

Hashiwokakero SDL

Généré par Doxygen 1.8.11

Table des matières

1	Index des classes	1
1.1	Liste des classes	1
2	Index des fichiers	3
2.1	Liste des fichiers	3
3	Documentation des classes	5
3.1	Référence de la structure <code>Env_t</code>	5
3.1.1	Description détaillée	6
4	Documentation des fichiers	7
4.1	Référence du fichier <code>model.h</code>	7
4.1.1	Description détaillée	8
4.1.2	Documentation des fonctions	8
4.1.2.1	<code>add_bridge_SDL(Env *env, SDL_Renderer *ren)</code>	8
4.1.2.2	<code>clean(SDL_Window *win, SDL_Renderer *ren, Env *env)</code>	8
4.1.2.3	<code>clean_game_SDL(Env *env)</code>	9
4.1.2.4	<code>del_bridge_SDL(Env *env, SDL_Renderer *ren, dir direction_add)</code>	9
4.1.2.5	<code>init(SDL_Window *win, SDL_Renderer *ren, int argc, char *argv[])</code>	9
4.1.2.6	<code>init_game_SDL(Env *env, char *argv)</code>	9
4.1.2.7	<code>process(SDL_Window *win, SDL_Renderer *ren, Env *env, SDL_Event *e)</code>	10
4.1.2.8	<code>process_gagner_souris(SDL_Window *win, SDL_Renderer *ren, Env *env, SDL_Event *e)</code>	10
4.1.2.9	<code>process_game_souris(SDL_Window *win, SDL_Renderer *ren, Env *env, SDL_Event *e)</code>	10

4.1.2.10	<code>process_menu_souris(SDL_Window *win, SDL_Renderer *ren, Env *env, SDL↵ _Event *e)</code>	11
4.1.2.11	<code>process_niveau_souris(SDL_Window *win, SDL_Renderer *ren, Env *env, S↵ DL_Event *e)</code>	11
4.1.2.12	<code>process_regle_souris(SDL_Window *win, SDL_Renderer *ren, Env *env, SDL↵ _Event *e)</code>	11
4.1.2.13	<code>render(SDL_Window *win, SDL_Renderer *ren, Env *env)</code>	12
4.1.2.14	<code>render_gagner_souris(SDL_Window *win, SDL_Renderer *ren, Env *env)</code>	12
4.1.2.15	<code>render_game_souris(SDL_Window *win, SDL_Renderer *ren, Env *env)</code>	12
4.1.2.16	<code>render_menu_souris(SDL_Window *win, SDL_Renderer *ren, Env *env)</code>	12
4.1.2.17	<code>render_niveau_souris(SDL_Window *win, SDL_Renderer *ren, Env *env)</code>	13
4.1.2.18	<code>render_regle_souris(SDL_Window *win, SDL_Renderer *ren, Env *env)</code>	13
Index		15

Chapitre 1

Index des classes

1.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

Env_t

La structure `Env_t` est une structure qui stock les `SDL_Surface` utilisées ainsi que les noeud et le jeu utilisé pour le jeu en version graphique

5

Chapitre 2

Index des fichiers

2.1 Liste des fichiers

Liste de tous les fichiers documentés avec une brève description :

model.h	Ce fichier contient les fonctions permettant d'initialiser la fenêtre, ainsi que le jeu	7
-------------------------	---	-------------------

Chapitre 3

Documentation des classes

3.1 Référence de la structure Env_t

La structure [Env_t](#) est une structure qui stock les SDL_Surface utilisées ainsi que les noeud et le jeu utilisé pour le jeu en version graphique.

```
#include <model.h>
```

Attributs publics

- SDL_Texture * **background**
- SDL_Texture * **titre**
- SDL_Texture * **nouv**
- SDL_Texture * **nouvbis**
- SDL_Texture * **nouvor**
- SDL_Texture * **regle**
- SDL_Texture * **regletitre**
- SDL_Texture * **reglebis**
- SDL_Texture * **regleor**
- SDL_Texture * **regle1**
- SDL_Texture * **regle2**
- SDL_Texture * **regle3**
- SDL_Texture * **regle3bis**
- SDL_Texture * **regle4**
- SDL_Texture * **regle4bis**
- SDL_Texture * **quitter**
- SDL_Texture * **quitterbis**
- SDL_Texture * **quitteror**
- SDL_Texture * **fleche**
- SDL_Texture * **flechebis**
- SDL_Texture * **flecheor**
- SDL_Texture * **choix**
- SDL_Texture * **choixbis**
- SDL_Texture * **easy**
- SDL_Texture * **easybis**
- SDL_Texture * **easyor**
- SDL_Texture * **medium**
- SDL_Texture * **mediumbis**
- SDL_Texture * **mediumor**
- SDL_Texture * **hard**
- SDL_Texture * **hardbis**
- SDL_Texture * **hardor**
- SDL_Texture * **help**
- SDL_Texture * **helpbis**
- SDL_Texture * **helpor**
- SDL_Texture * **sol**
- SDL_Texture * **solbis**
- SDL_Texture * **solor**

- `SDL_Texture * home`
- `SDL_Texture * homebis`
- `SDL_Texture * homeor`
- `SDL_Texture * gagner`
- `SDL_Texture * tmp`
- `SDL_Texture * degree [9]`
- `SDL_Texture * degreeor [9]`
- `SDL_Texture * nodes [500]`
- `SDL_Texture * un`
- `SDL_Texture * deux`
- `SDL_Texture * trois`
- `SDL_Texture * quatre`
- `SDL_Texture * cinq`
- `SDL_Texture * six`
- `SDL_Texture * sept`
- `SDL_Texture * huit`
- `SDL_Texture * pontsimpleh`
- `SDL_Texture * pontdoubleh`
- `SDL_Texture * pontsimplev`
- `SDL_Texture * pontdoublev`
- `SDL_Rect * coordonnes [500][4]`
- `game * n_game`
- `node * tab_n`
- `int * node_num`
- `int * x_max`
- `int * y_max`
- `int * nodes_to_add [2]`
- `int * lock`
- `int * load`
- `int ecran`

3.1.1 Description détaillée

La structure `Env_t` est une structure qui stock les `SDL_Surface` utilisées ainsi que les noeud et le jeu utilisé pour le jeu en version graphique.

La documentation de cette structure a été générée à partir du fichier suivant :

- `model.c`

Chapitre 4

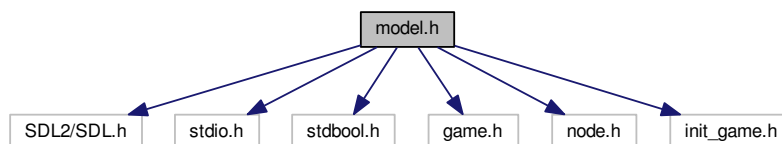
Documentation des fichiers

4.1 Référence du fichier model.h

Ce fichier contient les fonctions permettant d'initialiser la fenêtre, ainsi que le jeu.

```
#include <SDL2/SDL.h>
#include <stdio.h>
#include <stdbool.h>
#include "game.h"
#include "node.h"
#include "init_game.h"
```

Graphe des dépendances par inclusion de model.h :



Macros

- `#define PRINT(STR, ...) do { printf(STR, ## __VA_ARGS__); } while(0)`
- `#define ERROR(STR, ...) do { fprintf(stderr, STR, ## __VA_ARGS__); exit(EXIT_FAILURE); } while(0)`

Définitions de type

- `typedef struct Env_t Env`

Fonctions

- `Env * init` (SDL_Window *win, SDL_Renderer *ren, int argc, char *argv[])
Initialise les surfaces utilisées pour la fenêtre SDL.
- void `render_menu_souris` (SDL_Window *win, SDL_Renderer *ren, `Env` *env)
Affiche le menu (dans lequel on se déplacera avec la souris).
- void `render_niveau_souris` (SDL_Window *win, SDL_Renderer *ren, `Env` *env)
Affiche le menu de sélection du niveau.
- void `render_regle_souris` (SDL_Window *win, SDL_Renderer *ren, `Env` *env)
Affiche les règles.
- void `render_game_souris` (SDL_Window *win, SDL_Renderer *ren, `Env` *env)
Affiche le jeu et permet de jouer à la souris.
- void `render_gagner_souris` (SDL_Window *win, SDL_Renderer *ren, `Env` *env)
Affiche à l'écran l'inscription "Vous avez gagné".
- void `render` (SDL_Window *win, SDL_Renderer *ren, `Env` *env)
Utilise les autres fonctions render pour l'affichage.*
- bool `process` (SDL_Window *win, SDL_Renderer *ren, `Env` *env, SDL_Event *e)
Effectue telle ou telle action suivant ce que fait l'utilisateur.
- bool `process_menu_souris` (SDL_Window *win, SDL_Renderer *ren, `Env` *env, SDL_Event *e)
Effectue telle ou telle action suivant ce que fait l'utilisateur sur le menu principal.
- bool `process_niveau_souris` (SDL_Window *win, SDL_Renderer *ren, `Env` *env, SDL_Event *e)
Effectue telle ou telle action suivant ce que fait l'utilisateur sur le menu de sélection du niveau.
- bool `process_regle_souris` (SDL_Window *win, SDL_Renderer *ren, `Env` *env, SDL_Event *e)
Effectue telle ou telle action suivant ce que fait l'utilisateur sur la page indiquant les règles.
- bool `process_game_souris` (SDL_Window *win, SDL_Renderer *ren, `Env` *env, SDL_Event *e)
Effectue telle ou telle action suivant ce que fait l'utilisateur pendant qu'il joue.
- bool `process_gagner_souris` (SDL_Window *win, SDL_Renderer *ren, `Env` *env, SDL_Event *e)
Effectue telle ou telle action suivant ce que fait l'utilisateur sur l'écran lui disant qu'il a gagné.
- void `add_bridge SDL` (`Env` *env, SDL_Renderer *ren)
Permet d'ajouter un pont entre deux îles.
- void `del_bridge SDL` (`Env` *env, SDL_Renderer *ren, dir direction_add)
Permet de supprimer les ponts dans une direction si le degré maximal a été atteint.
- void `init_game SDL` (`Env` *env, char *argv)
Initialise le jeu par rapport au fichier passé en paramètre.
- void `clean_game SDL` (`Env` *env)
Détruit le tableau de noeud et le jeu.
- void `clean` (SDL_Window *win, SDL_Renderer *ren, `Env` *env)
Permet d'ajouter un pont entre deux îles.

4.1.1 Description détaillée

Ce fichier contient les fonctions permettant d'initialiser la fenêtre, ainsi que le jeu.

4.1.2 Documentation des fonctions

4.1.2.1 void add_bridge SDL (Env * env, SDL_Renderer * ren)

Permet d'ajouter un pont entre deux îles.

Paramètres

<code>env</code>	: structure <code>Env_t</code> contenant le jeu.
<code>ren</code>	: SDL_Renderer utilisé dans win.

4.1.2.2 void clean (SDL_Window * win, SDL_Renderer * ren, Env * env)

Permet d'ajouter un pont entre deux îles.

Paramètres

<i>win</i>	: SDL_Window fenêtre SDL principale.
<i>ren</i>	: SDL_Renderer utilisé dans win.
<i>env</i>	: structure Env_t contenant le jeu.

4.1.2.3 void clean_game_SDL (Env * env)

Détruit le tableau de noeud et le jeu.

Paramètres

<i>env</i>	: structure Env_t contenant le jeu et le tableau de noeud.
------------	--

4.1.2.4 void del_bridge_SDL (Env * env, SDL_Renderer * ren, dir direction_add)

Permet de supprimer les ponts dans une direction si le degré maximal a été atteint.

Paramètres

<i>env</i>	: structure Env_t contenant le jeu.
<i>ren</i>	: SDL_Renderer utilisé dans win.
<i>direction_add</i>	: direction dans laquelle le pont doit être supprimé.

4.1.2.5 Env* init (SDL_Window * win, SDL_Renderer * ren, int argc, char * argv[])

Initialise les surfaces utilisées pour la fenêtre SDL.

Paramètres

<i>win</i>	: SDL_Window qui sera initialisée.
<i>ren</i>	: SDL_Renderer utilisé dans win.
<i>argc</i>	: le nombre d'argument passé en paramètre.
<i>argv</i>	: ce paramètre contiendra le nom du fichier à ouvrir.

Renvoie

Une structure [Env_t](#).

4.1.2.6 void init_game_SDL (Env * env, char * argv)

Initialise le jeu par rapport au fichier passé en paramètre.

Paramètres

<i>env</i>	: structure Env_t contenant le jeu.
<i>argv</i>	: nom du fichier passé en paramètre.

4.1.2.7 `bool process (SDL_Window * win, SDL_Renderer * ren, Env * env, SDL_Event * e)`

Effectue telle ou telle action suivant ce que fait l'utilisateur.

Paramètres

<i>win</i>	: SDL_Window qui sera initialisée.
<i>ren</i>	: SDL_Renderer utilisé dans win.
<i>env</i>	: structure Env_t .
<i>e</i>	: SDL_Event qui sert à savoir l'action que l'utilisateur a fait.

Renvoie

Un booléen.

4.1.2.8 `bool process_gagner_souris (SDL_Window * win, SDL_Renderer * ren, Env * env, SDL_Event * e)`

Effectue telle ou telle action suivant ce que fait l'utilisateur sur l'écran lui disant qu'il a gagné.

Paramètres

<i>win</i>	: SDL_Window qui sera initialisée.
<i>ren</i>	: SDL_Renderer utilisé dans win.
<i>env</i>	: structure Env_t .
<i>e</i>	: SDL_Event qui sert à savoir l'action que l'utilisateur a fait.

Renvoie

Un booléen.

4.1.2.9 `bool process_game_souris (SDL_Window * win, SDL_Renderer * ren, Env * env, SDL_Event * e)`

Effectue telle ou telle action suivant ce que fait l'utilisateur pendant qu'il joue.

Paramètres

<i>win</i>	: SDL_Window qui sera initialisée.
<i>ren</i>	: SDL_Renderer utilisé dans win.
<i>env</i>	: structure Env_t .
<i>e</i>	: SDL_Event qui sert à savoir l'action que l'utilisateur a fait.

Renvoie

Un booléen.

4.1.2.10 bool process_menu_souris (SDL_Window * win, SDL_Renderer * ren, Env * env, SDL_Event * e)

Effectue telle ou telle action suivant ce que fait l'utilisateur sur le menu principal.

Paramètres

<i>win</i>	: SDL_Window qui sera initialisée.
<i>ren</i>	: SDL_Renderer utilisé dans win.
<i>env</i>	: structure Env_t .
<i>e</i>	: SDL_Event qui sert à savoir l'action que l'utilisateur à fait.

Renvoie

Un booléen.

4.1.2.11 bool process_niveau_souris (SDL_Window * win, SDL_Renderer * ren, Env * env, SDL_Event * e)

Effectue telle ou telle action suivant ce que fait l'utilisateur sur le menu de sélection du niveau.

Paramètres

<i>win</i>	: SDL_Window qui sera initialisée.
<i>ren</i>	: SDL_Renderer utilisé dans win.
<i>env</i>	: structure Env_t .
<i>e</i>	: SDL_Event qui sert à savoir l'action que l'utilisateur à fait.

Renvoie

Un booléen.

4.1.2.12 bool process_regle_souris (SDL_Window * win, SDL_Renderer * ren, Env * env, SDL_Event * e)

Effectue telle ou telle action suivant ce que fait l'utilisateur sur la page indiquant les règles.

Paramètres

<i>win</i>	: SDL_Window qui sera initialisée.
<i>ren</i>	: SDL_Renderer utilisé dans win.
<i>env</i>	: structure Env_t .
<i>e</i>	: SDL_Event qui sert à savoir l'action que l'utilisateur à fait.

Renvoie

Un booléen.

4.1.2.13 void render (SDL_Window * win, SDL_Renderer * ren, Env * env)

Utilise les autres fonctions render* pour l'affichage.

Paramètres

<i>win</i>	: SDL_Window qui sera initialisée.
<i>ren</i>	: SDL_Renderer utilisé dans win.
<i>env</i>	: structure Env_t .

4.1.2.14 void render_gagner_souris (SDL_Window * win, SDL_Renderer * ren, Env * env)

Affiche à l'écran l'inscription "Vous avez gagné" .

Paramètres

<i>win</i>	: SDL_Window qui sera initialisée.
<i>ren</i>	: SDL_Renderer utilisé dans win.
<i>env</i>	: structure Env_t .

4.1.2.15 void render_game_souris (SDL_Window * win, SDL_Renderer * ren, Env * env)

Affiche le jeu et permet de jouer à la souris.

Paramètres

<i>win</i>	: SDL_Window qui sera initialisée.
<i>ren</i>	: SDL_Renderer utilisé dans win.
<i>env</i>	: structure Env_t .

4.1.2.16 void render_menu_souris (SDL_Window * win, SDL_Renderer * ren, Env * env)

Affiche le menu (dans lequel on se déplacera avec la souris).

Paramètres

<i>win</i>	: SDL_Window qui sera initialisée.
<i>ren</i>	: SDL_Renderer utilisé dans win.
<i>env</i>	: structure Env_t .

4.1.2.17 void render_niveau_souris (SDL_Window * *win*, SDL_Renderer * *ren*, Env * *env*)

Affiche le menu de sélection du niveau.

Paramètres

<i>win</i>	: SDL_Window qui sera initialisée.
<i>ren</i>	: SDL_Renderer utilisé dans win.
<i>env</i>	: structure Env_t .

4.1.2.18 void render_regle_souris (SDL_Window * *win*, SDL_Renderer * *ren*, Env * *env*)

Affiche les règles.

Paramètres

<i>win</i>	: SDL_Window qui sera initialisée.
<i>ren</i>	: SDL_Renderer utilisé dans win.
<i>env</i>	: structure Env_t .

Index

add_bridge_SDL
model.h, [8](#)

clean
model.h, [8](#)
clean_game_SDL
model.h, [9](#)

del_bridge_SDL
model.h, [9](#)

Env_t, [5](#)

init
model.h, [9](#)
init_game_SDL
model.h, [9](#)

model.h, [7](#)
add_bridge_SDL, [8](#)
clean, [8](#)
clean_game_SDL, [9](#)
del_bridge_SDL, [9](#)
init, [9](#)
init_game_SDL, [9](#)
process, [10](#)
process_gagner_souris, [10](#)
process_game_souris, [10](#)
process_menu_souris, [11](#)
process_niveau_souris, [11](#)
process_regle_souris, [11](#)
render, [12](#)
render_gagner_souris, [12](#)
render_game_souris, [12](#)
render_menu_souris, [12](#)
render_niveau_souris, [12](#)
render_regle_souris, [13](#)

process
model.h, [10](#)
process_gagner_souris
model.h, [10](#)
process_game_souris
model.h, [10](#)
process_menu_souris
model.h, [11](#)
process_niveau_souris
model.h, [11](#)
process_regle_souris
model.h, [11](#)

render
model.h, [12](#)
render_gagner_souris
model.h, [12](#)
render_game_souris
model.h, [12](#)
render_menu_souris
model.h, [12](#)
render_niveau_souris
model.h, [12](#)
render_regle_souris
model.h, [13](#)