

Projet 1: Librairie de gestion d'images

1.0

Generated by Doxygen 1.8.18

1	Projet 1: Librairie de gestion d'images	1
2	Data Structure Index	3
2.1	Data Structures	3
3	File Index	5
3.1	File List	5
4	Data Structure Documentation	7
4.1	PNM_t Struct Reference	7
4.1.1	Detailed Description	7
5	File Documentation	9
5.1	pnm.h File Reference	9
5.1.1	Detailed Description	10
5.1.2	Function Documentation	10
5.1.2.1	create_matrix()	10
5.1.2.2	create_pnm()	11
5.1.2.3	destroy_matrix_columns()	11
5.1.2.4	destroy_matrix_rows()	12
5.1.2.5	destroy_pnm()	12
5.1.2.6	get_columns()	13
5.1.2.7	get_magicNumber()	13
5.1.2.8	get_matrix()	14
5.1.2.9	get_maxValuePixel()	14
5.1.2.10	get_rows()	15
5.1.2.11	load_matrix()	15
5.1.2.12	load_pnm()	16
5.1.2.13	manage_comments()	16
5.1.2.14	set_columns()	17
5.1.2.15	set_magicNumber()	17
5.1.2.16	set_matrix()	18
5.1.2.17	set_maxValuePixel()	19
5.1.2.18	set_rows()	20
5.1.2.19	write_matrix()	21
	Index	23

Chapter 1

Projet 1: Librairie de gestion d'images

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

PNM_t	7
-----------------	---

Chapter 3

File Index

3.1 File List

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

pnm.h	Librairie pour gérer les fichiers d'extension pnm (.pbm, .pgm, .ppm)	9
-----------------------	--	---

Chapter 4

Data Structure Documentation

4.1 PNM_t Struct Reference

Data Fields

- int **magicNumber**
- int **columns**
- int **rows**
- int **maxValuePixel**
- int ** **matrix**

4.1.1 Detailed Description

pnm.c

Ce fichier contient les définitions de types et les fonctions de manipulation d'images PNM.

Author

: Dumoulin Peissone S193957

Date

: 22/02/21 @projet: INFO0030 Projet 1 Définition du type opaque PNM

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

- pnm.c

Chapter 5

File Documentation

5.1 pnm.h File Reference

Librairie pour gérer les fichiers d'extension pnm (.pbm, .pgm, .ppm)

Typedefs

- typedef struct [PNM_t](#) **PNM**

Functions

- int [load_pnm](#) ([PNM](#) **image, char *filename)
Charge une image PNM depuis un fichier.
- int [write_pnm](#) ([PNM](#) *image, char *filename)
- [PNM](#) * [create_pnm](#) (void)
Crée et alloue dynamiquement une variable de type opaque PNM.*
- void [destroy_pnm](#) ([PNM](#) *image)
*Libère la mémoire allouée par *create_pnm.*
- int [get_magicNumber](#) ([PNM](#) *image)
Accesseur en lecture pour le champ magicNumber de image.*
- int [get_columns](#) ([PNM](#) *image)
Accesseur en lecture pour le champ columns de image.*
- int [get_rows](#) ([PNM](#) *image)
Accesseur en lecture pour le champ rows de image.*
- int [get_maxValuePixel](#) ([PNM](#) *image)
Accesseur en lecture pour le champ maxValuePixel de image.*
- int ** [get_matrix](#) ([PNM](#) *image)
Accesseur en lecture pour le champ matrix de image.*
- [PNM](#) * [set_magicNumber](#) ([PNM](#) *image, int magicNumber)
*Accesseur en écriture pour le champ magicNumber de *image.*
- [PNM](#) * [set_columns](#) ([PNM](#) *image, int columns)
*Accesseur en écriture pour le champ columns de *image.*
- [PNM](#) * [set_rows](#) ([PNM](#) *image, int rows)
*Accesseur en écriture pour le champ rows de *image.*

- `PNM * set_maxValuePixel (PNM *image, int maxValuePixel)`
*Accesseur en écriture pour le champ `maxValuePixel` de `*image`.*
- `PNM * set_matrix (PNM *image, int **matrix)`
*Accesseur en écriture pour le champ `matrix` de `*image`.*
- `int ** create_matrix (PNM *image)`
*Crée et alloue dynamiquement la matrice de `*image`.*
- `int load_matrix (PNM *image, FILE *fp)`
*Lecture dans un fichier et remplissage de la matrice de `*image`.*
- `int write_matrix (PNM *image, FILE *fp)`
*Ecriture de la matrice de `*image` dans un fichier.*
- `void destroy_matrix_rows (PNM *image)`
*Libère la mémoire allouée par `**create_matrix`.*
- `void destroy_matrix_columns (PNM *image)`
*Libère la mémoire allouée par `**create_matrix`.*
- `int manage_comments (FILE *fp)`
Permet de gérer une ligne pour savoir si on doit l'ignorer (celles commençant par '#')

5.1.1 Detailed Description

Librairie pour gérer les fichiers d'extension pnm (.pbm, .pgm, .ppm)

[pnm.h](#)

Ce fichier contient les déclarations de types et les prototypes des fonctions pour la manipulation d'images PNM.

Author

: Dumoulin Peissone S193957

Date

: 22/02/21 @projet: INFO0030 Projet 1

Author

Peissone Dumoulin - Université de Liège

Version

1.0

Date

22/02/2021

Déclaration du type opaque PNM

5.1.2 Function Documentation

5.1.2.1 create_matrix()

```
int** create_matrix (
    PNM * image )
```

Crée et alloue dynamiquement la matrice de `*image`.

Parameters

<i>image</i>	un pointeur sur PNM
--------------	---------------------

Precondition

: image != NULL

Postcondition

: matrice allouée

Returns

: image->matrix Succès NULL Erreur lors de l'allocation dynamique

5.1.2.2 create_pnm()

```
PNM* create_pnm ( )
```

Crée et alloue dynamiquement une variable de type opaque PNM*.

Parameters

<i>magicNumber</i>	l'entier représentant la chaîne de caractère initiale (1 pour "P1", 2 pour "P2" et 3 pour "P3")
<i>columns</i>	le nombre de pixels de hauteur
<i>rows</i>	le nombre de pixels de largeur
<i>maxValuePixel</i>	la valeur maximale que peut prendre un pixel

Precondition

: /

Postcondition

: *image alloué

Returns

: image Succès NULL Erreur lors de l'allocation dynamique

5.1.2.3 destroy_matrix_columns()

```
void destroy_matrix_columns (
    PNM * image )
```

Libère la mémoire allouée par **create_matrix.

Parameters

<i>image</i>	un pointeur sur PNM
--------------	---------------------

Precondition

: *image* != NULL

Postcondition

: la matrice colonne est libérée

Returns

: /

5.1.2.4 destroy_matrix_rows()

```
void destroy_matrix_rows (  
    PNM * image )
```

Libère la mémoire allouée par `**create_matrix`.

Parameters

<i>image</i>	un pointeur sur PNM
--------------	---------------------

Precondition

: *image* != NULL

Postcondition

: la matrice ligne est libérée

Returns

: /

5.1.2.5 destroy_pnm()

```
void destroy_pnm (  
    PNM * image )
```

Libère la mémoire allouée par `*create_pnm`.

Parameters

<i>image</i>	un pointeur sur PNM
--------------	---------------------

Precondition

: image != NULL

Postcondition

: *image libéré

Returns

: /

5.1.2.6 get_columns()

```
int get_columns (  
    PNM * image )
```

Accesseur en lecture pour le champ columns de image*.

Parameters

<i>image</i>	un pointeur sur PNM
--------------	---------------------

Precondition

: image != NULL

Postcondition

: accès en lecture au champ columns de *image

Returns

: image->columns Succès

5.1.2.7 get_magicNumber()

```
int get_magicNumber (  
    PNM * image )
```

Accesseur en lecture pour le champ magicNumber de image*.

Parameters

<i>image</i>	un pointeur sur PNM
--------------	---------------------

Precondition

: *image* != NULL

Postcondition

: accès en lecture au champ *magicNumber* de **image*

Returns

: *image*->*magicNumber* Succès

5.1.2.8 get_matrix()

```
int** get_matrix (
    PNM * image )
```

Accesseur en lecture pour le champ *matrix* de *image**.

Parameters

<i>image</i>	un pointeur sur PNM
--------------	---------------------

Precondition

: *image* != NULL

Postcondition

: accès en lecture au champ *matrix* de **image*

Returns

: *image*->*matrix* Succès

5.1.2.9 get_maxValuePixel()

```
int get_maxValuePixel (
    PNM * image )
```

Accesseur en lecture pour le champ *maxValuePixel* de *image**.

Parameters

<i>image</i>	un pointeur sur PNM
--------------	---------------------

Precondition

: *image* != NULL

Postcondition

: accès en lecture au champ `maxValuePixel` de **image*

Returns

: *image*->`getMaxValuePixel` Succès

5.1.2.10 get_rows()

```
int get_rows (  
    PNM * image )
```

Accesseur en lecture pour le champ `rows` de *image**.

Parameters

<i>image</i>	un pointeur sur PNM
--------------	---------------------

Precondition

: *image* != NULL

Postcondition

: accès en lecture au champ `rows` de **image*

Returns

: *image*->`rows` Succès

5.1.2.11 load_matrix()

```
int load_matrix (  
    PNM * image,  
    FILE * fp )
```

Lecture dans un fichier et remplissage de la matrice de **image*.

Parameters

<i>image</i>	un pointeur sur PNM
<i>fp</i>	un pointeur sur FILE

Precondition

: image != NULL, fp != NULL

Postcondition

: matrice chargée

Returns

: 0 Succès -3 Le contenu du fichier en input est mal formé (magicNumber)

5.1.2.12 load_pnm()

```
int load_pnm (
    PNM ** image,
    char * filename )
```

Charge une image PNM depuis un fichier.

Parameters

<i>image</i>	l'adresse d'un pointeur sur PNM à laquelle écrire l'adresse de l'image chargée.
<i>filename</i>	le chemin vers le fichier contenant l'image.

Precondition

: image != NULL, filename != NULL

Postcondition

: image pointe vers l'image chargée depuis le fichier.

Returns

: 0 Succès -1 Erreur à l'allocation de mémoire -2 Nom du fichier malformé -3 Contenu du fichier malformé

5.1.2.13 manage_comments()

```
int manage_comments (
    FILE * fp )
```

Permet de gérer une ligne pour savoir si on doit l'ignorer (celles commençant par '#')

Parameters

<i>fp</i>	un pointeur sur FILE
-----------	----------------------

Precondition

: *fp* != NULL

Postcondition

: la ligne est correctement ignorée

Returns

: 0 Succès -1 Echec

5.1.2.14 set_columns()

```
PNM* set_columns (
    PNM * image,
    int columns )
```

Accesseur en écriture pour le champ *columns* de **image*.

Parameters

<i>image</i>	un pointeur sur PNM
<i>columns</i>	nombre de pixels de hauteur

Precondition

: *image* != NULL

Postcondition

: accès en écriture au champ *columns* de **image*

Returns

: *image* Succès

5.1.2.15 set_magicNumber()

```
PNM* set_magicNumber (
    PNM * image,
    int magicNumber )
```

Accesseur en écriture pour le champ *magicNumber* de **image*.

Parameters

<i>image</i>	un pointeur sur PNM
<i>magicNumber</i>	nombre qui caractérise le type de fichier (1 pour pbm, 2 pour pgm, 3 pour ppm)

Precondition

: image != NULL

Postcondition

: accès en écriture au champ magicNumber de *image

Returns

: image Succès

5.1.2.16 set_matrix()

```
PNM* set_matrix (
    PNM * image,
    int ** matrix )
```

Accesseur en écriture pour le champ matrix de *image.

Parameters

<i>image</i>	un pointeur sur PNM
<i>matrix</i>	matrice contenant la valeur de chaque pixel de l'image

Precondition

: image != NULL

Postcondition

: accès en écriture au champ matrix de *image

Returns

: image Succès

5.1.2.17 set_maxValuePixel()

```
PNM* set_maxValuePixel (
    PNM * image,
    int maxValuePixel )
```

Accesseur en écriture pour le champ `maxValuePixel` de `*image`.

Parameters

<i>image</i>	un pointeur sur PNM
<i>maxValuePixel</i>	valeur maximale que peut prendre un pixel

Precondition

: image != NULL

Postcondition

: accès en écriture au champ *maxValuePixel* de *image

Returns

: image Succès

5.1.2.18 set_rows()

```
PNM* set_rows (
    PNM * image,
    int rows )
```

Accesseur en écriture pour le champ *rows* de *image.

Parameters

<i>image</i>	un pointeur sur PNM
<i>rows</i>	nombre de pixels de largeur

Precondition

: image != NULL

Postcondition

: accès en écriture au champ *rows* de *image

Returns

: image Succès

5.1.2.19 write_matrix()

```
int write_matrix (
    PNM * image,
    FILE * fp )
```

Ecriture de la matrice de *image dans un fichier.

Parameters

<i>image</i>	un pointeur sur PNM
<i>fp</i>	un pointeur sur FILE

Precondition

: *image* != NULL, *fp* != NULL

Postcondition

: *matrice*

Returns

: 0 Succès -2 La matrice n'a pas pu être écrite dans le fichier

Index

- create_matrix
 - pnm.h, [10](#)
- create_pnm
 - pnm.h, [11](#)
- destroy_matrix_columns
 - pnm.h, [11](#)
- destroy_matrix_rows
 - pnm.h, [12](#)
- destroy_pnm
 - pnm.h, [12](#)
- get_columns
 - pnm.h, [13](#)
- get_magicNumber
 - pnm.h, [13](#)
- get_matrix
 - pnm.h, [14](#)
- get_maxValuePixel
 - pnm.h, [14](#)
- get_rows
 - pnm.h, [15](#)
- load_matrix
 - pnm.h, [15](#)
- load_pnm
 - pnm.h, [16](#)
- manage_comments
 - pnm.h, [16](#)
- pnm.h, [9](#)
 - create_matrix, [10](#)
 - create_pnm, [11](#)
 - destroy_matrix_columns, [11](#)
 - destroy_matrix_rows, [12](#)
 - destroy_pnm, [12](#)
 - get_columns, [13](#)
 - get_magicNumber, [13](#)
 - get_matrix, [14](#)
 - get_maxValuePixel, [14](#)
 - get_rows, [15](#)
 - load_matrix, [15](#)
 - load_pnm, [16](#)
 - manage_comments, [16](#)
 - set_columns, [17](#)
 - set_magicNumber, [17](#)
 - set_matrix, [18](#)
 - set_maxValuePixel, [18](#)
 - set_rows, [20](#)
 - write_matrix, [20](#)
- PNM_t, [7](#)
- set_columns
 - pnm.h, [17](#)
- set_magicNumber
 - pnm.h, [17](#)
- set_matrix
 - pnm.h, [18](#)
- set_maxValuePixel
 - pnm.h, [18](#)
- set_rows
 - pnm.h, [20](#)
- write_matrix
 - pnm.h, [20](#)