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 /home/peps/Desktop/Univ/Projet de programmation/Projets/Projet 1/pnm/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()	11
5.1.2.4 get_columns()	12
5.1.2.5 get_magicNumber()	12
5.1.2.6 get_matrix()	13
5.1.2.7 get_maxValuePixel()	13
5.1.2.8 get_rows()	14
5.1.2.9 load_matrix()	14
5.1.2.10 load_pnm()	15
5.1.2.11 manage_comments()	15
5.1.2.12 manage_format_input()	16
5.1.2.13 set_columns()	17
5.1.2.14 set_magicNumber()	17
5.1.2.15 set_matrix()	18
5.1.2.16 set_maxValuePixel()	18
5.1.2.17 set_rows()	19
5.1.2.18 verify_output()	19
5.1.2.19 write_matrix()	20
Index	21

Projet 1: Librairie de gestion d'images

Data Structure Index

\sim 4	D-1-	O4	_1
2.1	Data	Stru	ctures

ere are the data structures with brief descriptions:	
PNM_t	

Data Structure Index

File Index

3.1 File List

Here is a list of all documented files with brief description

/home/peps/Desktop/Univ/Projet de programmation/Projets/Projet 1/pnm/pnm.h	
Librairie pour gérer les fichiers d'extension pnm (.pbm, .pgm, .ppm)	ç

6 File Index

Data Structure Documentation

4.1 PNM_t Struct Reference

Data Fields

- char magicNumber [2]
- unsigned short columns
- · unsigned short rows
- unsigned short maxValuePixel
- unsigned short * matrix

4.1.1 Detailed Description

Définition du type opaque PNM

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

• /home/peps/Desktop/Univ/Projet de programmation/Projets/Projet 1/pnm/pnm.c

File Documentation

5.1 /home/peps/Desktop/Univ/Projet de programmation/Projets/Projet 1/pnm/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*.

char * get_magicNumber (PNM *image)

Accesseur en lecture pour le champ magicNumber de image*.

unsigned short get_columns (PNM *image)

Accesseur en lecture pour le champ columns de image*.

unsigned short get_rows (PNM *image)

Accesseur en lecture pour le champ rows de image*.

• unsigned short get_maxValuePixel (PNM *image)

Accesseur en lecture pour le champ maxValuePixel de image*.

unsigned short * get_matrix (PNM *image)

Accesseur en lecture pour le champ matrix de image*.

PNM * set magicNumber (PNM *image, char *magicNumber)

Accesseur en écriture pour le champ magicNumber de *image.

PNM * set_columns (PNM *image, unsigned short columns)

Accesseur en écriture pour le champ columns de *image.

PNM * set rows (PNM *image, unsigned short rows)

Accesseur en écriture pour le champ rows de *image.

PNM * set_maxValuePixel (PNM *image, unsigned short maxValuePixel)

Accesseur en écriture pour le champ maxValuePixel de *image.

PNM * set_matrix (PNM *image, unsigned short *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 (PNM *image, unsigned short allocation_value)

Libère la mémoire en fonction de l'allocation.

int manage_comments (FILE *fp)

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

• int manage format input (PNM *image, char *format, char *input)

Gère si le format correspond au format de l'input.

int verify_output (PNM *image, char *output)

Vérifie si l'output contient des caractères spéciaux interdits.

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

: 05/03/21 @projet: INFO0030 Projet 1

Author

Peissone Dumoulin - Université de Liège

Version

1.0

Date

19/02/2021

Déclaration du type opaque PNM

5.1.2 Function Documentation

5.1.2.1 create_matrix()

Crée et alloue dynamiquement la matrice de *image.

Parameters

image un pointeur sur PNM

Precondition

: image != NULL

Postcondition

: matrice allouée

Returns

: image->matrix Succès -1 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



Precondition

:/

Postcondition

: *image alloué

Returns

: image Succès NULL Erreur lors de l'allocation dynamique

5.1.2.3 destroy()

```
void destroy (  \begin{array}{c} {\tt PNM} \, * \, image, \\ \\ {\tt unsigned \ short \ } allocation\_value \ ) \end{array}
```

Libère la mémoire en fonction de l'allocation.

Parameters

image	un pointeur sur PNM
allocation_value	le nombre de "couches" d'allocations

Precondition

```
: image != NULL, 0 < allocation_value < 4
```

Postcondition

: autant de libérations que d'allocations mémoires

Returns

:/

5.1.2.4 get_columns()

Accesseur en lecture pour le champ columns de image*.

Parameters

ſ	image	un pointeur sur PNM	
---	-------	---------------------	--

Precondition

: image != NULL

Postcondition

: accès en lecture au champ columns de *image

Returns

: image->columns Succès

5.1.2.5 get_magicNumber()

Accesseur en lecture pour le champ magicNumber de image*.

Parameters

image un po	ointeur sur PNM
-------------	-----------------

Precondition

: image != NULL

Postcondition

: accès en lecture au champ magicNumber de *image

Returns

: image->magicNumber Succès

5.1.2.6 get_matrix()

Accesseur en lecture pour le champ matrix de image*.

Parameters

image	un pointeur sur PNM
iiiiagc	an pointed out i ivivi

Precondition

: image != NULL

Postcondition

: accès en lecture au champ matrix de *image

Returns

: image->matrix Succès

5.1.2.7 get_maxValuePixel()

```
unsigned short get_maxValuePixel (  {\tt PNM} \, * \, image \, )
```

Accesseur en lecture pour le champ maxValuePixel de image*.

Parameters

image	un pointeur sur PNM
-------	---------------------

Precondition

```
: image != NULL
```

Postcondition

: accès en lecture au champ maxValuePixel de *image

Returns

: image->getMaxValuePixel Succès

5.1.2.8 get_rows()

```
unsigned short get_rows ( {\tt PNM} \, * \, image \, )
```

Accesseur en lecture pour le champ rows de image*.

Parameters

image	un pointeur sur PNM
-------	---------------------

Precondition

: image != NULL

Postcondition

: accès en lecture au champ rows de *image

Returns

: image->rows Succès

5.1.2.9 load_matrix()

```
int load_matrix (  \begin{array}{c} {\tt PNM} \, * \, image, \\ {\tt FILE} \, * \, fp \, ) \end{array} \label{eq:pnm}
```

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

Parameters

image	un pointeur sur PNM
fp	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

5.1.2.10 load_pnm()

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

Charge une image PNM depuis un fichier.

Parameters

image	l'adresse d'un pointeur sur PNM à laquelle écrire l'adresse de l'image chargée.
filename	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

5.1.2.11 manage_comments()

```
int manage_comments (  {\tt FILE} \, * \, fp \, ) \\
```

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

Parameters

```
fp un pointeur sur FILE
```

Precondition

: fp != NULL

Postcondition

: la ligne est correctement ignorée

Returns

: 0 Succès -1 Echec

5.1.2.12 manage_format_input()

Gère si le format correspond au format de l'input.

Parameters

image	un pointeur sur PNM	
format	le format du fichier	
input	le nom du fichier en entrée	

Precondition

: image != NULL, format != NULL, input != NULL

Postcondition

: format du fichier géré correctement

Returns

: 0 Succès -1 Mauvais format passé en argument

5.1.2.13 set_columns()

Accesseur en écriture pour le champ columns de *image.

Parameters

image	un pointeur sur PNM
columns	nombre de pixels de hauteur

Precondition

: image != NULL

Postcondition

: accès en écriture au champ columns de *image

Returns

: image Succès

5.1.2.14 set_magicNumber()

Accesseur en écriture pour le champ magicNumber de *image.

Parameters

image	un pointeur sur PNM
magicNumber	nombre magique qui caractérise le type de fichier ("P1"" pour pbm, "P2" pour pgm, "P3" pour ppm)

Precondition

: image != NULL

Postcondition

: accès en écriture au champ magicNumber de *image

Returns

: image Succès

5.1.2.15 set_matrix()

Accesseur en écriture pour le champ matrix de *image.

Parameters

image	un pointeur sur PNM
matrix	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

5.1.2.16 set_maxValuePixel()

Accesseur en écriture pour le champ maxValuePixel de *image.

Parameters

image	un pointeur sur PNM	
maxValuePixel	valeur maximale que peut prendre un pixel	

Precondition

: image != NULL

Postcondition

: accès en écriture au champ maxValuePixel de *image

Returns

: image Succès

5.1.2.17 set_rows()

Accesseur en écriture pour le champ rows de *image.

Parameters

image	un pointeur sur PNM
rows	nombre de pixels de largeur

Precondition

: image != NULL

Postcondition

: accès en écriture au champ rows de *image

Returns

: image Succès

5.1.2.18 verify_output()

```
int verify_output (
PNM * image,
char * output )
```

Vérifie si l'output contient des caractères spéciaux interdits.

Parameters

image	un pointeur sur PNM	
output	le nom du fichier en sortie	

Precondition

: image != NULL, output != NULL

Postcondition

: format du fichier géré correctement

Returns

: 0 Succès -1 Caractère invalide dans le nom du fichier

5.1.2.19 write_matrix()

```
int write_matrix (  \begin{array}{cccc} {\tt PNM} \ * \ image, \\ & & \\ {\tt FILE} \ * \ fp \ ) \end{array}
```

Ecriture de la matrice de *image dans un fichier.

Parameters

image	un pointeur sur PNM
fp	un pointeur sur FILE

Precondition

: image != NULL, fp != NULL

Postcondition

: matrice

Returns

: 0 Succès -2 L'image n'a pas pu être sauvée dans un fichier

Index

/home/peps/Desktop/Univ/Projet de programma- tion/Projets/Projet 1/pnm/pnm.h, 9	write_matrix, 20 PNM t, 7
create_matrix	set columns
pnm.h, 10	pnm.h, 16
•	•
create_pnm	set_magicNumber
pnm.h, 11	pnm.h, 17
destroy	set_matrix
pnm.h, 11	pnm.h, 18
μ	set_maxValuePixel
get columns	pnm.h, 18
pnm.h, 12	set_rows
get_magicNumber	pnm.h, 19
pnm.h, 12	verify_output
get_matrix	pnm.h, 19
pnm.h, 13	prim.n, 19
get maxValuePixel	write matrix
pnm.h, 13	pnm.h, 20
get_rows	p, 20
pnm.h, 14	
load matrix	
pnm.h, 14	
load_pnm	
pnm.h, 15	
manage_comments	
pnm.h, 15	
manage_format_input	
pnm.h, 16	
pnm.h	
create_matrix, 10	
create_pnm, 11	
destroy, 11	
get_columns, 12	
get_magicNumber, 12	
get_matrix, 13	
get_maxValuePixel, 13	
get_rows, 14	
load_matrix, 14	
load_pnm, 15	
manage_comments, 15	
manage_format_input, 16	
set_columns, 16	
set_magicNumber, 17	
set matrix, 18	
set_maxValuePixel, 18	
set_rows, 19	
verify_output, 19	