

Jeu 2D : Pokémon

Généré par Doxygen 1.8.6

Dimanche 17 Mai 2015 23 :22 :35



# Table des matières

<b>1</b>	<b>Jeu 2D type Pokémon</b>	<b>1</b>
<b>2</b>	<b>Index des classes</b>	<b>3</b>
2.1	Liste des classes . . . . .	3
<b>3</b>	<b>Index des fichiers</b>	<b>5</b>
3.1	Liste des fichiers . . . . .	5
<b>4</b>	<b>Documentation des classes</b>	<b>7</b>
4.1	Référence de la structure attaque . . . . .	7
4.1.1	Description détaillée . . . . .	7
4.1.2	Documentation des données membres . . . . .	7
4.1.2.1	degatBase . . . . .	7
4.1.2.2	imageAllie . . . . .	7
4.1.2.3	imageEnnemi . . . . .	7
4.1.2.4	nom . . . . .	7
4.1.2.5	precision . . . . .	8
4.1.2.6	type . . . . .	8
4.2	Référence de la structure pokemon . . . . .	8
4.2.1	Description détaillée . . . . .	8
4.2.2	Documentation des données membres . . . . .	8
4.2.2.1	imageAllie . . . . .	8
4.2.2.2	imageEnnemi . . . . .	8
4.2.2.3	nom . . . . .	8
4.2.2.4	type . . . . .	8
4.2.2.5	vie . . . . .	9
4.2.2.6	vieMax . . . . .	9
<b>5</b>	<b>Documentation des fichiers</b>	<b>11</b>
5.1	Référence du fichier constantes.h . . . . .	11
5.1.1	Description détaillée . . . . .	11
5.1.2	Documentation des macros . . . . .	12
5.1.2.1	HAUTEUR_FENETRE . . . . .	12

5.1.2.2	LARGEUR_FENETRE . . . . .	12
5.1.2.3	TAILLE_CASE . . . . .	12
5.1.3	Documentation du type de l'énumération . . . . .	12
5.1.3.1	orientationJoueur . . . . .	12
5.2	Référence du fichier credits.c . . . . .	12
5.2.1	Description détaillée . . . . .	12
5.2.2	Documentation des fonctions . . . . .	13
5.2.2.1	credits . . . . .	13
5.3	Référence du fichier credits.h . . . . .	13
5.3.1	Description détaillée . . . . .	13
5.3.2	Documentation des fonctions . . . . .	13
5.3.2.1	credits . . . . .	13
5.4	Référence du fichier discours.h . . . . .	14
5.4.1	Description détaillée . . . . .	14
5.5	Référence du fichier dresseur.c . . . . .	15
5.5.1	Description détaillée . . . . .	15
5.5.2	Documentation des fonctions . . . . .	15
5.5.2.1	coef . . . . .	15
5.5.2.2	egaliteAttaque . . . . .	16
5.5.2.3	egaliteStatistiques . . . . .	16
5.5.2.4	lancerCombat . . . . .	16
5.6	Référence du fichier dresseur.h . . . . .	16
5.6.1	Description détaillée . . . . .	17
5.6.2	Documentation des fonctions . . . . .	17
5.6.2.1	afficherTexteAttaque . . . . .	17
5.6.2.2	coef . . . . .	17
5.6.2.3	egaliteAttaque . . . . .	17
5.6.2.4	egaliteStatistiques . . . . .	17
5.6.2.5	lancerCombat . . . . .	17
5.7	Référence du fichier histoire.c . . . . .	18
5.7.1	Description détaillée . . . . .	18
5.7.2	Documentation des fonctions . . . . .	18
5.7.2.1	histoire . . . . .	18
5.7.2.2	peech . . . . .	18
5.8	Référence du fichier histoire.h . . . . .	18
5.8.1	Description détaillée . . . . .	19
5.8.2	Documentation des macros . . . . .	19
5.8.2.1	NB_FENETRE . . . . .	19
5.8.2.2	NB_MAX_CHARACTERE . . . . .	19
5.8.3	Documentation des fonctions . . . . .	19

5.8.3.1	histoire	19
5.8.3.2	peech	19
5.9	Référence du fichier jeu.c	20
5.9.1	Description détaillée	20
5.9.2	Documentation des fonctions	20
5.9.2.1	deplacerJoueur	20
5.9.2.2	jouer	20
5.9.2.3	remplirmap	21
5.9.2.4	remplirTableau	21
5.10	Référence du fichier jeu.h	21
5.10.1	Description détaillée	21
5.10.2	Documentation des fonctions	21
5.10.2.1	deplacerJoueur	22
5.10.2.2	jouer	22
5.10.2.3	remplirmap	22
5.10.2.4	remplirTableau	22
5.11	Référence du fichier main.c	22
5.11.1	Description détaillée	23
5.11.2	Documentation des fonctions	23
5.11.2.1	affichageMenu	23
5.11.2.2	chargementMenu	23
5.11.2.3	fermetureSDL	23
5.11.2.4	initialisationSDL	23
5.11.2.5	main	24
<b>Index</b>		<b>25</b>



# Chapitre 1

## Jeu 2D type Pokémon

Cette application est un jeu de rôle qui s'inspire grandement des jeux Pokémon.

Une fois l'application lancée le joueur est sur le menu, à partir de là il choisit de suivre l'histoire du jeu (h), de voir les crédits (c), ou bien de jouer au jeu (j).

Dans la partie jeu il y a deux interfaces entre lesquelles le joueur bascule de l'une à l'autre :

- Une interface carte où le joueur se déplace sur la carte et peut interagir avec l'extérieur
- Une interface combat où le joueur lance son Pokémon pour qu'il combatte avec un Pokémon ennemi





## Chapitre 2

# Index des classes

### 2.1 Liste des classes

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

<a href="#">attaque</a>	Structure d'une attaque . . . . .	7
<a href="#">pokemon</a>	Structure Pokémon . . . . .	8



## Chapitre 3

# Index des fichiers

### 3.1 Liste des fichiers

Liste de tous les fichiers documentés avec une brève description :

<a href="#">constantes.h</a>		
	Répertorie les constantes du projet . . . . .	11
<a href="#">credits.c</a>		
	Partie crédits du jeu Pokémon . . . . .	12
<a href="#">credits.h</a>		
	Interface des crédits du jeu Pokémon . . . . .	13
<a href="#">discours.h</a>		
	Fichier d'en-tête où sont inscrits toutes les phrases pour <a href="#">histoire.c</a> et <a href="#">credits.c</a> . . . . .	14
<a href="#">dresseur.c</a>		15
<a href="#">dresseur.h</a>		16
<a href="#">histoire.c</a>		
	Partie histoire du jeu Pokémon . . . . .	18
<a href="#">histoire.h</a>		
	Interface de l'histoire du jeu Pokémon . . . . .	18
<a href="#">jeu.c</a>		
	Fonction qui lance la partie jeu à partir du menu . . . . .	20
<a href="#">jeu.h</a>		21
<a href="#">main.c</a>		22



# Chapitre 4

## Documentation des classes

### 4.1 Référence de la structure attaque

Structure d'une attaque.

```
#include <dresseur.h>
```

#### Attributs publics

- `SDL_Surface *` `imageEnnemi`
- `SDL_Surface *` `imageAllie`
- `const char *` `nom`
- `Type` `type`
- `int` `precision`
- `int` `degatBase`
- `int` `degatParNiveau`
- `int` `degatParForce`

#### 4.1.1 Description détaillée

Structure d'une attaque.

#### 4.1.2 Documentation des données membres

##### 4.1.2.1 `int` `attaque` : `:degatBase`

Dégâts sans le coefficient de multiplication.

##### 4.1.2.2 `SDL_Surface*` `attaque` : `:imageAllie`

Surface de l'attaque si elle est lancée par le Pokémon allié sur le Pokémon ennemi.

##### 4.1.2.3 `SDL_Surface*` `attaque` : `:imageEnnemi`

Surface de l'attaque si elle est lancée par le Pokémon ennemi sur le Pokémon allié.

##### 4.1.2.4 `const char*` `attaque` : `:nom`

Nom de l'attaque.

#### 4.1.2.5 int attaque : :precision

Précision de l'attaque. C'est ce paramètre qui fait que des fois l'attaque rate.

#### 4.1.2.6 Type attaque : :type

Type de l'attaque.

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

— [dresseur.h](#)

## 4.2 Référence de la structure pokemon

Structure Pokémon.

```
#include <dresseur.h>
```

### Attributs publics

- SDL\_Surface \* [imageAllie](#)
- SDL\_Surface \* [imageEnnemi](#)
- const char \* [nom](#)
- int [vie](#)
- int [vieMax](#)
- int [force](#)
- int [vitesse](#)
- int [niveau](#)
- Type [type](#)
- int [experience](#)
- [Attaque](#) [attaque1](#)
- [Attaque](#) [attaque2](#)
- [Attaque](#) [attaque3](#)
- [Attaque](#) [attaque4](#)

### 4.2.1 Description détaillée

Structure Pokémon.

### 4.2.2 Documentation des données membres

#### 4.2.2.1 SDL\_Surface\* pokemon : :imageAllie

Surface du Pokémon s'il est du côté allié.

#### 4.2.2.2 SDL\_Surface\* pokemon : :imageEnnemi

Surface du Pokémon s'il est du côté ennemi.

#### 4.2.2.3 const char\* pokemon : :nom

Nom du Pokémon.

#### 4.2.2.4 Type pokemon : :type

Type du Pokémon.

### 4.2.2.5 int pokemon : :vie

Points de vie actuels du Pokémon.

### 4.2.2.6 int pokemon : :vieMax

Points de vie maximums.

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

— [dresseur.h](#)





## Chapitre 5

# Documentation des fichiers

### 5.1 Référence du fichier constantes.h

Répertorie les constantes du projet.

#### Macros

- #define `TAILLE_CASE` 100  
*Taille de la case.*
- #define `NB_CASES_LARGEUR` 10  
*Nombre de cases en largeur, ici 10.*
- #define `NB_CASES_HAUTEUR` 7  
*Nombre cases en hauteur, ici 7.*
- #define `LARGEUR_FENETRE` `TAILLE_CASE * NB_CASES_LARGEUR`  
*Largeur de la fenêtre.*
- #define `HAUTEUR_FENETRE` `TAILLE_CASE * NB_CASES_HAUTEUR`  
*Hauteur de la fenêtre.*

#### Énumérations

- enum `orientationJoueur` { `HAUT`, `BAS`, `GAUCHE`, `DROITE` }  
*Gère les quatre orientations du joueur.*

#### 5.1.1 Description détaillée

Répertorie les constantes du projet.

##### Version

1.0

##### Auteur

Antoine Falck

##### Date

Avril 2015

Ce fichier contient toutes les constantes liés au programmes.

Ce qui simplifie les choses lorsque l'on veut changer la taille de l'écran par exemple, il suffit de changer la valeur en question et de recompiler.

## 5.1.2 Documentation des macros

### 5.1.2.1 #define HAUTEUR\_FENETRE TAILLE\_CASE \* NB\_CASES\_HAUTEUR

Hauteur de la fenêtre.

Par définition  $l_f = T.N_{Ic}$

Avec  $l_f$  la hauteur de la fenêtre en pixel,  $T$  la taille de la case en pixel, et  $N_{Ic}$  le nombre de cases en hauteur.

### 5.1.2.2 #define LARGEUR\_FENETRE TAILLE\_CASE \* NB\_CASES\_LARGEUR

Largeur de la fenêtre.

Par définition  $L_f = T.N_{Lc}$

Avec  $L_f$  la largeur de la fenêtre en pixel,  $T$  la taille de la case en pixel, et  $N_{Lc}$  le nombre de cases en largeur.

### 5.1.2.3 #define TAILLE\_CASE 100

Taille de la case.

Taille de la case en pixel. La valeur par défaut est 100 pixels.

## 5.1.3 Documentation du type de l'énumération

### 5.1.3.1 enum orientationJoueur

Gère les quatre orientations du joueur.

Valeurs énumérées

**HAUT** Dans le cas où le joueur est orienté vers le haut.

**BAS** Dans le cas où le joueur est orienté vers le bas.

**GAUCHE** Dans le cas où le joueur est orienté vers la gauche.

**DROITE** Dans le cas où le joueur est orienté vers la droite.

## 5.2 Référence du fichier credits.c

Partie crédits du jeu Pokémon.

```
#include <SDL/SDL.h>
#include <SDL/SDL_ttf.h>
#include "credits.h"
#include "discours.h"
```

### Fonctions

- void `credits` (SDL\_Surface \*ecran)  
Fonction qui lance les crédits du jeu Pokémon.

### 5.2.1 Description détaillée

Partie crédits du jeu Pokémon. Petit commentaire sur le code, on remarque qu'on a dû créer 16 pointeurs sur SDL\_Surface pour chaque ligne de texte. En effet c'est la seule technique possible.

Il est impossible de créer un tableau de pointeur sur `SDL_Surface` ou bien d'inclure un symbole (type `"\n"`) dans le char pour pouvoir sauter une ligne.

**Version**

1.0

**Auteur**

Antoine Falck

**Date**

Avril 2015

## 5.2.2 Documentation des fonctions

### 5.2.2.1 void credits ( SDL\_Surface \* ecran )

Fonction qui lance les crédits du jeu Pokémon.

**Paramètres**

<code>in</code>	<code>ecran</code>	Fenêtre de l'application
-----------------	--------------------	--------------------------

## 5.3 Référence du fichier credits.h

Interface des crédits du jeu Pokémon.

**Fonctions**

- void `credits` (SDL\_Surface \*ecran)  
*Fonction qui lance les crédits du jeu Pokémon.*

### 5.3.1 Description détaillée

Interface des crédits du jeu Pokémon.

**Version**

1.1

**Auteur**

Antoine Falck

**Date**

Avril 2015

## 5.3.2 Documentation des fonctions

### 5.3.2.1 void credits ( SDL\_Surface \* ecran )

Fonction qui lance les crédits du jeu Pokémon.

## Paramètres

in	ecran	Fenêtre de l'application
----	-------	--------------------------

## 5.4 Référence du fichier discours.h

Fichier d'en-tête où sont inscrits toutes les phrases pour [histoire.c](#) et [credits.c](#).

## Macros

```

— #define TEXT0 "Salut jeune dresseur"
— #define TEXT1 "Euh... Je ne me souviens plus de ton prénom"
— #define TEXT2 "Quel est votre prénom ?"
— #define TEXT3 "Ah oui %s, pardonne mon viel âge"
— #define TEXT4 "L'aventure Pokémon commence pour toi"
— #define TEXT5 "Tu dois choisir un Pokémon pour ta quête"
— #define TEXT6 "Oups... Aucun Pokemon n'est disponible"
— #define TEXT7 "J'ai donné les 3 premiers Pokémon du labo"
— #define TEXT8 "Mais il m'en reste un, il est un peu spécial"
— #define TEXT9 "Il a un caractère bien trempé"
— #define TEXT10 "Son nom est Pikachu c'est un Pokémon souris"
— #define TEXT11 "Viens voir par là Pikachu voici %s"
— #define TEXT12 "Brzzz. *Bruits électriques*"
— #define TEXT13 "Rire. %s Piiikaaaaaaaaaaaaa"
— #define TEXT14 "Prends soin de ce Pikachu dans ton aventure"
— #define TEXT15 "Ton objectif est d'attraper tous les Pokemon"
— #define TEXT16 "La région Johto rassemble 8 arènes"
— #define TEXT17 "Une fois que tu auras obtenu tous les badges"
— #define TEXT18 "Tu pourras affronter la Ligue Pokémon"
— #define TEXT19 "Mais reste prudent, la route sera longue"
— #define TEXT20 "Gagne la confiance de tes Pokémon"
— #define TEXT21 "Fais les évoluer et protège les"
— #define TEXT22 "Une dernière chose avant de partir"
— #define TEXT23 "Voici un Pokédex"
— #define TEXT24 "Il te permettra d'identifier les Pokémon que"
— #define TEXT25 "tu rencontreras durant ton périple"
— #define TEXT26 "Bon vent %s, n'hésite pas à revenir"
— #define TEXT27 "me voir si tu as besoin de quelque chose"
— #define NOM1 "Antoine"
— #define NOM2 "Pierre-Antoine"
— #define NOM3 "Julien"
— #define CREDIT1 "CREDITS"
— #define CREDIT2 "Jeu 2D"
— #define CREDIT3 "Pokémon"
— #define CREDIT4 "CHEF DE PROJET"
— #define CREDIT5 "Julien Claus"
— #define CREDIT6 "EQUIPE TECHNIQUE"
— #define CREDIT7 "(ordre alphabétique)"
— #define CREDIT8 "Pierre-Antoine Chantal"
— #define CREDIT9 "Antoine Falck"
— #define CREDIT10 "Julien Violet"
— #define CREDIT11 "DIRECTEUR ARTISTIQUE"
— #define CREDIT12 "Antoine Falck"
— #define CREDIT13 "DIRECTEUR DE CREATION"
— #define CREDIT14 "Pierre-Antoine Chantal"
— #define CREDIT15 "TELECOM SUDPARIS"
— #define CREDIT16 "2015"

```

### 5.4.1 Description détaillée

Fichier d'en-tête où sont inscrits toutes les phrases pour [histoire.c](#) et [credits.c](#).

## Auteur

Antoine Falck

## Version

1.3

## Date

Avril 2015

## 5.5 Référence du fichier dresseur.c

```
#include <stdlib.h>
#include <stdio.h>
#include <SDL/SDL.h>
#include <SDL/SDL_image.h>
#include <SDL/SDL_ttf.h>
#include <time.h>
#include "dresseur.h"
```

### Fonctions

- void **lancerCombat** (SDL\_Surface \*ecran, int \*niveauPika, int numeroEnnemi, int niveauEnnemi, int \*ko)  
*Fonction qui lance le combat.*
- void **egaliteStatistiques** (Pokemon \*a, Pokemon b)  
*Fonction qui permet d'égaliser les statistiques des deux Pokémon qui combattent.*
- void **egaliteAttaque** (Attaque \*a, Attaque b)  
*Fonction qui permet d'égaliser les attaques des deux Pokémon qui combattent.*
- void **temporisation** (void)  
*Fonction qui fait la temporisation entre les événements du combat.*
- int **crit** ()  
*Fonction qui gère les coups critiques.*
- int **coef** (Type A, Type D)  
*Fonction qui gère le coefficient multiplicateur.*

### 5.5.1 Description détaillée

## Auteur

Pierre-Antoine Chantal et Julien Violet

## Date

Mai 2015

### 5.5.2 Documentation des fonctions

#### 5.5.2.1 int coef ( Type A, Type D )

Fonction qui gère le coefficient multiplicateur.

Suivant les types des Pokémon les dégâts sont plus ou moins important. Par exemple un Pokémon FEU ne fera pas beaucoup de dégâts sur un Pokémon EAU. Cette fonction permet donc de gérer le coefficient multiplicateur des dégâts d'une certaine attaque sur un certain type de Pokémon.

param[in] A param[in] B

### 5.5.2.2 void egaliteAttaque ( Attaque \* a, Attaque b )

Fonction qui permet d'égaliser les attaques des deux Pokémon qui combattent.

param[in, out] a param[in] b

### 5.5.2.3 void egaliteStatistiques ( Pokemon \* a, Pokemon b )

Fonction qui permet d'égaliser les statistiques des deux Pokémon qui combattent.

param[in, out] a param[in] b

### 5.5.2.4 void lancerCombat ( SDL\_Surface \* ecran, int \* niveauPika, int numeroEnnemi, int niveauEnnemi, int \* ko )

Fonction qui lance le combat.

param[in] ecran param[in, out] niveauPika param[in] numeroEnnemi param[in] niveauEnnemi param[in, out] ko

## 5.6 Référence du fichier dresseur.h

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

### Classes

- struct [attaque](#)  
*Structure d'une attaque.*
- struct [pokemon](#)  
*Structure Pokémon.*

### Définitions de type

- typedef enum [type](#) [Type](#)  
*Les différents types de Pokémon et d'attaques.*
- typedef struct [attaque](#) [Attaque](#)  
*Structure d'une attaque.*
- typedef struct [pokemon](#) [Pokemon](#)  
*Structure Pokémon.*

### Énumérations

- enum [type](#) {  
**TERRE, EAU, FEU, PLANTE,**  
**NORMAL, COMBAT, ROCHE, VOL,**  
**ELEK }**  
*Les différents types de Pokémon et d'attaques.*

### Fonctions

- void [lancerCombat](#) (SDL\_Surface \*ecran, int \*niveauPika, int numeroEnnemi, int niveauEnnemi, int \*ko)  
*Fonction qui lance le combat.*
- void [egaliteStatistiques](#) ([Pokemon](#) \*a, [Pokemon](#) b)  
*Fonction qui permet d'égaliser les statistiques des deux Pokémon qui combattent.*

- void `egaliteAttaque` (`Attaque *a`, `Attaque b`)  
*Fonction qui permet d'égaliser les attaques des deux Pokémon qui combattent.*
- void `temporisation` (void)  
*Fonction qui fait la temporisation entre les évènements du combat.*
- int `crit` ()  
*Fonction qui gère les coups critiques.*
- int `coef` (`Type A`, `Type D`)  
*Fonction qui gère le coefficient multiplicateur.*
- void `afficherTexteAttaque` (const char \*`nomAtak`, `SDL_Surface texteAtak`, `SDL_Rect position`, `TTF_Font police`, `SDL_Surface ecran`)  
*Fonction qui permet d'afficher le texte des attaques.*

### 5.6.1 Description détaillée

#### Auteur

Pierre-Antoine Chantal et Julien Violet

#### Date

Mai 2015

### 5.6.2 Documentation des fonctions

**5.6.2.1 void `afficherTexteAttaque` ( const char \* *nomAtak*, `SDL_Surface texteAtak`, `SDL_Rect position`, `TTF_Font police`, `SDL_Surface ecran` )**

Fonction qui permet d'afficher le texte des attaques.

param[in] `nomAtak` param[in] `texteAtak` param[in] `position` param[in] `police` param[in] `ecran`

**5.6.2.2 int `coef` ( `Type A`, `Type D` )**

Fonction qui gère le coefficient multiplicateur.

Suivant les types des Pokémon les dégâts sont plus ou moins important. Par exemple un Pokémon FEU ne fera pas beaucoup de dégâts sur un Pokémon EAU. Cette fonction permet donc de gérer le coefficient multiplicateur des dégâts d'une certaine attaque sur un certain type de Pokémon.

param[in] `A` param[in] `B`

**5.6.2.3 void `egaliteAttaque` ( `Attaque * a`, `Attaque b` )**

Fonction qui permet d'égaliser les attaques des deux Pokémon qui combattent.

param[in, out] `a` param[in] `b`

**5.6.2.4 void `egaliteStatistiques` ( `Pokemon * a`, `Pokemon b` )**

Fonction qui permet d'égaliser les statistiques des deux Pokémon qui combattent.

param[in, out] `a` param[in] `b`

**5.6.2.5 void `lancerCombat` ( `SDL_Surface * ecran`, int \* `niveauPika`, int `numeroEnnemi`, int `niveauEnnemi`, int \* `ko` )**

Fonction qui lance le combat.

param[in] `ecran` param[in, out] `niveauPika` param[in] `numeroEnnemi` param[in] `niveauEnnemi` param[in, out] `ko`

## 5.7 Référence du fichier histoire.c

Partie histoire du jeu Pokémon.

```
#include <SDL/SDL.h>
#include <SDL/SDL_image.h>
#include <SDL/SDL_ttf.h>
#include "histoire.h"
#include "discours.h"
```

### Fonctions

- void [histoire](#) (SDL\_Surface \*ecran)  
*Fonction qui lance l'histoire du jeu Pokémon.*
- void [peech](#) (char script[NB\_FENETRE][NB\_MAX\_CARACTERE])  
*Fonction qui charge le texte de [discours.h](#) dans un tableau de tableau de caractères.*

### 5.7.1 Description détaillée

Partie histoire du jeu Pokémon.

Version

2.3

Auteur

Antoine Falck

Date

Avril 2015

### 5.7.2 Documentation des fonctions

#### 5.7.2.1 void histoire ( SDL\_Surface \* *ecran* )

Fonction qui lance l'histoire du jeu Pokémon.

Paramètres

<i>in</i>	<i>ecran</i>	Fenêtre de l'application
-----------	--------------	--------------------------

#### 5.7.2.2 void peech ( char *script*[NB\_FENETRE][NB\_MAX\_CARACTERE] )

Fonction qui charge le texte de [discours.h](#) dans un tableau de tableau de caractères.

Paramètres

<i>in, out</i>	<i>script</i>	Tableau de tableau de caractères, qui est chargé dans texte (SDL_Surface) dès que le joueur appuie sur "Enter".
----------------	---------------	---

## 5.8 Référence du fichier histoire.h

Interface de l'histoire du jeu Pokémon.



## Macros

- `#define NB_FENETRE 100`
- `#define NB_MAX_CARACTERE 1024`

## Fonctions

- `void histoire (SDL_Surface *ecran)`  
*Fonction qui lance l'histoire du jeu Pokémon.*
- `void peech (char script[NB_FENETRE][NB_MAX_CARACTERE])`  
*Fonction qui charge le texte de `discours.h` dans un tableau de tableau de caractères.*

### 5.8.1 Description détaillée

Interface de l'histoire du jeu Pokémon.

#### Version

2.3

#### Auteur

Antoine Falck

#### Date

Avril 2015

### 5.8.2 Documentation des macros

#### 5.8.2.1 `#define NB_FENETRE 100`

Nombre de fenêtre maximum que l'on peut ouvrir dans le petit encart réservé à l'affichage du texte

#### 5.8.2.2 `#define NB_MAX_CARACTERE 1024`

Nombre maximum de caractère que l'on peut placer dans le texte qui s'affiche lors de l'histoire

### 5.8.3 Documentation des fonctions

#### 5.8.3.1 `void histoire ( SDL_Surface * ecran )`

Fonction qui lance l'histoire du jeu Pokémon.

##### Paramètres

<code>in</code>	<code><i>ecran</i></code>	Fenêtre de l'application
-----------------	---------------------------	--------------------------

#### 5.8.3.2 `void peech ( char script[NB_FENETRE][NB_MAX_CARACTERE] )`

Fonction qui charge le texte de `discours.h` dans un tableau de tableau de caractères.

## Paramètres

in, out	script	Tableau de tableau de caractères, qui est chargé dans texte (SDL_Surface) dès que le joueur appuie sur "Enter".
---------	--------	---

## 5.9 Référence du fichier jeu.c

Fonction qui lance la partie jeu à partir du menu.

```
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
#include "jeu.h"
#include "dresseur.h"
#include "constantes.h"
#include <SDL/SDL.h>
#include <SDL/SDL_image.h>
```

## Fonctions

- void **jouer** (SDL\_Surface \*ecran)  
*Fonction qui lance le jeu à partir du menu.*
- void **remplirTableau** (int tabcarte[10][7], FILE \*fichier)  
*Fonction qui remplit un tableau d'entiers.*
- void **deplacerJoueur** (SDL\_Surface \*ecran, int tabcarte[10][7], SDL\_Rect \*pos, int direction)  
*Fonction qui gère le déplacement du joueur.*
- void **remplirmap** (int tabcarte[10][7], SDL\_Rect \*pos, SDL\_Surface \*arbre, SDL\_Surface \*centre, SDL\_Surface \*herbe, SDL\_Surface \*ecran, SDL\_Surface \*maison1, SDL\_Surface \*maison2, SDL\_Surface \*pnj1, SDL\_Surface \*bureau, SDL\_Surface \*carrenoir, SDL\_Surface \*oss)  
*Fonction qui remplit la carte à partir du tableau précédemment rempli.*

### 5.9.1 Description détaillée

Fonction qui lance la partie jeu à partir du menu.

## Auteur

Julien Claus

## Date

Avril 2015

### 5.9.2 Documentation des fonctions

#### 5.9.2.1 void deplacerJoueur ( SDL\_Surface \* écran, int tabcarte[10][7], SDL\_Rect \* pos, int direction )

Fonction qui gère le déplacement du joueur.

param[in] écran param[in] tabcarte Tableau d'entier qui définit la carte sur laquelle le joueur est présent param[in, out] pos Position du joueur en pixels ; (0 ;0) en haut à gauche param[in] direction Direction du joueur, ou plutôt son orientation (haut, bas, gauche, droite)

#### 5.9.2.2 void jouer ( SDL\_Surface \* écran )

Fonction qui lance le jeu à partir du menu.

## Paramètres

in	ecran	Fenêtre de l'application
----	-------	--------------------------

5.9.2.3 void remplirmap ( int *tabcarte*[10][7], SDL\_Rect \* *pos*, SDL\_Surface \* *arbre*, SDL\_Surface \* *centre*, SDL\_Surface \* *herbe*, SDL\_Surface \* *ecran*, SDL\_Surface \* *maison1*, SDL\_Surface \* *maison2*, SDL\_Surface \* *pnj1*, SDL\_Surface \* *bureau*, SDL\_Surface \* *carrenoir*, SDL\_Surface \* *oss* )

Fonction qui remplit la carte à partir du tableau précédemment rempli.

param[in] tabcarte param[in] pos param[in] arbre param[in] centre param[in] herbe param[in] ekran param[in] maison1 param[in] maison2 param[in] pnj1 param[in] bureau param[in] carrenoir param[in] oss

5.9.2.4 void remplirTableau ( int *tabcarte*[10][7], FILE \* *fichier* )

Fonction qui remplit un tableau d'entiers.

Ce tableau d'entier rempli à partir d'un fichier a pour but de charger les cartes. Chaque caractère correspond à un élément de la carte qui est placée à la position souhaitée. Cette position est tout simplement l'ordre des caractères.

param[in, out] tabcarte Tableau d'entier à remplir param[in] fichier Fichier texte utilisé pour remplir le tableau

## 5.10 Référence du fichier jeu.h

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

### Fonctions

- void **jouer** (SDL\_Surface \*ecran)  
*Fonction qui lance le jeu à partir du menu.*
- void **remplirTableau** (int tabcarte[10][7], FILE \*fichier)  
*Fonction qui remplit un tableau d'entiers.*
- void **deplacerJoueur** (SDL\_Surface \*ecran, int tabcarte[10][7], SDL\_Rect \*pos, int direction)  
*Fonction qui gère le déplacement du joueur.*
- void **remplirmap** (int tabcarte[10][7], SDL\_Rect \*pos, SDL\_Surface \*arbre, SDL\_Surface \*centre, SDL\_Surface \*herbe, SDL\_Surface \*ecran, SDL\_Surface \*maison1, SDL\_Surface \*maison2, SDL\_Surface \*pnj1, SDL\_Surface \*bureau, SDL\_Surface \*carrenoir, SDL\_Surface \*oss)  
*Fonction qui remplit la carte à partir du tableau précédemment rempli.*

### 5.10.1 Description détaillée

#### Auteur

Julien Claus

#### Date

Avril 2015

### 5.10.2 Documentation des fonctions

### 5.10.2.1 void deplacerJoueur ( SDL\_Surface \* ecran, int tabcarte[10][7], SDL\_Rect \* pos, int direction )

Fonction qui gère le déplacement du joueur.

param[in] ecran param[in] tabcarte Tableau d'entier qui définit la carte sur laquelle le joueur est présent param[in, out] pos Position du joueur en pixels ; (0 ;0) en haut à gauche param[in] direction Direction du joueur, ou plutôt son orientation (haut, bas, gauche, droite)

### 5.10.2.2 void jouer ( SDL\_Surface \* ecran )

Fonction qui lance le jeu à partir du menu.

#### Paramètres

in	ecran	Fenêtre de l'application
----	-------	--------------------------

### 5.10.2.3 void remplirmap ( int tabcarte[10][7], SDL\_Rect \* pos, SDL\_Surface \* arbre, SDL\_Surface \* centre, SDL\_Surface \* herbe, SDL\_Surface \* ecran, SDL\_Surface \* maison1, SDL\_Surface \* maison2, SDL\_Surface \* pnj1, SDL\_Surface \* bureau, SDL\_Surface \* carrenoir, SDL\_Surface \* oss )

Fonction qui remplit la carte à partir du tableau précédemment rempli.

param[in] tabcarte param[in] pos param[in] arbre param[in] centre param[in] herbe param[in] ecran param[in] maison1 param[in] maison2 param[in] pnj1 param[in] bureau param[in] carrenoir param[in] oss

### 5.10.2.4 void remplirTableau ( int tabcarte[10][7], FILE \* fichier )

Fonction qui remplit un tableau d'entiers.

Ce tableau d'entier rempli à partir d'un fichier a pour but de charger les cartes. Chaque caractère correspond à un élément de la carte qui est placée à la position souhaitée. Cette position est tout simplement l'ordre des caractères.

param[in, out] tabcarte Tableau d'entier à remplir param[in] fichier Fichier texte utilisé pour remplir le tableau

## 5.11 Référence du fichier main.c

```
#include <stdio.h>
#include <stdlib.h>
#include <SDL/SDL.h>
#include <SDL/SDL_image.h>
#include "histoire.h"
#include "jeu.h"
#include "credits.h"
#include "constantes.h"
```

## Fonctions

- void [initialisationSDL](#) (SDL\_Surface \*\*ecran)  
*Fonction qui fait appel à la SDL et charge l'écran (au sens interface graphique) de l'application.*
- void [chargementMenu](#) (SDL\_Surface \*\*menu, SDL\_Rect \*positionMenu)  
*Fonction qui a pour fonction de charger le menu du jeu.*
- void [affichageMenu](#) (SDL\_Surface \*ecran, SDL\_Surface \*menu, SDL\_Rect positionMenu)  
*Fonction qui permet d'afficher le menu chargé précédemment.*
- void [fermetureSDL](#) (SDL\_Surface \*ecran, SDL\_Surface \*menu)  
*Fonction qui ferme la SDL et désalloue les surfaces (SDL\_Surface) allouées.*
- int [main](#) (int argc, char \*argv[])

Fonction principale. Elle lance l'application et fait l'appel de la SDL.

### 5.11.1 Description détaillée

Programme principal qui gère entre autre le lancement et l'arrêt de la bibliothèque SDL.

Version

1.0

Auteur

Antoine Falck

Date

Avril 2015

### 5.11.2 Documentation des fonctions

5.11.2.1 void affichageMenu ( SDL\_Surface \* *ecran*, SDL\_Surface \* *menu*, SDL\_Rect *positionMenu* )

Fonction qui permet d'afficher le menu chargé précédemment.

Paramètres

in	<i>ecran</i>	La surface qui recevra le collage.
in	<i>menu</i>	La surface collée.
in	<i>positionMenu</i>	: La position de la surface collée.

5.11.2.2 void chargementMenu ( SDL\_Surface \*\* *menu*, SDL\_Rect \* *positionMenu* )

Fonction qui a pour fonction de charger le menu du jeu.

Paramètres

in	<i>&amp;menu</i>	Pointeur sur menu qui est en fait l'image du menu que l'on collera plus tard sur écran (la fenêtre de l'application).
in	<i>&amp;positionMenu</i>	Pointeur sur positionMenu qui gère la position de l'image "menu" à coller sur la fenêtre.

5.11.2.3 void fermetureSDL ( SDL\_Surface \* *ecran*, SDL\_Surface \* *menu* )

Fonction qui ferme la SDL et désalloue les surfaces (SDL\_Surface) allouées.

Paramètres

in	<i>ecran</i>	Première surface allouée.
in	<i>menu</i>	Deuxième surface allouée.

5.11.2.4 void initialisationSDL ( SDL\_Surface \*\* *ecran* )

Fonction qui fait appel à la SDL et charge l'écran (au sens interface graphique) de l'application.

**Paramètres**

<code>in</code>	<code>&amp;ecran</code>	Pointeur sur ecran, ecran étant la fenêtre de l'application.
-----------------	-------------------------	--

**5.11.2.5** `int main ( int argc, char * argv[] )`

Fonction principale. Elle lance l'application et fait l'appel de la SDL.

**Paramètres**

<code>in</code>	<code><i>argc</i></code>	
<code>in</code>	<code><i>argv</i></code>	

**Renvoie**

EXIT\_SUCCESS ou EXIT\_FAILURE

# Index

- affichageMenu
  - main.c, 23
- afficherTexteAttaque
  - dresseur.h, 17
- attaque, 7
  - degatBase, 7
  - imageAllie, 7
  - imageEnnemi, 7
  - nom, 7
  - precision, 7
  - type, 8
- BAS
  - constantes.h, 12
- chargementMenu
  - main.c, 23
- coef
  - dresseur.c, 15
  - dresseur.h, 17
- constantes.h
  - BAS, 12
  - DROITE, 12
  - GAUCHE, 12
  - HAUT, 12
- constantes.h, 11
  - HAUTEUR\_FENETRE, 12
  - LARGEUR\_FENETRE, 12
  - orientationJoueur, 12
  - TAILLE\_CASE, 12
- credits
  - credits.c, 13
  - credits.h, 13
- credits.c, 12
  - credits, 13
- credits.h, 13
  - credits, 13
- DROITE
  - constantes.h, 12
- degatBase
  - attaque, 7
- deplacerJoueur
  - jeu.c, 20
  - jeu.h, 21
- discours.h, 14
- dresseur.c, 15
  - coef, 15
  - egaliteAttaque, 15
  - egaliteStatistiques, 16
  - lancerCombat, 16
- dresseur.h, 16
  - afficherTexteAttaque, 17
  - coef, 17
  - egaliteAttaque, 17
  - egaliteStatistiques, 17
  - lancerCombat, 17
- egaliteAttaque
  - dresseur.c, 15
  - dresseur.h, 17
- egaliteStatistiques
  - dresseur.c, 16
  - dresseur.h, 17
- fermetureSDL
  - main.c, 23
- GAUCHE
  - constantes.h, 12
- HAUT
  - constantes.h, 12
- HAUTEUR\_FENETRE
  - constantes.h, 12
- histoire
  - histoire.c, 18
  - histoire.h, 19
- histoire.c, 18
  - histoire, 18
  - peech, 18
- histoire.h, 18
  - histoire, 19
  - NB\_FENETRE, 19
  - NB\_MAX\_CARACTERE, 19
  - peech, 19
- imageAllie
  - attaque, 7
  - pokemon, 8
- imageEnnemi
  - attaque, 7
  - pokemon, 8
- initialisationSDL
  - main.c, 23
- jeu.c, 20
  - deplacerJoueur, 20
  - jouer, 20
  - remplirTableau, 21
  - remplirmap, 21

jeu.h, 21  
    deplacerJoueur, 21  
    jouer, 22  
    remplirTableau, 22  
    remplirmap, 22  
jouer  
    jeu.c, 20  
    jeu.h, 22  
  
LARGEUR\_FENETRE  
    constantes.h, 12  
lancerCombat  
    dresseur.c, 16  
    dresseur.h, 17  
  
main  
    main.c, 24  
main.c, 22  
    affichageMenu, 23  
    chargementMenu, 23  
    fermetureSDL, 23  
    initialisationSDL, 23  
    main, 24  
  
NB\_FENETRE  
    histoire.h, 19  
NB\_MAX\_CARACTERE  
    histoire.h, 19  
nom  
    attaque, 7  
    pokemon, 8  
  
orientationJoueur  
    constantes.h, 12  
  
peech  
    histoire.c, 18  
    histoire.h, 19  
pokemon, 8  
    imageAllie, 8  
    imageEnnemi, 8  
    nom, 8  
    type, 8  
    vie, 8  
    vieMax, 9  
precision  
    attaque, 7  
  
remplirTableau  
    jeu.c, 21  
    jeu.h, 22  
remplirmap  
    jeu.c, 21  
    jeu.h, 22  
  
TAILLE\_CASE  
    constantes.h, 12  
type  
    attaque, 8  
    pokemon, 8  
  
vie  
    pokemon, 8  
vieMax  
    pokemon, 9