

Super Martin

Generated by Doxygen 1.8.6

Thu Mar 20 2014 14:59:36

Contents

1	Data Structure Index	1
1.1	Data Structures	1
2	File Index	3
2.1	File List	3
3	Data Structure Documentation	5
3.1	Character Struct Reference	5
3.1.1	Field Documentation	5
3.1.1.1	isJumping	5
3.1.1.2	isOnGround	5
3.1.1.3	life	5
3.2	Input Struct Reference	5
3.3	Level Struct Reference	6
3.4	Map Struct Reference	6
3.5	Sound Struct Reference	7
4	File Documentation	9
4.1	const.h File Reference	9
4.1.1	Detailed Description	10
4.2	file.c File Reference	10
4.2.1	Detailed Description	11
4.2.2	Function Documentation	11
4.2.2.1	closeFile	11
4.2.2.2	openFile	11
4.2.2.3	readFileSize	11
4.3	file.h File Reference	11
4.3.1	Detailed Description	13
4.3.2	Function Documentation	13
4.3.2.1	closeFile	13
4.3.2.2	openFile	13
4.3.2.3	readFileSize	13

4.4	file_level.h File Reference	14
4.4.1	Detailed Description	15
4.4.2	Function Documentation	16
4.4.2.1	closeLevel	16
4.4.2.2	closeLevelList	16
4.4.2.3	initLevel	16
4.4.2.4	openLevel	16
4.4.2.5	readLevelFile	17
4.4.2.6	writeLevel	17
4.5	game.c File Reference	18
4.5.1	Detailed Description	18
4.5.2	Function Documentation	19
4.5.2.1	play	19
4.5.2.2	printGameOver	20
4.5.2.3	printPause	21
4.5.2.4	updateSpeed	22
4.6	game.h File Reference	23
4.6.1	Detailed Description	24
4.6.2	Function Documentation	24
4.6.2.1	play	24
4.6.2.2	printGameOver	25
4.6.2.3	printPause	26
4.6.2.4	updateSpeed	27
4.7	image.c File Reference	28
4.7.1	Detailed Description	28
4.7.2	Function Documentation	28
4.7.2.1	imageLoad	28
4.7.2.2	imageLoadAlpha	29
4.8	image.h File Reference	29
4.8.1	Detailed Description	30
4.8.2	Function Documentation	30
4.8.2.1	imageLoad	30
4.8.2.2	imageLoadAlpha	31
4.9	input.c File Reference	32
4.9.1	Detailed Description	32
4.9.2	Function Documentation	32
4.9.2.1	keyboardActionGame	32
4.9.2.2	keyboardActionMenu	33
4.9.2.3	updateEvents	33
4.9.2.4	updateWaitEvents	33

4.10	main.c File Reference	33
4.10.1	Detailed Description	34
4.11	map.c File Reference	34
4.11.1	Detailed Description	35
4.11.2	Function Documentation	35
4.11.2.1	initMap	35
4.11.2.2	scrolling	35
4.11.2.3	updateScreenMap	36
4.12	map.h File Reference	36
4.12.1	Detailed Description	37
4.12.2	Function Documentation	37
4.12.2.1	initMap	37
4.12.2.2	scrolling	38
4.12.2.3	updateScreenMap	38
4.13	menu.c File Reference	39
4.13.1	Detailed Description	39
4.13.2	Function Documentation	39
4.13.2.1	blinkText	39
4.14	menu.h File Reference	39
4.14.1	Detailed Description	40
4.14.2	Function Documentation	41
4.14.2.1	blinkText	41
4.15	menu_level.c File Reference	41
4.15.1	Detailed Description	41
4.15.2	Function Documentation	42
4.15.2.1	menuLevel	42
4.16	menu_level.h File Reference	42
4.16.1	Detailed Description	43
4.16.2	Function Documentation	44
4.16.2.1	menuLevel	44
4.17	player.c File Reference	44
4.17.1	Detailed Description	45
4.17.2	Function Documentation	45
4.17.2.1	blitCharacter	45
4.17.2.2	collisionSprite	46
4.17.2.3	creatorCharacter	46
4.17.2.4	moveCharacter	46
4.17.2.5	movementVector	47
4.18	player.h File Reference	47
4.18.1	Detailed Description	49

4.18.2	Macro Definition Documentation	49
4.18.2.1	ABS	49
4.18.2.2	SGN	49
4.18.3	Function Documentation	49
4.18.3.1	blitCharacter	49
4.18.3.2	collisionSprite	49
4.18.3.3	creatorCharacter	50
4.18.3.4	moveCharacter	51
4.18.3.5	movementVector	51
4.19	share.h File Reference	52
4.19.1	Detailed Description	53
4.19.2	Function Documentation	53
4.19.2.1	waitFPS	54
4.20	sound.c File Reference	55
4.20.1	Detailed Description	55
4.20.2	Function Documentation	56
4.20.2.1	createSound	56
4.20.2.2	freeSound	56
4.20.2.3	playMusic	56
4.20.2.4	playMusicOnce	56
4.20.2.5	soundVolume	56
4.20.2.6	stopSound	56
4.21	sound.h File Reference	57
4.21.1	Detailed Description	58
4.21.2	Function Documentation	58
4.21.2.1	createSound	58
4.21.2.2	freeSound	58
4.21.2.3	playMusic	58
4.21.2.4	playMusicOnce	58
4.21.2.5	soundVolume	59
4.21.2.6	stopSound	59
4.22	text.c File Reference	59
4.22.1	Detailed Description	59
4.22.2	Function Documentation	60
4.22.2.1	printText	60
4.23	text.h File Reference	60
4.23.1	Detailed Description	61
4.23.2	Function Documentation	61
4.23.2.1	printText	61

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Character	5
Input	5
Level	6
Map	6
Sound	7

Chapter 2

File Index

2.1 File List

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

const.h	Contient les constantes du programme	9
file.c	Fonctions d'accès aux fichiers	10
file.h	Prototypes des fonctions d'accès aux fichiers	11
file_level.h	Gestion des fichiers de carte	14
game.c	Contient les fonctions liées au jeu	18
game.h	Header de game.c	23
image.c	Contient les fonctions liées aux images	28
image.h	Contient les fonctions liées aux images	29
input.c	32
input.h	??
main.c	33
map.c	Contient les fonctions liées au chargement et à l'affichage de la carte	34
map.h	Header de map.c	36
menu.c	Contient les fonctions liées au menu	39
menu.h	Header de menu.c	39
menu_level.c	Menu gérant le choix du niveau	41
menu_level.h	Menu gérant le choix du niveau	42
player.c	Contient les fonctions pour manipuler le joueur	44
player.h	Header de player.c	47
share.h	Gestion des fichiers de carte	52

sound.c	Contient les fonction pour jouer du son	55
sound.h	Header de sound.c	57
text.c	Contient les fonction pour afficher du texte à l'écran	59
text.h	Header de text.c	60

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](#)
- int [life](#)

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)

3.1.1.3 int life

0 when not, height remaning between character and max height if jumping

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

- [player.h](#)

3.2 Input Struct Reference

Data Fields

- char **key** [SDLK_LAST]
- int **quit**

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

- `input.h`

3.3 Level Struct Reference

Data Fields

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

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

- [const.h](#)

3.4 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](#)

3.5 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](#)

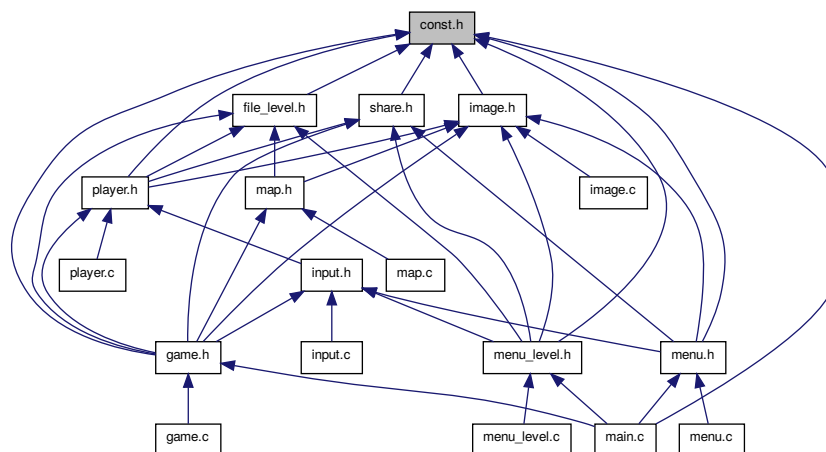
Chapter 4

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 33`
- `#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 0`
- `#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

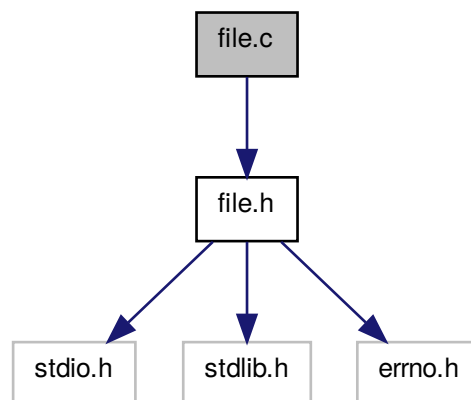
2014-02-27

4.2 file.c File Reference

Fonctions d'accès aux 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'accès aux 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	<i>*ptr_fichier</i>	le fichier
----	---------------------	------------

Returns

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

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

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

Parameters

in	<i>nom</i> []	le nom du fichier
in	<i>mode</i> []	le mode voulu

Returns

un pointeur sur le fichier ouvert, NULL s'il y a eu un problème

4.2.2.3 int readFileSize (FILE * *ptr_fichier*)

Lis la taille du fichier

Parameters

in	<i>*ptr_fichier</i>	le fichier
----	---------------------	------------

Returns

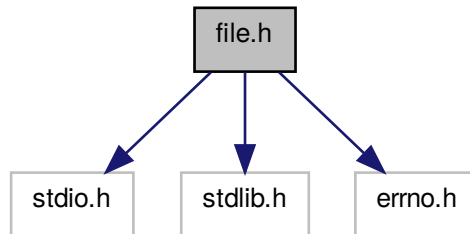
entier ayant la taille du fichier

4.3 file.h File Reference

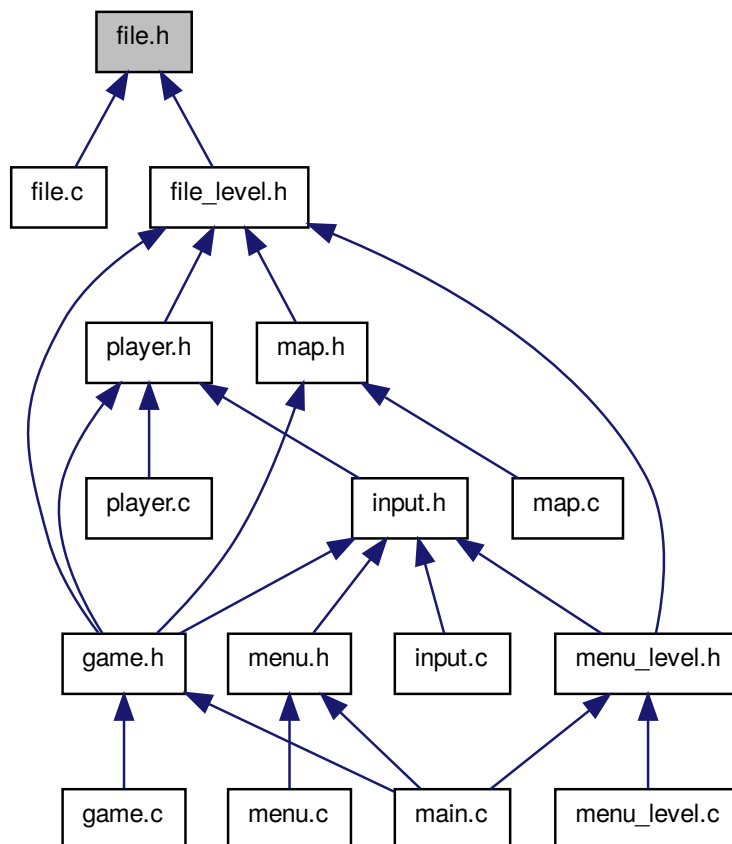
Prototypes des fonctions d'accès aux fichiers.

```
#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'accès 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	<i>*ptr_fichier</i>	le fichier
----	---------------------	------------

Returns

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

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

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

Parameters

in	<i>nom</i> []	le nom du fichier
in	<i>mode</i> []	le mode voulu

Returns

un pointeur sur le fichier ouvert, NULL s'il y a eu un problème

4.3.2.3 int readFileSize (FILE * *ptr_fichier*)

Lis la taille du fichier

Parameters

in	<i>*ptr_fichier</i>	le fichier
----	---------------------	------------

Returns

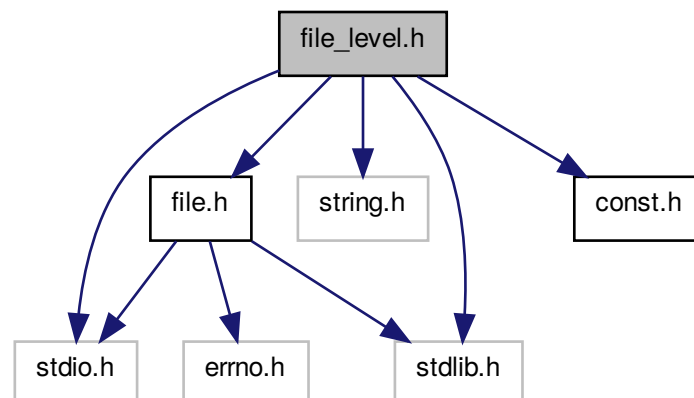
entier ayant la taille du fichier

4.4 file_level.h File Reference

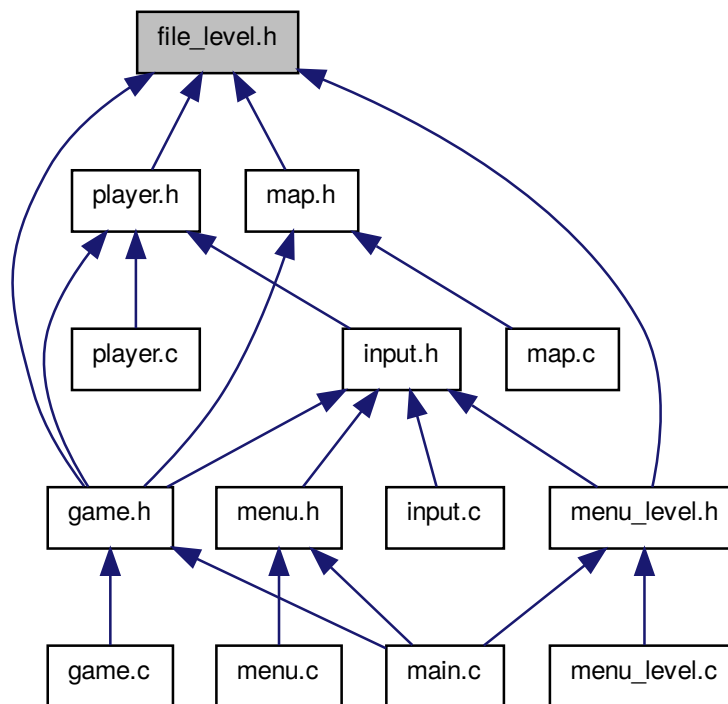
Gestion des fichiers de carte.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "file.h"
#include "const.h"
```

Include dependency graph for file_level.h:



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



Macros

- `#define TAILLE_MAX_NB_BLOCS_LARGEUR 5`
- `#define TAILLE_MAX_NB_BLOCS_HAUTEUR 4`
- `#define TAILLE_BUFFER 2`

Functions

- `Level * openLevel (char *file_name)`
- `void closeLevel (Level *lvl)`
- `Level * initLevel (Level *lvl)`
- `void writeLevel (char *file_name, Level *lvl)`
- `char ** readLevelFile (int *nb_lvl)`
- `void closeLevelList (char **level_names, int nb_lvl)`

4.4.1 Detailed Description

Gestion des fichiers de carte.

Author

Remi BERTHO

Date

15/03/14

Version

1.0

4.4.2 Function Documentation

4.4.2.1 void closeLevel (Level * lvl)

Ferme un niveau

Parameters

out	lvl	le niveau
-----	-----	-----------

4.4.2.2 void closeLevelList (char ** level_names, int nb_lvl)

Desalloue la liste des noms de niveau

Parameters

in, out	level_names	la liste des noms de niveau
in	nb_lvl	le nombre de niveau

4.4.2.3 Level* initLevel (Level * lvl)

Initialise un niveau en supposant que sa largeur et sa hauteur sont deja dans le niveau

Parameters

out	lvl	le niveau
-----	-----	-----------

Returns

un pointeur sur le niveau

4.4.2.4 Level* openLevel (char * file_name)

Ouvre un fichier map, et le stocke

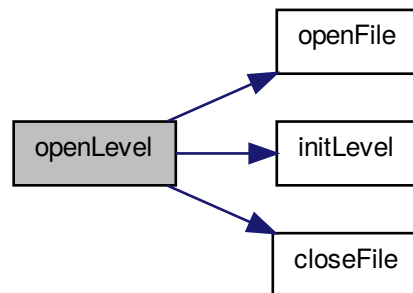
Parameters

in	file_name	le nom du fichier
----	-----------	-------------------

Returns

un pointeur sur le niveau

Here is the call graph for this function:

**4.4.2.5 char** readLevelFile (int * nb_lv)**

Lis le fichier level

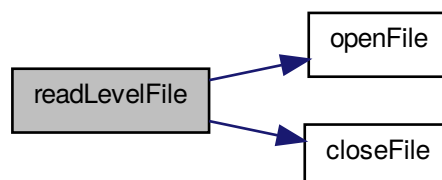
Parameters

out	<i>nb_lv</i>	le nombre de niveau
-----	--------------	---------------------

Returns

un pointeur sur la liste des niveaux cree

Here is the call graph for this function:

**4.4.2.6 void writeLevel (char * file_name, Level * lv)**

Ecrit le niveau dans un fichier

Date

2014-02-27

4.5.2 Function Documentation

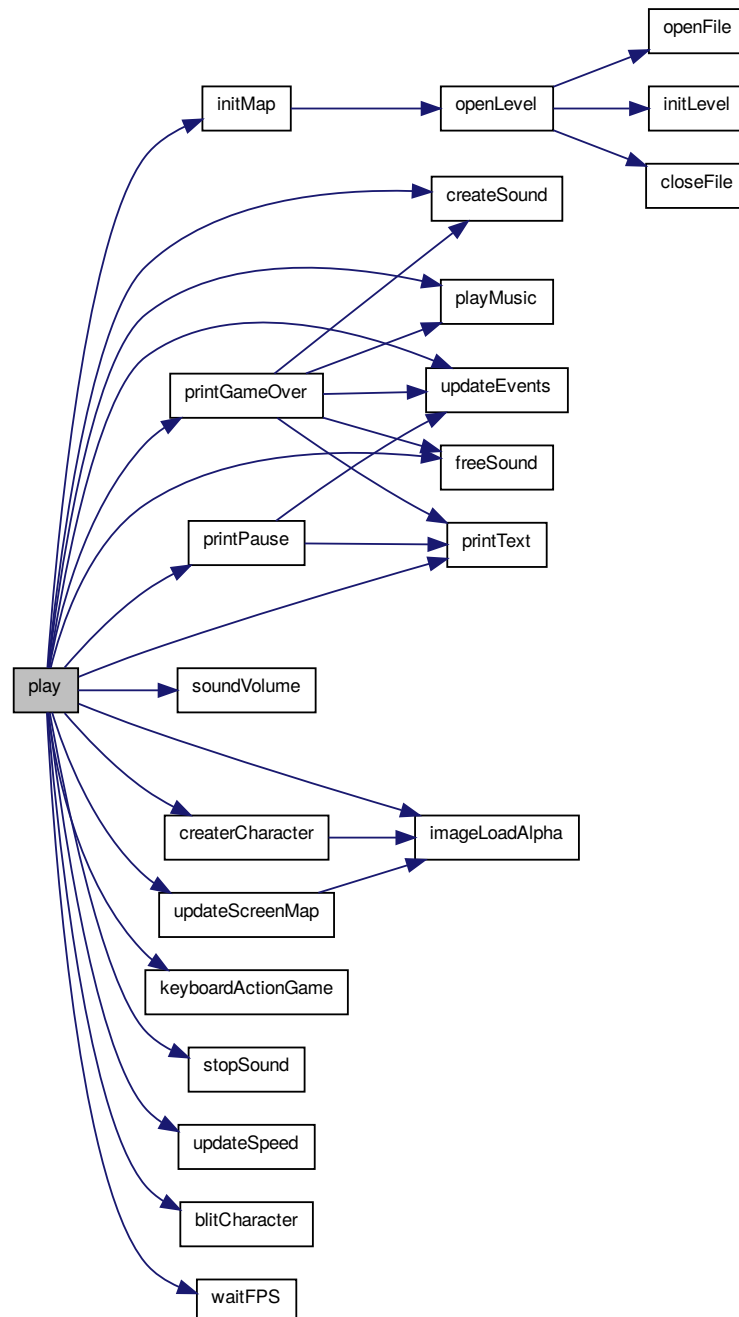
4.5.2.1 void play (SDL_Surface * *screen*, char * *level_name*)

contient la boucle principale du jeu qui appelle les fonctions

Parameters

<i>in, out</i>	<i>screen</i>	L'écran de jeu
<i>in</i>	<i>lvel_name</i>	le nom du niveau

Here is the call graph for this function:



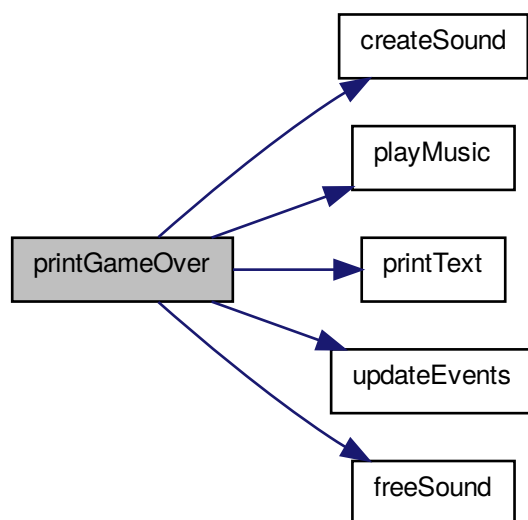
4.5.2.2 void printGameOver (SDL_Surface * screen, int * continuer, Input * in)

affiche le message de game overflow_error

Parameters

out	<i>screen</i>	l'écran de jeu
-----	---------------	----------------

Here is the call graph for this function:



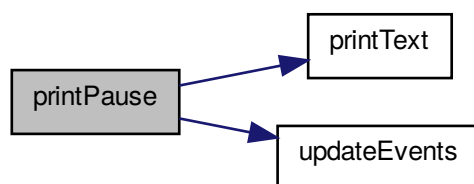
4.5.2.3 void printPause (SDL_Surface * screen, Input * in, int * time, int * continuer)

Met en pause le jeu

Parameters

out	<i>screen</i>	l'écran de jeu
out	<i>time</i>	le temps restant
in	<i>in</i>	la structure input
out	<i>le</i>	booléen de main loop de la fonction jouer

Here is the call graph for this function:



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

Met a jour la vitesse

Parameters

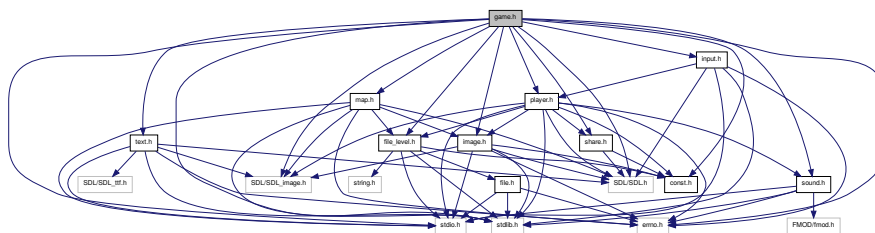
out	<i>float</i>	la vitesse
out	<i>acceleration</i>	l'acceleration

4.6 game.h File Reference

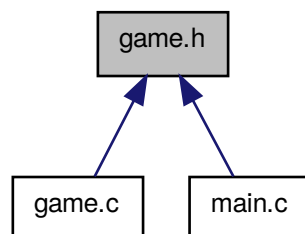
header de [game.c](#)

```
#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 "input.h"
```

Include dependency graph for game.h:



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



Functions

- void [play](#) (SDL_Surface *screen, char *level_name)
- void [printGameOver](#) (SDL_Surface *screen, int *continuer, [Input](#) *in)
- 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, [Input](#) *in, int *time, int *continuer)
- Uint32 **decomptage** (Uint32 intervalle, void *parametre)

4.6.1 Detailed Description

header de [game.c](#)

Author

Xavier COPONET

Date

2014-02-27

4.6.2 Function Documentation

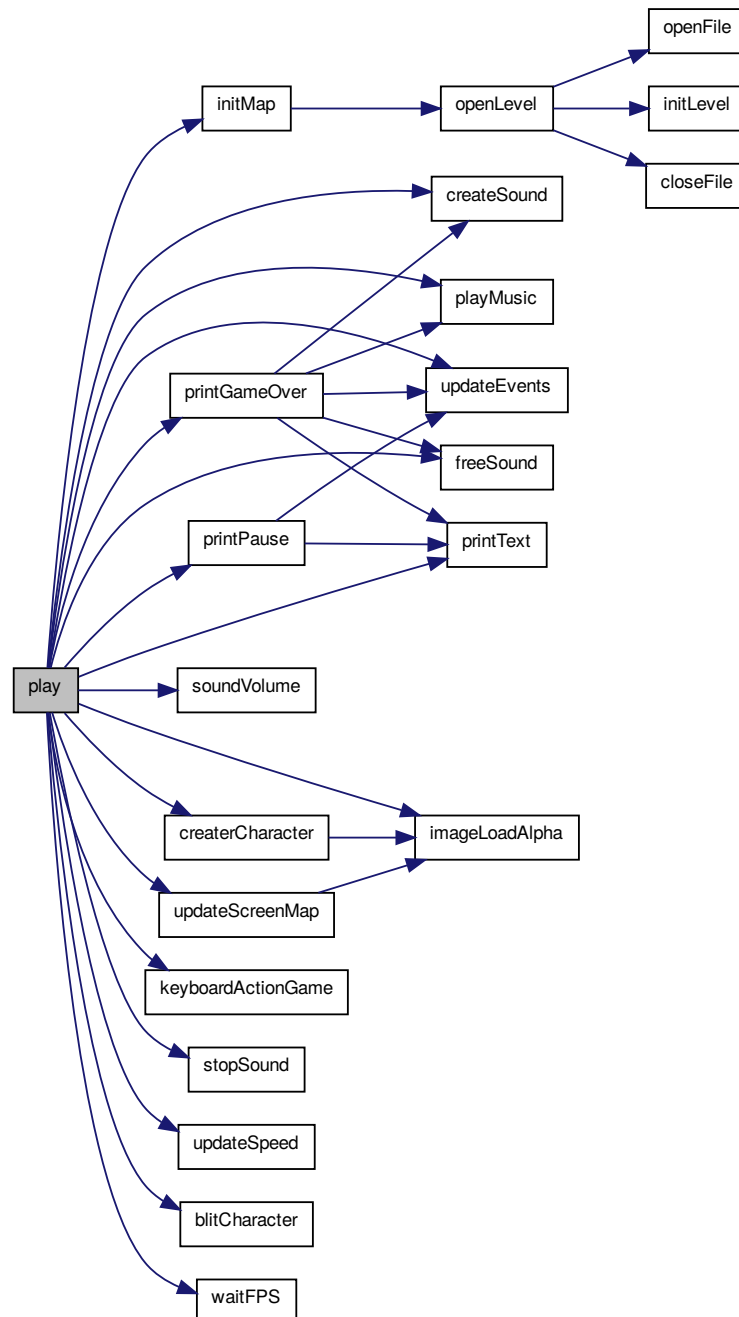
4.6.2.1 void play (SDL_Surface * screen, char * level_name)

contient la boucle principale du jeu qui appelle les fonctions

Parameters

in, out	<i>screen</i>	L'écran de jeu
in	<i>lvel_name</i>	le nom du niveau

Here is the call graph for this function:



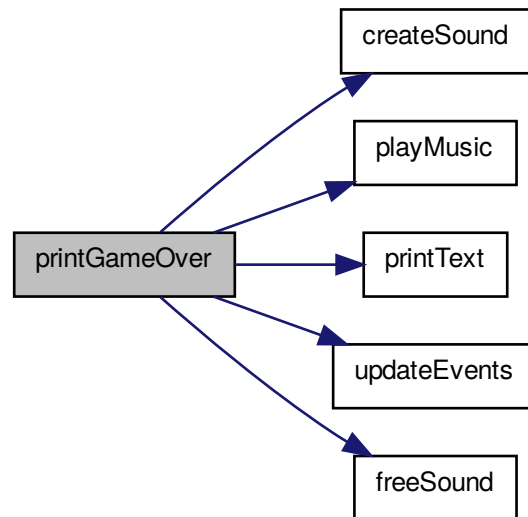
4.6.2.2 void printGameOver (SDL_Surface * screen, int * continuer, Input * in)

affiche le message de game overflow_error

Parameters

out	<i>screen</i>	l'écran de jeu
-----	---------------	----------------

Here is the call graph for this function:



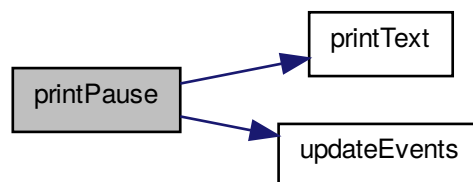
4.6.2.3 void printPause (SDL_Surface * screen, Input * in, int * time, int * continuer)

Met en pause le jeu

Parameters

out	<i>screen</i>	l'écran de jeu
out	<i>time</i>	le temps restant
in	<i>in</i>	la structure input
out	<i>le</i>	booléen de main loop de la fonction jouer

Here is the call graph for this function:



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

Met a jour la vitesse

Parameters

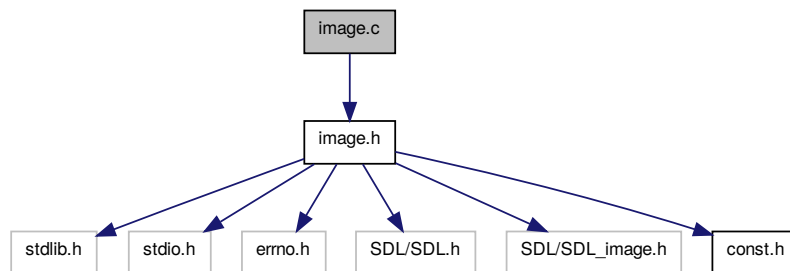
out	<i>float</i>	la vitesse
out	<i>acceleration</i>	l'acceleration

4.7 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`)

4.7.1 Detailed Description

contient les fonction liées aux images

Author

Rémi BERTHO

Date

2014-02-27

4.7.2 Function Documentation

4.7.2.1 `SDL_Surface *` `imageLoad` (`char * file_name`)

Charge une image

Parameters

in	<i>file_name</i>	le nom du fichier
----	------------------	-------------------

Returns

un pointeur sur une SDL_Surface

4.7.2.2 SDL_Surface * imageLoadAlpha (char * file_name)

Charge une image

Parameters

in	<i>file_name</i>	le nom du fichier
----	------------------	-------------------

Returns

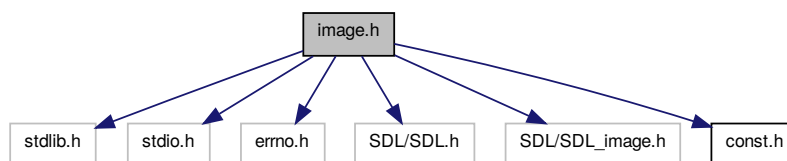
un pointeur sur une SDL_Surface

4.8 image.h File Reference

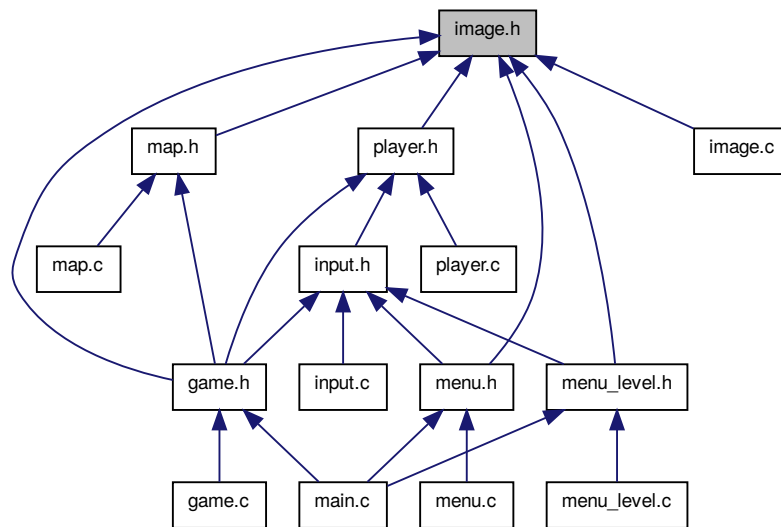
contient les fonction liées aux images

```
#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`)

4.8.1 Detailed Description

contient les fonction liées aux images

Author

Rémi BERTHO

Date

2014-02-27

4.8.2 Function Documentation

4.8.2.1 `SDL_Surface* imageLoad (char * file_name)`

Charge une image

Parameters

<code>in</code>	<code>file_name</code>	le nom du fichier
-----------------	------------------------	-------------------

Returns

un pointeur sur une `SDL_Surface`

4.8.2.2 `SDL_Surface*` `imageLoadAlpha (char * file_name)`

Charge une image

Parameters

in	file_name	le nom du fichier
----	-----------	-------------------

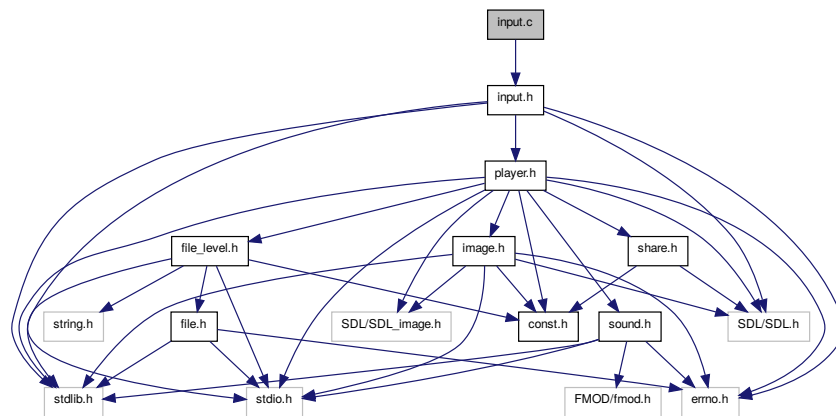
Returns

un pointeur sur une SDL_Surface

4.9 input.c File Reference

```
#include "input.h"
```

Include dependency graph for input.c:



Functions

- int [updateEvents](#) (Input *in)
- void [keyboardActionGame](#) (Input *in, int *move_left, int *move_right, int *jump, int *pause, [Character](#) *player)
- int [updateWaitEvents](#) (Input *in)
- void [keyboardActionMenu](#) (Input *in, int *cursorPos, int *play_level, int nb_lvl)

4.9.1 Detailed Description

Author

Xavier COPONET

Date

2014-03-18

4.9.2 Function Documentation

4.9.2.1 void [keyboardActionGame](#) (Input * *in*, int * *move_left*, int * *move_right*, int * *jump*, int * *pause*, [Character](#) * *player*)

perform action command by keyboard action

Parameters

in	<i>in</i>	the input structure
out	<i>move_left</i>	the left movement boolean
out	<i>move_right</i>	the right movement boolean
out	<i>jump</i>	the jump boolean
out	<i>pause</i>	the pause boolean
in	<i>player</i>	the Player

4.9.2.2 void keyboardActionMenu (Input * *in*, int * *cursorPos*, int * *play_level*, int *nb_lvl*)

perform action command by keyboard action

Parameters

in	<i>in</i>	the input structure
out	<i>cursorPos</i>	cursor position
out	<i>play_level</i>	boolean about playing the level or quit to title screen
in	<i>nb_lvl</i>	the number of levels

4.9.2.3 int updateEvents (Input * *in*)

recuperate keyboard input with a SDL_PollEvent

Parameters

out	<i>in</i>	the input structure
-----	-----------	---------------------

Returns

1 if a key is activated

4.9.2.4 int updateWaitEvents (Input * *in*)

recuperate keyboard input with a SDL_WaitEvent

Parameters

out	<i>in</i>	the input structure
-----	-----------	---------------------

Returns

1 if a key is activated

4.10 main.c File Reference

```
#include "game.h"
#include "const.h"
#include "menu.h"
#include "menu_level.h"
#include "sound.h"
```


4.11.1 Detailed Description

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

Author

Xavier COPONET

Date

2014-03-18

4.11.2 Function Documentation

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

initialise la carte

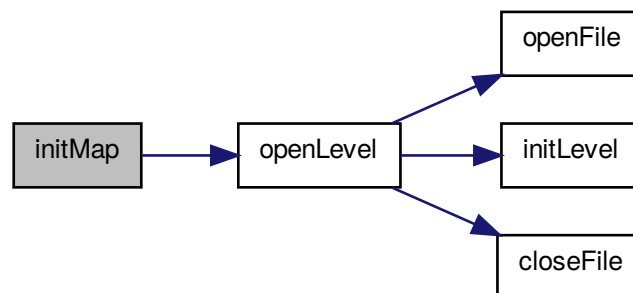
Parameters

in	<i>screen</i>	l'écran de jeu
in	<i>level_name</i>	le nom du niveau

Returns

un pointeur sur la carte initialisée

Here is the call graph for this function:



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

effectue un scrolling

Parameters

in, out	<i>map</i>	Le niveau à gérer
in	<i>direction</i>	La direction de scrolling
in	<i>speed</i>	la vitesse de scrolling

4.11.2.3 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	<i>screen</i>	of the game
in	<i>Map</i>	*m The map
in	<i>tileset</i>	lvl tileset

Here is the call graph for this function:



4.12 map.h File Reference

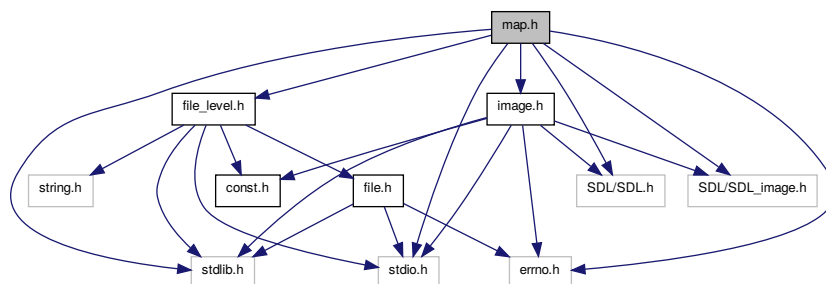
header de [map.c](#)

```

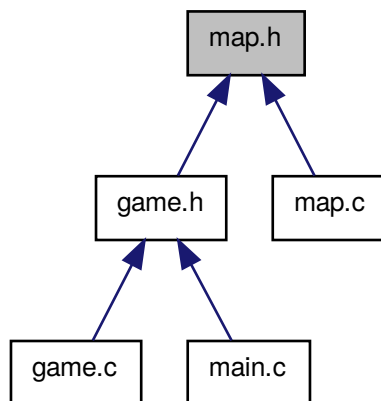
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <SDL/SDL.h>
#include <SDL/SDL_image.h>
#include "image.h"
#include "file_level.h"

```

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)

4.12.1 Detailed Description

header de [map.c](#)

Author

Xavier COPONET

Date

2014-03-18

4.12.2 Function Documentation

4.12.2.1 [Map](#)* [initMap](#) (SDL_Surface * *screen*, char * *level_name*)

initialise la carte

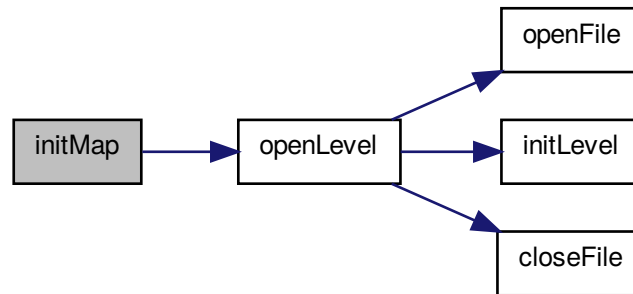
Parameters

in	<i>screen</i>	l'écran de jeu
in	<i>level_name</i>	le nom du niveau

Returns

un pointeur sur la carte initialisée

Here is the call graph for this function:



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

effectue un scrolling

Parameters

in, out	<i>map</i>	Le niveau à gérer
in	<i>direction</i>	La direction de scrolling
in	<i>speed</i>	la vitesse de scrolling

4.12.2.3 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	<i>screen</i>	of the game
in	<i>Map</i>	*m The map
in	<i>tileset</i>	lvl tileset

Here is the call graph for this function:

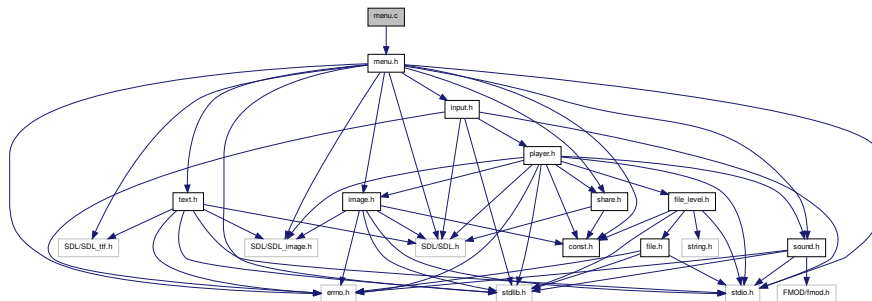


4.13 menu.c File Reference

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 interval, void *param)

4.13.1 Detailed Description

contient les fonction liées au menu

Author

Xavier COPONET

Date

2014-02-27

4.13.2 Function Documentation

4.13.2.1 Uint32 blinkText (Uint32 interval, void * param)

toggle the printing text boolean

Parameters

in	interval	the interval between two callback of the function
----	----------	---

Returns

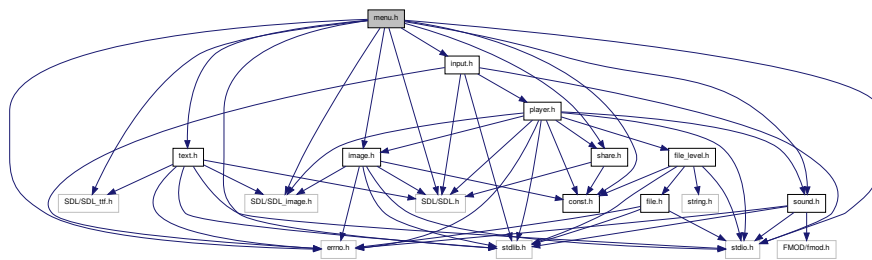
1000 if the boolean is right, 600 if not

4.14 menu.h File Reference

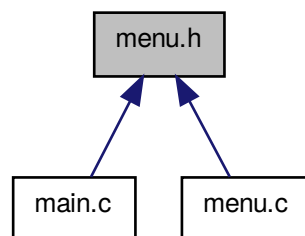
header de [menu.c](#)

```
#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 "input.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 interval, void *param)

4.14.1 Detailed Description

header de [menu.c](#)

Author

Xavier COPONET

Date

2014-02-27

4.14.2 Function Documentation

4.14.2.1 Uint32 blinkText (Uint32 *interval*, void * *param*)

toggle the printing text boolean

Parameters

in	<i>interval</i>	the interval between two callback of the function
----	-----------------	---

Returns

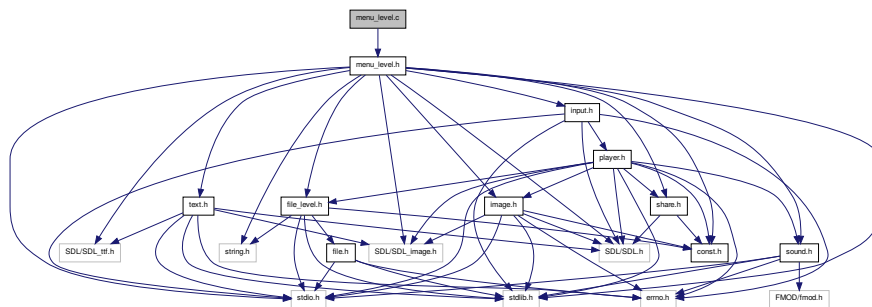
1000 if the boolean is right, 600 if not

4.15 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)

4.15.1 Detailed Description

Menu gerant le choix du niveau.

Author

Remi BERTHO

Date

15/03/14

Version

2.0 (implémentation de la gestion des event)

4.15.2 Function Documentation

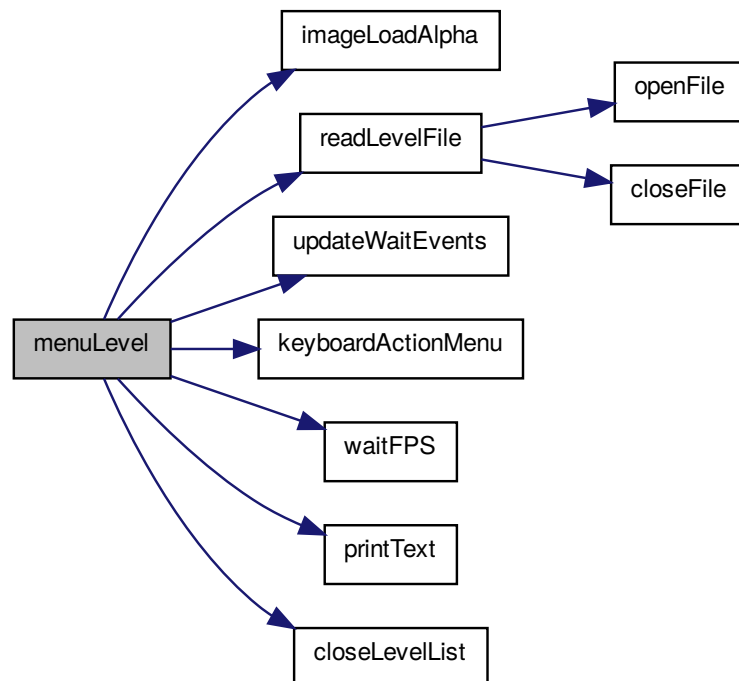
4.15.2.1 `int menuLevel (SDL_Surface * screen, char level_name[TAILLE_MAX_NOM_FICHIER], Sound * s)`

Menu pour choisir le niveau

Parameters

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

Here is the call graph for this function:



4.16 menu_level.h File Reference

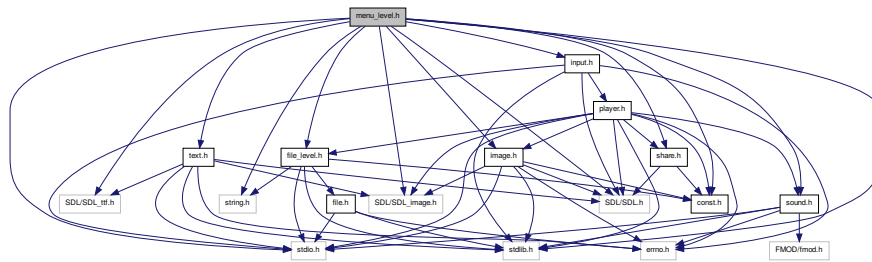
Menu gerant le choix du niveau.

```

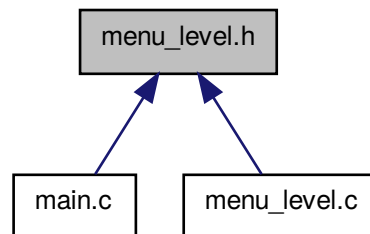
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <SDL/SDL.h>
#include <SDL/SDL_image.h>
#include <SDL/SDL_ttf.h>
#include "const.h"
#include "file_level.h"
#include "share.h"
#include "text.h"
#include "sound.h"
#include "image.h"
#include "input.h"

```

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)

4.16.1 Detailed Description

Menu gerant le choix du niveau.

Author

Remi BERTHO

Date

15/03/14

Version

2.0

4.16.2 Function Documentation

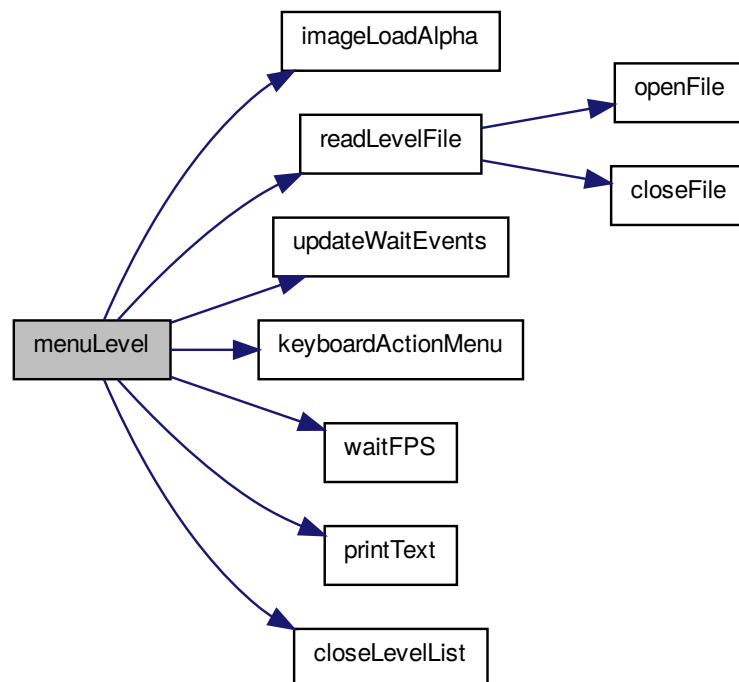
4.16.2.1 `int menuLevel (SDL_Surface * screen, char level_name[TAILLE_MAX_NOM_FICHIER], Sound * s)`

Menu pour choisir le niveau

Parameters

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

Here is the call graph for this function:

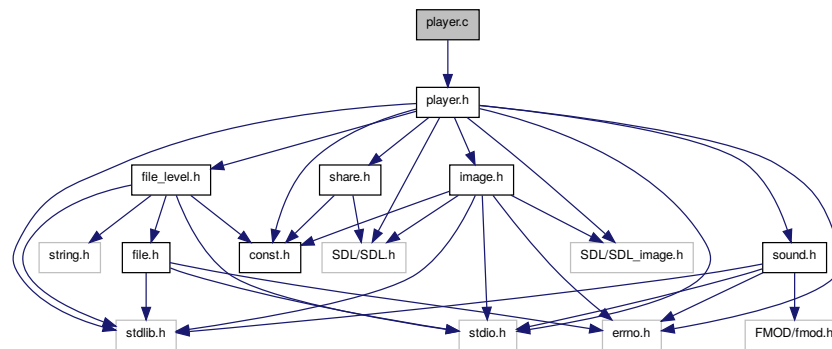


4.17 player.c File Reference

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, [Sound](#) *jump_sound)

4.17.1 Detailed Description

contient les fonction pour manipuler le joueur

Author

Xavier COPONET

Date

2014-02-27

4.17.2 Function Documentation

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

blit le personnage à l'écran

Parameters

in, out	screen	L'écran
---------	--------	---------

in	<i>c</i>	Le personnage
in	<i>m</i>	la carte du jeu

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

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

Parameters

in	<i>r</i>	le SDL_Rect correspondant à la sprite
in	<i>m</i>	la carte contenant le décor

Returns

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

4.17.2.3 Character * creatorCharacter (char * *spR*, char * *spL*)

créer un personnage

Parameters

in	<i>spR</i>	l'adresse de la sprite droite
in	<i>spL</i>	l'adresse de la sprite gauche

Returns

le pointeur sur la structure créée

Here is the call graph for this function:



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

déplace le personnage selon la direction

Parameters

in, out	<i>c</i>	Le personnage
in	<i>direction</i>	La direction du déplacement
in	<i>m</i>	la carte sur laquelle le personnage se déplace

in	<i>speed</i>	la vitesse de déplacement
----	--------------	---------------------------

Returns

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

Here is the call graph for this function:



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

create a movement vector

Parameters

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

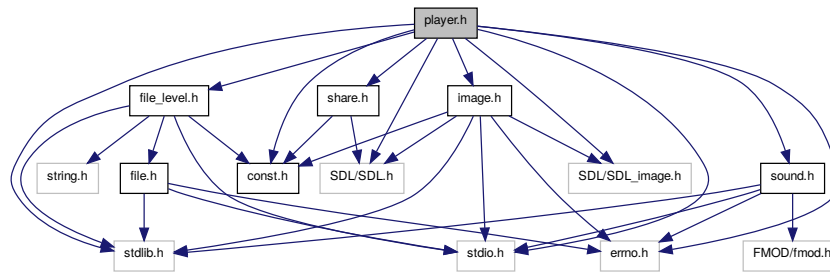
4.18 player.h File Reference

header de [player.c](#)

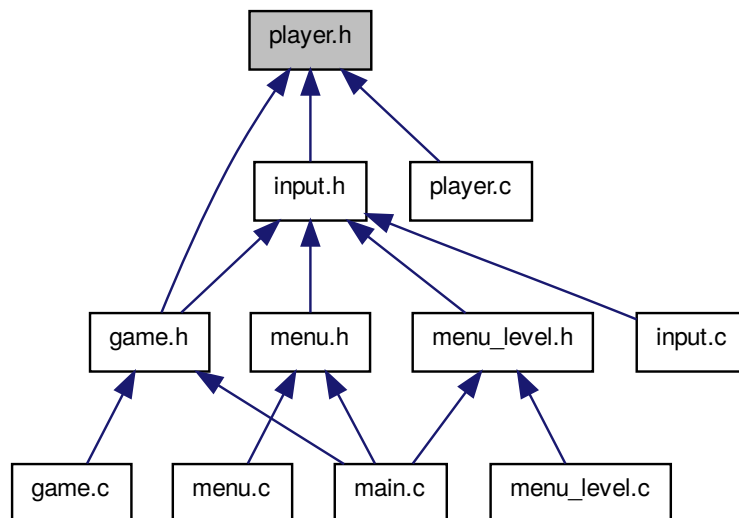
```

#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 "sound.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](#) * [creatorCharacter](#) (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, [Sound](#) *jump_sound)

4.18.1 Detailed Description

header de [player.c](#)

Author

Xavier COPONET

Date

2014-02-27

4.18.2 Macro Definition Documentation

4.18.2.1 `#define ABS(X) (((X)<0)?(-(X)):(X))`

X absolute value

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

X sign

4.18.3 Function Documentation

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

blit le personnage à l'écran

Parameters

<i>in, out</i>	<i>screen</i>	L'écran
<i>in</i>	<i>c</i>	Le personnage
<i>in</i>	<i>m</i>	la carte du jeu

4.18.3.2 `int collisionSprite (SDL_Rect r, Map * m)`

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

Parameters

<i>in</i>	<i>r</i>	le SDL_Rect correspondant à la sprite
<i>in</i>	<i>m</i>	la carte contenant le décor

Returns

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

4.18.3.3 Character* creatorCharacter (char * *spR*, char * *spL*)

créer un personnage

Parameters

in	<i>spR</i>	l'adresse de la sprite droite
in	<i>spL</i>	l'adresse de la sprite gauche

Returns

le pointeur sur la structure créée

Here is the call graph for this function:



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

déplace le personnage selon la direction

Parameters

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

Returns

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

Here is the call graph for this function:



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

create a movement vector

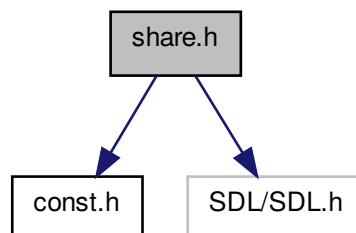
Parameters

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

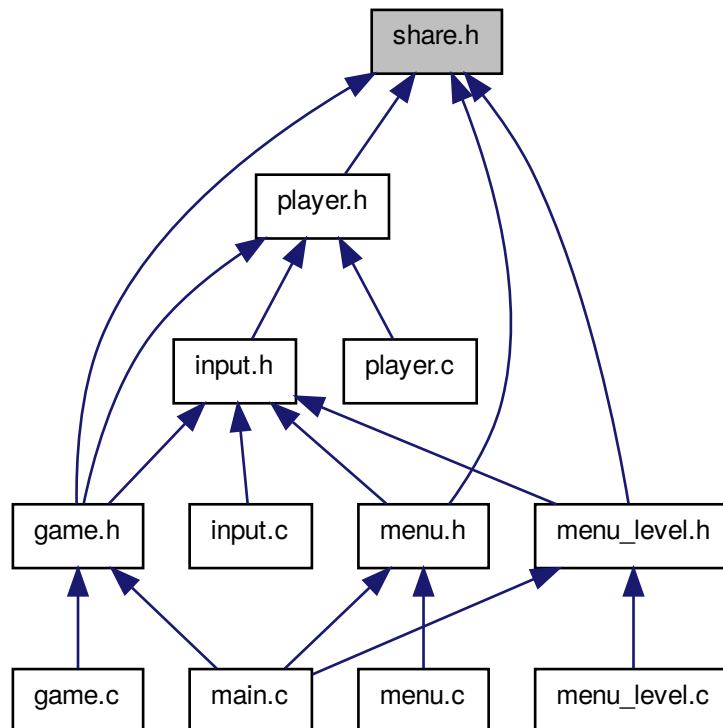
4.19 share.h File Reference

Gestion des fichiers de carte.

```
#include "const.h"  
#include <SDL/SDL.h>  
Include dependency graph for share.h:
```



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



Functions

- void [waitFPS](#) (int *previous_time, int *current_time)

4.19.1 Detailed Description

Gestion des fichiers de carte.

Author

Remi BERTHO

Date

15/03/14

Version

1.0

4.19.2 Function Documentation

4.19.2.1 void waitFPS (int * *previous_time*, int * *current_time*)

Permet d'attendre le temps necessaire pour avoir le nombre de PFS souhaité

Parameters

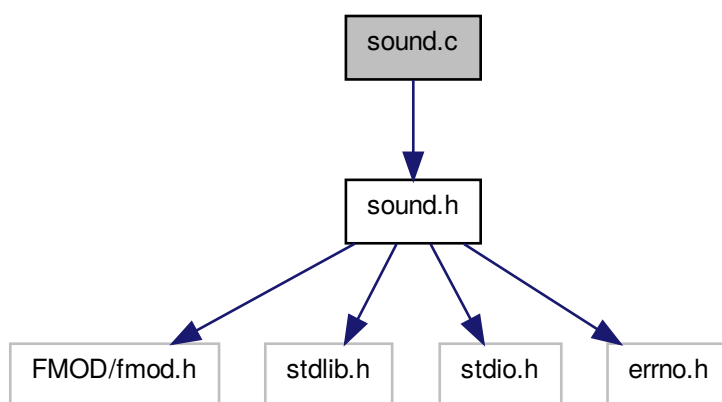
in, out	<i>previous_time</i>	le temps precedent
in, out	<i>previous_time</i>	le temps actuel

4.20 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)`

4.20.1 Detailed Description

contient les fonction pour jouer du son

Author

Xavier COPONET

Date

2014-02-27

4.20.2 Function Documentation

4.20.2.1 `sound * createSound (void)`

créer une structure son

Returns

la structure son

4.20.2.2 `void freeSound (Sound * s)`

release the sound

Parameters

out	s	the sound
-----	---	-----------

4.20.2.3 `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

4.20.2.4 `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

4.20.2.5 `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]

4.20.2.6 `void stopSound (Sound * s)`

stop the sound

Parameters

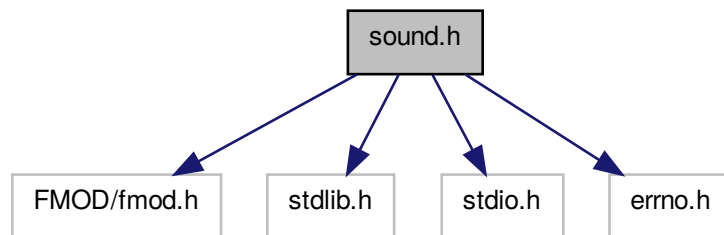
out	the	sound to stop
-----	-----	---------------

4.21 sound.h File Reference

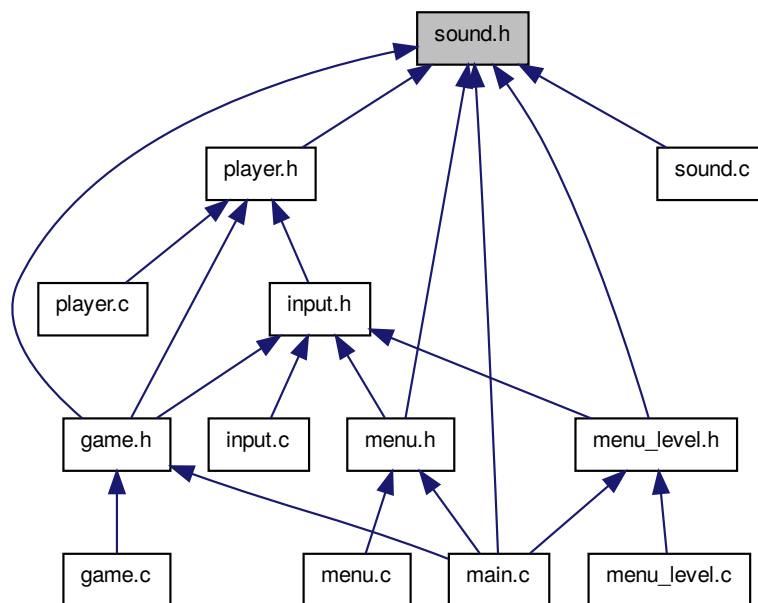
header de [sound.c](#)

```
#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)

4.21.1 Detailed Description

header de [sound.c](#)

Author

Xavier COPONET

Date

2014-02-27

4.21.2 Function Documentation

4.21.2.1 [Sound*](#) [createSound](#) (void)

créer une structure son

Returns

la structure son

4.21.2.2 void [freeSound](#) ([Sound *](#) s)

release the sound

Parameters

out	s	the sound
-----	---	-----------

4.21.2.3 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

4.21.2.4 void [playMusicOnce](#) ([Sound *](#) s, char * *file*)

lit un fichier long une fois

Parameters

in, out	<i>s</i>	la structure son que l'on manipule
in	<i>file</i>	Le fichier son à lire

4.21.2.5 void soundVolume (Sound * s, float volume)

set the sound volume

Parameters

out	<i>s</i>	the sound
in	<i>volume</i>	the sound volume : [0.0 : no sound ; 1.0 (default) max power]

4.21.2.6 void stopSound (Sound * s)

stop the sound

Parameters

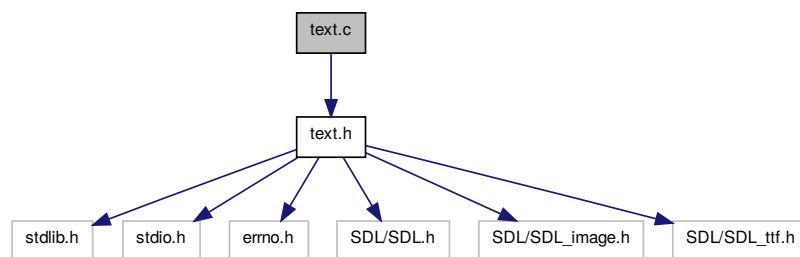
out	<i>the</i>	sound to stop
-----	------------	---------------

4.22 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, int r, int g, int b, char *font, int ptSize, int mode)

4.22.1 Detailed Description

contient les fonction pour afficher du texte à l'écran

Author

Xavier COPONET

Date

2014-02-27

4.22.2 Function Documentation

4.22.2.1 `void printText (SDL_Surface * screen, SDL_Rect * posText, char * text, int r, int g, int b, char * font, int ptSize, int mode)`

affiche le texte sur l'écran à la position donnée

Parameters

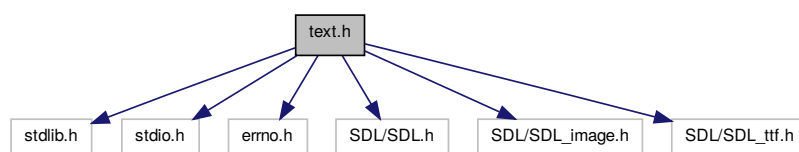
out	<i>screen</i>	L'écran
in	<i>posText</i>	La position du texte à afficher ; si NULL, centré en largeur et hauteur
in	<i>text</i>	Le texte à afficher
in	<i>int</i>	r red value
in	<i>int</i>	g green value
in	<i>int</i>	b blue value
in	<i>font</i>	L'adresse de la police d'affichage (.ttf)
in	<i>ptSize</i>	la taille du texte à afficher
in	<i>mode</i>	Le mode d'écriture : 0 (Solid), 1 (Blended)

4.23 text.h File Reference

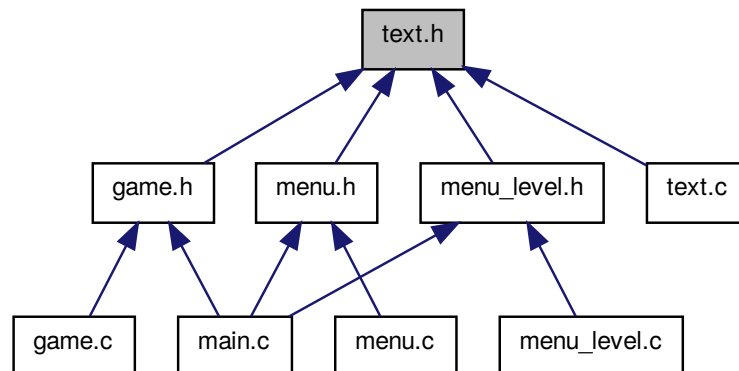
header de [text.c](#)

```
#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, int r, int g, int b, char *font, int ptSize, int mode)

4.23.1 Detailed Description

header de [text.c](#)

Author

Xavier COPONET

Date

2014-02-27

4.23.2 Function Documentation

4.23.2.1 void [printText](#) (SDL_Surface * *screen*, SDL_Rect * *posText*, char * *text*, int *r*, int *g*, int *b*, char * *font*, int *ptSize*, int *mode*)

affiche le texte sur l'écran à la position donnée

Parameters

out	<i>screen</i>	L'écran
in	<i>posText</i>	La position du texte à afficher ; si NULL, centré en largeur et hauteur
in	<i>text</i>	Le texte à afficher
in	<i>int</i>	r red value

in	<i>int</i>	g green value
in	<i>int</i>	b blue value
in	<i>font</i>	L'adresse de la police d'affichage (.ttf)
in	<i>ptSize</i>	la taille du texte à afficher
in	<i>mode</i>	Le mode d'écriture : 0 (Solid), 1 (Blended)

Index

ABS
 player.h, 49

blinkText
 menu.c, 39
 menu.h, 41

blitCharacter
 player.c, 45
 player.h, 49

Character, 5
 isJumping, 5
 isOnGround, 5
 life, 5

closeFile
 file.c, 11
 file.h, 13

closeLevel
 file_level.h, 16

closeLevelList
 file_level.h, 16

collisionSprite
 player.c, 46
 player.h, 49

const.h, 9

createSound
 sound.c, 56
 sound.h, 58

creatorCharacter
 player.c, 46
 player.h, 49

file.c, 10
 closeFile, 11
 openFile, 11
 readFileSize, 11

file.h, 11
 closeFile, 13
 openFile, 13
 readFileSize, 13

file_level.h, 14
 closeLevel, 16
 closeLevelList, 16
 initLevel, 16
 openLevel, 16
 readLevelFile, 17
 writeLevel, 17

freeSound
 sound.c, 56
 sound.h, 58

game.c, 18
 play, 19
 printGameOver, 20
 printPause, 21
 updateSpeed, 21

game.h, 23
 play, 24
 printGameOver, 25
 printPause, 26
 updateSpeed, 26

image.c, 28
 imageLoad, 28
 imageLoadAlpha, 29

image.h, 29
 imageLoad, 30
 imageLoadAlpha, 30

imageLoad
 image.c, 28
 image.h, 30

imageLoadAlpha
 image.c, 29
 image.h, 30

initLevel
 file_level.h, 16

initMap
 map.c, 35
 map.h, 37

Input, 5

input.c, 32
 keyboardActionGame, 32
 keyboardActionMenu, 33
 updateEvents, 33
 updateWaitEvents, 33

isJumping
 Character, 5

isOnGround
 Character, 5

keyboardActionGame
 input.c, 32

keyboardActionMenu
 input.c, 33

Level, 6

life
 Character, 5

main.c, 33

Map, 6

- map.c, 34
 - initMap, 35
 - scrolling, 35
 - updateScreenMap, 36
- map.h, 36
 - initMap, 37
 - scrolling, 38
 - updateScreenMap, 38
- menu.c, 39
 - blinkText, 39
- menu.h, 39
 - blinkText, 41
- menu_level.c, 41
 - menuLevel, 42
- menu_level.h, 42
 - menuLevel, 44
- menuLevel
 - menu_level.c, 42
 - menu_level.h, 44
- moveCharacter
 - player.c, 46
 - player.h, 51
- movementVector
 - player.c, 47
 - player.h, 51
- openFile
 - file.c, 11
 - file.h, 13
- openLevel
 - file_level.h, 16
- play
 - game.c, 19
 - game.h, 24
- playMusic
 - sound.c, 56
 - sound.h, 58
- playMusicOnce
 - sound.c, 56
 - sound.h, 58
- player.c, 44
 - blitCharacter, 45
 - collisionSprite, 46
 - creatorCharacter, 46
 - moveCharacter, 46
 - movementVector, 47
- player.h, 47
 - ABS, 49
 - blitCharacter, 49
 - collisionSprite, 49
 - creatorCharacter, 49
 - moveCharacter, 51
 - movementVector, 51
 - SGN, 49
- printGameOver
 - game.c, 20
 - game.h, 25
- printPause
 - game.c, 21
 - game.h, 26
- printText
 - text.c, 60
 - text.h, 61
- readFileSize
 - file.c, 11
 - file.h, 13
- readLevelFile
 - file_level.h, 17
- SGN
 - player.h, 49
- scrolling
 - map.c, 35
 - map.h, 38
- share.h, 52
 - waitFPS, 53
- Sound, 7
- sound.c, 55
 - createSound, 56
 - freeSound, 56
 - playMusic, 56
 - playMusicOnce, 56
 - soundVolume, 56
 - stopSound, 56
- sound.h, 57
 - createSound, 58
 - freeSound, 58
 - playMusic, 58
 - playMusicOnce, 58
 - soundVolume, 59
 - stopSound, 59
- soundVolume
 - sound.c, 56
 - sound.h, 59
- stopSound
 - sound.c, 56
 - sound.h, 59
- text.c, 59
 - printText, 60
- text.h, 60
 - printText, 61
- updateEvents
 - input.c, 33
- updateScreenMap
 - map.c, 36
 - map.h, 38
- updateSpeed
 - game.c, 21
 - game.h, 26
- updateWaitEvents
 - input.c, 33
- waitFPS
 - share.h, 53

writeLevel
 file_level.h, [17](#)