Super Martin

Generated by Doxygen 1.8.6

Tue Mar 18 2014 18:00:51

Contents

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

Data Structure Documentation

3.1 Character Struct Reference

Data Fields

- SDL_Surface * **spriteR**
- SDL_Surface * **spriteL**
- SDL_Rect **location**
- int **isRight**
- int isOnGround
- int isJumping

3.1.1 Field Documentation

3.1.1.1 int isJumping

indique si le perso est au sol

3.1.1.2 int isOnGround

indique la direction de regard du personnage (1 droite, 0 gauche)

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

- player.h

3.2 Level Struct Reference

Data Fields

- unsigned char ** **map**
- int **width**
- int **height**
- int **timer_level**
- char **background** [TAILLE_MAX_NOM_FICHIER]
- char **music** [TAILLE_MAX_NOM_FICHIER]

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

- const.h

3.3 Map Struct Reference

Collaboration diagram for Map:

Data Fields

- Level * **lvl**
- int **xScroll**
- int **screenWidth**
- int **screenHeight**

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

- const.h

3.4 Sound Struct Reference

Data Fields

- FMOD_SYSTEM * **sys**
- FMOD_SOUND * **sound**
- FMOD_CHANNELGROUP * **channel**

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

- sound.h

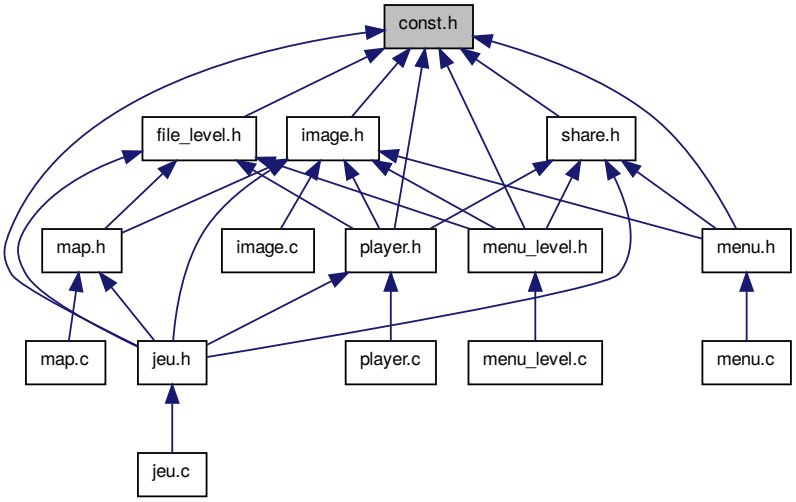
Chapter 4

File Documentation

4.1 const.h File Reference

contient les constantes du programme

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



Data Structures

- struct Level
- struct Map

Macros

- #define **TAILLE_BLOC** 16
- #define **NB_BLOCS_LARGEUR** 60
- #define **NB_BLOCS_HAUTEUR** 34
- #define **LARGEUR_FENETRE** TAILLE_BLOC ∗ NB_BLOCS_LARGEUR
- #define **HAUTEUR_FENETRE** TAILLE_BLOC ∗ NB_BLOCS_HAUTEUR

- #define **FPS** 60
- #define **TAILLE_MAX_NOM_FICHIER** 100
- #define **TAILLE_SAUT** 17
- #define **MARGE_SCROLLING** 2
- #define **POURCENTAGE_DEPLACEMENT** 20
- #define **TILE_MAX** 8

Enumerations

- enum { **VOID** =0, **GRASS1** =1, **GROUND1** =2, **GREY_WALL** =3 }
- enum { **RIGHT**, **LEFT**, **UP**, **DOWN** }

4.1.1 Detailed Description

contient les constantes du programme

Author

Xavier COPONET

Date

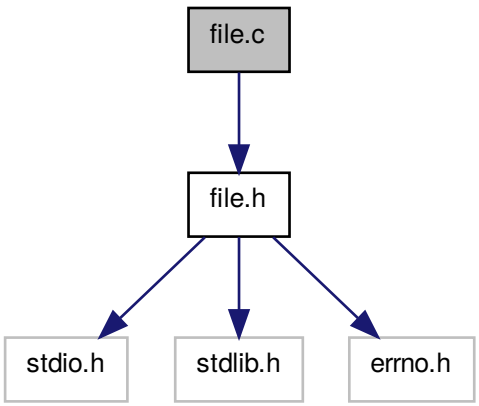
2014-02-27

4.2 file.c File Reference

Fonctions d'acces au fichiers.

```
#include "file.h"
```

Include dependency graph for file.c:



Functions

- FILE * openFile (char nom[], char mode[])
- int closeFile (FILE *ptr_fichier)
- int readFileSize (FILE *ptr_fichier)

4.2.1 Detailed Description

Fonctions d'acces au fichiers.

Author

Remi BERTHO

Date

15/03/14

4.2.2 Function Documentation

4.2.2.1 int closeFile (FILE * *ptr_fichier*)

Ferme le fichier

Parameters

in	*ptr_fichier	le fichier

Returns

entier 0 si tout s'est bien passe, 1 sinon

4.2.2.2 FILE * openFile (char *nom[]*, char *mode[]*)

Ouvre un fichier a partir de son nom (nom[]) et du mode voulu (mode[])

Parameters

in	*nom[]*	le nom du fichier
in	*mode[]*	le mode voulu

Returns

un pointeur sur le fichier ouvert, NULL s'il y a eut un probleme

4.2.2.3 int readFileSize (FILE * *ptr_fichier*)

Lis la taille du fichier

Parameters

in	*ptr_fichier	le fichier

Returns

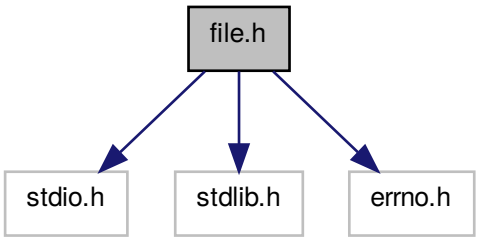
entier ayant la taille du fichier

4.3 file.h File Reference

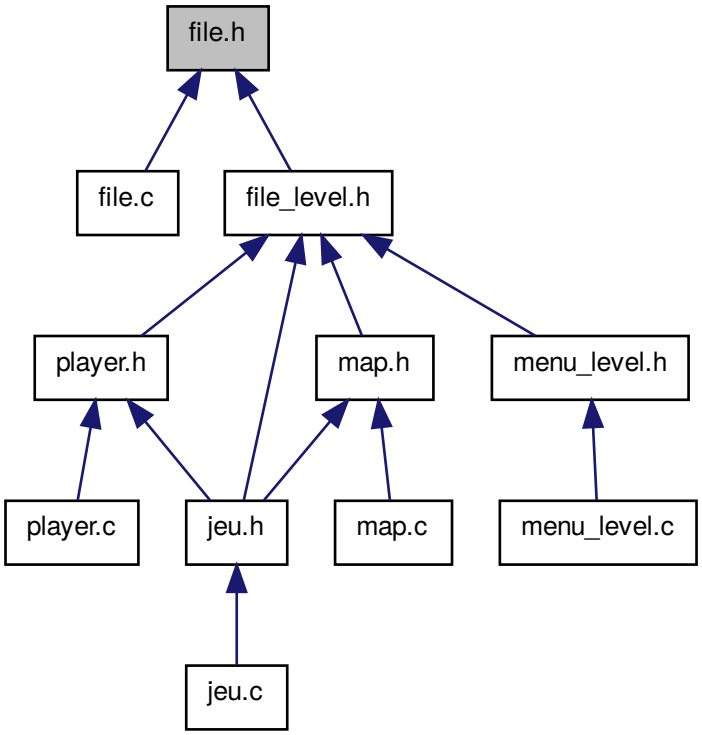
Prototypes des fonctions d'acces aux fichiers.

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
```

Include dependency graph for file.h:



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



Functions

- FILE * openFile (char nome[], char mode[])
- int closeFile (FILE *ptr_fichier)
- int readFileSize (FILE *ptr_fichier)

4.3.1 Detailed Description

Prototypes des fonctions d'acces aux fichiers.

Author

Remi BERTHO

Date

15/03/14

4.3.2 Function Documentation

4.3.2.1 int closeFile (FILE * *ptr_fichier*)

Ferme le fichier

Parameters

in	*ptr_fichier	le fichier

Returns

entier 0 si tout s'est bien passe, 1 sinon

4.3.2.2 FILE* openFile (char *nom[]*, char *mode[]*)

Ouvre un fichier a partir de son nom (nom[]) et du mode voulu (mode[])

Parameters

in	*nom[]*	le nom du fichier
in	*mode[]*	le mode voulu

Returns

un pointeur sur le fichier ouvert, NULL s'il y a eut un probleme

4.3.2.3 int readFileSize (FILE * *ptr_fichier*)

Lis la taille du fichier

Parameters

in	*ptr_fichier	le fichier

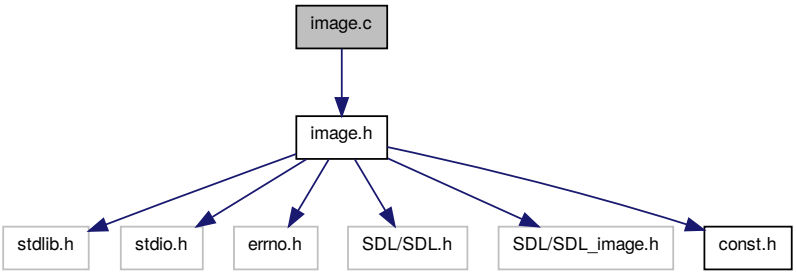
Returns

entier ayant la taille du fichier

4.4 image.c File Reference

contient les fonction liées aux images

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



Functions

- SDL_Surface ∗ imageLoad (char ∗file_name)
- SDL_Surface ∗ imageLoadAlpha (char ∗file_name)

4.4.1 Detailed Description

contient les fonction liées aux images

Author

Rémi BERTHO

Date

2014-02-27

4.4.2 Function Documentation

4.4.2.1 SDL_Surface ∗ imageLoad (char ∗ *file_name*)

Charge une image

Parameters

in	*file_name*	le nom du fichier

Returns

un pointeur sur une SDL_Surface

4.4.2.2 SDL_Surface ∗ imageLoadAlpha (char ∗ *file_name*)

Charge une image

Parameters

in	*file_name*	le nom du fichier

Returns

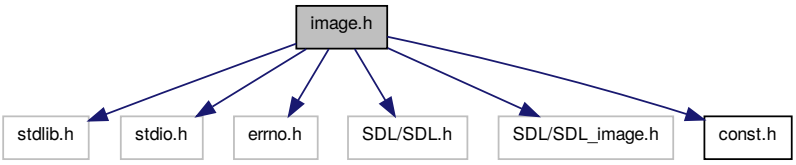
un pointeur sur une SDL_Surface

4.5 image.h File Reference

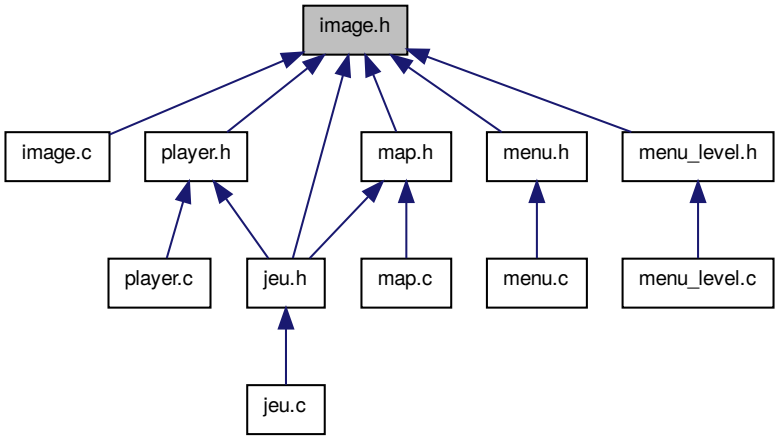
contient les fonction liées aux images

```
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <SDL/SDL.h>
#include <SDL/SDL_image.h>
#include "const.h"
```

Include dependency graph for image.h:



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



Functions

- SDL_Surface ∗ imageLoad (char ∗file_name)

- SDL_Surface ∗ imageLoadAlpha (char ∗file_name)

4.5.1 Detailed Description

contient les fonction liées aux images

Author

Rémi BERTHO

Date

2014-02-27

4.5.2 Function Documentation

4.5.2.1 SDL_Surface∗ imageLoad (char ∗ *file_name*)

Charge une image

Parameters

in	*file_name*	le nom du fichier

Returns

un pointeur sur une SDL_Surface

4.5.2.2 SDL_Surface∗ imageLoadAlpha (char ∗ *file_name*)

Charge une image

Parameters

in	*file_name*	le nom du fichier

Returns

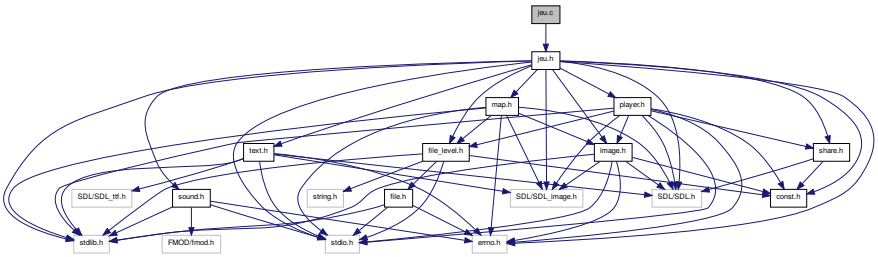
un pointeur sur une SDL_Surface

4.6 jeu.c File Reference

contient les fonction liées au jeu

```
#include "jeu.h"
```

Include dependency graph for jeu.c:



Functions

- void jouer (SDL_Surface ∗screen, char ∗level_name)
- void printGameOver (SDL_Surface ∗screen, int ∗continuer)
- void **move** (int move_left, int move_right, Character ∗player, Map ∗m, float speed, int ∗acceleration)
- void updateSpeed (float ∗speed, int acceleration)
- void printPause (SDL_Surface ∗screen, SDL_Event ∗event, int ∗time)
- Uint32 **decomptage** (Uint32 intervalle, void ∗parametre)

4.6.1 Detailed Description

contient les fonction liées au jeu

Author

Xavier COPONET

Date

2014-02-27

4.6.2 Function Documentation

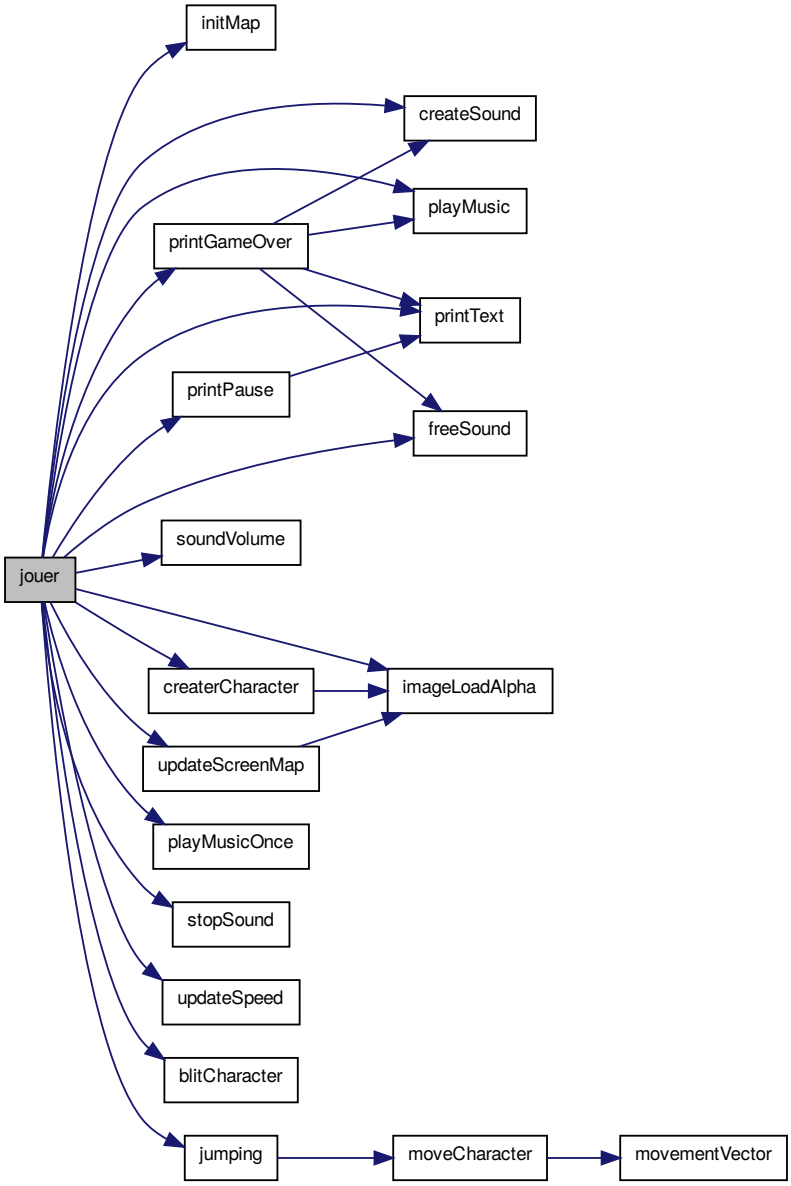
4.6.2.1 void jouer (SDL_Surface ∗ *screen*, char ∗ *level_name*)

contient la boucle principale du jeu qui appelle les fonctions

Parameters

in,out	*screen*	L'écran de jeu
in	*lvel_name*	le nom du niveau

Here is the call graph for this function:



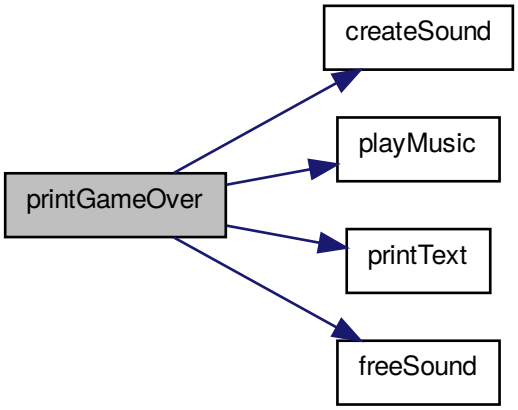
4.6.2.2 void printGameOver (SDL_Surface * *screen*, int * *continuer*)

affiche le message de game overflow_error

Parameters

out	*screen*	l'écran de jeu

Here is the call graph for this function:



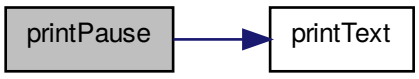
4.6.2.3 void printPause (SDL_Surface * *screen*, SDL_Event * *event*, int * *time*)

Met en pause le jeu

Parameters

out	*screen*	l'écran de jeu
out	*time*	le temps restant
out	*event*	l'evenement en cours

Here is the call graph for this function:



4.6.2.4 void updateSpeed (float * *speed*, int *acceleration*)

Met a jour la vitesse

Parameters

out	*float*	la vitesse

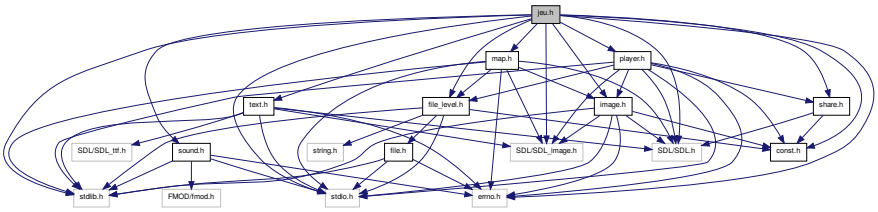
out	*acceleration*	l'acceleration

4.7 jeu.h File Reference

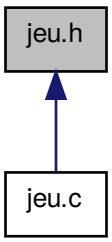
header de jeu.c

```
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <SDL/SDL.h>
#include <SDL/SDL_image.h>
#include "const.h"
#include "text.h"
#include "sound.h"
#include "share.h"
#include "player.h"
#include "file_level.h"
#include "image.h"
#include "map.h"
```

Include dependency graph for jeu.h:



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



Functions

- void jouer (SDL_Surface ∗screen, char ∗level_name)
- void printGameOver (SDL_Surface ∗screen, int ∗continuer)
- void **move** (int move_left, int move_right, Character ∗player, Map ∗m, float speed, int ∗acceleration)

- void updateSpeed (float ∗speed, int acceleration)
- void printPause (SDL_Surface ∗screen, SDL_Event ∗event, int ∗time)
- Uint32 **decomptage** (Uint32 intervalle, void ∗parametre)

4.7.1 Detailed Description

header de jeu.c

Author

Xavier COPONET

Date

2014-02-27

4.7.2 Function Documentation

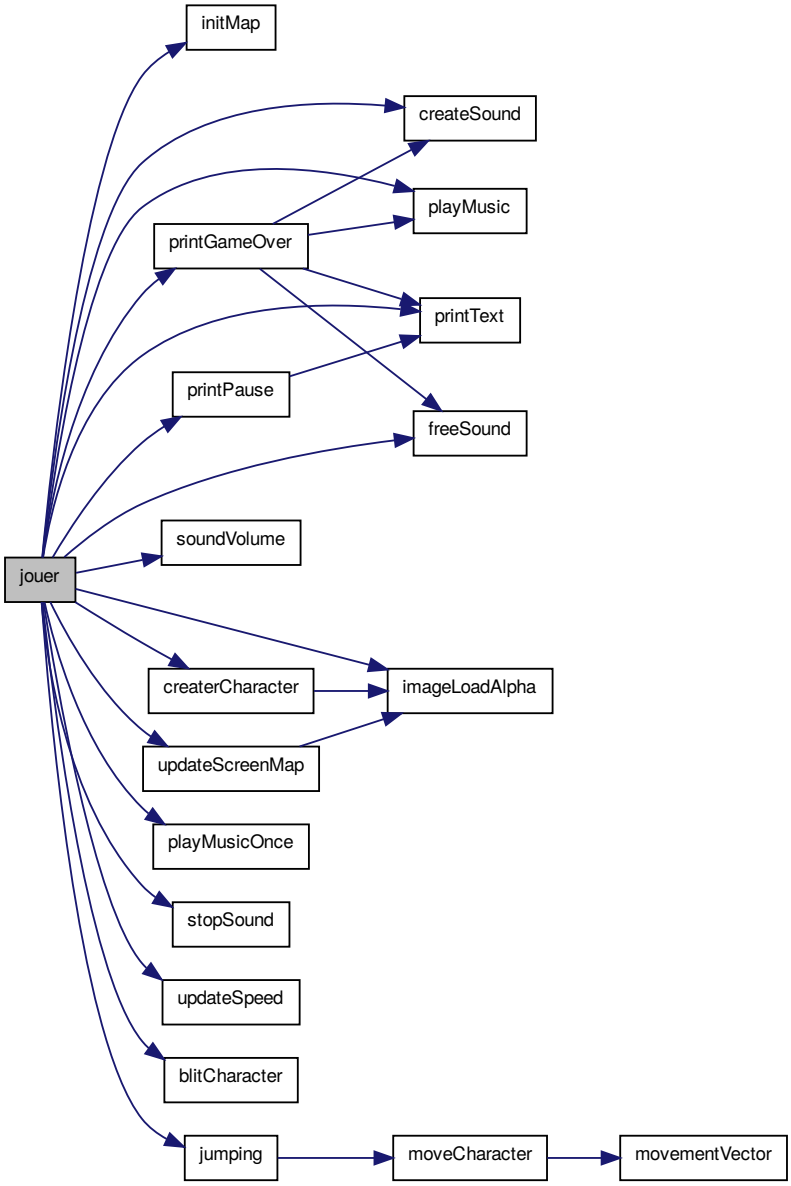
4.7.2.1 void jouer (SDL_Surface ∗ *screen*, char ∗ *level_name*)

contient la boucle principale du jeu qui appelle les fonctions

Parameters

in,out	*screen*	L'écran de jeu
in	*lvel_name*	le nom du niveau

Here is the call graph for this function:



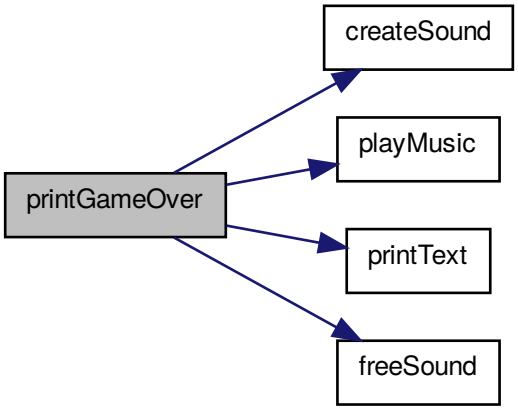
4.7.2.2 void printGameOver (SDL_Surface * *screen*, int * *continuer*)

affiche le message de game overflow_error

Parameters

out	*screen*	l'écran de jeu

Here is the call graph for this function:



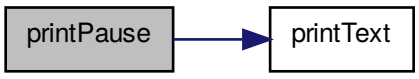
4.7.2.3 void printPause (SDL_Surface * *screen*, SDL_Event * *event*, int * *time*)

Met en pause le jeu

Parameters

out	*screen*	l'écran de jeu
out	*time*	le temps restant
out	*event*	l'evenement en cours

Here is the call graph for this function:



4.7.2.4 void updateSpeed (float * *speed*, int *acceleration*)

Met a jour la vitesse

Parameters

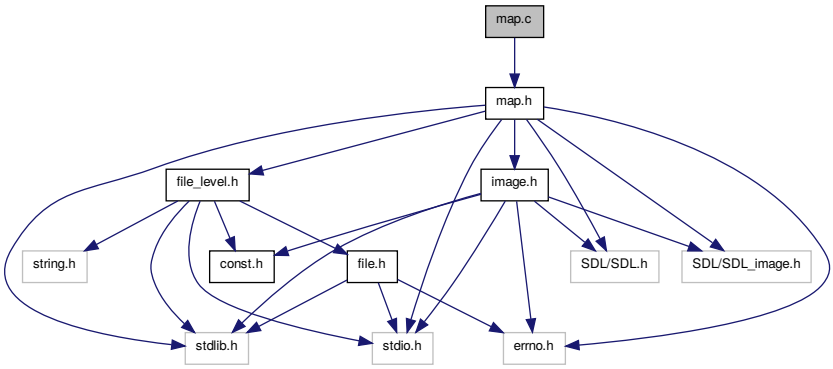
out	*float*	la vitesse

out	*acceleration*	l'acceleration

4.8 map.c File Reference

contient les fonction liées au chargement et à l'affichage de la carte

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



Functions

- void updateScreenMap (SDL_Surface ∗screen, Map ∗m, char ∗tileset)
- void scrolling (Map ∗m, int direction, float speed)
- Map ∗ initMap (SDL_Surface ∗screen, char ∗level_name)
- void **freeMap** (Map ∗m)

4.8.1 Detailed Description

contient les fonction liées au chargement et à l'affichage de la carte

Author

Xavier COPONET

Date

2014-03-18

4.8.2 Function Documentation

4.8.2.1 Map ∗ initMap (SDL_Surface ∗ *screen,* char ∗ *level_name*)

initialise la carte

Parameters

in	*screen*	l'écran de jeu
in	*level_name*	le nom du niveau

Returns

un pointeur sur la carte initialisée

4.8.2.2 void scrolling (Map * *m*, int *direction*, float *speed*)

effectue un scrolling

Parameters

in,out	*map*	Le niveau à gérer
in	*direction*	La direction de scrolling
in	*speed*	la vitesse de scrolling

4.8.2.3 void updateScreenMap (SDL_Surface * *screen*, Map * *m*, char * *tileset*)

met à jour l'écran avec les données de la carte (ignore les personnages)

Parameters

in,out	*screen*	of the game
in	*Map*	*m The map
in	*tileset*	lvl tileset

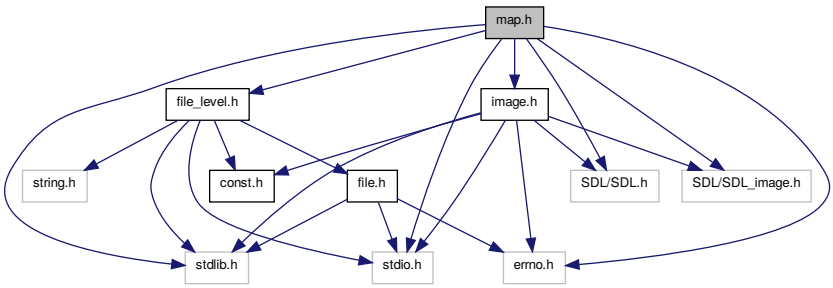
Here is the call graph for this function:

4.9 map.h File Reference

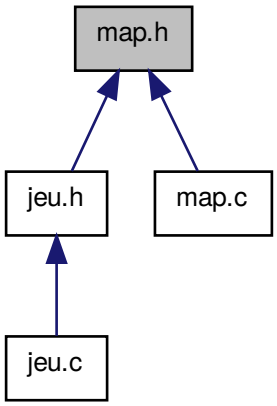
header de map.c

```
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <SDL/SDL.h>
#include <SDL/SDL_image.h>
#include "image.h"
#include "file_level.h"
```

Include dependency graph for map.h:



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



Functions

- void updateScreenMap (SDL_Surface ∗screen, Map ∗m, char ∗tileset)
- void scrolling (Map ∗m, int direction, float speed)
- Map ∗ initMap (SDL_Surface ∗screen, char ∗level_name)
- void **freeMap** (Map ∗m)

4.9.1 Detailed Description

header de map.c

Author

Xavier COPONET

Date

2014-03-18

4.9.2 Function Documentation

4.9.2.1 Map* initMap (SDL_Surface * *screen*, char * *level_name*)

initialise la carte

Parameters

in	*screen*	l'écran de jeu
in	*level_name*	le nom du niveau

Returns

un pointeur sur la carte initialisée

4.9.2.2 void scrolling (Map * *m*, int *direction*, float *speed*)

effectue un scrolling

Parameters

in,out	*map*	Le niveau à gérer
in	*direction*	La direction de scrolling
in	*speed*	la vitesse de scrolling

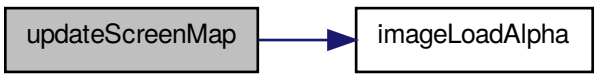
4.9.2.3 void updateScreenMap (SDL_Surface * *screen*, Map * *m*, char * *tileset*)

met à jour l'écran avec les données de la carte (ignore les personnages)

Parameters

in,out	*screen*	of the game
in	*Map*	*m The map
in	*tileset*	lvl tileset

Here is the call graph for this function:

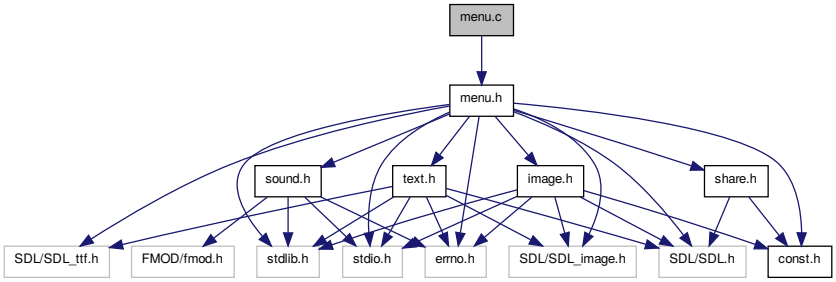


4.10 menu.c File Reference

contient les fonction liées au menu

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

Include dependency graph for menu.c:



Functions

- int **menu** (SDL_Surface *screen, int *continuer, Sound *s)
- Uint32 **blinkText** (Uint32 intervalle, void *parametre)

4.10.1 Detailed Description

contient les fonction liées au menu

Author

Xavier COPONET

Date

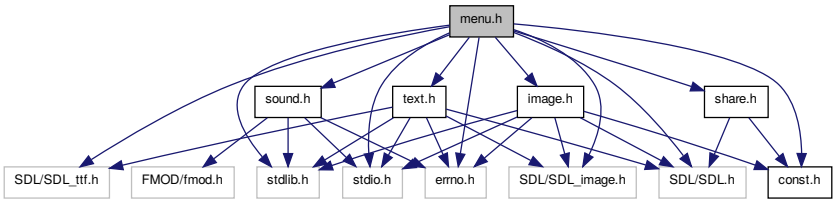
2014-02-27

4.11 menu.h File Reference

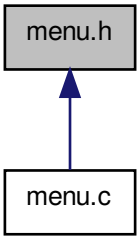
header de menu.c

```
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <SDL/SDL.h>
#include <SDL/SDL_image.h>
#include <SDL/SDL_ttf.h>
#include "const.h"
#include "text.h"
#include "sound.h"
#include "share.h"
#include "image.h"
```

Include dependency graph for menu.h:



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



Functions

- int **menu** (SDL_Surface ∗screen, int ∗continuer, Sound ∗s)
- Uint32 **blinkText** (Uint32 intervalle, void ∗parametre)

4.11.1 Detailed Description

header de menu.c

Author

Xavier COPONET

Date

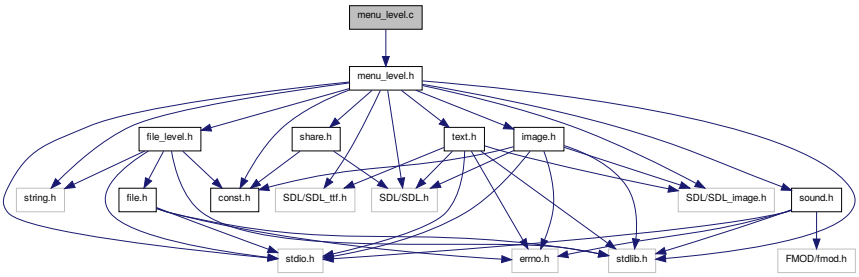
2014-02-27

4.12 menu_level.c File Reference

Menu gerant le choix du niveau.

```
#include "menu_level.h"
```

Include dependency graph for menu_level.c:



Functions

- int menuLevel (SDL_Surface ∗screen, char level_name[TAILLE_MAX_NOM_FICHIER], Sound ∗s)

4.12.1 Detailed Description

Menu gerant le choix du niveau.

Author

Remi BERTHO

Date

15/03/14

Version

1.0

4.12.2 Function Documentation

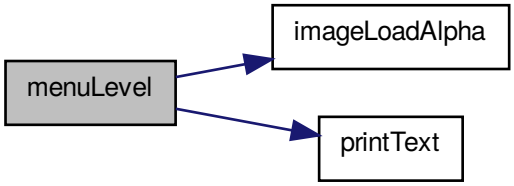
4.12.2.1 int menuLevel (SDL_Surface ∗ *screen,* char *level_name[TAILLE_MAX_NOM_FICHIER],* Sound ∗ *s*)

Menu pour choisr le niveau

Parameters

out	*screen*	l'écran de jeu
out	*level_name*	le nom du level que l'on va vouloir lancer
in	*s*	la musique de fond

Here is the call graph for this function:

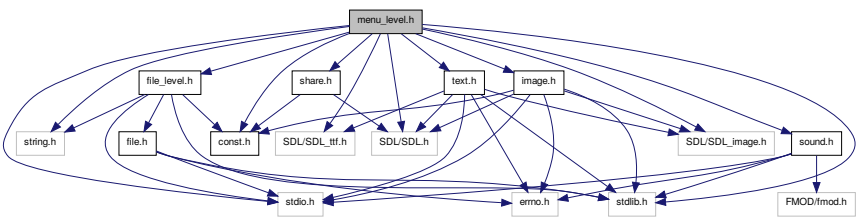


4.13 menu_level.h File Reference

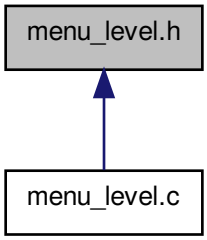
Menu gerant le choix du niveau.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <SDL/SDL.h>
#include <SDL/SDL_image.h>
#include <SDL/SDL_ttf.h>
#include "const.h"
#include "file_level.h"
#include "share.h"
#include "text.h"
#include "sound.h"
#include "image.h"
```

Include dependency graph for menu_level.h:



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



Functions

- int menuLevel (SDL_Surface ∗screen, char level_name[TAILLE_MAX_NOM_FICHIER], Sound ∗s)

4.13.1 Detailed Description

Menu gerant le choix du niveau.

Author

Remi BERTHO

Date

15/03/14

Version

1.0

4.13.2 Function Documentation

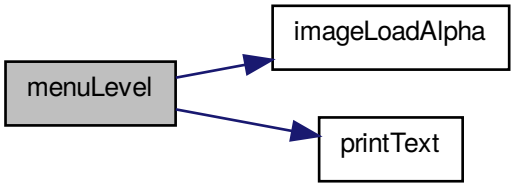
4.13.2.1 int menuLevel (SDL_Surface ∗ *screen*, char *level_name[TAILLE_MAX_NOM_FICHIER]*, Sound ∗ *s*)

Menu pour choisr le niveau

Parameters

out	*screen*	l'écran de jeu
out	*level_name*	le nom du level que l'on va vouloir lancer
in	*s*	la musique de fond

Here is the call graph for this function:

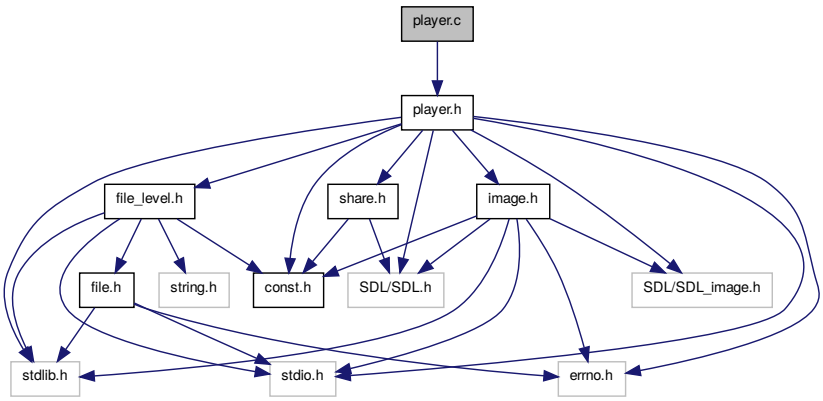


4.14 player.c File Reference

contient les fonction pour manipuler le joueur

```
#include "player.h"
```

Include dependency graph for player.c:



Functions

- Character ∗ createrCharacter (char ∗spR, char ∗spL)
- int moveCharacter (Character ∗c, int direction, Map ∗m, float speed)
- int **tryMovement** (Character ∗c, int vx, int vy, Map ∗m)
- void movementVector (int direction, int ∗vx, int ∗vy, int speed, Character ∗c)
- void blitCharacter (SDL_Surface ∗screen, Character ∗c, Map ∗m)
- int collisionSprite (SDL_Rect r, Map ∗m)
- void **gravity** (Character ∗c, Map ∗m, SDL_Surface ∗screen)
- void **presiseMoveCharacter** (Character ∗c, int vx, int vy, Map ∗m)
- void jumping (Character ∗c, Map ∗m)

4.14.1 Detailed Description

contient les fonction pour manipuler le joueur

Author

Xavier COPONET

Date

2014-02-27

4.14.2 Function Documentation

4.14.2.1 void blitCharacter (SDL_Surface * *screen*, Character * *c*, Map * *m*)

blit le personnage à l'écran

Parameters

in,out	*screen*	L'écran
in	*c*	Le personnage
in	*m*	la carte du jeu

4.14.2.2 int collisionSprite (SDL_Rect *r*, Map * *m*)

détermine s'il y a collision entre une sprite et le décor

Parameters

in	*r*	le SDL_Rect correspondant à la sprite
in	*m*	la carte contenant le décor

Returns

1 s'il y a collision ou si en dehors du monde, 0 sinon

4.14.2.3 Character * createrCharacter (char * *spR*, char * *spL*)

créer un personnage

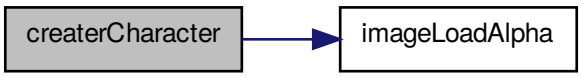
Parameters

in	*spR*	l'adresse de la sprite droite
in	*spL*	l'adresse de la sprite gauche

Returns

le pointeur sur la structure créée

Here is the call graph for this function:



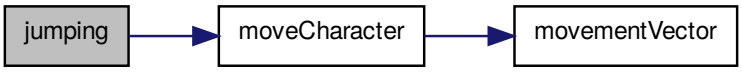
4.14.2.4 void jumping (Character * *c*, Map * *m*)

make the character jump

Parameters

in,out	*c*	the Character
in	*m*	The map the Character is on

Here is the call graph for this function:



4.14.2.5 void moveCharacter (Character * *c*, int *direction*, Map * *m*, float *speed*)

déplace le personnage selon la direction

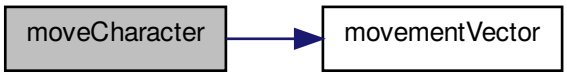
Parameters

in,out	*c*	Le personnage
in	*direction*	La direction du déplacement
in	*m*	la carte sur laquelle le personnage se déplace
in	*speed*	la vitesse de déplacement

Returns

1 si le personnage a pu se deplacer normalement, 0 s'il a fallut affiner

Here is the call graph for this function:



4.14.2.6 void movementVector (int *direction*, int * *vx*, int * *vy*, int *speed*, Character * *c*)

create a movement vector

Parameters

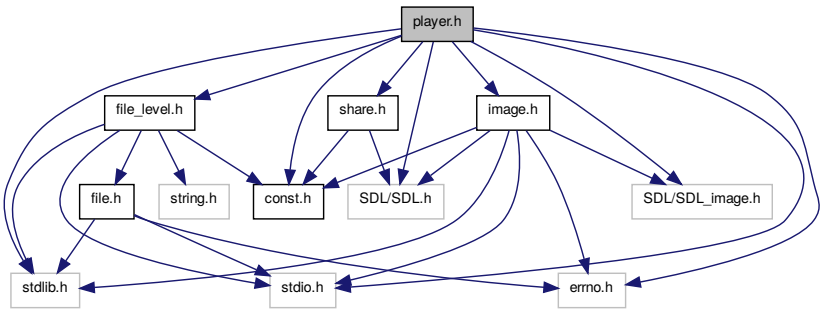
in	*direction*	The movement's direction
out	*vx*	the horizontal component of the vector
out	*vy*	the vertical component of the vector
in	*speed*	the speed of the move
out	*c*	the Character you have to move

4.15 player.h File Reference

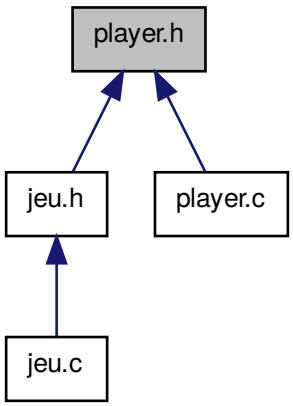
header de player.c

```
#include "const.h"
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <SDL/SDL.h>
#include <SDL/SDL_image.h>
#include "file_level.h"
#include "share.h"
#include "image.h"
```

Include dependency graph for player.h:



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



Data Structures

- struct Character

Macros

- #define SGN(X) (((X)==0)?(0):(((X)<0)?(-1):(1)))
- #define ABS(X) ((((X)<0)?(-(X)):(X)))

Functions

- Character * createrCharacter (char *spR, char *spL)
- int moveCharacter (Character *c, int direction, Map *m, float speed)
- void blitCharacter (SDL_Surface *screen, Character *c, Map *m)
- int collisionSprite (SDL_Rect r, Map *m)

- void **gravity** (Character ∗c, Map ∗m, SDL_Surface ∗screen)
- void movementVector (int direction, int ∗vx, int ∗vy, int speed, Character ∗c)
- int **tryMovement** (Character ∗c, int vx, int vy, Map ∗m)
- void **presiseMoveCharacter** (Character ∗c, int vx, int vy, Map ∗m)
- void jumping (Character ∗c, Map ∗m)

4.15.1 Detailed Description

header de player.c

Author

Xavier COPONET

Date

2014-02-27

4.15.2 Macro Definition Documentation

4.15.2.1 #define ABS(*X*) ((((X)<0)?(-(X)):(X)))

X absolute value

4.15.2.2 #define SGN(*X*) (((X)==0)?(0):(((X)<0)?(-1):(1)))

X sign

4.15.3 Function Documentation

4.15.3.1 void blitCharacter (SDL_Surface ∗ *screen*, **Character** ∗ *c*, **Map** ∗ *m*)

blit le personnage à l'écran

Parameters

in,out	*screen*	L'écran
in	*c*	Le personnage
in	*m*	la carte du jeu

4.15.3.2 int collisionSprite (SDL_Rect *r*, **Map** ∗ *m*)

détermine s'il y a collision entre une sprite et le décor

Parameters

in	*r*	le SDL_Rect correspondant à la sprite
in	*m*	la carte contenant le décor

Returns

1 s'il y a collision ou si en dehors du monde, 0 sinon

4.15.3.3 Character* createrCharacter (char * *spR*, char * *spL*)

créer un personnage

Parameters

in	*spR*	l'adresse de la sprite droite
in	*spL*	l'adresse de la sprite gauche

Returns

le pointeur sur la structure créée

Here is the call graph for this function:

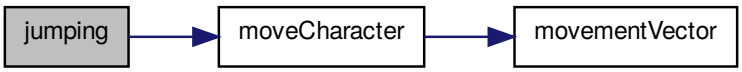
4.15.3.4 void jumping (Character * *c*, Map * *m*)

make the character jump

Parameters

in,out	*c*	the Character
in	*m*	The map the Character is on

Here is the call graph for this function:



4.15.3.5 int moveCharacter (Character * *c*, int *direction*, Map * *m*, float *speed*)

déplace le personnage selon la direction

Parameters

in,out	*c*	Le personnage
in	*direction*	La direction du déplacement
in	*m*	la carte sur laquelle le personnage se déplace

in	*speed*	la vitesse de déplacement

Returns

1 si le personnage a pu se deplacer normalement, 0 s'il a fallut affiner

Here is the call graph for this function:

4.15.3.6 void movementVector (int *direction*, int * *vx*, int * *vy*, int *speed*, Character * *c*)

create a movement vector

Parameters

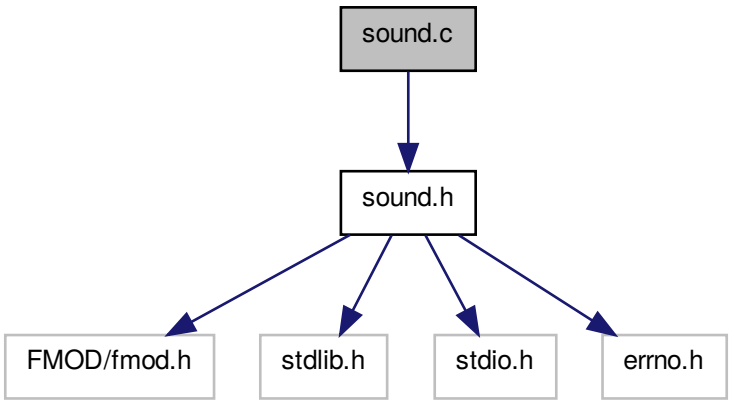
in	*direction*	The movement's direction
out	*vx*	the horizontal component of the vector
out	*vy*	the vertical component of the vector
in	*speed*	the speed of the move
out	*c*	the Character you have to move

4.16 sound.c File Reference

contient les fonction pour jouer du son

```
#include "sound.h"
```

Include dependency graph for sound.c:



Functions

- Sound * createSound (void)
- void playMusic (Sound *s, char *file)
- void playMusicOnce (Sound *s, char *file)
- void freeSound (Sound *s)
- void stopSound (Sound *s)
- void soundVolume (Sound *s, float volume)

4.16.1 Detailed Description

contient les fonction pour jouer du son

Author

Xavier COPONET

Date

2014-02-27

4.16.2 Function Documentation

4.16.2.1 sound * createSound (void)

créer une structure son

Returns

la structure son

4.16.2.2 void freeSound (Sound * s)

release the sound

Parameters

out	*s*	the sound

4.16.2.3 void playMusic (Sound * *s*, char * *file*)

lit un fichier long (musique)

Parameters

in,out	*s*	la structure son que l'on manipule
in	*file*	Le fichier son à lire

4.16.2.4 void playMusicOnce (Sound * *s*, char * *file*)

lit un fichier long une fois

Parameters

in,out	*s*	la structure son que l'on manipule
in	*file*	Le fichier son à lire

4.16.2.5 void soundVolume (Sound * *s*, float *volume*)

set the sound volume

Parameters

out	*s*	the sound
in	*volume*	the sound volume : [0.0 : no sound ; 1.0 (default) max power]

4.16.2.6 void stopSound (Sound * *s*)

stop the sound

Parameters

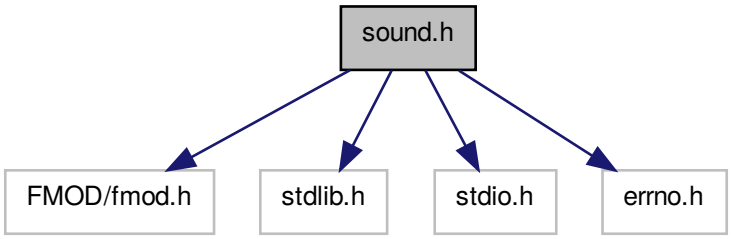
out	*the*	sound to stop

4.17 sound.h File Reference

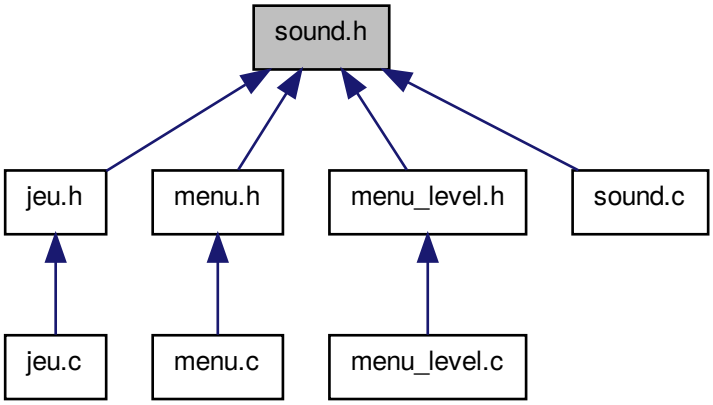
header de sound.c

```
#include <FMOD/fmod.h>
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
```

Include dependency graph for sound.h:



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



Data Structures

- struct Sound

Functions

- Sound * createSound (void)
- void playMusic (Sound *s, char *file)
- void playMusicOnce (Sound *s, char *file)
- void freeSound (Sound *s)
- void stopSound (Sound *s)
- void soundVolume (Sound *s, float volume)

4.17.1 Detailed Description

header de sound.c

Author

Xavier COPONET

Date

2014-02-27

4.17.2 Function Documentation

4.17.2.1 Sound* createSound (void)

créer une structure son

Returns

la structure son

4.17.2.2 void freeSound (Sound * s)

release the sound

Parameters

out	*s*	the sound

4.17.2.3 void playMusic (Sound * s, char * file)

lit un fichier long (musique)

Parameters

in,out	*s*	la structure son que l'on manipule
in	*file*	Le fichier son à lire

4.17.2.4 void playMusicOnce (Sound * s, char * file)

lit un fichier long une fois

Parameters

in,out	*s*	la structure son que l'on manipule
in	*file*	Le fichier son à lire

4.17.2.5 void soundVolume (Sound * s, float volume)

set the sound volume

Parameters

out	*s*	the sound
in	*volume*	the sound volume : [0.0 : no sound ; 1.0 (default) max power]

4.17.2.6 void stopSound (Sound * *s*)

stop the sound

Parameters

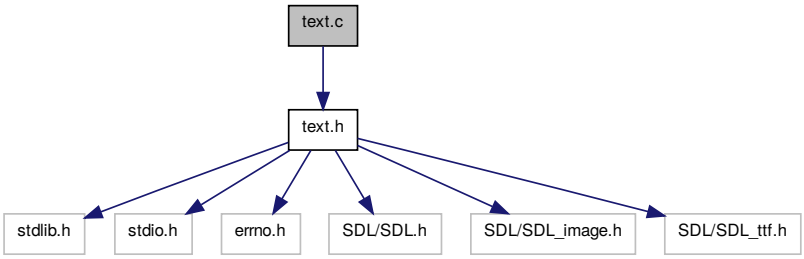
out	*the*	sound to stop

4.18 text.c File Reference

contient les fonction pour afficher du texte à l'écran

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

Include dependency graph for text.c:



Functions

- void printText (SDL_Surface *screen, SDL_Rect *posText, char *text, SDL_Color color, char *font, int ptSize, int mode)

4.18.1 Detailed Description

contient les fonction pour afficher du texte à l'écran

Author

Xavier COPONET

Date

2014-02-27

4.18.2 Function Documentation

4.18.2.1 void printText (SDL_Surface * *screen,* SDL_Rect * *posText,* char * *text,* SDL_Color *color,* char * *font,* int *ptSize,* int *mode*)

affiche le texte sur l'écran à la position donnée

Parameters

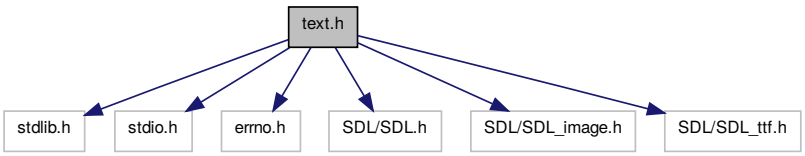
out	*screen*	L'écran
in	*posText*	La position du texte à afficher ; si NULL, centré en largeur et hauteur
in	*text*	Le texte à afficher
in	*color*	la couleur du texte
in	*font*	L'adresse de la police d'affichage (.ttf)
in	*ptSize*	la taille du texte à afficher
in	*mode*	Le mode d'écriture : 0 (Solid), 1 (Blended)

4.19 text.h File Reference

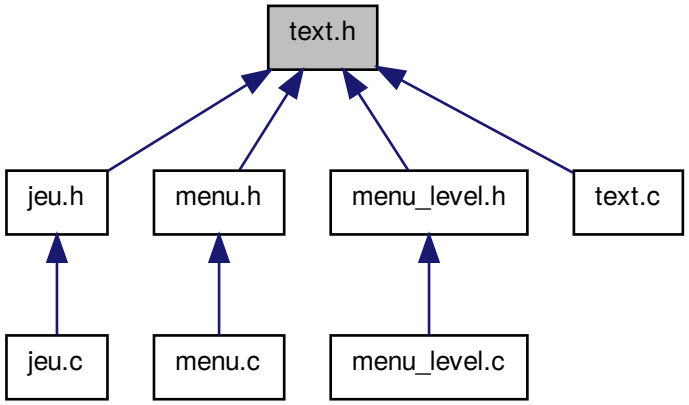
header de text.c

```
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <SDL/SDL.h>
#include <SDL/SDL_image.h>
#include <SDL/SDL_ttf.h>
```

Include dependency graph for text.h:



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



Functions

- void printText (SDL_Surface ∗screen, SDL_Rect ∗posText, char ∗text, SDL_Color color, char ∗font, int ptSize, int mode)

4.19.1 Detailed Description

header de text.c

Author

Xavier COPONET

Date

2014-02-27

4.19.2 Function Documentation

4.19.2.1 void printText (SDL_Surface ∗ *screen,* SDL_Rect ∗ *posText,* char ∗ *text,* SDL_Color *color,* char ∗ *font,* int *ptSize,* int *mode*)

affiche le texte sur l'écran à la position donnée

Parameters

out	*screen*	L'écran
in	*posText*	La position du texte à afficher ; si NULL, centré en largeur et hauteur
in	*text*	Le texte à afficher
in	*color*	la couleur du texte
in	*font*	L'adresse de la police d'affichage (.ttf)
in	*ptSize*	la taille du texte à afficher
in	*mode*	Le mode d'écriture : 0 (Solid), 1 (Blended)

Index