

Projet Champignon 2016

Généré par Doxygen 1.8.13

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 carte	5
3.1.1	Description détaillée	5
3.2	Référence de la structure champignon	5
3.2.1	Description détaillée	6
3.3	Référence de la structure cueilleur	6
3.3.1	Description détaillée	6
3.4	Référence de la structure sanglier	6
3.4.1	Description détaillée	6
4	Documentation des fichiers	7
4.1	Référence du fichier lanceur_projet_champignon.c	7
4.1.1	Description détaillée	7
4.1.2	Documentation des fonctions	7
4.1.2.1	main()	7
4.2	Référence du fichier projet_champignon.c	8
4.2.1	Description détaillée	8
4.2.2	Documentation des fonctions	8
4.2.2.1	afficherChampignon()	8

4.2.2.2	<code>afficherCueilleur()</code>	9
4.2.2.3	<code>afficherSanglier()</code>	9
4.2.2.4	<code>aleatoire()</code>	9
4.2.2.5	<code>init_champignon()</code>	10
4.2.2.6	<code>init_cueilleur()</code>	10
4.2.2.7	<code>init_sanglier()</code>	10
4.3	Référence du fichier <code>projet_champignon.h</code>	10
4.3.1	Description détaillée	11
4.3.2	Documentation des fonctions	11
4.3.2.1	<code>afficherChampignon()</code>	11
4.3.2.2	<code>afficherCueilleur()</code>	12
4.3.2.3	<code>afficherSanglier()</code>	12
4.3.2.4	<code>aleatoire()</code>	12
4.3.2.5	<code>init_champignon()</code>	13
4.3.2.6	<code>init_cueilleur()</code>	13
4.3.2.7	<code>init_sanglier()</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 :

carte	Structure de données de la carte, contient un cueilleur, un compteur de nombre de tours, un tableau de champignons et un tableau de sangliers	5
champignon	Structure de données d'un champignon, le champignon à une position x, y sur la carte et un compteur pour déterminer le tour auquel il a été ramassé	5
cueilleur	Structure de données du cueilleur, dispose d'une position x, y, un compteur à champignons ainsi que d'un état de boost	6
sanglier	Structure de données d'un sanglier	6

Chapitre 2

Index des fichiers

2.1 Liste des fichiers

Liste de tous les fichiers documentés avec une brève description :

lanceur_projet_champignon.c	
Fichier principal du projet Champignon	7
projet_champignon.c	
Fichier source du projet champignon	8
projet_champignon.h	
Header du projet Champignon	10

Chapitre 3

Documentation des classes

3.1 Référence de la structure carte

Structure de données de la carte, contient un cueilleur, un compteur de nombre de tours, un tableau de champignons et un tableau de sangliers.

```
#include <projet_champignon.h>
```

Attributs publics

- struct [cueilleur](#) **joueur**
- int **compteur**
- int **champignons** []

3.1.1 Description détaillée

Structure de données de la carte, contient un cueilleur, un compteur de nombre de tours, un tableau de champignons et un tableau de sangliers.

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

- [projet_champignon.h](#)

3.2 Référence de la structure champignon

Structure de données d'un champignon, le champignon à une position x, y sur la carte et un compteur pour déterminer le tour auquel il a été ramassé.

```
#include <projet_champignon.h>
```

Attributs publics

- char **type**
- int **x**
- int **y**
- int **ramasse**

3.2.1 Description détaillée

Structure de données d'un champignon, le champignon à une position x, y sur la carte et un compteur pour déterminer le tour auquel il a été ramassé.

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

— [projet_champignon.h](#)

3.3 Référence de la structure cueilleur

Structure de données du cueilleur, dispose d'une position x, y, un compteur à champignons ainsi que d'un état de boost.

```
#include <projet_champignon.h>
```

Attributs publics

- int **x**
- int **y**
- int **champignons**
- int **boost**
- int **dernierBoost**

3.3.1 Description détaillée

Structure de données du cueilleur, dispose d'une position x, y, un compteur à champignons ainsi que d'un état de boost.

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

— [projet_champignon.h](#)

3.4 Référence de la structure sanglier

Structure de données d'un sanglier.

```
#include <projet_champignon.h>
```

Attributs publics

- int **x**
- int **y**
- int **speed**

3.4.1 Description détaillée

Structure de données d'un sanglier.

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

— [projet_champignon.h](#)

Chapitre 4

Documentation des fichiers

4.1 Référence du fichier lanceur_projet_champignon.c

fichier principal du projet Champignon.

```
#include "projet_champignon.h"  
#include <stdlib.h>  
#include <time.h>
```

Fonctions

— int `main` ()

4.1.1 Description détaillée

fichier principal du projet Champignon.

Auteur

benjamin Fichier contenant le programme principal.

4.1.2 Documentation des fonctions

4.1.2.1 `main()`

```
int main ( )
```

déclaration de la variable joueur de structure cueilleur.

déclaration de la variable champignon de structure champignon.

déclaration de la variable sanglier de structure sanglier

affichage des valeurs de la structure du joueur.

affichage des valeurs de la structure du champignon.

affichage des valeurs de la structure du sanglier.

4.2 Référence du fichier projet_champignon.c

fichier source du projet champignon.

```
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
#include "projet_champignon.h"
```

Fonctions

- unsigned int `aleatoire` (unsigned int limite)
La fonction aleatoire retourne un entier généré de manière pseudo-aléatoire compris entre 0 et la valeur de son paramètre.
- struct `cueilleur` `init_cueilleur` ()
Fonction permettant d'initialiser un cueilleur (structure), la position initiale est définie aléatoirement.
- struct `champignon` `init_champignon` ()
Fonction permettant d'initialiser un champignon (structure), la position initiale est définie aléatoirement.
- struct `sanglier` `init_sanglier` ()
Fonction permettant d'initialiser un sanglier (structure), la position initiale et la vitesse de déplacement sont définies aléatoirement.
- void `afficherCueilleur` (struct `cueilleur` c)
Affiche l'ensemble des données du joueur.
- void `afficherChampignon` (struct `champignon` c)
Affiche l'ensemble des données d'un champignon.
- void `afficherSanglier` (struct `sanglier` c)
Affiche l'ensemble des données d'un sanglier.

4.2.1 Description détaillée

fichier source du projet champignon.

Auteur

benjamin Boboul
Le fichier contient du code, c'est déjà bien.

4.2.2 Documentation des fonctions

4.2.2.1 `afficherChampignon()`

```
void afficherChampignon (
    struct champignon c )
```

Affiche l'ensemble des données d'un champignon.

Paramètres

in	c	struct <code>champignon</code> du champignon.
----	---	---

4.2.2.2 afficherCueilleur()

```
void afficherCueilleur (
    struct cueilleur c )
```

Affiche l'ensemble des données du joueur.

Paramètres

in	<i>c</i>	struct cueilleur du joueur.
----	----------	-----------------------------

4.2.2.3 afficherSanglier()

```
void afficherSanglier (
    struct sanglier c )
```

Affiche l'ensemble des données d'un sanglier.

Paramètres

in	<i>c</i>	struct sanglier du sanglier.
----	----------	------------------------------

4.2.2.4 aleatoire()

```
unsigned int aleatoire (
    unsigned int limite )
```

La fonction aleatoire retourne un entier généré de manière pseudo-aléatoire compris entre 0 et la valeur de son paramètre.

Pour initialiser le générateur pseudo-aléatoire, il faut ajouter l'instruction `srand(time(NULL))` ; au début de la fonction main et inclure les bibliothèques `time.h` et `stdlib.h`.

Paramètres

in	<i>limite</i>	Entier qui définit la limite à ne pas dépasser.
----	---------------	---

Renvoie

entier non signé.

4.2.2.5 init_champignon()

```
struct champignon init_champignon ( )
```

Fonction permettant d'initialiser un champignon (structure), la position initiale est définie aléatoirement.

Le type du champignon est représenté par un caractère pouvant avoir la valeur 'n' pour normal, 'b' pour boost et 'e' pour empoisonné.

Exemple d'utilisation : struct champignon champ ; champ= init_champignon(char type);

Renvoie

struct champignon

4.2.2.6 init_cueilleur()

```
struct cueilleur init_cueilleur ( )
```

Fonction permettant d'initialiser un cueilleur (structure), la position initiale est définie aléatoirement.

Exemple d'utilisation : struct cueilleur c; c=init_cueilleur();

Renvoie

struct cueilleur

4.2.2.7 init_sanglier()

```
struct sanglier init_sanglier ( )
```

Fonction permettant d'initialiser un sanglier (structure), la position initiale et la vitesse de déplacement sont définies aléatoirement.

Exemple d'utilisation : struct sangliers ;c=init_sanglier();

Renvoie

struct champignon

4.3 Référence du fichier projet_champignon.h

Header du projet Champignon.

Classes

- struct `cueilleur`
Structure de données du cueilleur, dispose d'une position x, y, un compteur à champignons ainsi que d'un état de boost.
- struct `champignon`
Structure de données d'un champignon, le champignon à une position x, y sur la carte et un compteur pour déterminer le tour auquel il a été ramassé.
- struct `sanglier`
Structure de données d'un sanglier.
- struct `carte`
Structure de données de la carte, contient un cueilleur, un compteur de nombre de tours, un tableau de champignons et un tableau de sangliers.

Fonctions

- struct `cueilleur init_cueilleur ()`
Fonction permettant d'initialiser un cueilleur (structure), la position initiale est définie aléatoirement.
- struct `champignon init_champignon ()`
Fonction permettant d'initialiser un champignon (structure), la position initiale est définie aléatoirement.
- struct `sanglier init_sanglier ()`
Fonction permettant d'initialiser un sanglier (structure), la position initiale et la vitesse de déplacement sont définies aléatoirement.
- unsigned int `aleatoire (unsigned int limite)`
La fonction aleatoire retourne un entier généré de manière pseudo-aléatoire compris entre 0 et la valeur de son paramètre.
- void `afficherCueilleur (struct cueilleur c)`
Affiche l'ensemble des données du joueur.
- void `afficherChampignon (struct champignon c)`
Affiche l'ensemble des données d'un champignon.
- void `afficherSanglier (struct sanglier c)`
Affiche l'ensemble des données d'un sanglier.

4.3.1 Description détaillée

Header du projet Champignon.

Auteur

benjamin

Le fichier contient les déclarations des structures, fonctions et procédures.

4.3.2 Documentation des fonctions

4.3.2.1 afficherChampignon()

```
void afficherChampignon (
    struct champignon c )
```

Affiche l'ensemble des données d'un champignon.

Paramètres

in	<i>c</i>	struct champignon du champignon.
----	----------	----------------------------------

4.3.2.2 afficherCueilleur()

```
void afficherCueilleur (
    struct cueilleur c )
```

Affiche l'ensemble des données du joueur.

Paramètres

in	<i>c</i>	struct cueilleur du joueur.
----	----------	-----------------------------

4.3.2.3 afficherSanglier()

```
void afficherSanglier (
    struct sanglier c )
```

Affiche l'ensemble des données d'un sanglier.

Paramètres

in	<i>c</i>	struct sanglier du sanglier.
----	----------	------------------------------

4.3.2.4 aleatoire()

```
unsigned int aleatoire (
    unsigned int limite )
```

La fonction aleatoire retourne un entier généré de manière pseudo-aléatoire compris entre 0 et la valeur de son paramètre.

Pour initialiser le générateur pseudo-aléatoire, il faut ajouter l'instruction `srand(time(NULL))` ; au début de la fonction main et inclure les bibliothèques `time.h` et `stdlib.h`.

Paramètres

in	<i>limite</i>	Entier qui définit la limite à ne pas dépasser.
----	---------------	---

Renvoie

entier non signé.

4.3.2.5 init_champignon()

```
struct champignon init_champignon ( )
```

Fonction permettant d'initialiser un champignon (structure), la position initiale est définie aléatoirement.

Le type du champignon est représenté par un caractère pouvant avoir la valeur 'n' pour normal, 'b' pour boost et 'e' pour empoisonné.

Exemple d'utilisation : struct champignon champ ; champ= init_champignon(char type) ;

Renvoie

struct champignon

4.3.2.6 init_cueilleur()

```
struct cueilleur init_cueilleur ( )
```

Fonction permettant d'initialiser un cueilleur (structure), la position initiale est définie aléatoirement.

Exemple d'utilisation : struct cueilleur c ; c=init_cueilleur() ;

Renvoie

struct cueilleur

4.3.2.7 init_sanglier()

```
struct sanglier init_sanglier ( )
```

Fonction permettant d'initialiser un sanglier (structure), la position initiale et la vitesse de déplacement sont définies aléatoirement.

Exemple d'utilisation : struct sangliers ;c=init_sanglier() ;

Renvoie

struct champignon

Index

- afficherChampignon
 - projet_champignon.c, [8](#)
 - projet_champignon.h, [11](#)
- afficherCueilleur
 - projet_champignon.c, [9](#)
 - projet_champignon.h, [12](#)
- afficherSanglier
 - projet_champignon.c, [9](#)
 - projet_champignon.h, [12](#)
- aleatoire
 - projet_champignon.c, [9](#)
 - projet_champignon.h, [12](#)

- carte, [5](#)
- champignon, [5](#)
- cueilleur, [6](#)

- init_champignon
 - projet_champignon.c, [9](#)
 - projet_champignon.h, [13](#)
- init_cueilleur
 - projet_champignon.c, [10](#)
 - projet_champignon.h, [13](#)
- init_sanglier
 - projet_champignon.c, [10](#)
 - projet_champignon.h, [13](#)

- lanceur_projet_champignon.c, [7](#)
 - main, [7](#)

- main
 - lanceur_projet_champignon.c, [7](#)

- projet_champignon.c, [8](#)
 - afficherChampignon, [8](#)
 - afficherCueilleur, [9](#)
 - afficherSanglier, [9](#)
 - aleatoire, [9](#)
 - init_champignon, [9](#)
 - init_cueilleur, [10](#)
 - init_sanglier, [10](#)
- projet_champignon.h, [10](#)
 - afficherChampignon, [11](#)
 - afficherCueilleur, [12](#)
 - afficherSanglier, [12](#)
 - aleatoire, [12](#)
 - init_champignon, [13](#)
 - init_cueilleur, [13](#)
 - init_sanglier, [13](#)

- sanglier, [6](#)