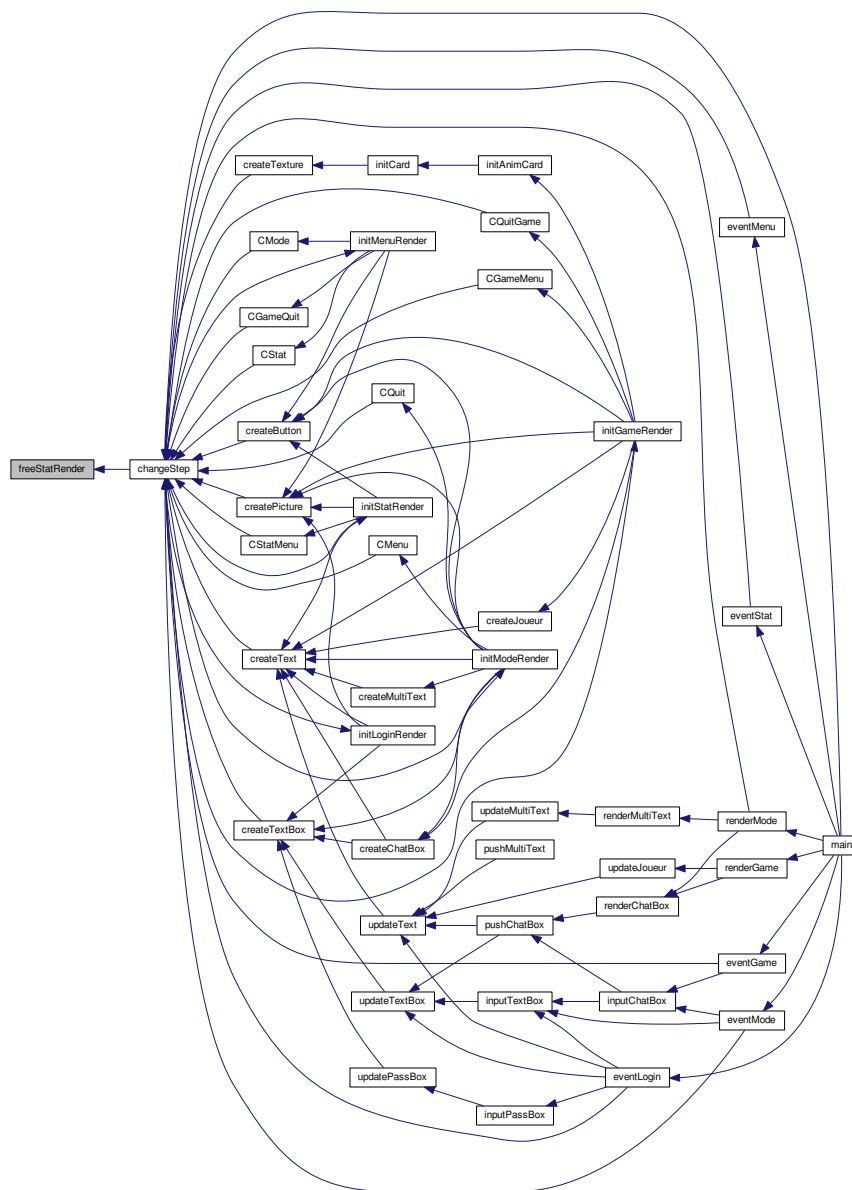
Libere les ressources graphiques initialisees par initStatRender.

Definition at line 114 of file stat.c.

Here is the call graph for this function:



Here is the caller graph for this function:



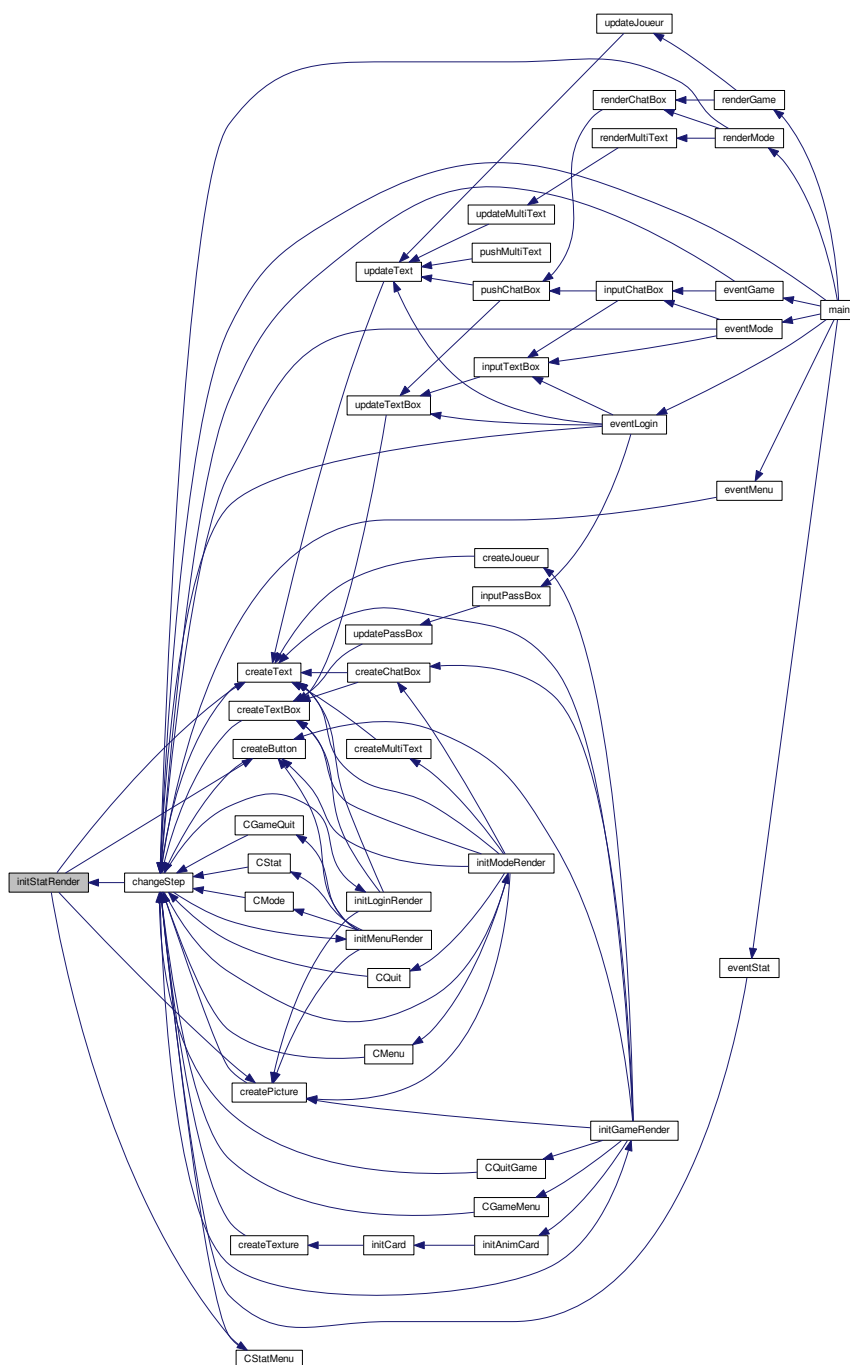
4.17.3.4 void initStatRender ()

Initialisation des ressources graphiques spécifique au statistiques.

Definition at line 55 of file stat.c.

Here is the call graph for this function:

Here is the caller graph for this function:

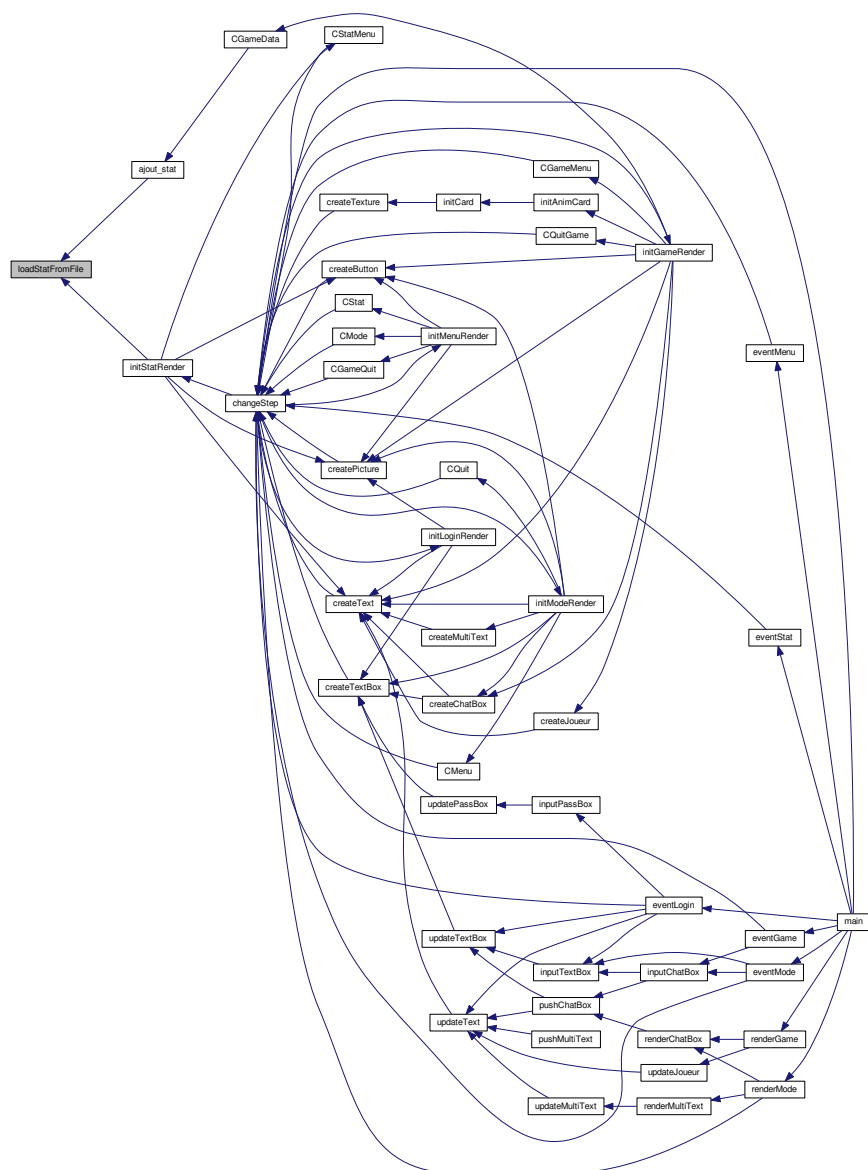


4.17.3.5 void loadStatFromFile ()

charge le tableau de statistique

Definition at line 140 of file stat.c.

Here is the caller graph for this function:



4.17.3.6 void renderStat ()

dessine ressources graphiques spécifique au statistiques

Definition at line 90 of file stat.c.

- void `freeButton` (`Button` *img)

libère les ressources graphiques de l'élément

Variables

- `SDL_Renderer` * `render`
- `TTF_Font` * `font`

4.18.1 Function Documentation

4.18.1.1 `Button` `createButton` (`char str[]`, `int x`, `int y`, `int size`)

créer les ressources du bouton Attention le callback n'est pas initialisé

Parameters

<code>str</code>	contenu du bouton (chaîne de caractère)
<code>x</code>	position x du bouton
<code>y</code>	position y du bouton
<code>size</code>	rapport entre la taille de la surface (dépend de la police) et la texture

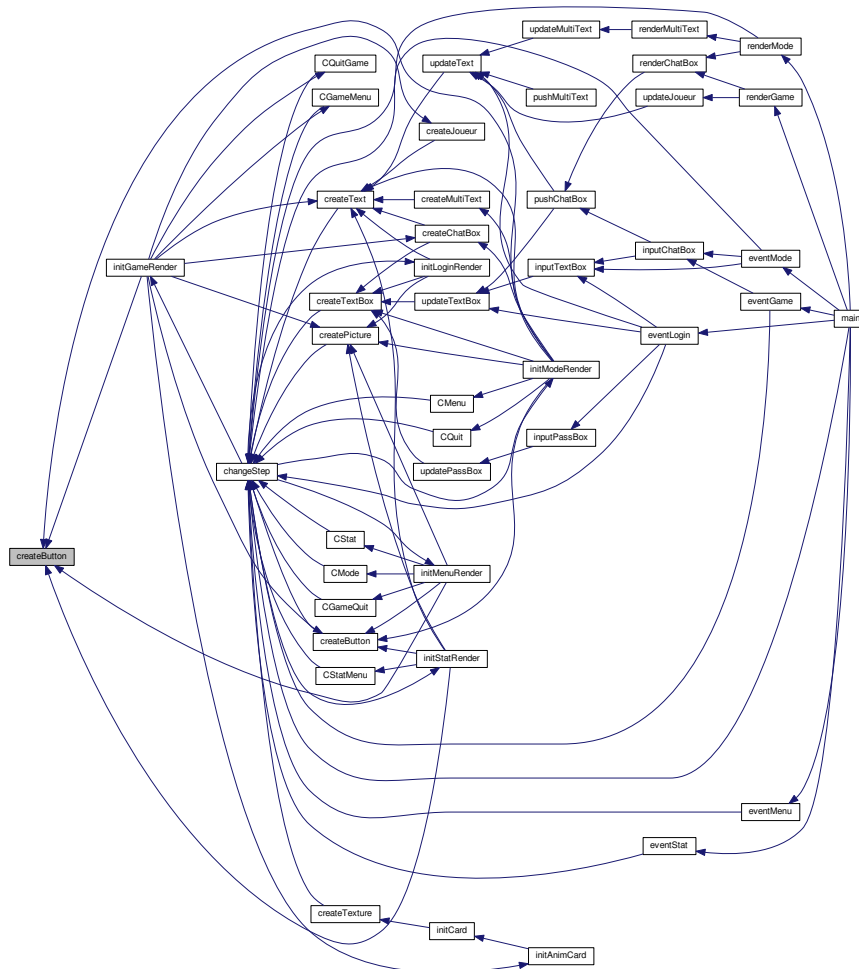
Returns

l'objet initialisé

Definition at line 9 of file button.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.18.1.2 void freeButton (Button * *img*)

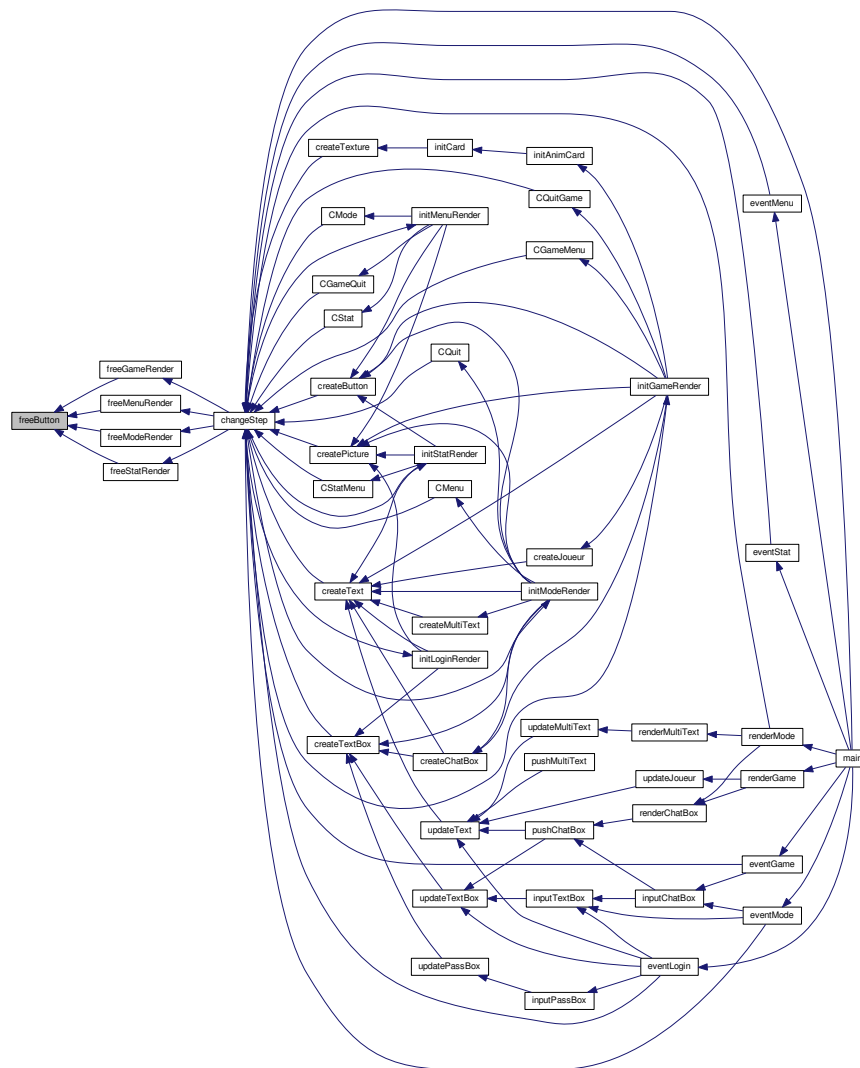
libère les ressources graphiques de l'élément

Parameters

<i>img</i>	pointeur sur le bouton
------------	------------------------

Definition at line 94 of file button.c.

Here is the caller graph for this function:



4.18.1.3 void inputButton (Button * ptr, SDL_Event * event)

gère les événements sur le bouton

Parameters

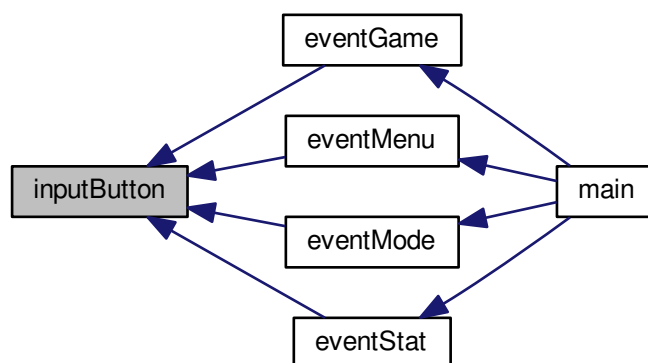
<i>ptr</i>	pointeur sur le bouton
<i>event</i>	variable contenant l'événement à prendre en compte

Definition at line 74 of file button.c.

Here is the call graph for this function:



Here is the caller graph for this function:



4.18.1.4 void renderButton (Button * img)

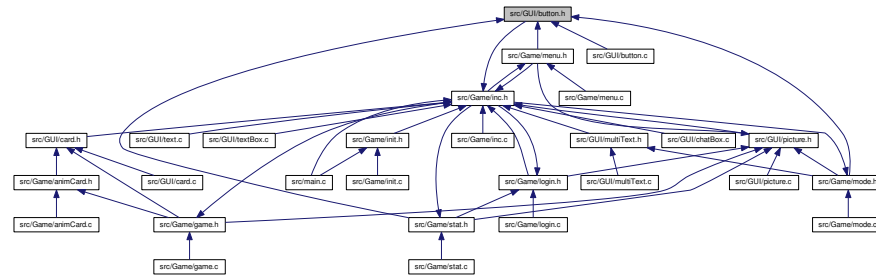
affiche le bouton sur l'écran

Parameters

<i>img</i>	pointeur sur le bouton
------------	------------------------

Definition at line 86 of file button.c.

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [Button](#)
élément graphique, cliquable.

Macros

- `#define` [SIZESTR](#) 21

Functions

- [Button](#) `createButton` (char str[], int x, int y, int size)
créer les ressources du bouton Attention le callback n'est pas initialisé
- void [inputButton](#) ([Button](#) *ptr, SDL_Event *event)
gère les événement sur le bouton
- void [renderButton](#) ([Button](#) *img)
affiche le bouton sur l'écran
- void [freeButton](#) ([Button](#) *img)
libère les ressources graphiques de l'élément

4.19.1 Detailed Description

élément de l'interface graphique utilisateur: le bouton.

Author

Aubin Detrez

4.19.2 Macro Definition Documentation

4.19.2.1 `#define` SIZESTR 21

Definition at line 10 of file button.h.

4.19.3 Function Documentation

4.19.3.1 [Button](#) `createButton` (char str[], int x, int y, int size)

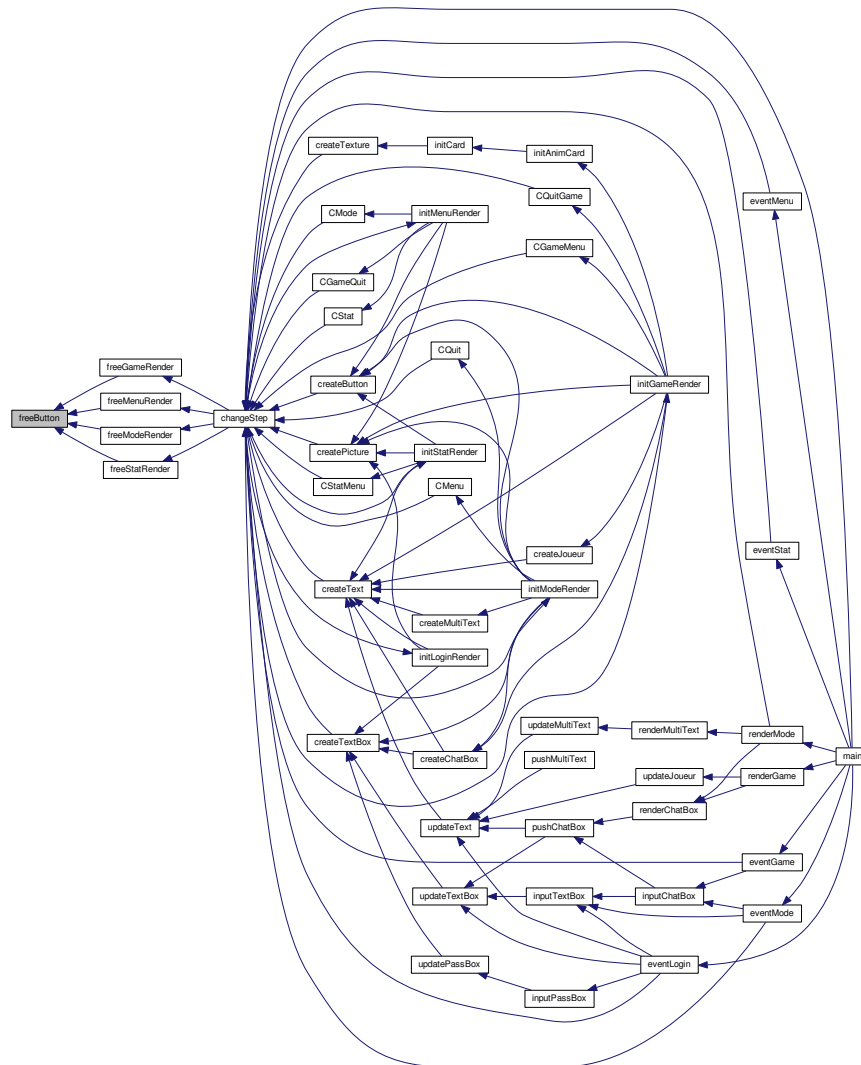
créer les ressources du bouton Attention le callback n'est pas initialisé

Parameters

<i>img</i>	pointeur sur le bouton
------------	------------------------

Definition at line 94 of file button.c.

Here is the caller graph for this function:



4.19.3.3 void inputButton (Button * ptr, SDL_Event * event)

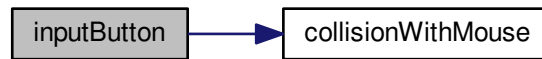
gère les événements sur le bouton

Parameters

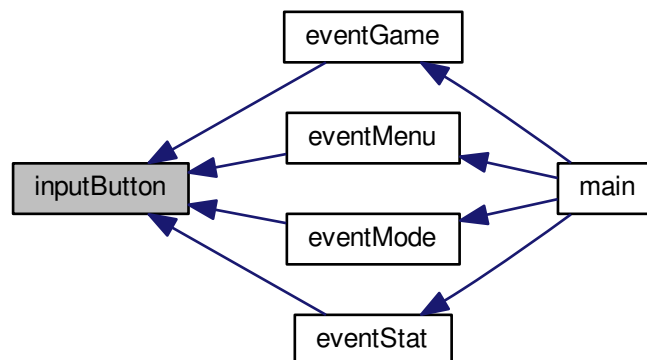
<i>ptr</i>	pointeur sur le bouton
<i>event</i>	variable contenant l'événement à prendre en compte

Definition at line 74 of file button.c.

Here is the call graph for this function:



Here is the caller graph for this function:



4.19.3.4 void renderButton (Button * *img*)

affiche le bouton sur l'écran

Parameters

<i>img</i>	pointeur sur le bouton
------------	------------------------

Definition at line 86 of file button.c.

Variables

- `SDL_Texture *` [cards](#) [[NB_CARD](#)]
- `SDL_Renderer *` [renderer](#)

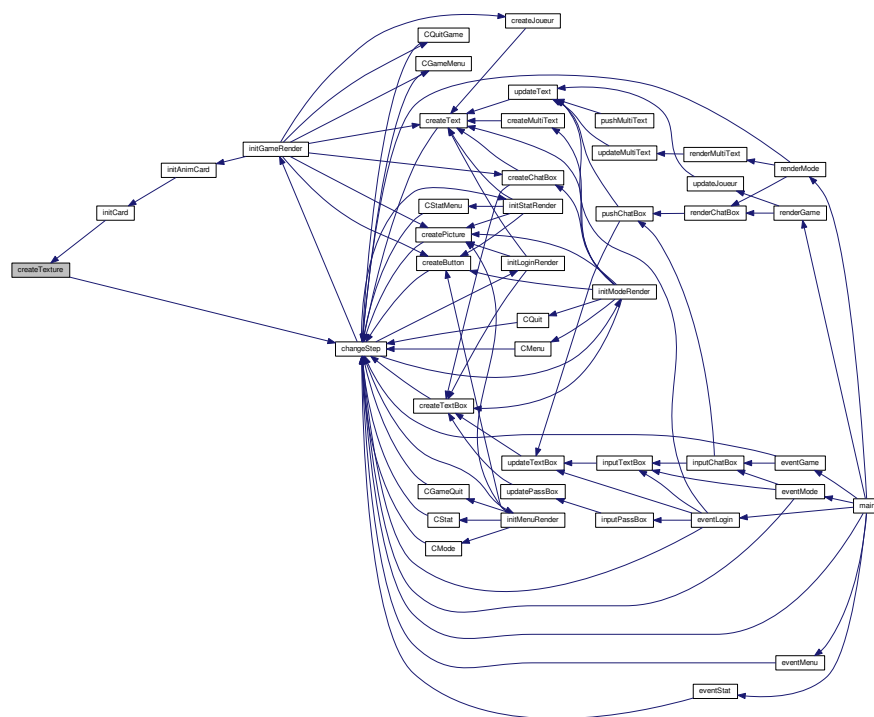
4.20.1 Function Documentation

4.20.1.1 `SDL_Texture* createTexture (char path[], int x, int y, int size)`

Definition at line 7 of file `card.c`.

Here is the call graph for this function:

Here is the caller graph for this function:

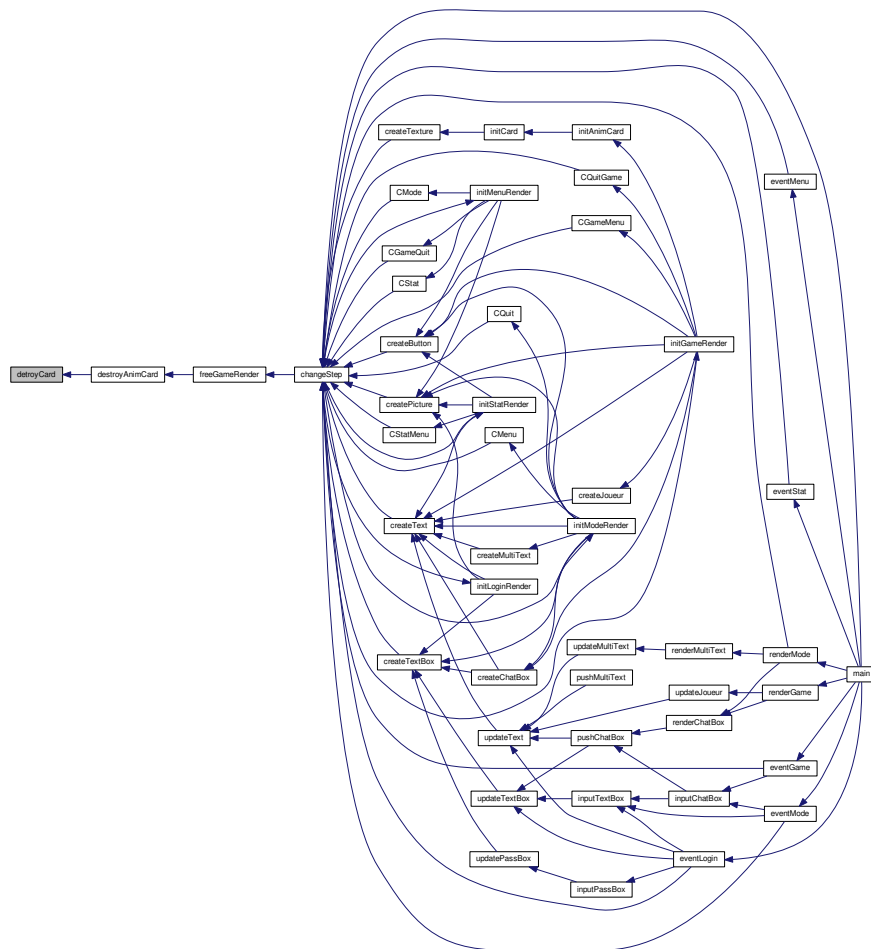


4.20.1.2 `void detroyCard ()`

libère les ressources chargées dans `initCard`

Definition at line 40 of file `card.c`.

Here is the caller graph for this function:



4.20.1.3 void eventCard (SDL_Event * event, Card * card, void (*)(int) callback)

gère les événements sur une carte si une carte est cliquée la fonction appelle le callback avec le numéro de la carte en paramètre.

Parameters

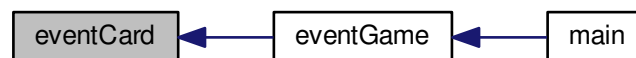
<i>event</i>	pointeur vers l'événement à prendre en compte
<i>card</i>	pointeur vers la carte
<i>callback</i>	pointeur vers la fonction à appeler en cas de clic

Definition at line 58 of file card.c.

Here is the call graph for this function:



Here is the caller graph for this function:



4.20.1.4 void initCard ()

initialisation des ressources graphiques pour les cartes les ressources sont les mêmes quelques soit le nombre de carte qui seront affichées

Definition at line 29 of file card.c.

Here is the call graph for this function:

Functions

- void `initCard` ()

initialisation des ressources graphiques pour les cartes les ressources sont les mêmes quelques soit le nombre de carte qui seront affichées

- void `renderCard` (`Card` *card)

afficher à l'écran une carte l'image affichée est choisit selon le numéro de la carte depuis les textures chargées par `initCard`

- void `eventCard` (`SDL_Event` *event, `Card` *card, void(*callback)(int))

gère les événement sur une carte si une carte est cliquée la fonction appelle le callback avec le numéro de la carte en paramètre.

- void `detroyCard` ()

libère les ressources chargées dans `initCard`

4.21.1 Detailed Description

Composant de l'interface graphique utilisateur: la carte.

Author

Aubin Detrez

4.21.2 Macro Definition Documentation

4.21.2.1 `#define CARD_H 80`

Definition at line 13 of file card.h.

4.21.2.2 `#define CARD_W 52`

Definition at line 14 of file card.h.

4.21.2.3 `#define NB_CARD 104`

Definition at line 11 of file card.h.

4.21.2.4 `#define PATH_CARD "/media/%d.bmp"`

Definition at line 12 of file card.h.

4.21.3 Function Documentation

4.21.3.1 void `detroyCard` ()

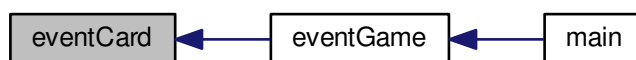
libère les ressources chargées dans `initCard`

Definition at line 40 of file card.c.

Here is the call graph for this function:



Here is the caller graph for this function:



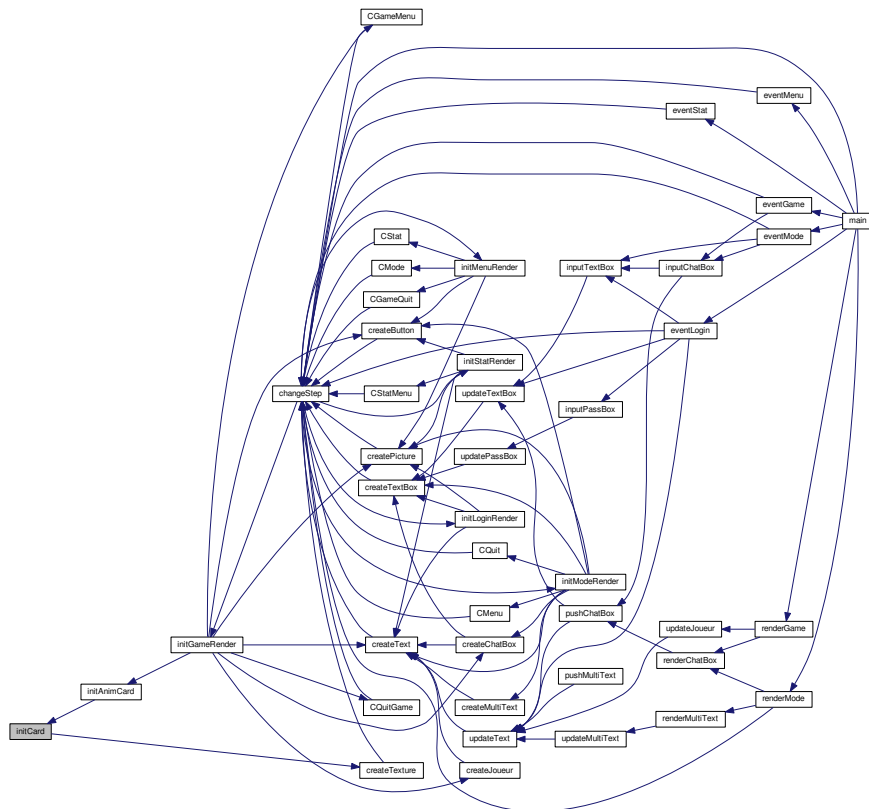
4.21.3.3 void initCard ()

initialisation des ressources graphiques pour les cartes les ressources sont les mêmes quelques soit le nombre de carte qui seront affichées

Definition at line 29 of file card.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.21.3.4 void renderCard (Card * card)

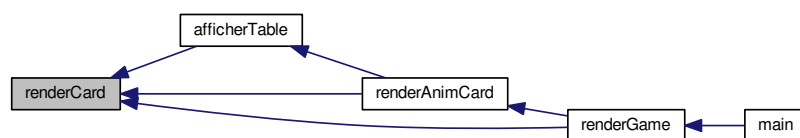
afficher à l'écran une carte l'image affichée est choisit selon le numéro de la carte depuis les textures chargées par initCard

Parameters

<i>card</i>	pointeur vers la carte à afficher
-------------	-----------------------------------

Definition at line 48 of file card.c.

Here is the caller graph for this function:



4.22 src/GUI/chatBox.c File Reference

```
#include "../Game/inc.h"
```

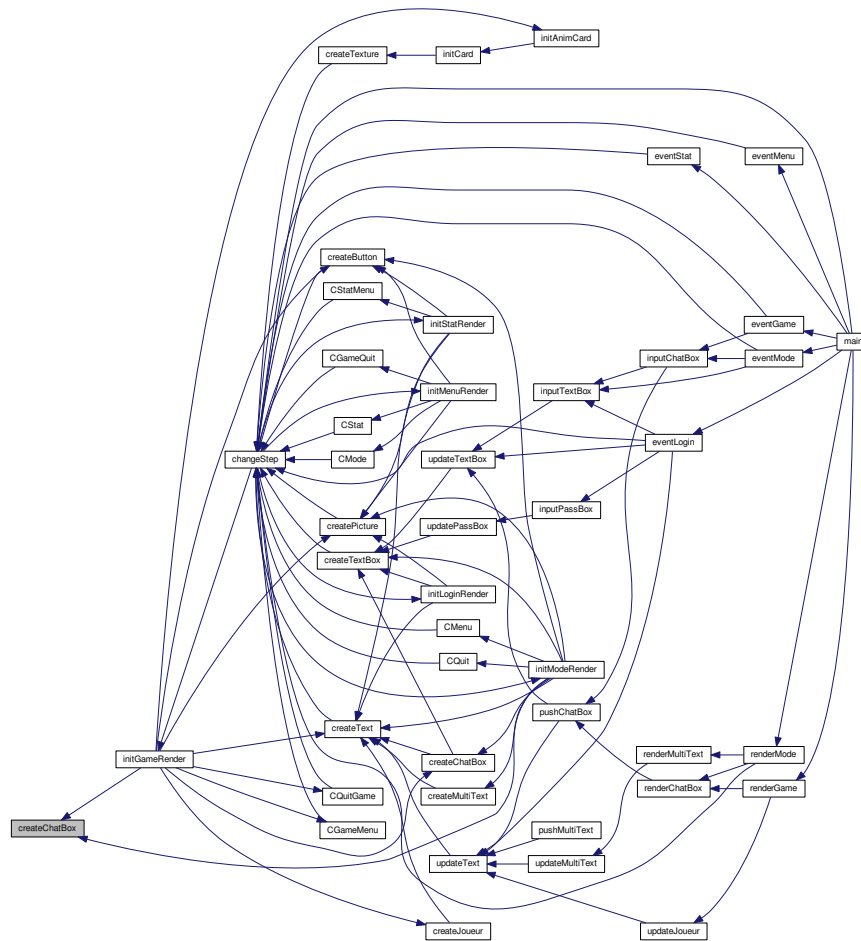

Returns

objet chat

Definition at line 4 of file chatBox.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.22.1.2 void freeChatBox (ChatBox * *img*)

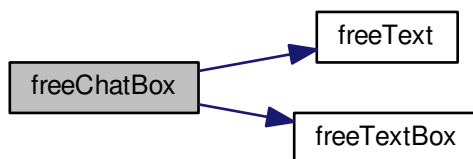
libération des ressources graphiques associées au chat

Parameters

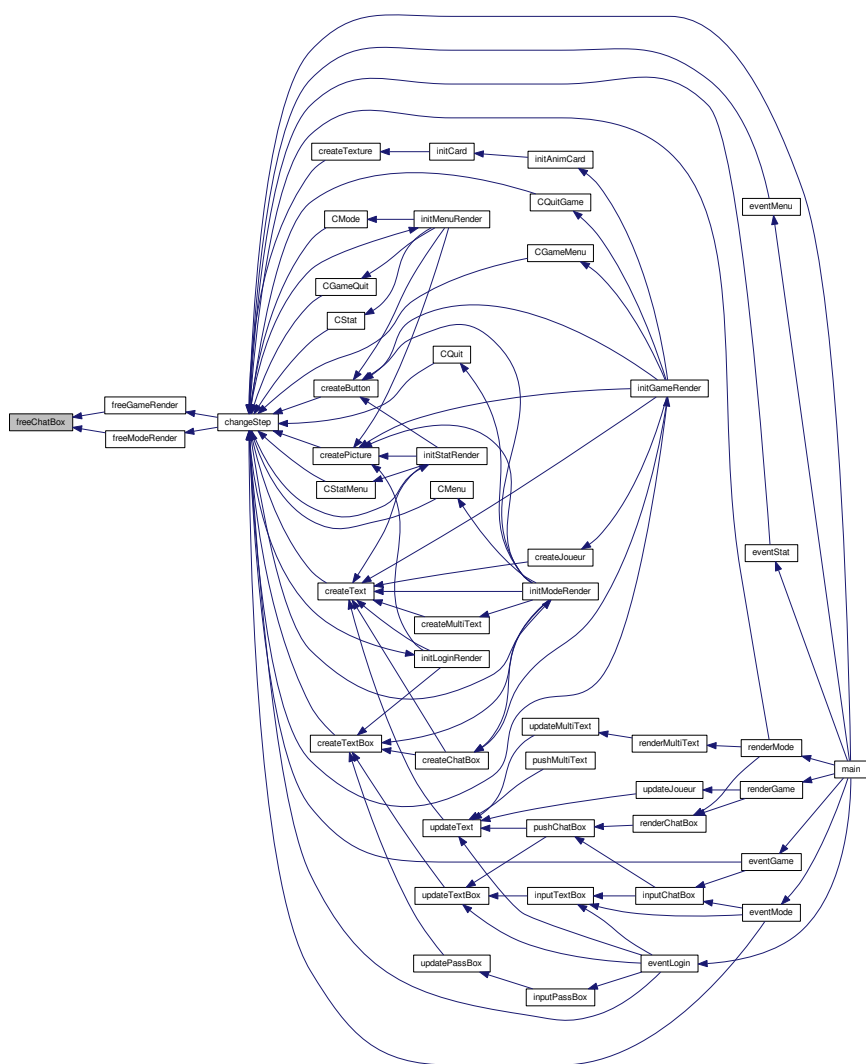
<i>img</i>	pointeur vers le chat
------------	-----------------------

Definition at line 51 of file chatBox.c.

Here is the call graph for this function:



Here is the caller graph for this function:



4.22.1.3 void inputChatBox (ChatBox * *ptr*, SDL_Event * *event*)

gestion des événements (entrées)

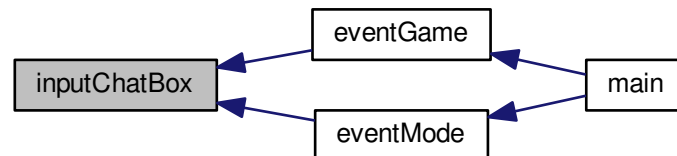
Parameters

<i>ptr</i>	pointeur vers le chat
<i>event</i>	événement à traiter

Definition at line 39 of file chatBox.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.22.1.4 void pushChatBox (ChatBox * ptr)

ajouter un message au chat le message en question est dans input.text, tout les messages précédent vont être dessiné plus haut

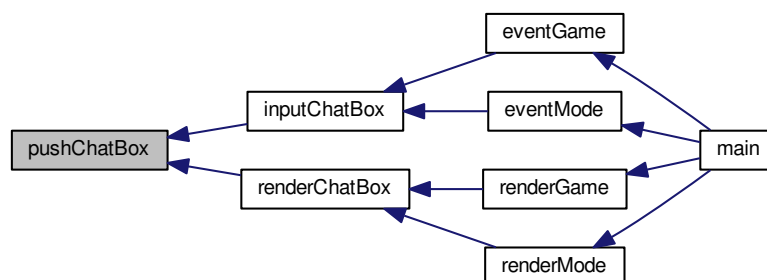
Parameters

<i>ptr</i>	pointeur sur le chat
------------	----------------------

Definition at line 24 of file chatBox.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.22.1.5 void renderChatBox (ChatBox * img)

affichage à l'écran du chat

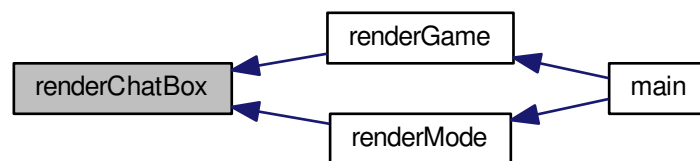
Parameters

<i>img</i>	pointeur vers le chat
------------	-----------------------

Definition at line 59 of file chatBox.c.

Here is the call graph for this function:

Here is the caller graph for this function:



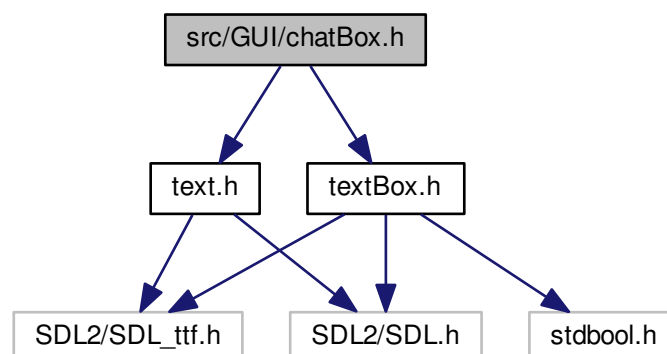
4.23 src/GUI/chatBox.h File Reference

élément de l'interface graphique utilisateur: le chat.

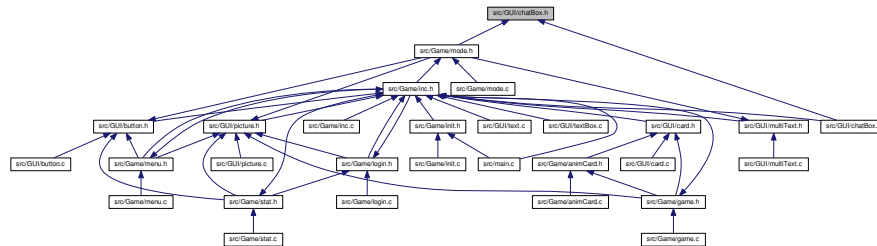
```
#include "text.h"
```

```
#include "textBox.h"
```

Include dependency graph for chatBox.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [ChatBox](#)
regroupement de composants graphiques pour faire un chat

Macros

- #define [NB_MSG](#) 6
- #define [DY](#) 30
- #define [CHAT_W](#) 300

Functions

- [ChatBox](#) [createChatBox](#) (int x, int y)
initialisation des ressources de chaque composantes et parametrage
- void [pushChatBox](#) ([ChatBox](#) *ptr)
ajouter un message au chat le message en question est dans input.text, tout les messages précédent vont être dessiné plus haut
- void [inputChatBox](#) ([ChatBox](#) *ptr, [SDL_Event](#) *event)
gestion des événements (entrées)
- void [freeChatBox](#) ([ChatBox](#) *img)
libération des ressources graphiques associées au chat
- void [renderChatBox](#) ([ChatBox](#) *img)
affichage à l'écran du chat

4.23.1 Detailed Description

élément de l'interface graphique utilisateur: le chat.

Author

Aubin Detrez

4.23.2 Macro Definition Documentation

4.23.2.1 #define CHAT_W 300

Definition at line 14 of file chatBox.h.

4.23.2.2 #define DY 30

Definition at line 13 of file chatBox.h.

4.23.2.3 #define NB_MSG 6

Definition at line 12 of file chatBox.h.

4.23.3 Function Documentation

4.23.3.1 ChatBox createChatBox (int x, int y)

initialisation des ressources de chaque composantes et parametrage

Parameters

<i>x</i>	position selon x de l'ensemble
<i>y</i>	position selon y de l'ensemble

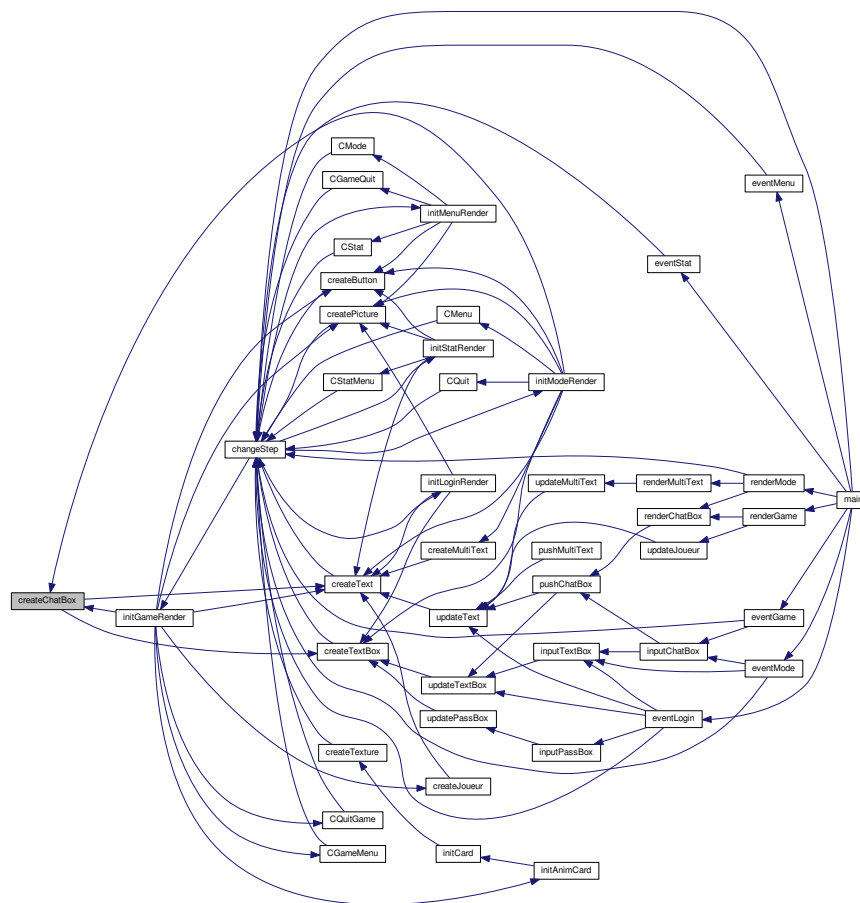
Returns

objet chat

Definition at line 4 of file chatBox.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.23.3.2 void freeChatBox (ChatBox * img)

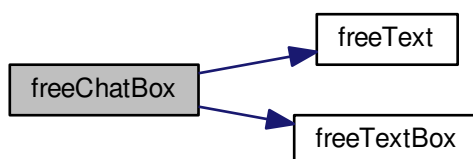
libération des ressources graphiques associées au chat

Parameters

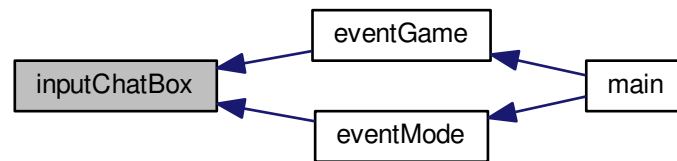
<i>img</i>	pointeur vers le chat
------------	-----------------------

Definition at line 51 of file chatBox.c.

Here is the call graph for this function:



Here is the caller graph for this function:



4.23.3.4 void pushChatBox (ChatBox * ptr)

ajouter un message au chat le message en question est dans input.text, tout les messages précédent vont être dessiné plus haut

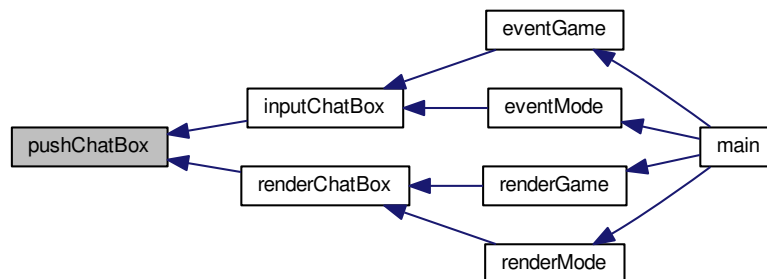
Parameters

<i>ptr</i>	pointeur sur le chat
------------	----------------------

Definition at line 24 of file chatBox.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.23.3.5 void renderChatBox (ChatBox * img)

affichage à l'écran du chat

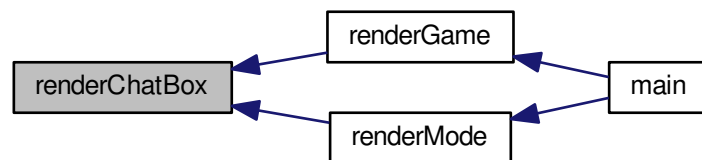
Parameters

<i>img</i>	pointeur vers le chat
------------	-----------------------

Definition at line 59 of file chatBox.c.

Here is the call graph for this function:

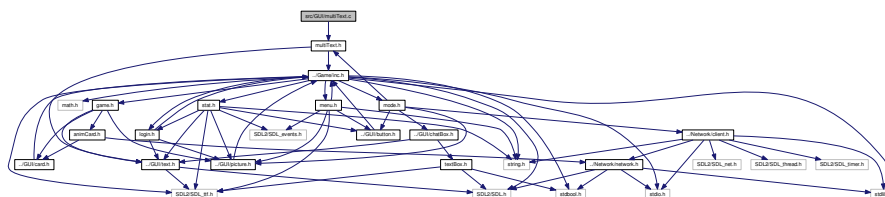
Here is the caller graph for this function:



4.24 src/GUI/multiText.c File Reference

```
#include "multiText.h"
```

Include dependency graph for multiText.c:



Macros

- #define **DY** 30

Functions

- **MultiText createMultiText** (int x, int y)
creation d'un texte sur plusieurs lignes
- void **pushMultiText** (**MultiText** *ptr, char *msg)
ajout d'une ligne les autres lignes vont étre affichées plus haut et dernière ligne va étre perdu
- void **updateMultiText** (**MultiText** *ptr)
- void **inputMultiText** (**MultiText** *ptr, SDL_Event *event)
gestion des évenement sur le texte appelle la fonction callback si il y a un clic sur une ligne et si initialised vaut true
- void **freeMultiText** (**MultiText** *img)
libère les ressources de chaque lignes
- void **renderMultiText** (**MultiText** *img)
affiche à l'écran toutes les lignes

Variables

- SDL_Renderer * [renderer](#)

4.24.1 Macro Definition Documentation

4.24.1.1 #define DY 30

Definition at line 2 of file multiText.c.

4.24.2 Function Documentation

4.24.2.1 MultiText createMultiText (int x, int y)

creation d'un texte sur plusieurs lignes

Parameters

<i>x</i>	position x de l'ensemble
<i>y</i>	position y de l'ensemble

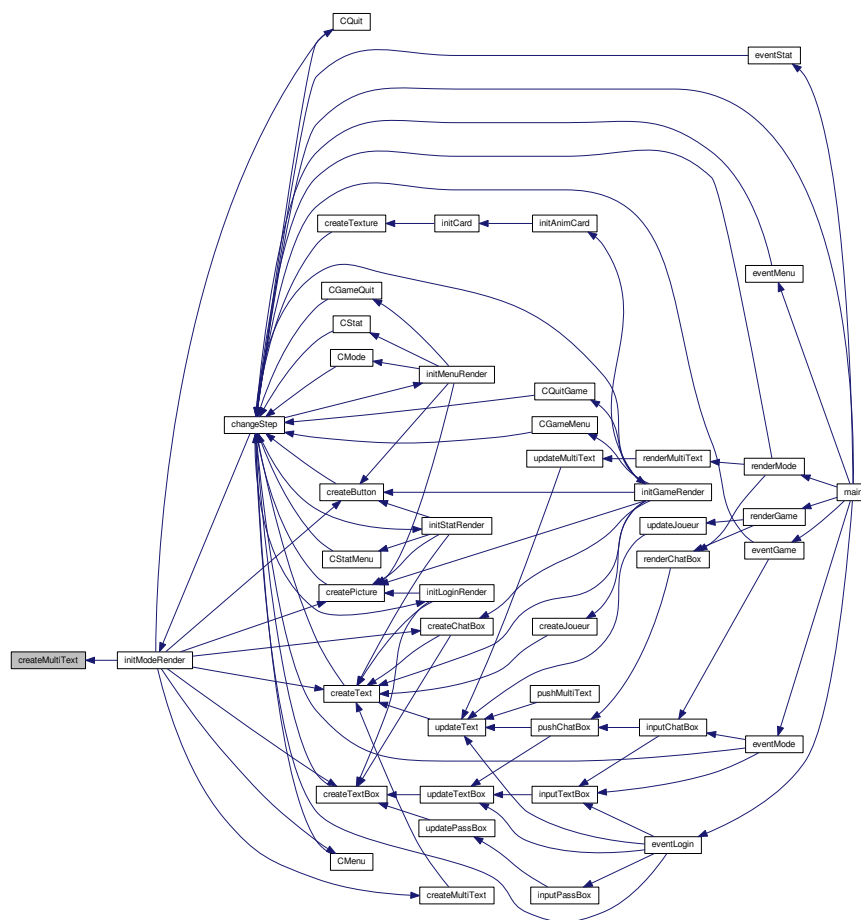
Returns

objet initialisé correspondant

Definition at line 4 of file multiText.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.24.2.2 void freeMultiText (MultiText * img)

libère les ressources de chaque ligne

Parameters

<i>img</i>	pointeur sur le texte
------------	-----------------------

Definition at line 48 of file multiText.c.

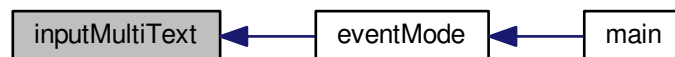
Here is the call graph for this function:



Here is the call graph for this function:



Here is the caller graph for this function:



4.24.2.4 void pushMultiText (MultiText * ptr, char * msg)

ajout d'une ligne les autres lignes vont être affichées plus haut et dernière ligne va être perdu

Parameters

<i>ptr</i>	pointeur sur le texte
<i>message</i>	à ajouter

Definition at line 18 of file multiText.c.

Here is the call graph for this function:

4.24.2.5 void renderMultiText (MultiText * img)

affiche à l'écran toutes les lignes

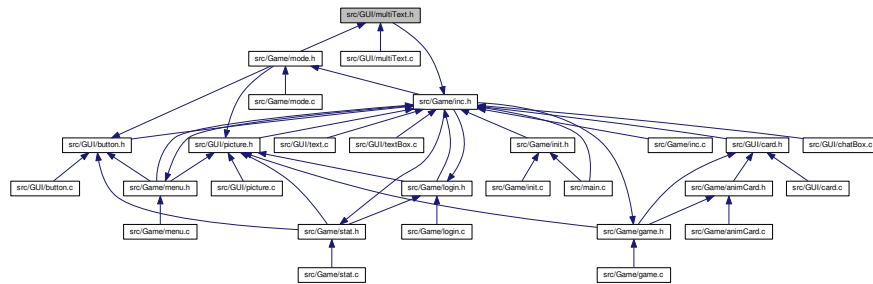
Parameters

<i>pointeur</i>	sur le texte
-----------------	--------------

Definition at line 55 of file multiText.c.

Here is the call graph for this function:

This graph shows which files directly or indirectly include this file:



Data Structures

- struct [MultiText](#)
Objet pour l'affichage sur plusieurs lignes.

Macros

- `#define` [NB_LINE](#) 10

Functions

- [MultiText](#) `createMultiText` (int x, int y)
creation d'un texte sur plusieurs lignes
- void [pushMultiText](#) ([MultiText](#) *ptr, char *msg)
ajout d'une ligne les autres lignes vont étre affichées plus haut et dernière ligne va étre perdu
- void [inputMultiText](#) ([MultiText](#) *ptr, SDL_Event *event)
gestion des événement sur le texte appelle la fonction callback si il y a un clic sur une ligne et si initialised vaut true
- void [freeMultiText](#) ([MultiText](#) *img)
libère les ressources de chaque lignes
- void [renderMultiText](#) ([MultiText](#) *img)
affiche à l'écran toutes les lignes

4.25.1 Detailed Description

élément de l'interface graphique utilisateur: l'affichage sur plusieurs lignes.

Author

Aubin Detrez

4.25.2 Macro Definition Documentation

4.25.2.1 `#define` [NB_LINE](#) 10

Definition at line 11 of file `multiText.h`.

4.25.3 Function Documentation

4.25.3.1 MultiText createMultiText (int x, int y)

creation d'un texte sur plusieurs lignes

Parameters

<i>x</i>	position x de l'ensemble
<i>y</i>	position y de l'ensemble

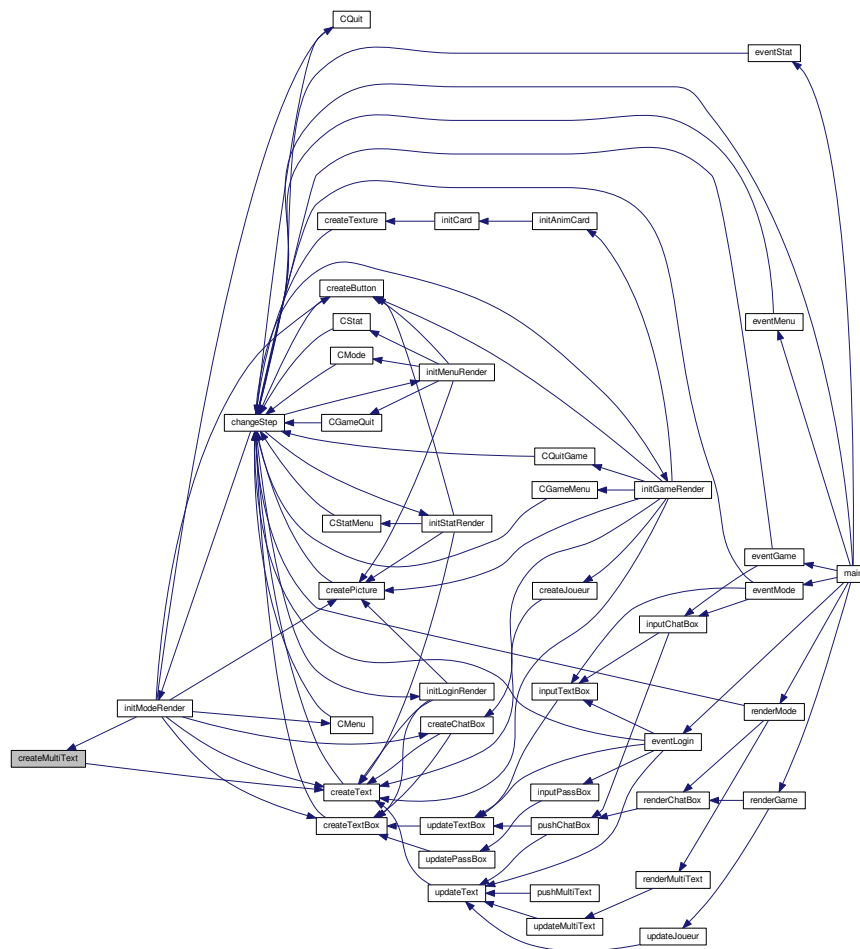
Returns

objet initialisé correspondant

Definition at line 4 of file multiText.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.25.3.2 void freeMultiText (MultiText * img)

libère les ressources de chaque ligne

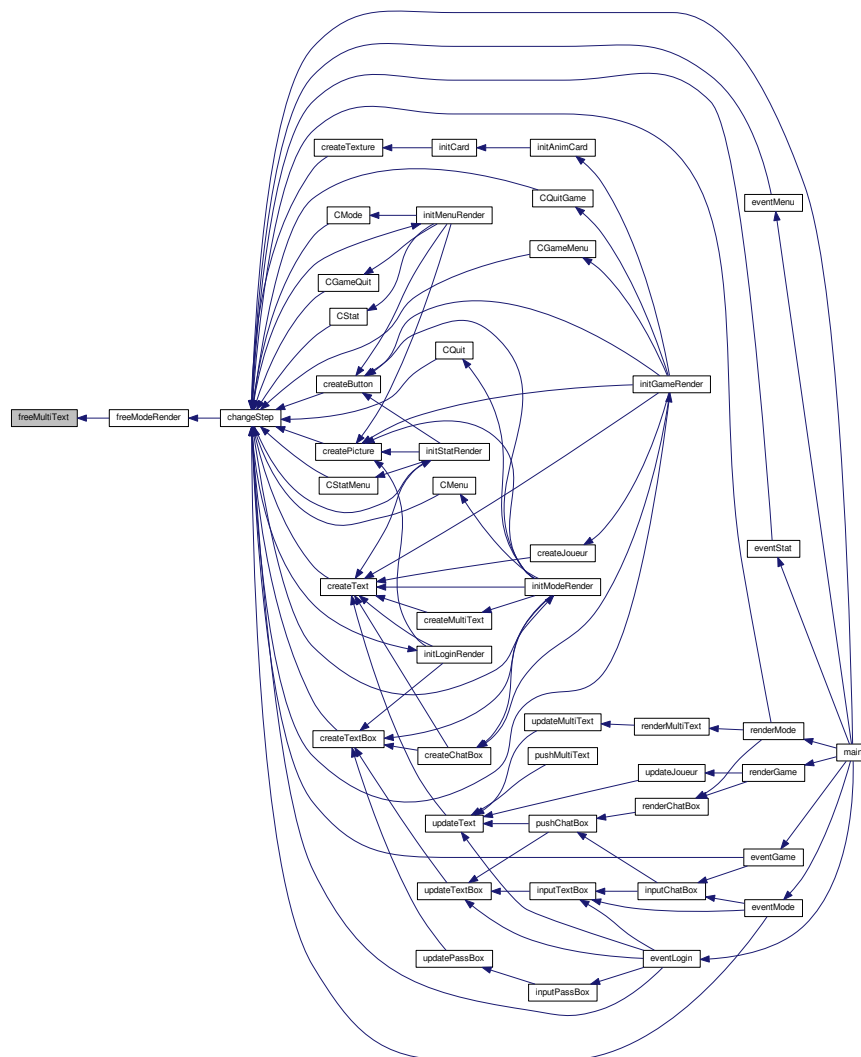
Parameters

<i>img</i>	pointeur sur le texte
------------	-----------------------

Here is the call graph for this function:



Here is the caller graph for this function:



4.25.3.3 void inputMultiText (MultiText * *ptr*, SDL_Event * *event*)

gestion des événement sur le texte appelle la fonction callback si il y a un clic sur une ligne et si initialised vaut true

Parameters

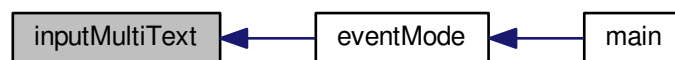
<i>ptr</i>	pointeur sur le text
<i>event</i>	événement à prendre en compte

Definition at line 38 of file multiText.c.

Here is the call graph for this function:



Here is the caller graph for this function:



4.25.3.4 void pushMultiText (MultiText * ptr, char * msg)

ajout d'une ligne les autres lignes vont étre affichées plus haut et dernière ligne va étre perdu

Parameters

<i>ptr</i>	pointeur sur le texte
<i>message</i>	à ajouter

Definition at line 18 of file multiText.c.

Here is the call graph for this function:

4.25.3.5 void renderMultiText (MultiText * img)

affiche à l'écran toutes les lignes

Parameters

<i>pointeur</i>	sur le texte
-----------------	--------------

Definition at line 55 of file multiText.c.

Here is the call graph for this function:

<i>path</i>	chemin vers le fichier de l'image (format bitmap)
<i>x</i>	position x de l'image
<i>y</i>	position y de l'image
<i>size</i>	rapport entre les dimension de l'image charg celles souhait

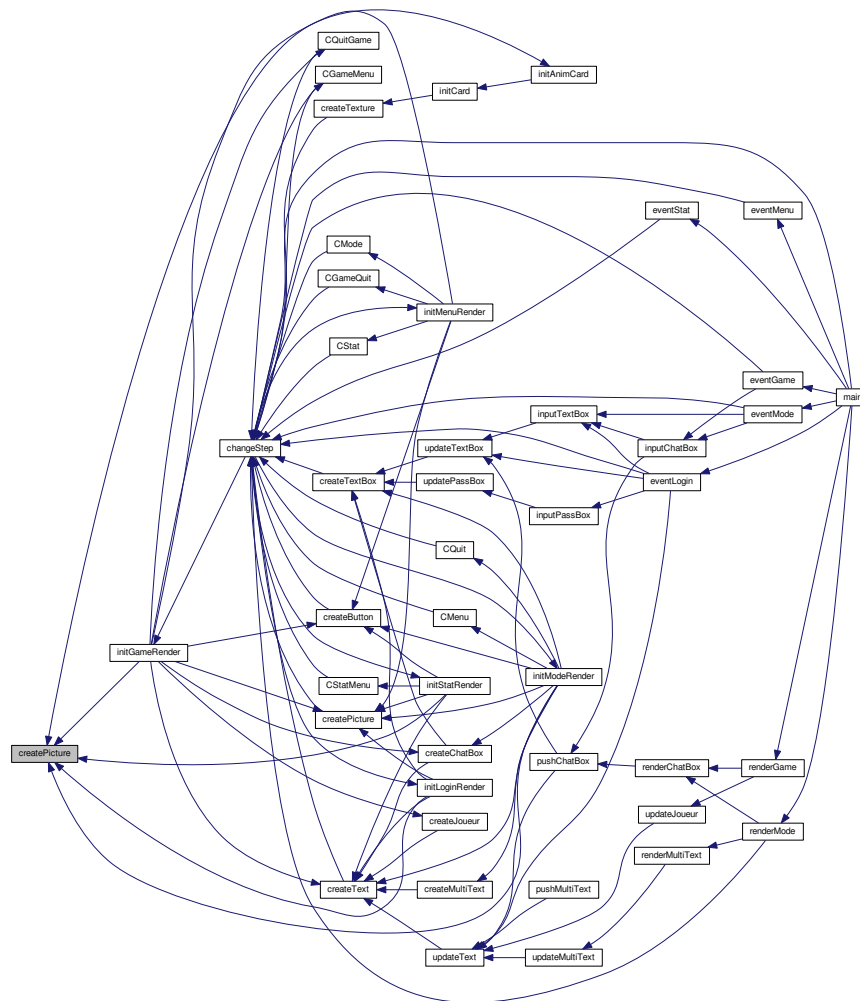
Returns

l'image initialisDoxyReturn

Definition at line 5 of file picture.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.26.1.2 void freePicture (Picture * img)

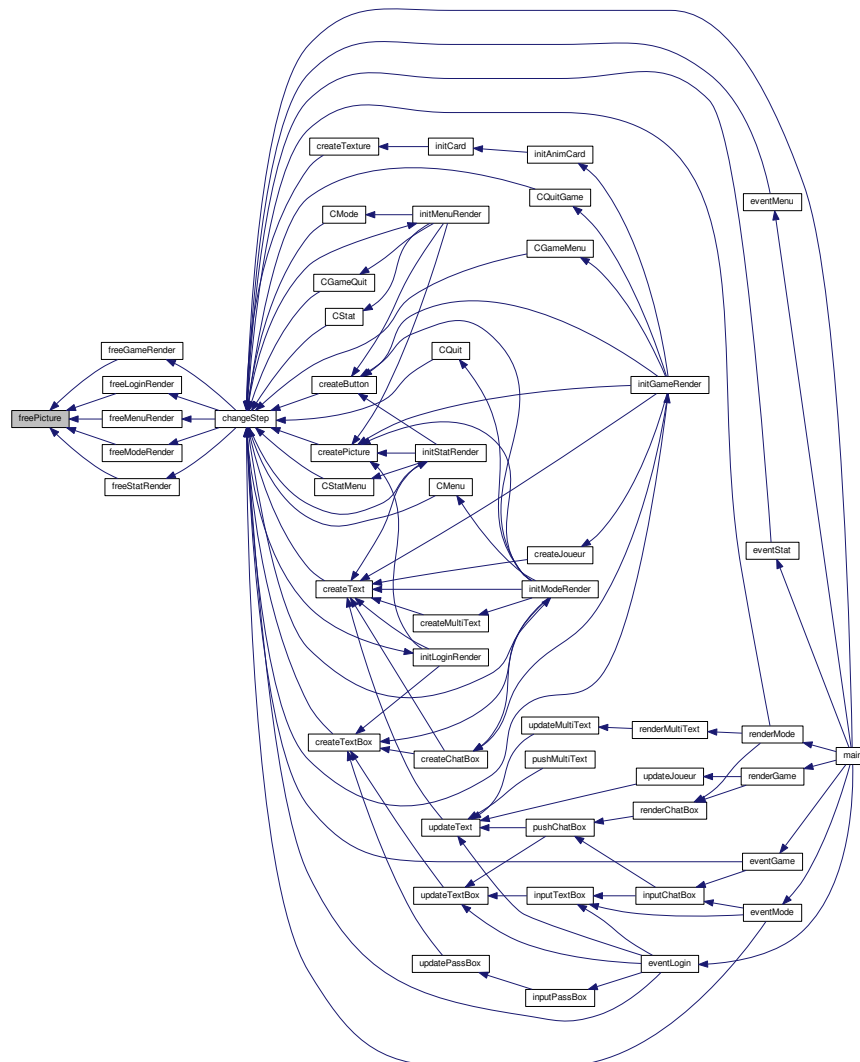
libr les ressources graphiques

Parameters

<i>img</i>	pointeur sur l'image
------------	----------------------

Definition at line 33 of file picture.c.

Here is the caller graph for this function:



4.26.1.3 void renderPicture (Picture * img)

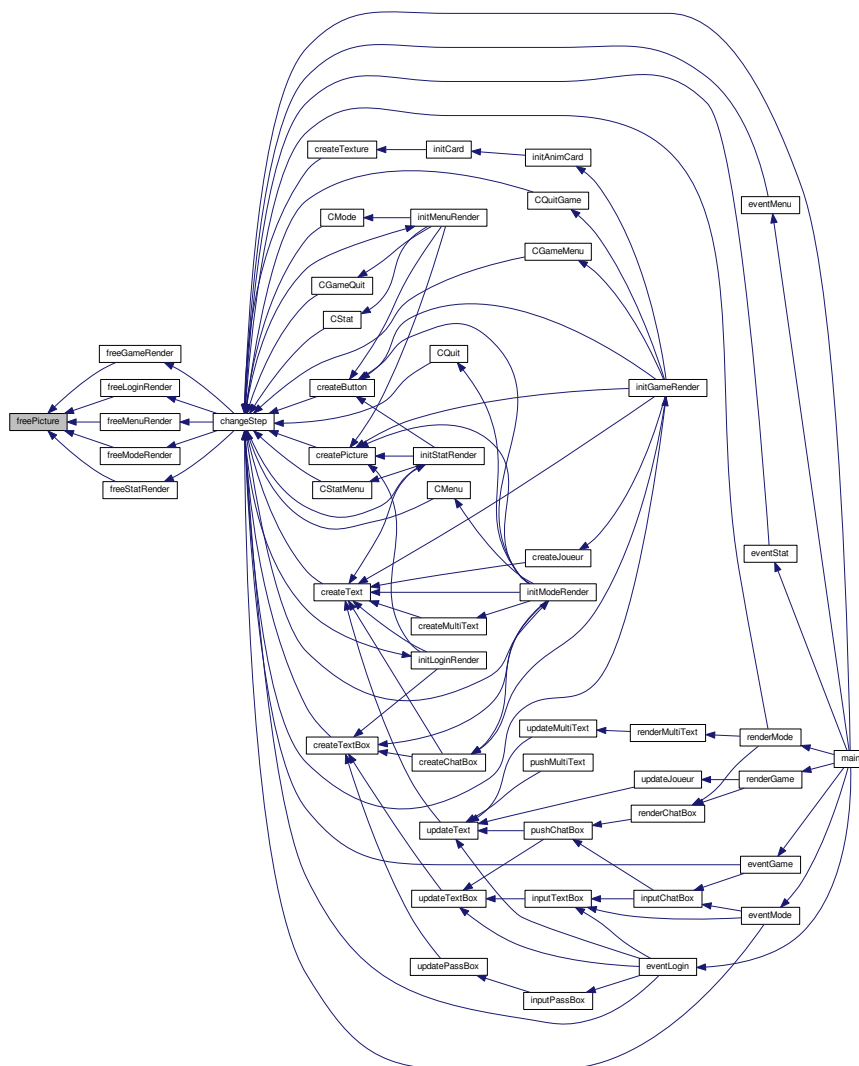
afficher an l'image

Parameters

<i>img</i>	pointeur sur l'image
------------	----------------------

Definition at line 39 of file picture.c.

Here is the caller graph for this function:



4.27.2.3 void renderPicture (Picture * *img*)

afficher an l'image

Parameters

<i>img</i>	pointeur sur l'image
------------	----------------------

Definition at line 39 of file picture.c.

4.28.1.1 Text createText (char *str*[], int *x*, int *y*, int *size*)

création de la texture et paramétrage de la structure

Parameters

<i>str</i>	chaîne de caractère à afficher
<i>x</i>	position x du texte
<i>y</i>	position y du texte
<i>size</i>	rapport entre les dimension de la surface (police) et de la texture(affiché à l'écran)

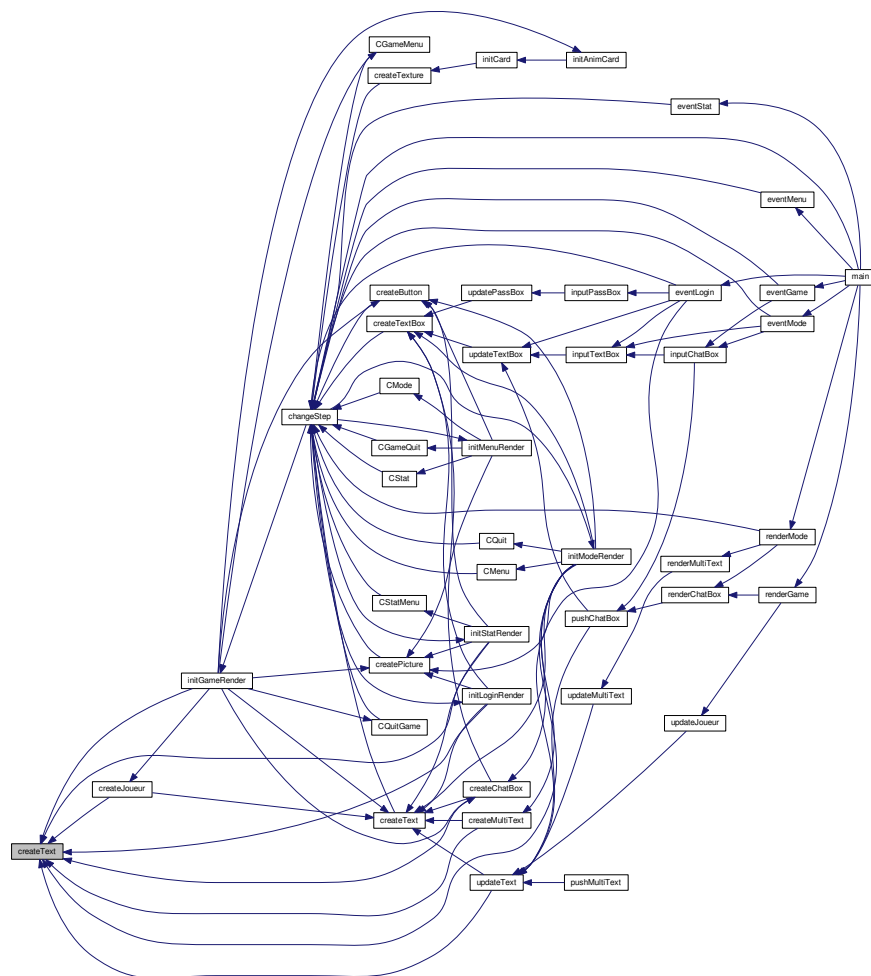
Returns

l'objet initialisé

Definition at line 16 of file text.c.

Here is the call graph for this function:

Here is the caller graph for this function:



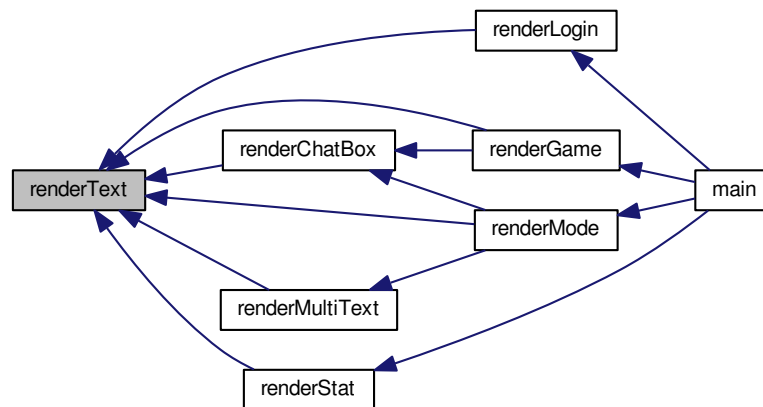
4.28.1.2 void freeText (Text * img)

libération des ressources du texte

pointeur vers le texte

Definition at line 70 of file text.c.

Here is the caller graph for this function:



4.28.1.4 void setTextColor (int *r*, int *g*, int *b*)

change la couleur du texte note: tout les text créer et/ou mis à jour après auront la couleur

Parameters

<i>r</i>	taux de rouge [0,255]
<i>g</i>	taux de vert [0,255]
<i>b</i>	taux de bleu [0,255]

Definition at line 9 of file text.c.

4.28.1.5 void updateText (Text * *ptr*, char *str*[])

met à jour l'objet pour afficher la chaine str

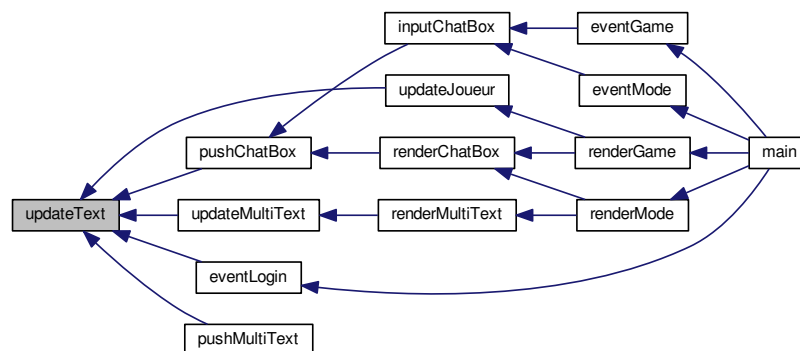
Parameters

<i>ptr</i>	pointeur vers le texte
<i>str</i>	chaîne de caractère à afficher

Definition at line 59 of file text.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.29.1 Detailed Description

élément de l'interface graphique utilisateur: le texte.

Author

Aubin Detrez

4.29.2 Function Documentation

4.29.2.1 Text createText (char *str*[], int *x*, int *y*, int *size*)

création de la texture et paramétrage de la structure

Parameters

<i>str</i>	chaîne de caractère à afficher
<i>x</i>	position x du texte
<i>y</i>	position y du texte
<i>size</i>	rapport entre les dimension de la surface (police) et de la texture(affiché à l'écran)

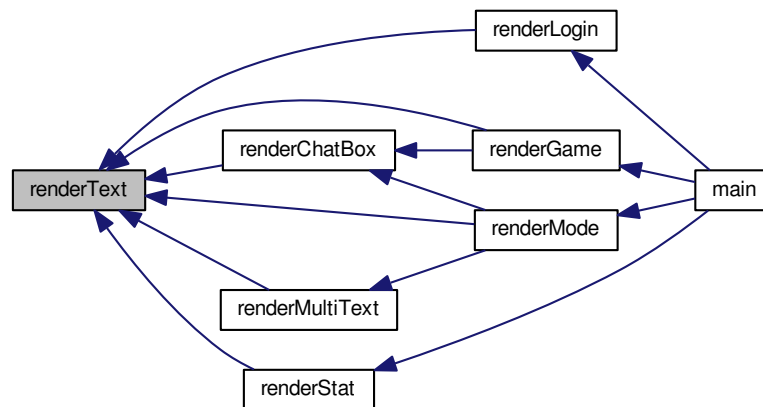
Returns

l'objet initialisé

Definition at line 16 of file text.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.29.2.4 void setTextColor (int *r*, int *g*, int *b*)

change la couleur du texte note: tout les text créer et/ou mis à jour après auront la couleur

Parameters

<i>r</i>	taux de rouge [0,255]
<i>g</i>	taux de vert [0,255]
<i>b</i>	taux de bleu [0,255]

Definition at line 9 of file text.c.

4.29.2.5 void updateText (Text * *ptr*, char *str*[])

met à jour l'objet pour afficher la chaine str

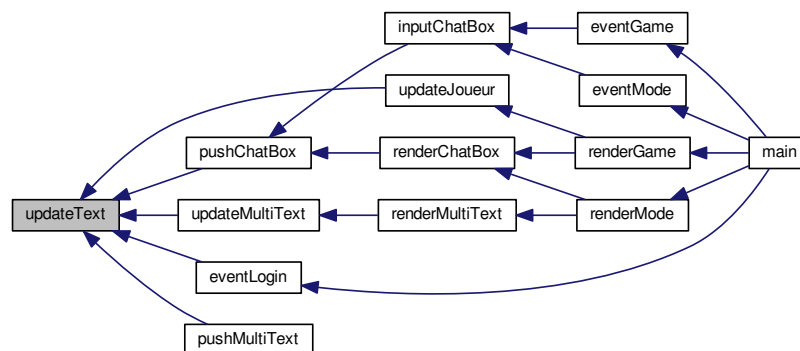
Parameters

<i>ptr</i>	pointeur vers le texte
<i>str</i>	chaîne de caractère à afficher

Definition at line 59 of file text.c.

Here is the call graph for this function:

Here is the caller graph for this function:



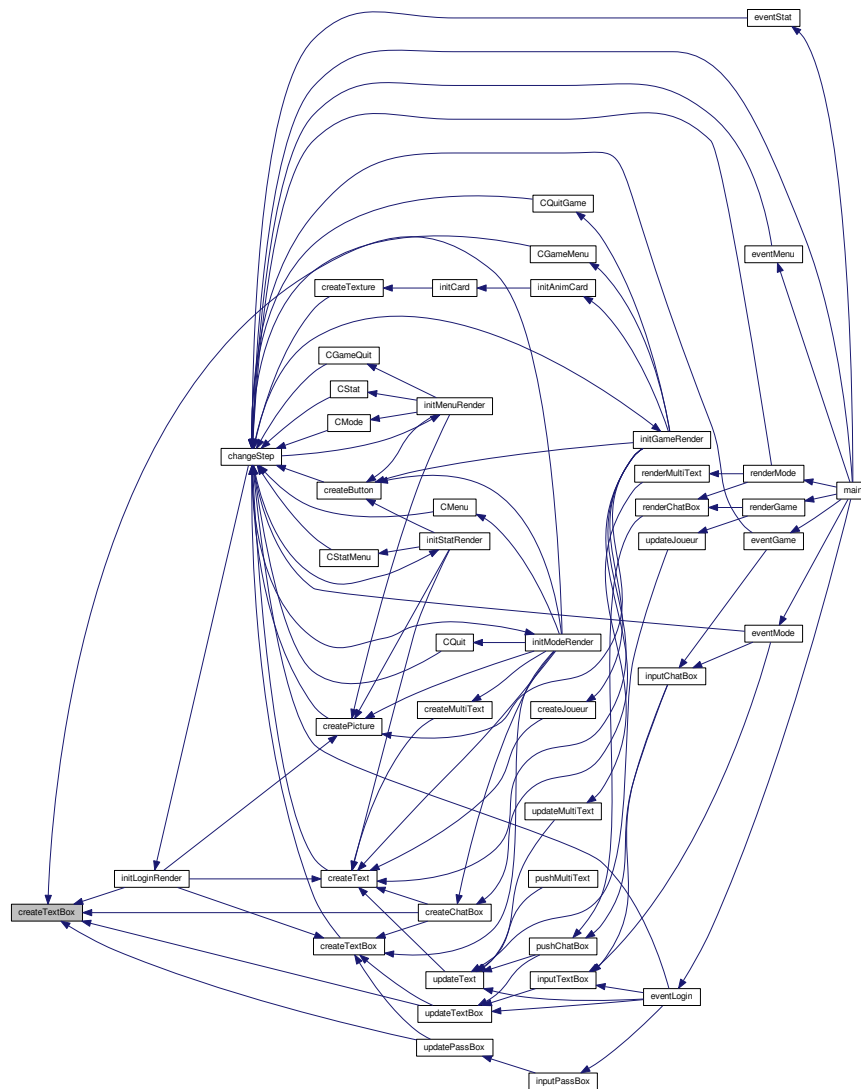
Returns

l'objet initialisé

Definition at line 9 of file textBox.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.30.1.2 void freeTextBox (TextBox * *img*)

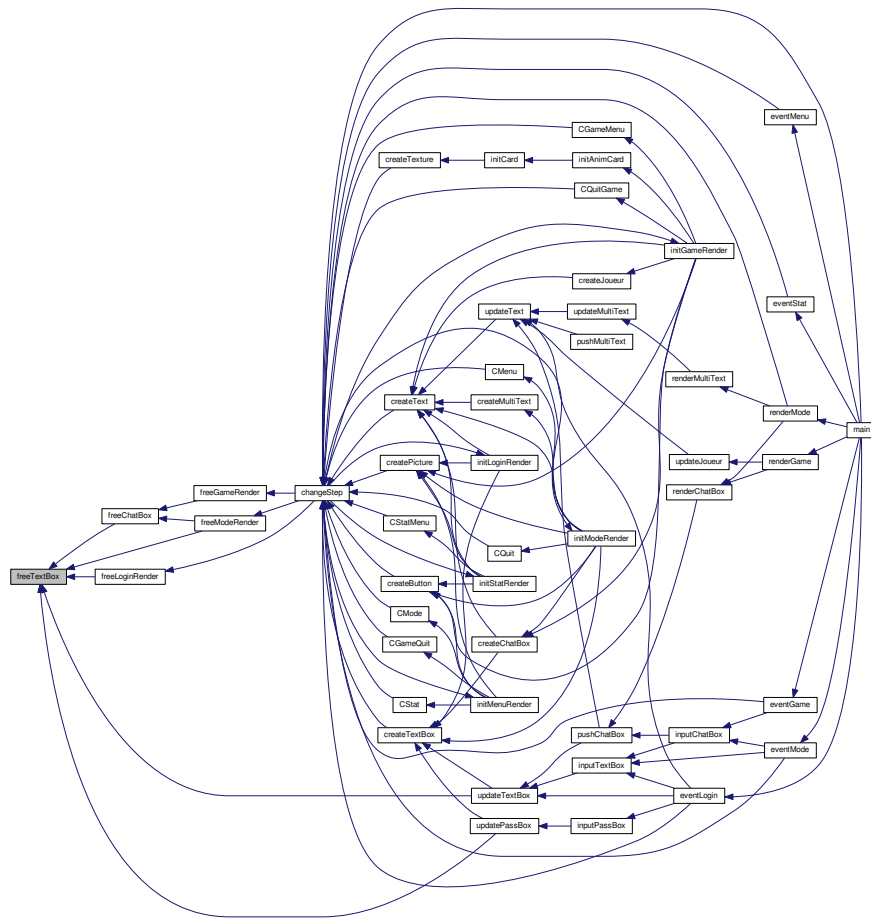
libère les ressources utilisées par la zone de saisie de texte

Parameters

<i>img</i>	pointeur sur l'objet
------------	----------------------

Definition at line 101 of file textBox.c.

Here is the caller graph for this function:



4.30.1.3 void inputPassBox (TextBox * ptr, SDL_Event * event)

gère les entrée pour la saisie de mot de passe pour qu'il y ait saisie la zone doit être sélectionnée soit par un clic sur maxRect ou par default la variable text contient le mot de passe à tout instant mais la texture ne comprend que des étoiles

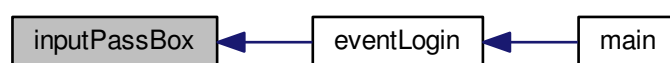
Parameters

<i>ptr</i>	pointeur sur l'objet
<i>event</i>	événement à prendre en compte

Definition at line 138 of file textBox.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.30.1.4 void inputTextBox (TextBox * *ptr*, SDL_Event * *event*)

gère les entrée pour la saisie de texte pour qu'il y ait saisie la zone doit être sélectionnée soit par un clic sur maxRect ou par default

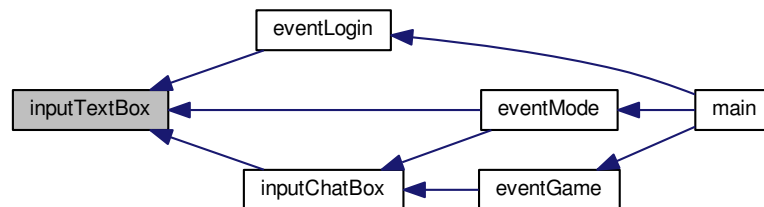
Parameters

<i>ptr</i>	pointeur sur l'objet
<i>event</i>	événement à prendre en compte

Definition at line 120 of file textBox.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.30.1.5 void renderTextBox (TextBox * img)

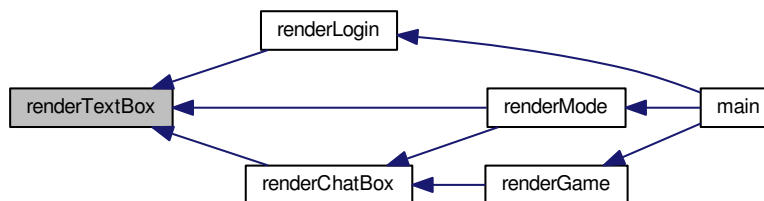
afficher la saisie de texte à l'écran

Parameters

<i>img</i>	pointeur sur l'objet
------------	----------------------

Definition at line 107 of file textBox.c.

Here is the caller graph for this function:

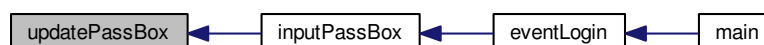


4.30.1.6 void updatePassBox (TextBox * ptr)

Definition at line 82 of file textBox.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.30.1.7 void updateTextBox (TextBox * ptr)

mise à jour d'une zone de saisie de texte note: à partir de la variable text

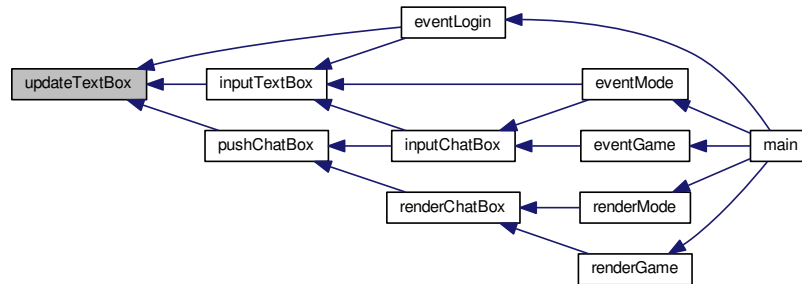
Parameters

<i>ptr</i>	pointeur sur l'objet
------------	----------------------

Definition at line 71 of file textBox.c.

Here is the call graph for this function:

Here is the caller graph for this function:



4.30.2 Variable Documentation

4.30.2.1 TTF_Font* font

Definition at line 54 of file inc.h.

4.30.2.2 SDL_Renderer* renderer

Definition at line 56 of file inc.h.

4.31 src/GUI/textBox.h File Reference

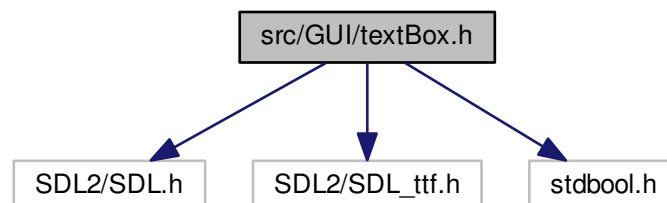
élément de l'interface graphique utilisateur: la saisie de texte.

```
#include <SDL2/SDL.h>
```

```
#include <SDL2/SDL_ttf.h>
```

```
#include <stdbool.h>
```

Include dependency graph for textBox.h:



<i>maxcar</i>	nombre maximum de caractère saisie
<i>arg0</i>	si true la zone de saisie est selectionnée par default

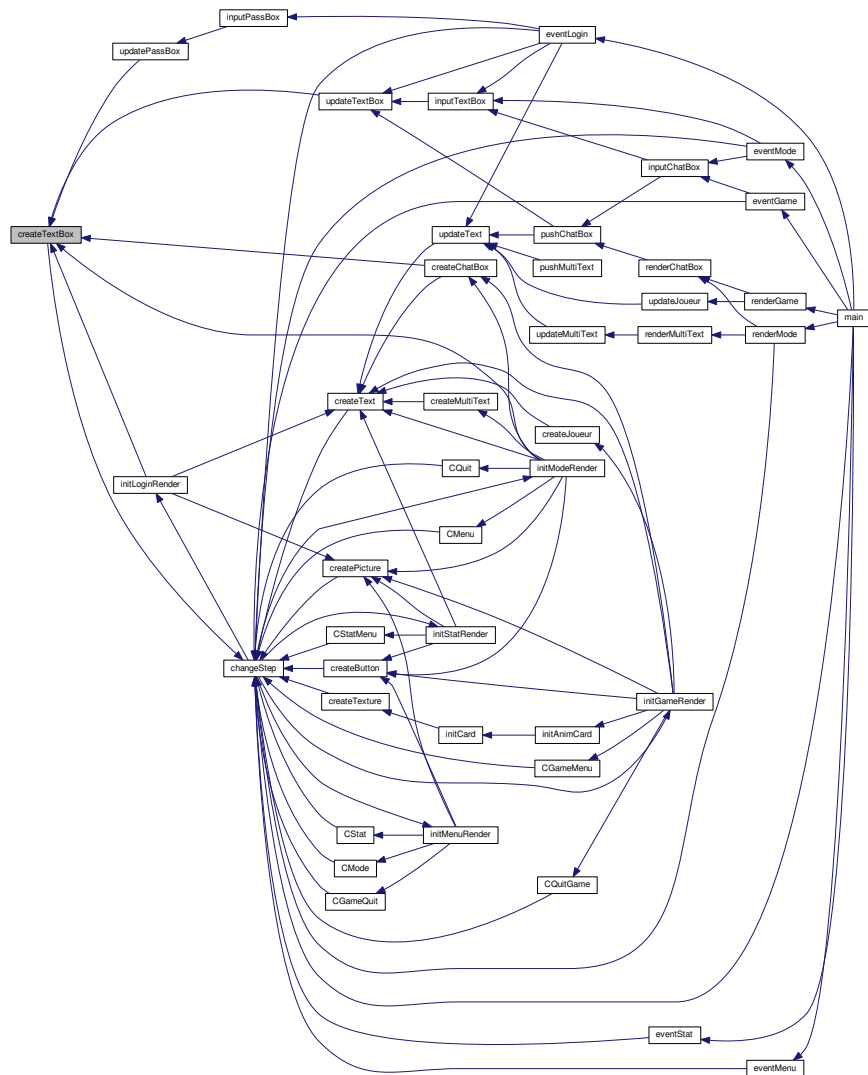
Returns

l'objet initialisé

Definition at line 9 of file textBox.c.

Here is the call graph for this function:

Here is the caller graph for this function:

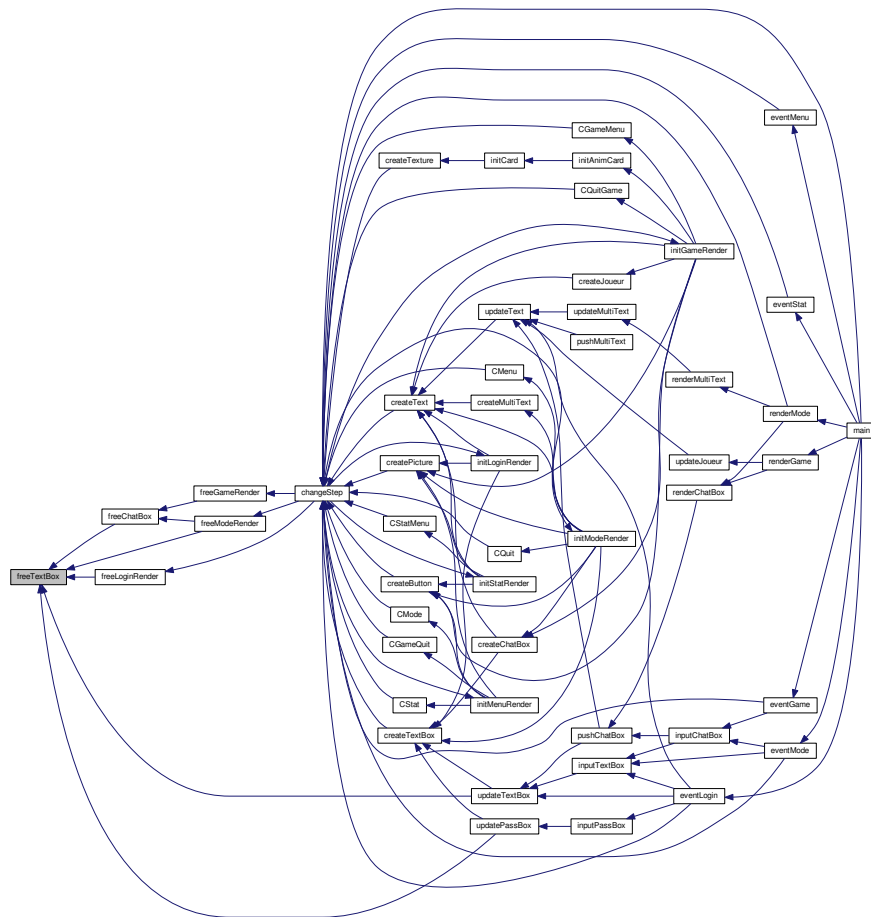
**4.31.2.2 void freeTextBox (TextBox * img)**

libère les ressources utilisées par la zone de saisie de texte

Parameters

<i>img</i>	pointeur sur l'objet
------------	----------------------

Definition at line 101 of file textBox.c.
Here is the caller graph for this function:



4.31.2.3 void inputPassBox (TextBox * ptr, SDL_Event * event)

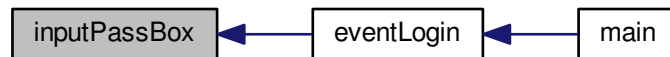
gère les entrée pour la saisie de mot de passe pour qu'il y ait saisie la zone doit être sélectionnée soit par un clic sur maxRect ou par default la variable text contient le mot de passe à tout instant mais la texture ne comprend que des étoiles

Parameters

<i>ptr</i>	pointeur sur l'objet
<i>event</i>	événement à prendre en compte

Definition at line 138 of file textBox.c.
Here is the call graph for this function:

Here is the caller graph for this function:



4.31.2.4 void inputTextBox (TextBox * ptr, SDL_Event * event)

gère les entrée pour la saisie de texte pour qu'il y ait saisie la zone doit être sélectionnée soit par un clic sur `maxRect` ou par default

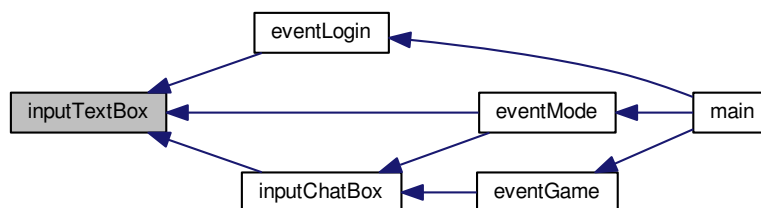
Parameters

<i>ptr</i>	pointeur sur l'objet
<i>event</i>	événement à prendre en compte

Definition at line 120 of file `textBox.c`.

Here is the call graph for this function:

Here is the caller graph for this function:



4.31.2.5 void renderTextBox (TextBox * img)

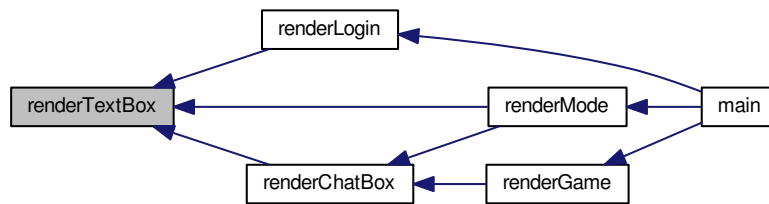
afficher la saisie de texte à l'écran

Parameters

<i>img</i>	pointeur sur l'objet
------------	----------------------

Definition at line 107 of file `textBox.c`.

Here is the caller graph for this function:



4.31.2.6 void updateTextBox (TextBox * ptr)

mise à jour d'une zone de saisie de texte note: à partir de la variable text

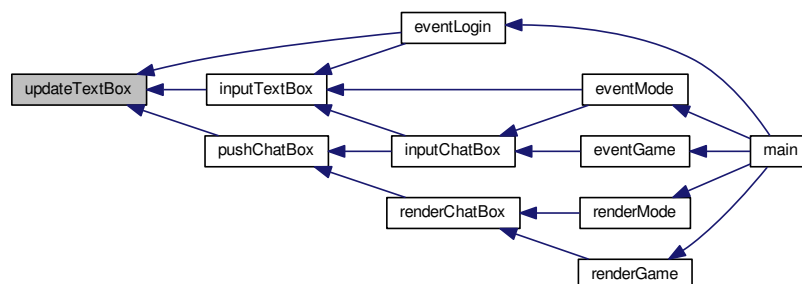
Parameters

<i>ptr</i>	pointeur sur l'objet
------------	----------------------

Definition at line 71 of file textBox.c.

Here is the call graph for this function:

Here is the caller graph for this function:



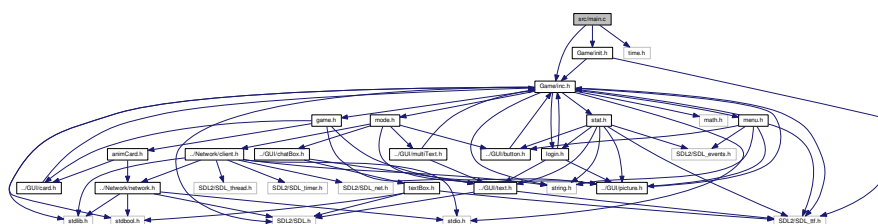
4.32 src/main.c File Reference

```

#include "Game/inc.h"
#include "Game/init.h"
#include "time.h"

```

Include dependency graph for main.c:



Functions

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

Variables

- SDL_Renderer * [renderer](#)

4.32.1 Function Documentation

4.32.1.1 int main (int argc, char ** argv)

Definition at line 12 of file main.c.

Here is the call graph for this function:

4.32.2 Variable Documentation

4.32.2.1 SDL_Renderer* renderer

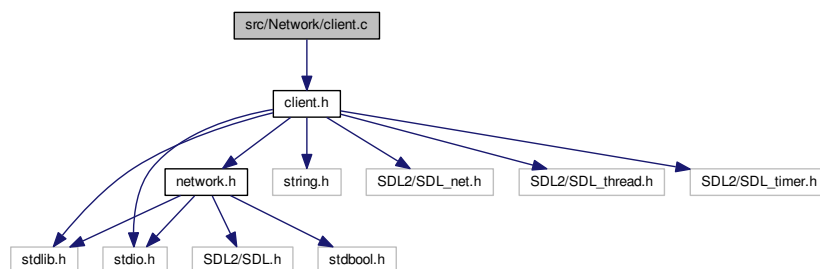
Definition at line 56 of file inc.h.

4.33 src/Network/client.c File Reference

implémentation réseau

```
#include "client.h"
```

Include dependency graph for client.c:



Functions

- void [initialisationReseau](#) (char *strip, void(*backData)([Data](#) *), void(*backList)([DataList](#) *), void(*backGame)([DataGame](#) *))
initialisation du réseau initialise la connexion avec le serveur
- void [updateCallback](#) (void(*backData)([Data](#) *), void(*backList)([DataList](#) *), void(*backGame)([DataGame](#) *))
changement de callback rediriger les retour du thread de réception vers de nouvelles fonctions
- void [reception](#) ()
lancer la réception dans un nouveau thread les données sont retournées par des callback définis dans initialisationReseau ou updateCallback
- void [sendMsg](#) (char *msg)
transmettre le message au serveur les messages sont afficher dans les chats des clients de la même salle (d'attente comme de jeu)
- void [askList](#) ()
demande au serveur de transmettre la liste des salles non bloquante et la réponse du serveur sera transmise au callback associé à son type (par le thread de réception)
- void [join](#) (char *str, int passwd)
demande à rejoindre une salle de jeu réponse du serveur via callback
- void [create](#) (char *str, int passwd)
demande à créer une salle de jeu réponse du serveur via callback

- void `startGame` ()
demande à commencer une partie réponse du serveur via callback
- void `choice` (int choice)
envoie le choix du joueur au serveur réponse du serveur via callback
- void `sendQuit` ()
demande au serveur à quitter proprement lorsque le thread de réception recevra la réponse du serveur il se terminera
- void `freeRessourcesReseau` ()
libération des ressources si elles sont initialisées
- int `isNetInitialised` ()
vérifier que le réseau est initialisé
- void `setWait` (bool arg0)
fait attendre le thread client pour la réception d'une `DataGame`
- void `printData` (`Data` *data)
- void `identifySrv` (char *name)
s'identifier sur le serveur note: seulement une fois dans une salle de jeu

Variables

- bool `isQuitting`

4.33.1 Detailed Description

implémentation réseau

Author

Aubin Detrez

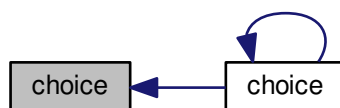
Permet de se connecter au serveur de jeu, lancer un thread d'écoute et recevoir les données du serveur pas l'intermédiaire de callback

4.33.2 Function Documentation

4.33.2.1 void askList ()

demande au serveur de transmettre la liste des salles non bloquante et la réponse du serveur sera transmise au callback associé à son type (par le thread de réception)
Definition at line 174 of file client.c.

Here is the caller graph for this function:



4.33.2.3 void create (char * *str*, int *passwd*)

demande à creer une salle de jeu réponce du serveur via callback

Parameters

<i>str</i>	chaîne de caractère pour le nom de la salle
<i>passwd</i>	mot de passe (-1 si non utilisé)

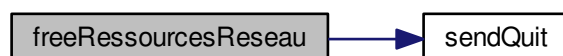
Definition at line 203 of file client.c.

4.33.2.4 void freeRessourcesReseau ()

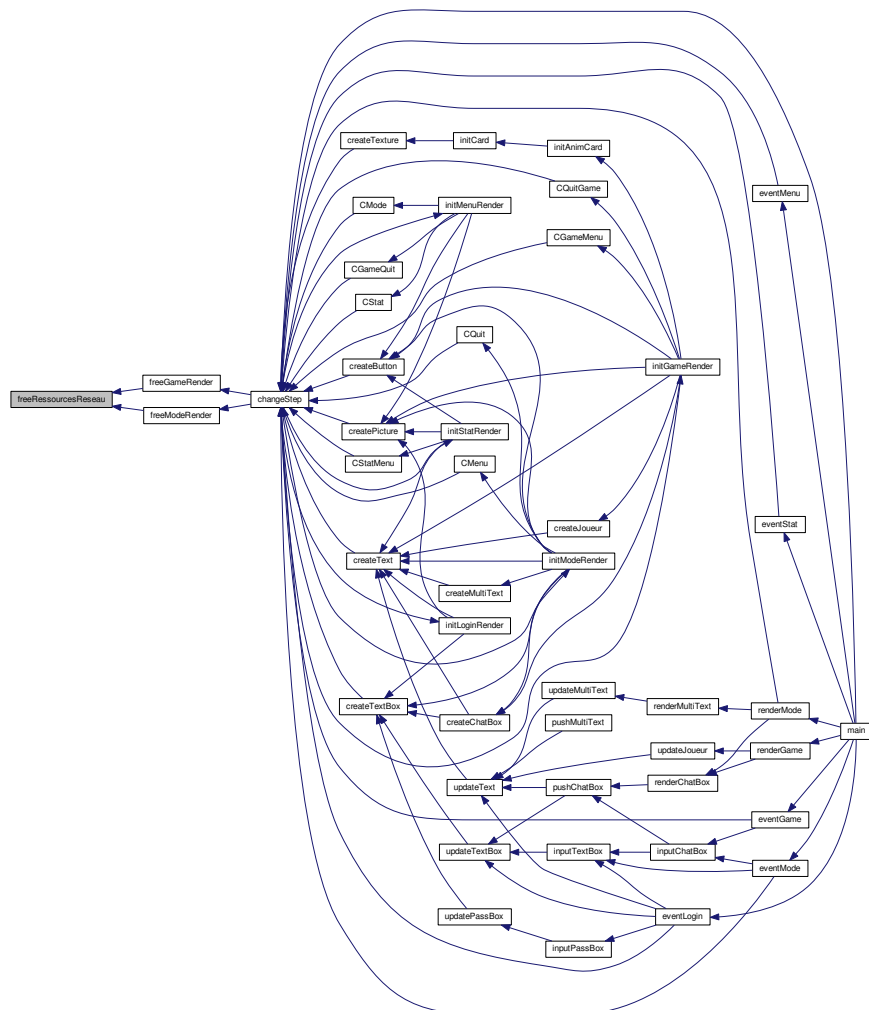
libération des ressources si elles sont initialisées

Definition at line 264 of file client.c.

Here is the call graph for this function:



Here is the caller graph for this function:



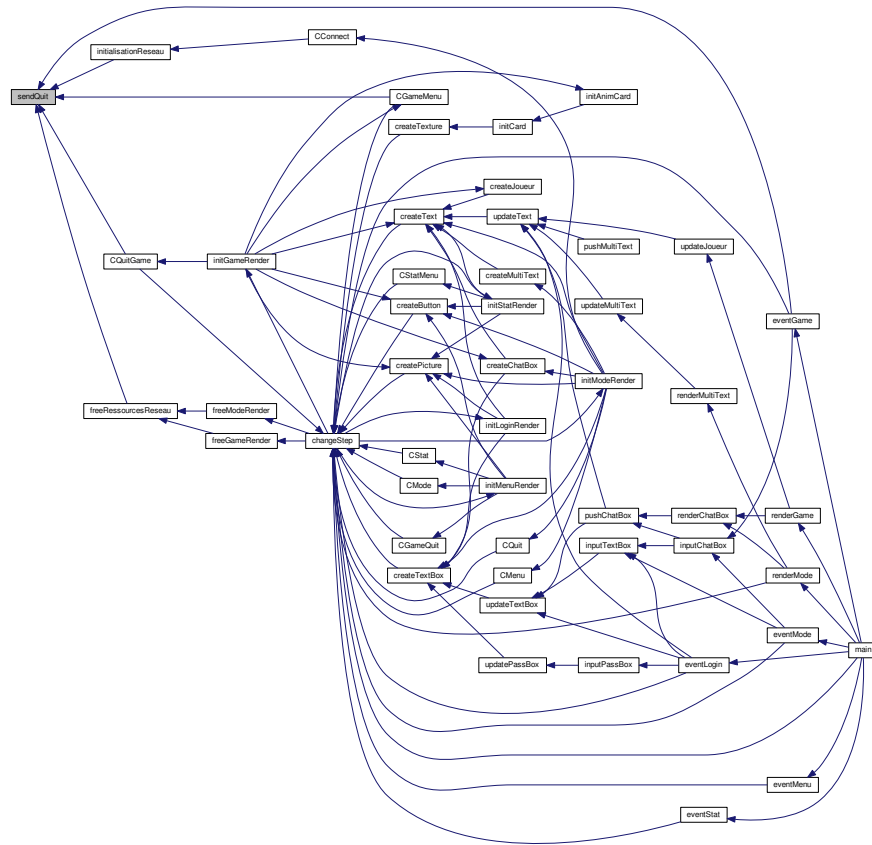
4.33.2.5 void identifySrv (char * name)

s'identifier sur le serveur note: seulement une fois dans une salle de jeu

- name chaine de caractère correspondant à son nom

Definition at line 356 of file client.c.

Here is the caller graph for this function:



4.33.2.13 void setWait (bool arg0)

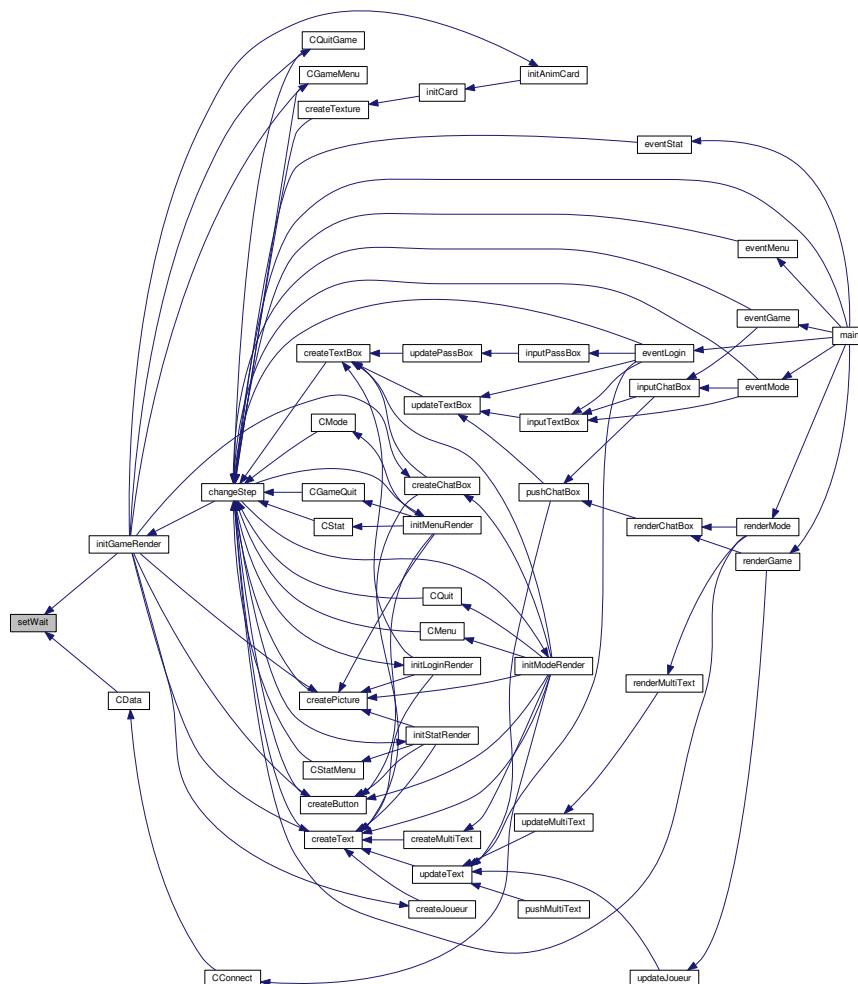
fait attendre le thread client pour la réception d'une [DataGame](#)

Parameters

<i>true</i>	si peut réceptionner, false sinon
-------------	-----------------------------------

Definition at line 288 of file client.c.

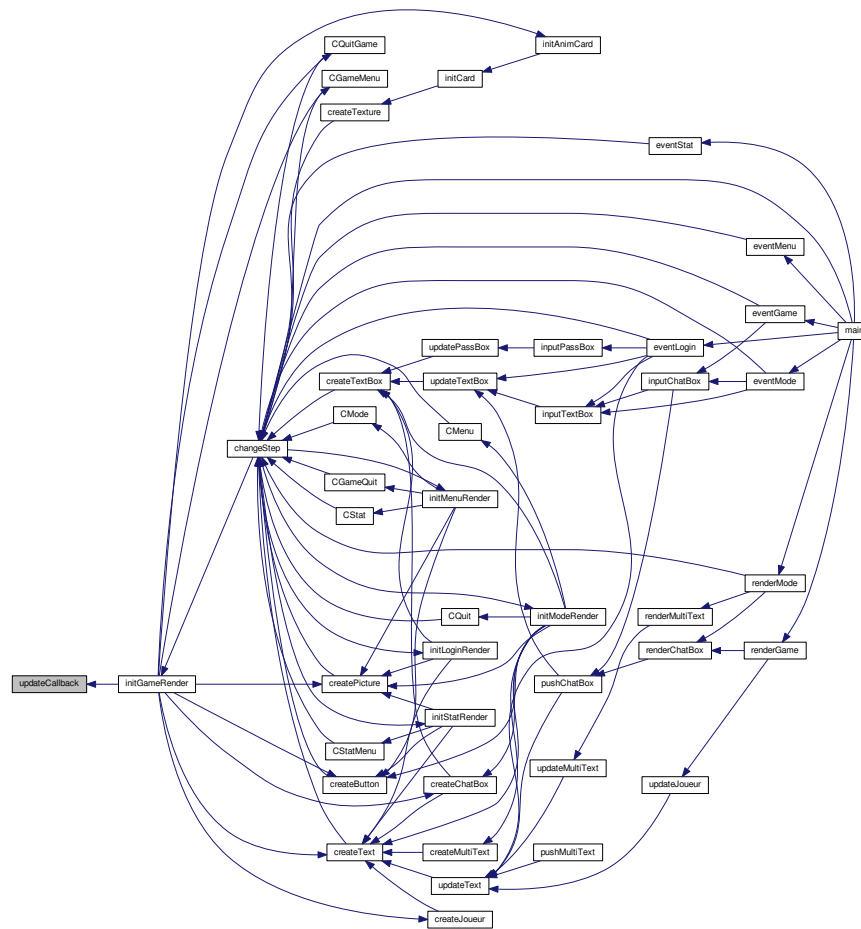
Here is the caller graph for this function:



4.33.2.14 void startGame ()

demande à commencer une partie réponse du serveur via callback
Definition at line 218 of file client.c.

Here is the caller graph for this function:



4.33.3 Variable Documentation

4.33.3.1 bool isQuitting

Definition at line 23 of file client.c.

4.34 src/Network/client.h File Reference

réseau

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <SDL2/SDL_net.h>
#include <SDL2/SDL_thread.h>
#include <SDL2/SDL_timer.h>
#include "network.h"
```


- void `setWait` (bool arg0)
fait attendre le thread client pour la réception d'une `DataGame`
- void `identifySrv` (char *name)
s'identifier sur le serveur note: seulement une fois dans une salle de jeu

4.34.1 Detailed Description

réseau
 Author

Aubin Detrez

Permet de se connecter au serveur de jeu, lancer un thread d'écoute et recevoir les données du serveur pas l'intermédiaire de callback

4.34.2 Macro Definition Documentation

4.34.2.1 `#define PORT 2000`

Definition at line 20 of file client.h.

4.34.3 Function Documentation

4.34.3.1 void `askList` ()

demande au serveur de transmettre la liste des salles non bloquante et la réponse du serveur sera transmise au callback associé à son type (par le thread de réception)
 Definition at line 174 of file client.c.

4.34.3.2 void `choice` (int *choice*)

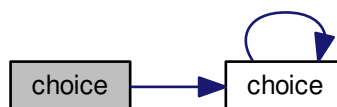
envoie le choix du joueur au serveur réponse du serveur via callback

Parameters

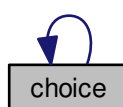
<i>le</i>	numéro de la carte choisit ([1;104])
-----------	--------------------------------------

Definition at line 231 of file client.c.

Here is the call graph for this function:



Here is the caller graph for this function:



4.34.3.3 void create (char * *str*, int *passwd*)

demande à créer une salle de jeu réponse du serveur via callback

Parameters

<i>str</i>	chaîne de caractère pour le nom de la salle
<i>passwd</i>	mot de passe (-1 si non utilisé)

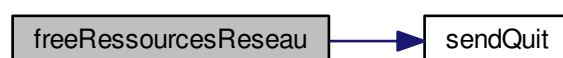
Definition at line 203 of file client.c.

4.34.3.4 void freeRessourcesReseau ()

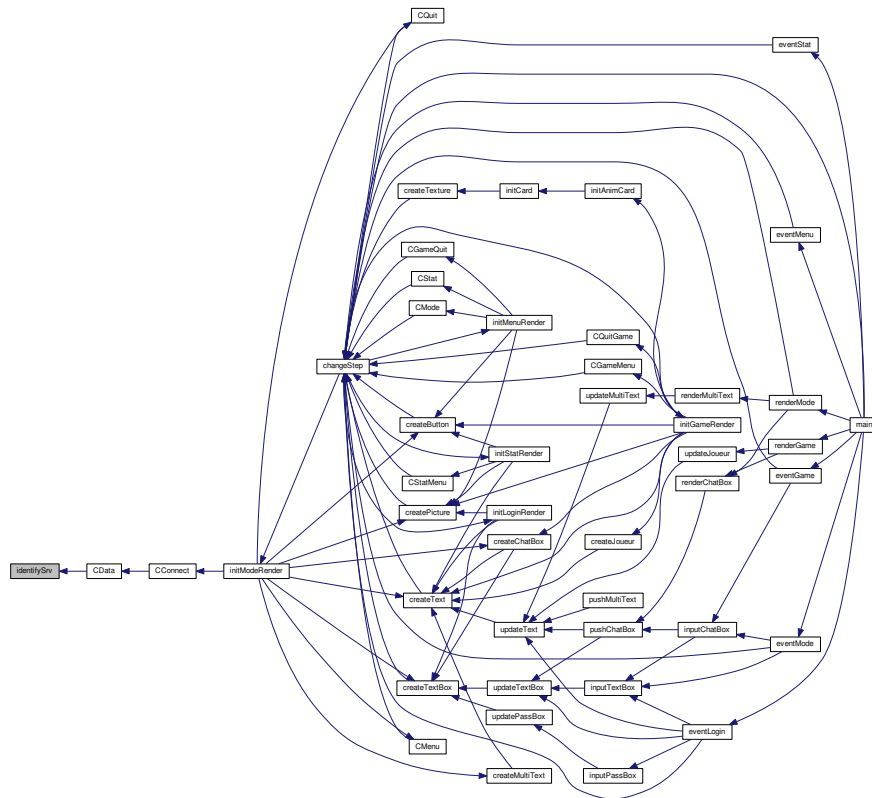
libération des ressources si elles sont initialisées

Definition at line 264 of file client.c.

Here is the call graph for this function:



Here is the caller graph for this function:



4.34.3.6 void initialisationReseau (char * *strip*, void(*)(**Data** *) *backData*, void(*)(**DataList** *) *backList*, void(*)(**DataGame** *) *backGame*)

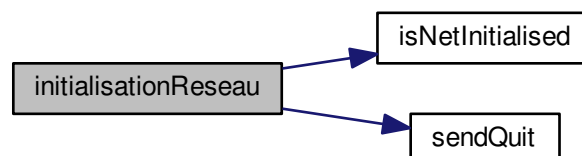
initialisation du réseau initialise la connexion avec le serveur

Parameters

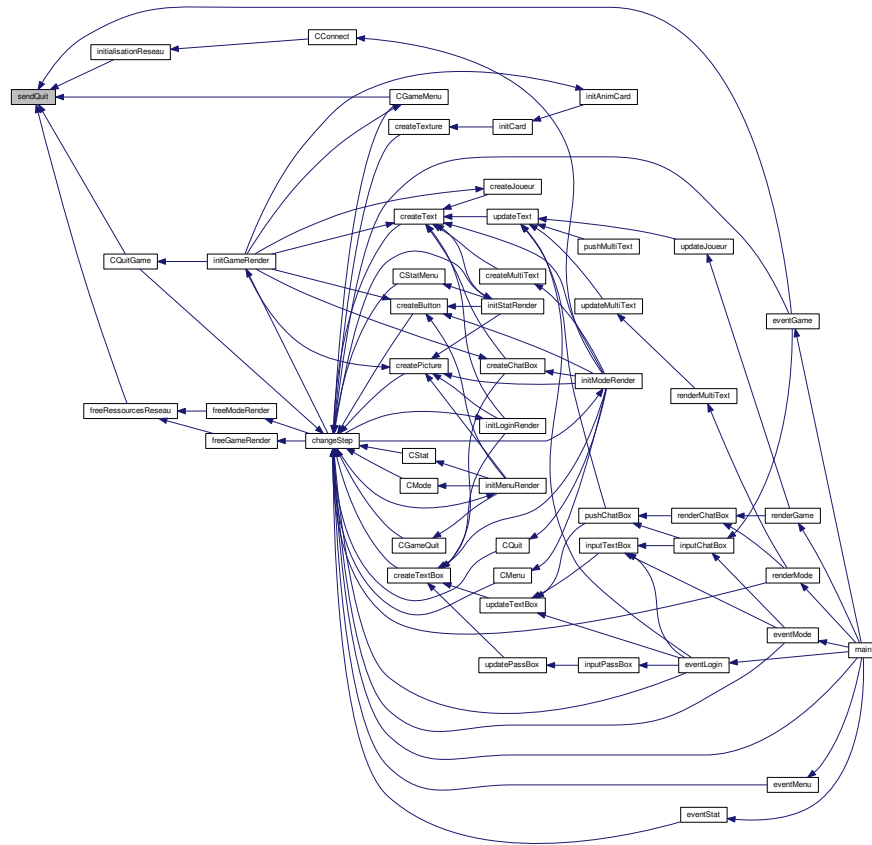
<i>strip</i>	chaîne de caractère correspondant à l'ip du serveur (ex: "127.0.0.1")
<i>backData</i>	callback pour la réception de données de type Data
<i>backList</i>	callback pour la réception de données de type DataList
<i>backGame</i>	callback pour la réception de données de type DataGame

Definition at line 32 of file client.c.

Here is the call graph for this function:



Here is the caller graph for this function:



4.34.3.13 void setWait (bool *arg0*)

fait attendre le thread client pour la réception d'une [DataGame](#)

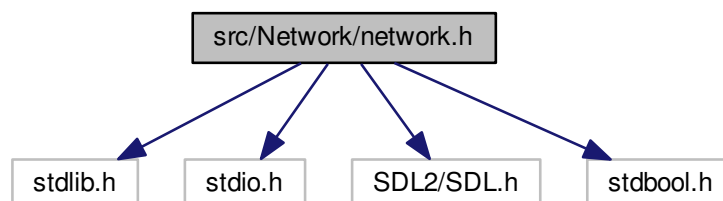
Parameters

<i>true</i>	si peut réceptionner, false sinon
-------------	-----------------------------------

Definition at line 288 of file client.c.

Fichier correspondant au structures et types de donnée pour le reseau doit être le même sur le serveur et le client (note: sur le serveur c'est inc.hpp)

Include dependency graph for network.h:



4.35.2.2 #define BUF_SIZE 100

Definition at line 15 of file network.h.

4.35.2.3 #define CONN_CREATE 6

Definition at line 48 of file network.h.

4.35.2.4 #define CONN_ERROR 0

Definition at line 23 of file network.h.

4.35.2.5 #define CONN_FULL 1

Definition at line 26 of file network.h.

4.35.2.6 #define CONN_JOIN 5

Definition at line 42 of file network.h.

4.35.2.7 #define CONN_LIST 4

Definition at line 35 of file network.h.

4.35.2.8 #define CONN_OK 3

Definition at line 32 of file network.h.

4.35.2.9 #define CONN_QUIT 15

Definition at line 86 of file network.h.

4.35.2.10 #define CONN_STOP 2

Definition at line 29 of file network.h.

4.35.2.11 #define END_GAME 13

Definition at line 80 of file network.h.

4.35.2.12 #define END_TURN 12

Definition at line 75 of file network.h.

4.35.2.13 #define GAME_CHOICE 11

Definition at line 70 of file network.h.

4.35.2.14 #define GAME_ERROR 9

Definition at line 59 of file network.h.

4.35.2.15 #define GAME_LOGIN 8

Definition at line 56 of file network.h.

4.35.2.16 #define GAME_OK 7

Definition at line 53 of file network.h.

4.35.2.17 `#define GAME_START 10`

Definition at line 65 of file network.h.

4.35.2.18 `#define MAXCONN 100`

Definition at line 17 of file network.h.

4.35.2.19 `#define SWITCH 14`

Definition at line 83 of file network.h.

4.35.3 Enumeration Type Documentation

4.35.3.1 enum `DataType`

les types de paquet selon leurs fonction correspond au champ `dataType` de la structure [Data](#)

Les types de donnée

Enumerator

CONN pour la connexion
MSG pour le chat (message)
GAME pour le jeu
CONN
MSG
GAME

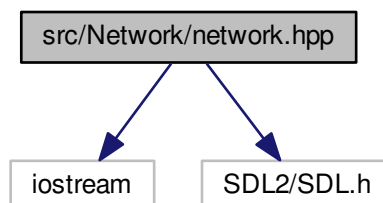
Definition at line 95 of file network.h.

4.36 `src/Network/network.hpp` File Reference

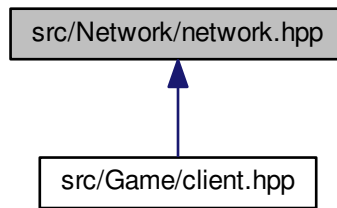
```
#include <iostream>
```

```
#include <SDL2/SDL.h>
```

Include dependency graph for `network.hpp`:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct [__attribute__](#)
- struct [__attribute__](#)
- struct [__attribute__](#)

Macros

- #define [BUF_SIZE](#) 100
- #define [BUF_LIST](#) 640
- #define [MAXCONN](#) 100
- #define [CONN_ERROR](#) 0
- #define [CONN_FULL](#) 1
- #define [CONN_STOP](#) 2
- #define [CONN_OK](#) 3
- #define [CONN_LIST](#) 4
- #define [CONN_JOIN](#) 5
- #define [CONN_CREATE](#) 6
- #define [GAME_OK](#) 7
- #define [GAME_LOGIN](#) 8
- #define [GAME_ERROR](#) 9
- #define [GAME_START](#) 10
- #define [GAME_CHOICE](#) 11
- #define [END_TURN](#) 12
- #define [END_GAME](#) 13

Enumerations

- enum [DataType](#) {
[CONN](#) =0, [MSG](#) =1, [GAME](#) =2, [CONN](#) =0,
[MSG](#) =1, [GAME](#) =2 }

4.36.1 Macro Definition Documentation

4.36.1.1 #define BUF_LIST 640

Definition at line 7 of file network.hpp.

4.36.1.2 #define BUF_SIZE 100

Definition at line 6 of file network.hpp.

4.36.1.3 #define CONN_CREATE 6

Definition at line 39 of file network.hpp.

4.36.1.4 `#define` `CONN_ERROR` 0

Les valeurs possibles de `[Data].car` selon le `[Data].dataType`
Definition at line 14 of file `network.hpp`.

4.36.1.5 `#define` `CONN_FULL` 1

Definition at line 17 of file `network.hpp`.

4.36.1.6 `#define` `CONN_JOIN` 5

Definition at line 33 of file `network.hpp`.

4.36.1.7 `#define` `CONN_LIST` 4

Definition at line 26 of file `network.hpp`.

4.36.1.8 `#define` `CONN_OK` 3

Definition at line 23 of file `network.hpp`.

4.36.1.9 `#define` `CONN_STOP` 2

Definition at line 20 of file `network.hpp`.

4.36.1.10 `#define` `END_GAME` 13

Definition at line 71 of file `network.hpp`.

4.36.1.11 `#define` `END_TURN` 12

Definition at line 66 of file `network.hpp`.

4.36.1.12 `#define` `GAME_CHOICE` 11

Definition at line 61 of file `network.hpp`.

4.36.1.13 `#define` `GAME_ERROR` 9

Definition at line 50 of file `network.hpp`.

4.36.1.14 `#define` `GAME_LOGIN` 8

Definition at line 47 of file `network.hpp`.

4.36.1.15 `#define` `GAME_OK` 7

Definition at line 44 of file `network.hpp`.

4.36.1.16 `#define` `GAME_START` 10

Definition at line 56 of file `network.hpp`.

4.36.1.17 `#define` `MAXCONN` 100

Definition at line 8 of file `network.hpp`.

4.36.2 Enumeration Type Documentation

4.36.2.1 enum `DataType`

Les types de donnée

Enumerator

CONN	pour la connection
MSG	pour le chat (message)
GAME	pour le jeu
CONN	
MSG	
GAME	

Definition at line 75 of file network.hpp.

