

Super Martin

Generated by Doxygen 1.8.6

Tue Mar 18 2014 18:00:51

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
3.2	Level Struct Reference	5
3.3	Map Struct Reference	6
3.4	Sound Struct Reference	6
4	File Documentation	7
4.1	const.h File Reference	7
4.1.1	Detailed Description	8
4.2	file.c File Reference	8
4.2.1	Detailed Description	9
4.2.2	Function Documentation	9
4.2.2.1	closeFile	9
4.2.2.2	openFile	9
4.2.2.3	readFileSize	9
4.3	file.h File Reference	10
4.3.1	Detailed Description	11
4.3.2	Function Documentation	11
4.3.2.1	closeFile	11
4.3.2.2	openFile	11
4.3.2.3	readFileSize	11
4.4	image.c File Reference	12
4.4.1	Detailed Description	12

4.4.2	Function Documentation	12
4.4.2.1	imageLoad	12
4.4.2.2	imageLoadAlpha	12
4.5	image.h File Reference	13
4.5.1	Detailed Description	14
4.5.2	Function Documentation	14
4.5.2.1	imageLoad	14
4.5.2.2	imageLoadAlpha	14
4.6	jeu.c File Reference	14
4.6.1	Detailed Description	15
4.6.2	Function Documentation	15
4.6.2.1	jouer	15
4.6.2.2	printGameOver	16
4.6.2.3	printPause	17
4.6.2.4	updateSpeed	17
4.7	jeu.h File Reference	18
4.7.1	Detailed Description	19
4.7.2	Function Documentation	19
4.7.2.1	jouer	19
4.7.2.2	printGameOver	20
4.7.2.3	printPause	21
4.7.2.4	updateSpeed	21
4.8	map.c File Reference	22
4.8.1	Detailed Description	22
4.8.2	Function Documentation	22
4.8.2.1	initMap	22
4.8.2.2	scrolling	23
4.8.2.3	updateScreenMap	23
4.9	map.h File Reference	23
4.9.1	Detailed Description	24
4.9.2	Function Documentation	25
4.9.2.1	initMap	25
4.9.2.2	scrolling	25
4.9.2.3	updateScreenMap	25
4.10	menu.c File Reference	25
4.10.1	Detailed Description	26
4.11	menu.h File Reference	26
4.11.1	Detailed Description	27
4.12	menu_level.c File Reference	27
4.12.1	Detailed Description	28

4.12.2	Function Documentation	28
4.12.2.1	menuLevel	28
4.13	menu_level.h File Reference	29
4.13.1	Detailed Description	30
4.13.2	Function Documentation	30
4.13.2.1	menuLevel	30
4.14	player.c File Reference	31
4.14.1	Detailed Description	32
4.14.2	Function Documentation	32
4.14.2.1	blitCharacter	32
4.14.2.2	collisionSprite	32
4.14.2.3	creatorCharacter	32
4.14.2.4	jumping	33
4.14.2.5	moveCharacter	33
4.14.2.6	movementVector	34
4.15	player.h File Reference	34
4.15.1	Detailed Description	36
4.15.2	Macro Definition Documentation	36
4.15.2.1	ABS	36
4.15.2.2	SGN	36
4.15.3	Function Documentation	36
4.15.3.1	blitCharacter	36
4.15.3.2	collisionSprite	36
4.15.3.3	creatorCharacter	37
4.15.3.4	jumping	38
4.15.3.5	moveCharacter	38
4.15.3.6	movementVector	39
4.16	sound.c File Reference	39
4.16.1	Detailed Description	40
4.16.2	Function Documentation	40
4.16.2.1	createSound	40
4.16.2.2	freeSound	41
4.16.2.3	playMusic	42
4.16.2.4	playMusicOnce	42
4.16.2.5	soundVolume	42
4.16.2.6	stopSound	42
4.17	sound.h File Reference	42
4.17.1	Detailed Description	44
4.17.2	Function Documentation	44
4.17.2.1	createSound	44

4.17.2.2	freeSound	44
4.17.2.3	playMusic	44
4.17.2.4	playMusicOnce	44
4.17.2.5	soundVolume	44
4.17.2.6	stopSound	45
4.18	text.c File Reference	45
4.18.1	Detailed Description	45
4.18.2	Function Documentation	45
4.18.2.1	printText	46
4.19	text.h File Reference	47
4.19.1	Detailed Description	48
4.19.2	Function Documentation	48
4.19.2.1	printText	48
Index		49

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Character	5
Level	5
Map	6
Sound	6

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	7
file.c	Fonctions d'accès aux fichiers	8
file.h	Prototypes des fonctions d'accès aux fichiers	10
file_level.h	??
image.c	Contient les fonctions liées aux images	12
image.h	Contient les fonctions liées aux images	13
jeu.c	Contient les fonctions liées au jeu	14
jeu.h	Header de jeu.c	18
map.c	Contient les fonctions liées au chargement et à l'affichage de la carte	22
map.h	Header de map.c	23
menu.c	Contient les fonctions liées au menu	25
menu.h	Header de menu.c	26
menu_level.c	Menu gérant le choix du niveau	27
menu_level.h	Menu gérant le choix du niveau	29
player.c	Contient les fonctions pour manipuler le joueur	31
player.h	Header de player.c	34
share.h	??
sound.c	Contient les fonctions pour jouer du son	39
sound.h	Header de sound.c	42
text.c	Contient les fonctions pour afficher du texte à l'écran	45

[text.h](#)Header de [text.c](#) 47

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](#)

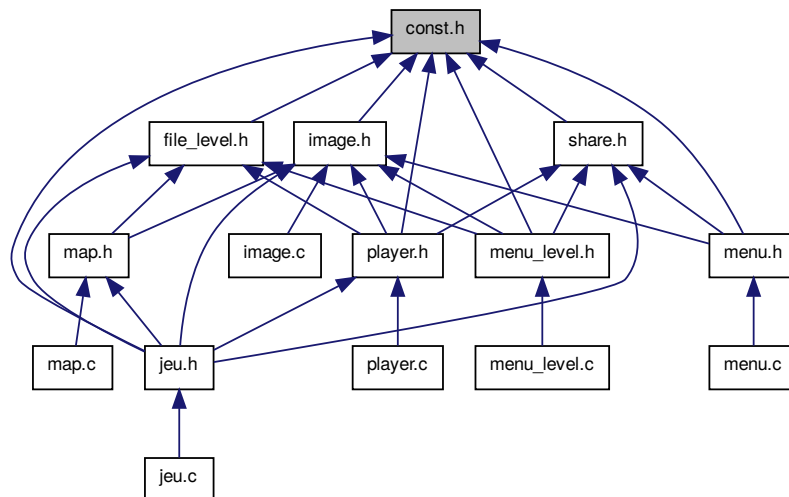
Chapter 4

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

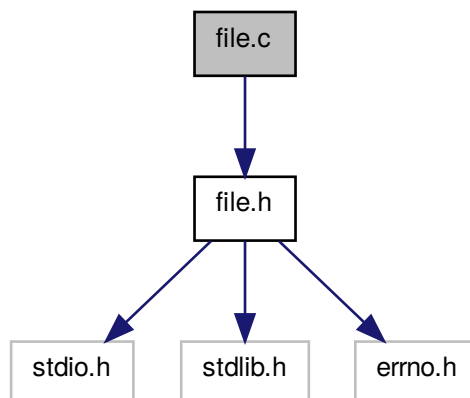
2014-02-27

4.2 file.c File Reference

Fonctions d'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	*ptr_fichier	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	nom[]	le nom du fichier
in	mode[]	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	*ptr_fichier	le fichier
----	--------------	------------

Returns

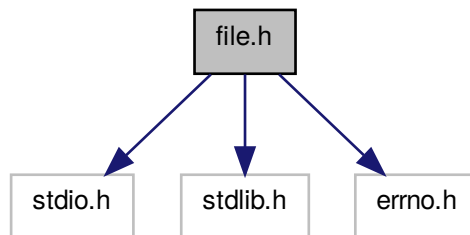
entier ayant la taille du fichier

4.3 file.h File Reference

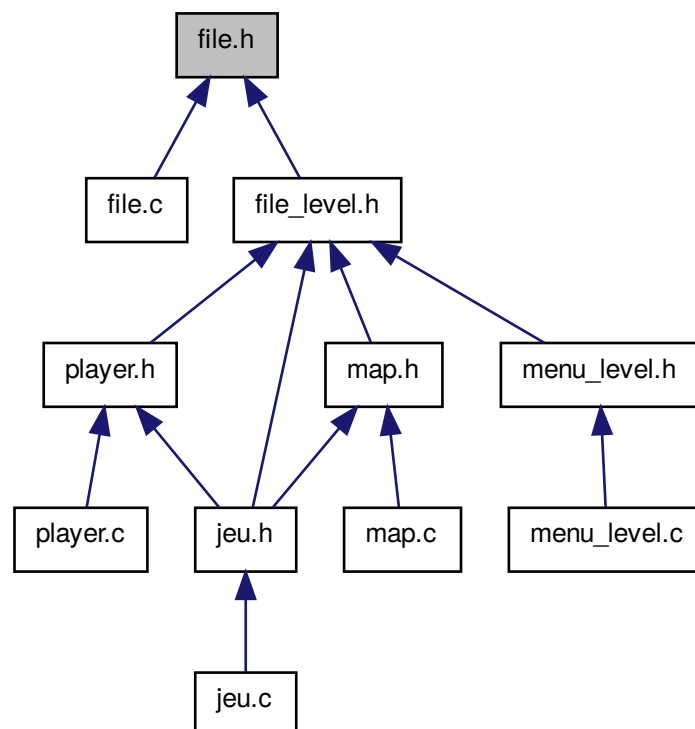
Prototypes des fonctions d'accès aux fichiers.

```
#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

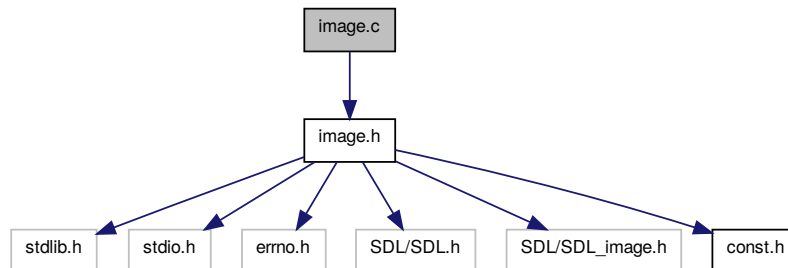
entier ayant la taille du fichier

4.4 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

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

Returns

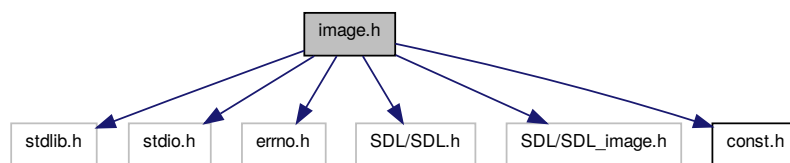
un pointeur sur une SDL_Surface

4.5 image.h File Reference

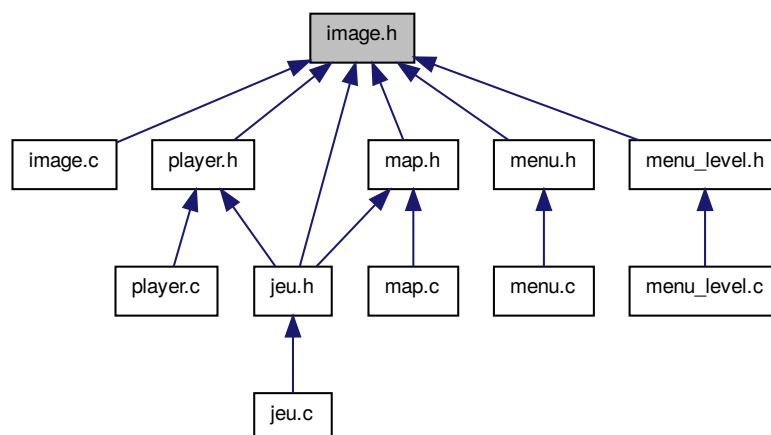
contient les fonction liées aux images

```
#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	<i>file_name</i>	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	<i>file_name</i>	le nom du fichier
----	------------------	-------------------

Returns

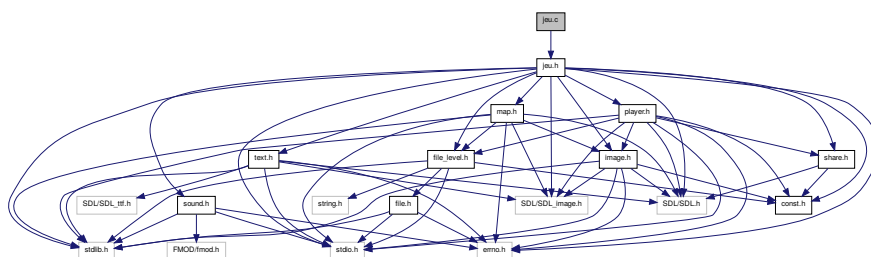
un pointeur sur une `SDL_Surface`

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

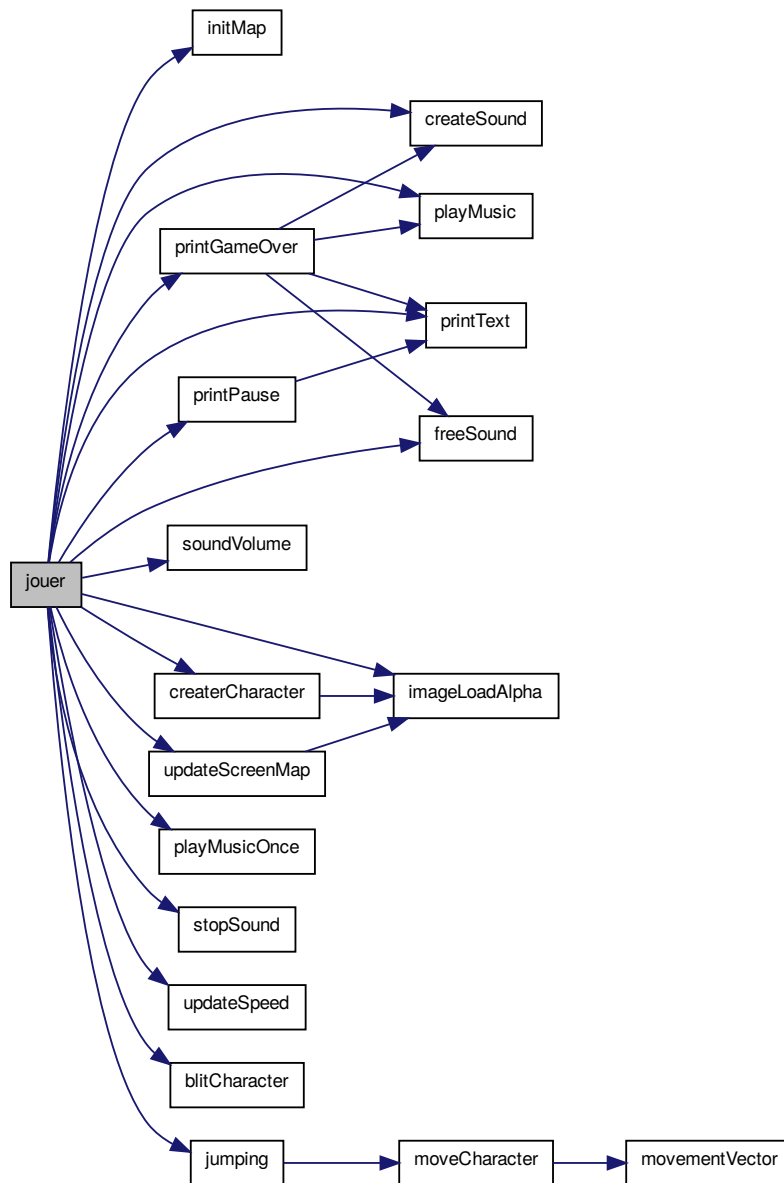
4.6.2.1 void jouer (SDL_Surface * screen, char * level_name)

contient la boucle principale du jeu qui appelle les fonctions

Parameters

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

Here is the call graph for this function:



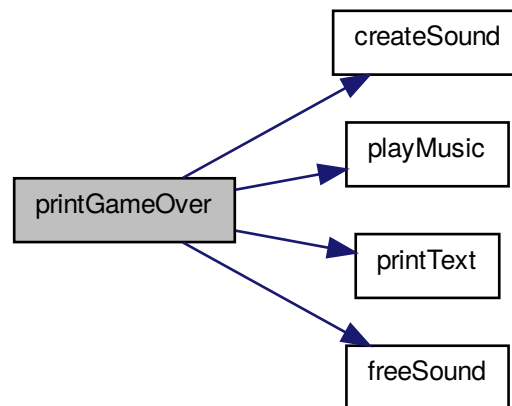
4.6.2.2 void printGameOver (SDL_Surface * screen, int * continuer)

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	<i>screen</i>	l'écran de jeu
out	<i>time</i>	le temps restant
out	<i>event</i>	l'événement en cours

Here is the call graph for this function:



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

Met à jour la vitesse

Parameters

out	<i>float</i>	la vitesse
-----	--------------	------------

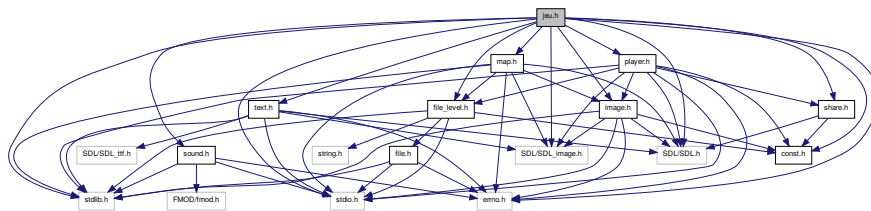
out	<i>acceleration</i>	l'acceleration
-----	---------------------	----------------

4.7 jeu.h File Reference

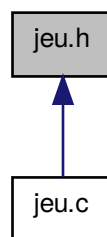
header de [jeu.c](#)

```
#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

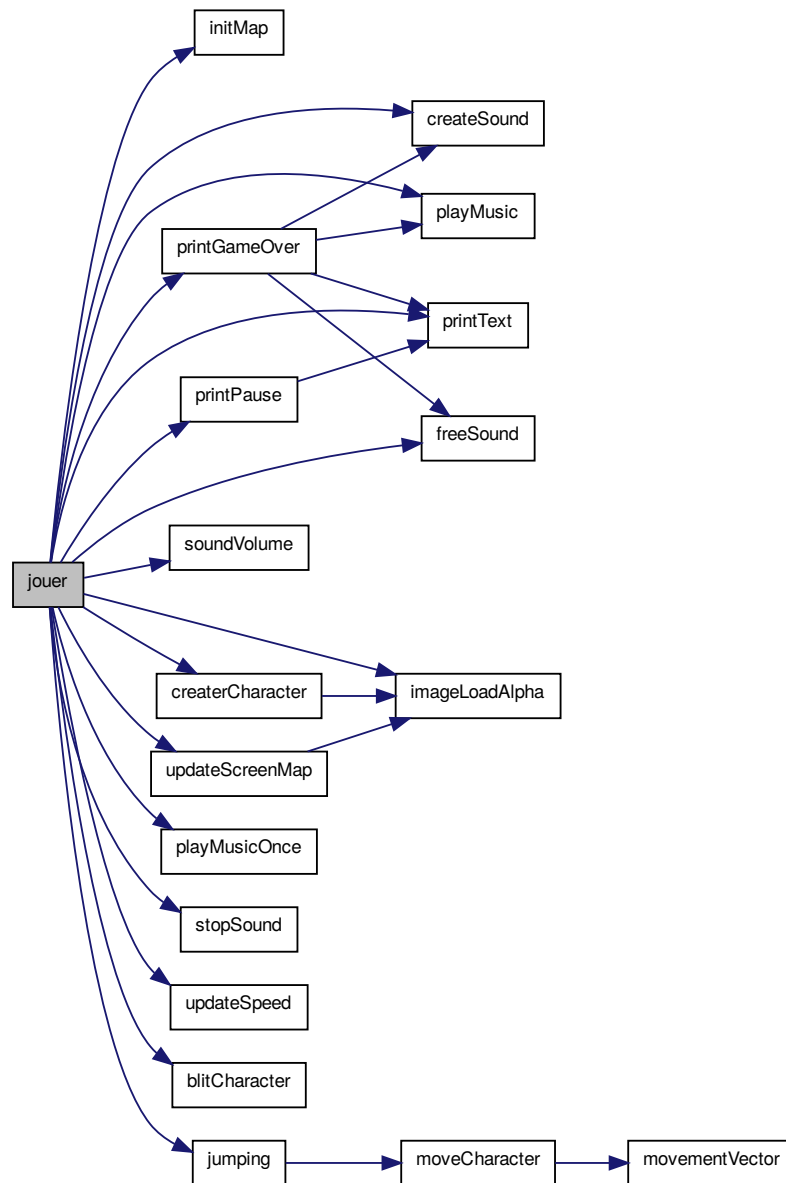
4.7.2.1 void jouer (SDL_Surface * screen, char * level_name)

contient la boucle principale du jeu qui appelle les fonctions

Parameters

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

Here is the call graph for this function:



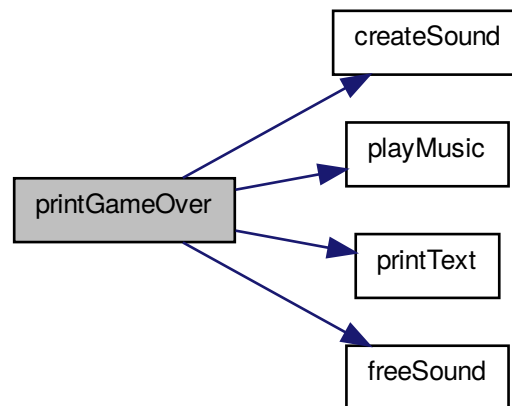
4.7.2.2 void printGameOver (SDL_Surface * screen, int * continuer)

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	<i>screen</i>	l'écran de jeu
out	<i>time</i>	le temps restant
out	<i>event</i>	l'événement en cours

Here is the call graph for this function:



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

Met à jour la vitesse

Parameters

out	<i>float</i>	la vitesse
-----	--------------	------------

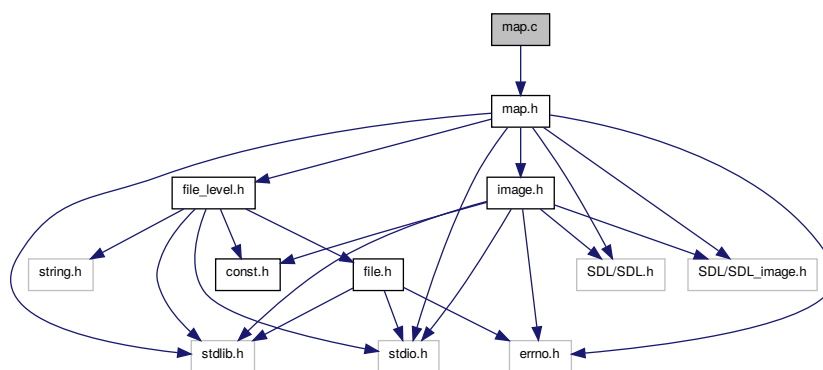
out	<i>acceleration</i>	l'acceleration
-----	---------------------	----------------

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	<i>screen</i>	l'écran de jeu
in	<i>level_name</i>	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	<i>map</i>	Le niveau à gérer
in	<i>direction</i>	La direction de scrolling
in	<i>speed</i>	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	<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:



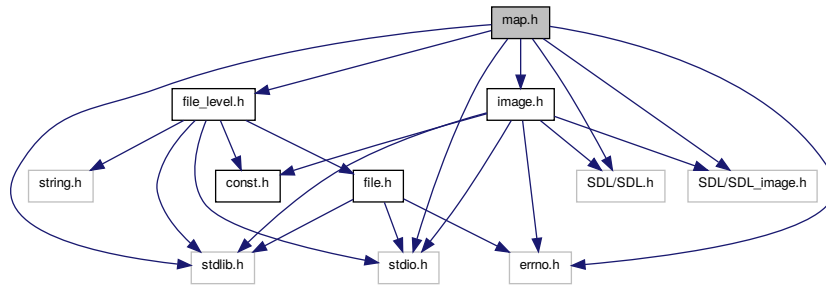
4.9 map.h File Reference

header de [map.c](#)

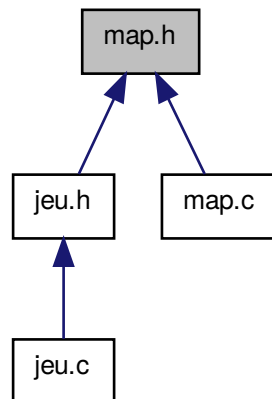
```

#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	<i>screen</i>	l'écran de jeu
in	<i>level_name</i>	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	<i>map</i>	Le niveau à gérer
in	<i>direction</i>	La direction de scrolling
in	<i>speed</i>	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	<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:

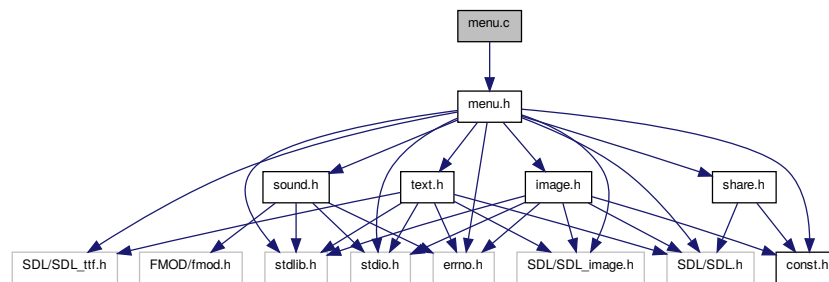


4.10 menu.c File Reference

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

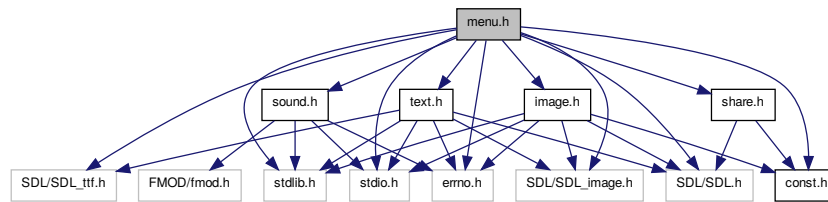
header de [menu.c](#)

```

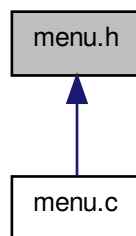
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <SDL/SDL.h>
#include <SDL/SDL_image.h>
#include <SDL/SDL_ttf.h>
#include "const.h"
#include "text.h"
#include "sound.h"
#include "share.h"
#include "image.h"

```


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

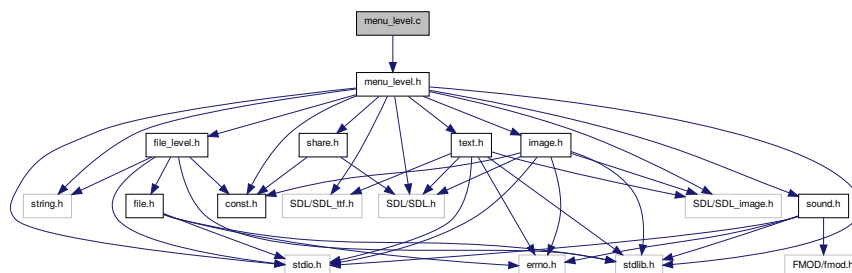
2014-02-27

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

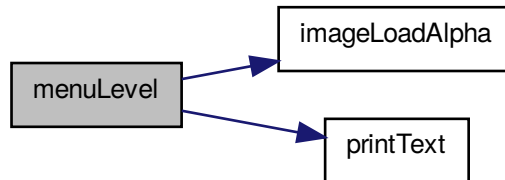
4.12.2.1 int [menuLevel](#) (SDL_Surface * *screen*, char *level_name*[TAILLE_MAX_NOM_FICHIER], [Sound](#) * *s*)

Menu pour 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:

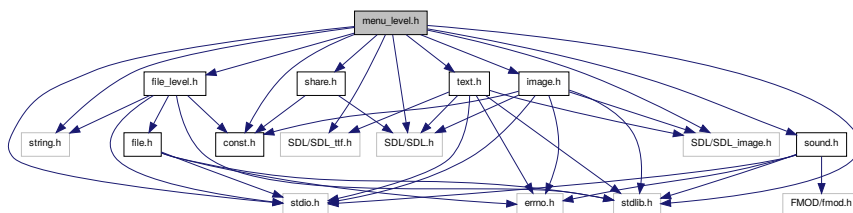


4.13 menu_level.h File Reference

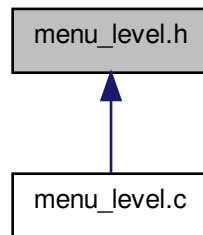
Menu gerant le choix du niveau.

```
#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

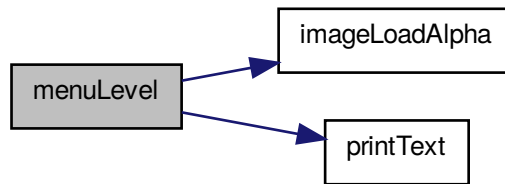
4.13.2.1 int `menuLevel` (`SDL_Surface` * *screen*, char *level_name*[`TAILLE_MAX_NOM_FICHIER`], `Sound` * *s*)

Menu pour 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:

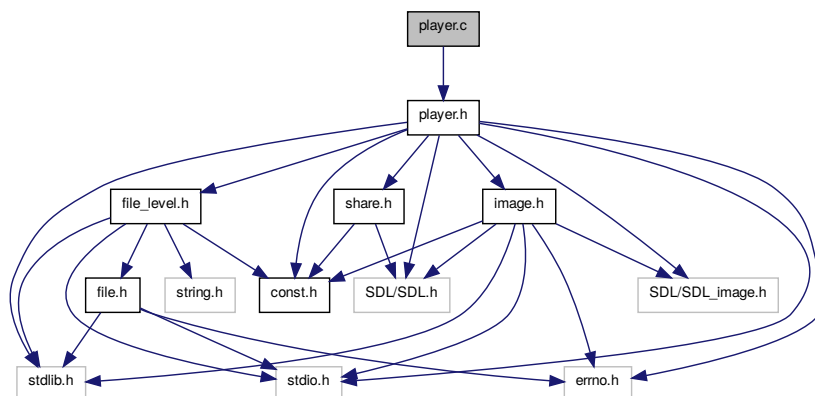


4.14 player.c File Reference

contient les fonction pour manipuler le joueur

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

Include dependency graph for player.c:



Functions

- `Character * creatorCharacter` (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	<i>screen</i>	L'écran
in	<i>c</i>	Le personnage
in	<i>m</i>	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	<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

4.14.2.3 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:

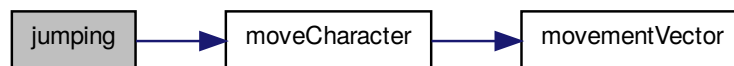
4.14.2.4 void jumping (Character * *c*, Map * *m*)

make the character jump

Parameters

in, out	<i>c</i>	the Character
in	<i>m</i>	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	<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:



4.14.2.6 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	vx	the horizontal component of the vector
out	vy	the vertical component of the vector
in	<i>speed</i>	the speed of the move
out	c	the Character you have to move

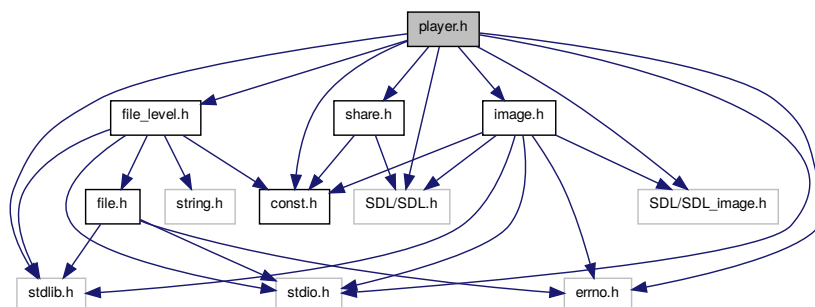
4.15 player.h File Reference

header de [player.c](#)

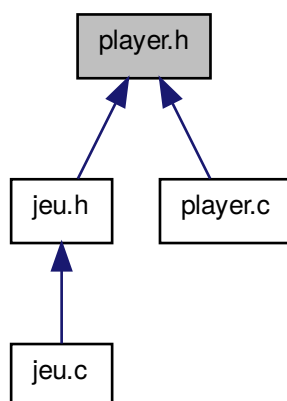
```

#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](#) * [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)

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

<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

4.15.3.2 **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

4.15.3.3 **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:

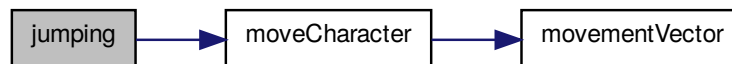
**4.15.3.4 void jumping (Character * c, Map * m)**

make the character jump

Parameters

in, out	<i>c</i>	the Character
in	<i>m</i>	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	<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:



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

create a movement vector

Parameters

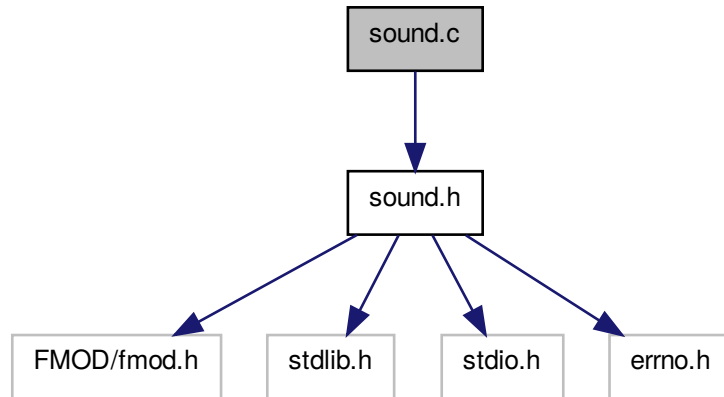
in	<i>direction</i>	The movement's direction
out	<i>vx</i>	the horizontal component of the vector
out	<i>vy</i>	the vertical component of the vector
in	<i>speed</i>	the speed of the move
out	<i>c</i>	the Character you have to move

4.16 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

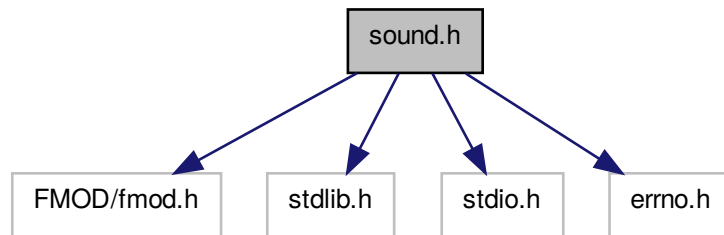
out	the	sound to stop
-----	-----	---------------

4.17 sound.h File Reference

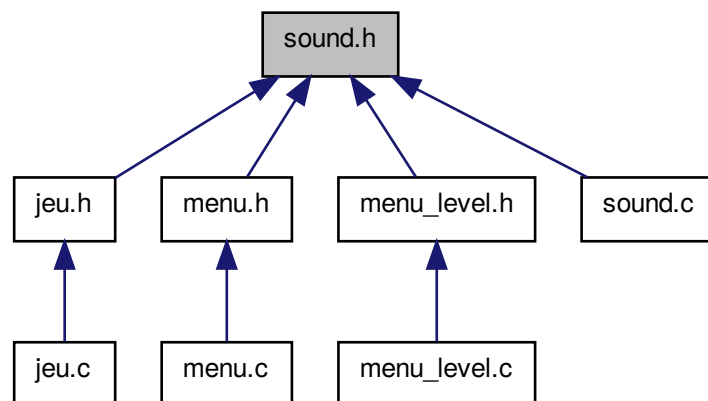
header de [sound.c](#)

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

4.17.2.6 void stopSound (Sound * s)

stop the sound

Parameters

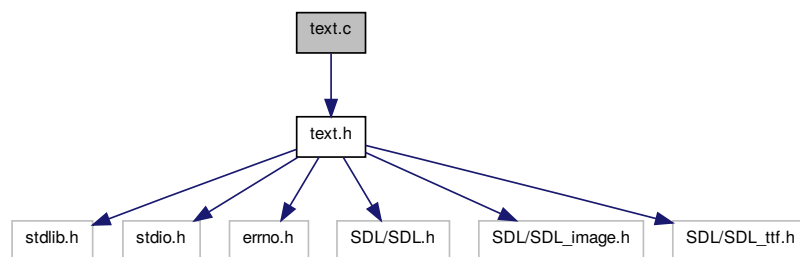
out	<i>the</i>	sound to stop
-----	------------	---------------

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

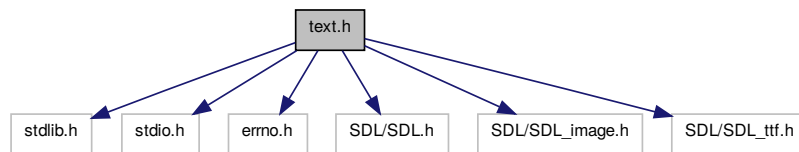
out	<i>screen</i>	L'écran
in	<i>posText</i>	La position du texte à afficher ; si NULL, centré en largeur et hauteur
in	<i>text</i>	Le texte à afficher
in	<i>color</i>	la couleur du texte
in	<i>font</i>	L'adresse de la police d'affichage (.ttf)
in	<i>ptSize</i>	la taille du texte à afficher
in	<i>mode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

4.19 text.h File Reference

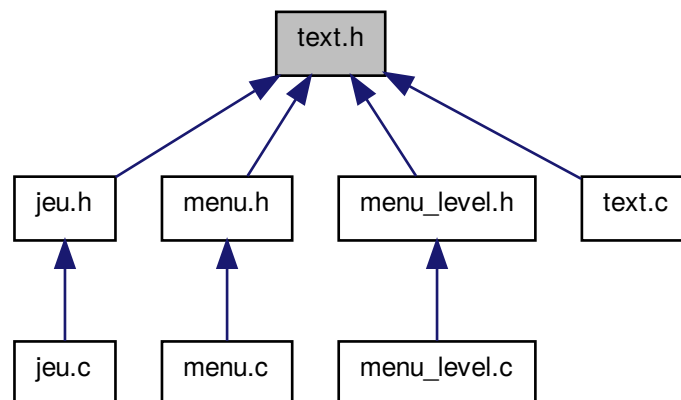
header de [text.c](#)

```
#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	<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>color</i>	la couleur du texte
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, 36
- blitCharacter
 - player.c, 32
 - player.h, 36
- Character, 5
 - isJumping, 5
 - isOnGround, 5
- closeFile
 - file.c, 9
 - file.h, 11
- collisionSprite
 - player.c, 32
 - player.h, 36
- const.h, 7
- createSound
 - sound.c, 40
 - sound.h, 44
- createCharacter
 - player.c, 32
 - player.h, 36
- file.c, 8
 - closeFile, 9
 - openFile, 9
 - readFileSize, 9
- file.h, 10
 - closeFile, 11
 - openFile, 11
 - readFileSize, 11
- freeSound
 - sound.c, 40
 - sound.h, 44
- image.c, 12
 - imageLoad, 12
 - imageLoadAlpha, 12
- image.h, 13
 - imageLoad, 14
 - imageLoadAlpha, 14
- imageLoad
 - image.c, 12
 - image.h, 14
- imageLoadAlpha
 - image.c, 12
 - image.h, 14
- initMap
 - map.c, 22
 - map.h, 25
- isJumping
 - Character, 5
- isOnGround
 - Character, 5
- jeu.c, 14
 - jouer, 15
 - printGameOver, 16
 - printPause, 17
 - updateSpeed, 17
- jeu.h, 18
 - jouer, 19
 - printGameOver, 20
 - printPause, 21
 - updateSpeed, 21
- jouer
 - jeu.c, 15
 - jeu.h, 19
- jumping
 - player.c, 33
 - player.h, 38
- Level, 5
- Map, 6
 - map.c, 22
 - initMap, 22
 - scrolling, 23
 - updateScreenMap, 23
 - map.h, 23
 - initMap, 25
 - scrolling, 25
 - updateScreenMap, 25
- menu.c, 25
- menu.h, 26
- menu_level.c, 27
 - menuLevel, 28
- menu_level.h, 29
 - menuLevel, 30
- menuLevel
 - menu_level.c, 28
 - menu_level.h, 30
- moveCharacter
 - player.c, 33
 - player.h, 38
- movementVector
 - player.c, 34
 - player.h, 39
- openFile

- file.c, [9](#)
- file.h, [11](#)
- playMusic
 - sound.c, [42](#)
 - sound.h, [44](#)
- playMusicOnce
 - sound.c, [42](#)
 - sound.h, [44](#)
- player.c, [31](#)
 - blitCharacter, [32](#)
 - collisionSprite, [32](#)
 - creatorCharacter, [32](#)
 - jumping, [33](#)
 - moveCharacter, [33](#)
 - movementVector, [34](#)
- player.h, [34](#)
 - ABS, [36](#)
 - blitCharacter, [36](#)
 - collisionSprite, [36](#)
 - creatorCharacter, [36](#)
 - jumping, [38](#)
 - moveCharacter, [38](#)
 - movementVector, [39](#)
 - SGN, [36](#)
- printGameOver
 - jeu.c, [16](#)
 - jeu.h, [20](#)
- printPause
 - jeu.c, [17](#)
 - jeu.h, [21](#)
- printText
 - text.c, [45](#)
 - text.h, [48](#)
- readFileSize
 - file.c, [9](#)
 - file.h, [11](#)
- SGN
 - player.h, [36](#)
- scrolling
 - map.c, [23](#)
 - map.h, [25](#)
- Sound, [6](#)
- sound.c, [39](#)
 - createSound, [40](#)
 - freeSound, [40](#)
 - playMusic, [42](#)
 - playMusicOnce, [42](#)
 - soundVolume, [42](#)
 - stopSound, [42](#)
- sound.h, [42](#)
 - createSound, [44](#)
 - freeSound, [44](#)
 - playMusic, [44](#)
 - playMusicOnce, [44](#)
 - soundVolume, [44](#)
 - stopSound, [45](#)
- soundVolume
 - sound.c, [42](#)
 - sound.h, [44](#)
- stopSound
 - sound.c, [42](#)
 - sound.h, [45](#)
- text.c, [45](#)
 - printText, [45](#)
- text.h, [47](#)
 - printText, [48](#)
- updateScreenMap
 - map.c, [23](#)
 - map.h, [25](#)
- updateSpeed
 - jeu.c, [17](#)
 - jeu.h, [21](#)