PGM Image Processor

1.0

Generated by Doxygen 1.8.2

Fri Dec 21 2012 12:22:20

Contents

1 The mainpage documentation

This is a simple example of a mainpage you can create yourself. Place this inside of a file called `mainpage.dox` and use doxygen. If you specified `INPUT` or `FILE_PATTERNS` in your Doxyfile please add `.dox` to your file patterns or `mainpage.dox` to your INPUT files.

2 Todo List

Global getFileName (char *fileName)

filename validation

3 Module Index

3.1 Modules

Here is a list of all modules:

Stdin Operations 4

UI 2

4 Data Structure Index

4.1 Data Structures

Here are the data structures with brief descriptions:

PGM 5

5 File Index

5.1 File List

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

src/CPGM.c
PGM(P2) Image Library Implementation 5

src/CPGM.h
PGM(P2) Image Library 7

src/main.c
PGM(P2) Image Processor 10

6 Module Documentation

6.1 UI

User Interface functions.

Functions

- void negative (PGM ∗image)

  Negative Effect.
- void horizontalFlip (PGM ∗image)

  Horizontal Flip Effect.
- void verticalFlip (PGM ∗image)

  Vertical Effect.
- void rotate90C (PGM ∗image)

  Rotate90C Effect.
- void cProcess (PGM ∗image)

  Option 'c' - Create PGM.
- void eProcess (PGM ∗image)

  Option 'e' - Effects apply to PGM file.
- void mProcess (PGM ∗image)

  Option 'm' - ID Marking.
- void rProcess (PGM ∗image)

  Option 'r' - Read PGM file.
- void vProcess (const PGM ∗image)

  Option 'v' - Character-View.
- void wProcess (const PGM ∗image)

  Option 'w' - Write to PGM file.
- void printMainMenu ()

  Print the main menu and the option list.

6.1.1 Detailed Description

User Interface functions. All UI functions are strong exception safety. Which means operations can fail(i.e. user inputs the wrong data), but failed operations are guaranteed to have no side effects so all data retain original values.

6.1.2 Function Documentation

6.1.2.1 void cProcess (PGM ∗ *image*)

Option 'c' - Create PGM.

Create a PGM through stdin, store in memory and write to file.

Parameters

`out`	*image*	The memory area to hold the image data.

Definition at line 190 of file main.c.

6.1.2.2 void eProcess (PGM ∗ *image*)

Option 'e' - Effects apply to PGM file.

Sub-menu for applying Effect to image.

Parameters

`in,out`	*image*	The memory area to hold the image data.

Definition at line 331 of file main.c.

6.1.2.3 void mProcess (PGM ∗ *image*)

Option 'm' - ID Marking.

Steganography to embeds course code and student ID into the image.

Parameters

`in,out`	*image*	The memory area to hold the image data.

Definition at line 299 of file main.c.

6.1.2.4 void rProcess (PGM ∗ *image*)

Option 'r' - Read PGM file.

Load PGM file into memory.

Parameters

`out`	*image*	The memory area to hold the image data.

Definition at line 163 of file main.c.

6.1.2.5 void vProcess (const PGM ∗ *image*)

Option 'v' - Character-View.

View the stored image with user-specified characters that represent pixel values.

Parameters

`in`	*image*	The memory area to hold the image data.

Definition at line 281 of file main.c.

6.1.2.6 void wProcess (const PGM ∗ *image*)

Option 'w' - Write to PGM file.

Write the PGM to file. Strong exception safety.

Parameters

`in`	*image*	The memory area to hold the image data.

Definition at line 237 of file main.c.

6.2 Stdin Operations

Functions to handle stdin operations.

Functions

- void getFileName (char ∗fileName)

  Get filename from stdin & some validation works.
- char ∗ safeGetString (char ∗str, int buffSize)

  fgets from stdin and clear the remaining char in the stream
- int safeGetInt (char ∗message, int min, int max)

  sending a prompt and get a valid integer
- int CLIReadNum (char ∗string)

  get a number from CLI

6.2.1 Detailed Description

Functions to handle stdin operations. Handle the tricky things of using the standard input functions.

6.2.2 Function Documentation

6.2.2.1 int CLIReadNum (char ∗ *string*)

get a number from CLI

Return values

-1	invalid input

Definition at line 438 of file main.c.

6.2.2.2 void getFileName (char ∗ *fileName*)

Get filename from stdin & some validation works.

Parameters

`out`	*fileName*	

Todo filename validation

Definition at line 397 of file main.c.

6.2.2.3 char ∗ safeGetString (char ∗ *str,* int *buffSize*)

fgets from stdin and clear the remaining char in the stream

Parameters

`out`	*str*	Pointer to an array of chars where the string read is copied.
`out`	*buffSize*	Maximum number of characters to be copied into str (including the terminating null-character).

Definition at line 404 of file main.c.

7 Data Structure Documentation

7.1 PGM Struct Reference

```
#include <CPGM.h>
```

Data Fields

- char comment [MAX_COMMENT_LENGTH]
- int **width**
- int **height**
- int **greyMax**
- unsigned char pixelData [DEF_MAX_PIXEL_W ∗DEF_MAX_PIXEL_H]

7.1.1 Detailed Description

A structure to represent a PGM (P2) file

Definition at line 44 of file CPGM.h.

7.1.2 Field Documentation

7.1.2.1 char comment[MAX_COMMENT_LENGTH]

Comments show in the second line of the file

Definition at line 46 of file CPGM.h.

7.1.2.2 unsigned char pixelData[DEF_MAX_PIXEL_W ∗DEF_MAX_PIXEL_H]

Pixel data range from 0-255.

Definition at line 50 of file CPGM.h.

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

- src/CPGM.h

8 File Documentation

8.1 src/CPGM.c File Reference

PGM(P2) Image Library Implementation.

```
#include "CPGM.h"
```

Macros

- #define **MAX_FILE_BUFFER** 255

Functions

- static int **readNum** (FILE ∗file)
- int isNullPGM (const PGM ∗image)

Determine the input image is null or not.

- void setNullPGM (PGM ∗image)

  Set the input image to null.

- int readFilePGM (FILE ∗file, PGM ∗image)

  Read PGM file into memory.

- int **writeFilePGM** (FILE ∗file, const PGM ∗image, int useGroupComment)
- void **printPixelPGM** (FILE ∗file, const PGM ∗image, char ∗specChar)
- int **embedInfoPGM** (PGM ∗image, char ∗info)
- void **printAttPGM** (FILE ∗file, const PGM ∗image)
- void **reset** (PGM ∗image)
- void negative (PGM ∗image)

  Negative Effect.

- void horizontalFlip (PGM ∗image)

  Horizontal Flip Effect.

- void verticalFlip (PGM ∗image)

  Vertical Effect.

- void rotate90C (PGM ∗image)

  Rotate90C Effect.

Variables

- static const PGM nullImg ={"", -1, -1, -1, {0}}

8.1.1 Detailed Description

PGM(P2) Image Library Implementation.

Author

Oneonestar oneonestar@gmail.com

Version

1.0

Date

2012-10-27

Copyright

2012 Oneonestar

8.1.2 LICENSE

Definition in file CPGM.c.

8.1.3 Function Documentation

8.1.3.1 int isNullPGM (const PGM * *image*)

Determine the input image is null or not.

Parameters

image	the input image

Return values

1	is equal to null
0	is not equal to null

Definition at line 69 of file CPGM.c.

8.1.3.2 int readFilePGM (FILE * *file,* PGM * *image*)

Read PGM file into memory.

Parameters

`in`	*file*	opened file pointer
`out`	*memory*	location of the image

Definition at line 79 of file CPGM.c.

8.1.3.3 void setNullPGM (PGM * *image*)

Set the input image to null.

Precondition

image pointer is allocated

Parameters

`out`	*null*	image

Definition at line 74 of file CPGM.c.

8.1.4 Variable Documentation

8.1.4.1 const PGM nullImg ={"", -1, -1, -1, {0}} `[static]`

Null PGM

Definition at line 29 of file CPGM.c.

8.2 src/CPGM.h File Reference

PGM(P2) Image Library.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <ctype.h>
```

Data Structures

- struct PGM

Macros

- #define **DEF_MAX_PIXEL_W** 300
- #define **DEF_MAX_PIXEL_H** 300
- #define MAX_COMMENT_LENGTH 255

Functions

- int isNullPGM (const PGM ∗image)

  Determine the input image is null or not.
- void setNullPGM (PGM ∗image)

  Set the input image to null.
- int readFilePGM (FILE ∗file, PGM ∗image)

  Read PGM file into memory.
- int **writeFilePGM** (FILE ∗file, const PGM ∗image, int useGroupComment)
- int **embedInfoPGM** (PGM ∗image, char ∗info)
- void negative (PGM ∗image)

  Negative Effect.
- void horizontalFlip (PGM ∗image)

  Horizontal Flip Effect.
- void verticalFlip (PGM ∗image)

  Vertical Effect.
- void rotate90C (PGM ∗image)

  Rotate90C Effect.
- void **reset** (PGM ∗image)
- void **printPixelPGM** (FILE ∗file, const PGM ∗image, char ∗specChar)
- void **printAttPGM** (FILE ∗file, const PGM ∗image)

8.2.1 Detailed Description

PGM(P2) Image Library.

Author

Oneonestar `oneonestar@gmail.com`

Version

1.0

Date

2012-10-27

Copyright

2012 Oneonestar

8.2.2 LICENSE

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see `http://www.gnu.org/licenses/`.

Definition in file CPGM.h.

8.2.3 Macro Definition Documentation

8.2.3.1 #define MAX_COMMENT_LENGTH 255

Max comment length in the PGM file

Definition at line 40 of file CPGM.h.

8.2.4 Function Documentation

8.2.4.1 int isNullPGM (const PGM ∗ *image*)

Determine the input image is null or not.

Parameters

image	the input image

Return values

1	is equal to null
0	is not equal to null

Definition at line 69 of file CPGM.c.

8.2.4.2 int readFilePGM (FILE ∗ *file,* PGM ∗ *image*)

Read PGM file into memory.

Parameters

`in`	*file*	opened file pointer
`out`	*memory*	location of the image

Definition at line 79 of file CPGM.c.

8.2.4.3 void setNullPGM (PGM ∗ *image*)

Set the input image to null.

Precondition

image pointer is allocated

Parameters

out	*null*	image

Definition at line 74 of file CPGM.c.

8.3 src/main.c File Reference

PGM(P2) Image Processor.

```
#include "CPGM.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Macros

- #define MAX_STRING_BUFFER 255

Functions

- void cProcess (PGM ∗image)

  Option 'c' - Create PGM.
- void eProcess (PGM ∗image)

  Option 'e' - Effects apply to PGM file.
- void mProcess (PGM ∗image)

  Option 'm' - ID Marking.
- void rProcess (PGM ∗image)

  Option 'r' - Read PGM file.
- void vProcess (const PGM ∗image)

  Option 'v' - Character-View.
- void wProcess (const PGM ∗image)

  Option 'w' - Write to PGM file.
- void printMainMenu ()

  Print the main menu and the option list.
- void getFileName (char ∗fileName)

  Get filename from stdin & some validation works.
- char ∗ safeGetString (char ∗str, int buffSize)

  fgets from stdin and clear the remaining char in the stream
- int safeGetInt (char ∗message, int min, int max)

  sending a prompt and get a valid integer
- int CLIReadNum (char ∗string)

  get a number from CLI
- int **main** ()

8.3.1 Detailed Description

PGM(P2) Image Processor.

Author

Oneonestar oneonestar@gmail.com

Version

1.0

Date

2012-10-27

Copyright

2012 Oneonestar

8.3.2 LICENSE

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see http://www.gnu.org/licenses/.

Definition in file main.c.

8.3.3 Macro Definition Documentation

8.3.3.1 #define MAX_STRING_BUFFER 255

Max length of the command that may input

Definition at line 35 of file main.c.

Index