INF2 - Labo 2

Generated by Doxygen 1.8.12

Contents

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

File Index

2.1 File List

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

Chapter 3

Class Documentation

3.1 bmp_color_index_t Struct Reference

Public Attributes

- uint8_t **r**
- uint8_t **g**
- uint8_t **b**
- uint8_t **pad**

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

- bmp.h

3.2 bmp_file_header_t Struct Reference

Public Attributes

- uint16_t **bfType**
- uint32_t **bfSize**
- uint16_t **bfReserved1**
- uint16_t **bfReserved2**
- uint32_t **bfOffBits**

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

- bmp.h

3.3 bmp_info_header_t Struct Reference

Public Attributes

- uint32_t **biSize**
- int32_t **biWidth**
- int32_t **biHeight**
- uint16_t **biPlanes**
- uint16_t **biBitCount**
- uint32_t **biCompression**
- uint32_t **biSizeImage**
- int32_t **biXPelsPerMeter**
- int32_t **biYPelsPerMeter**
- uint32_t **biClrUsed**
- uint32_t **biClrImportant**

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

- bmp.h

Chapter 4

File Documentation

4.1 improc.h File Reference

: Functions able to make modification on images. The images are send in a vector of unsigned whole number representing the color of each pixels

```
#include <vector>
#include <cstdint>
```

Functions

- void invert (vector< uint8_t > &img_bw, const uint32_t w, const uint32_t h)

  Invert the color of a image.
- void binary (vector< uint8_t > &img_bw, const uint32_t w, const uint32_t h, const uint8_t mid_value)

  Convert the ligth color in white and dark color in black.
- void draw_diag (vector< uint8_t > &img_lines, const uint32_t w, const uint32_t h, const uint32_t w_center, const uint32_t h_center, const int direction, const uint8_t color)

  Draw a diagonal on a image.
- void draw_hline (vector< uint8_t > &img_lines, const uint32_t w, const uint32_t h, const uint32_t pos_vertical, const uint32_t thickness, const uint8_t color)

  draw a horizontal line on a image
- void draw_vline (vector< uint8_t > &img_lines, const uint32_t w, const uint32_t h, const uint32_t pos_horizontal, const uint32_t thickness, const uint8_t color)

  draw a vertical line on a image
- void checker (vector< uint8_t > &img_checker, const uint32_t w, const uint32_t h, const uint32_t size_square)

  draw a checker on a image
- void life_game (vector< uint8_t > &img_life, const uint32_t w_life, const uint32_t h_life, const uint32_t nb_iteration)

  draw the result of a number of iteration of the game of life

4.1.1 Detailed Description

: Functions able to make modification on images. The images are send in a vector of unsigned whole number representing the color of each pixels

Projet : Labo2

Author

: James Smith

Date

: 01.03.2017

Remarque(s) : < à="" compléter>="">

Compilateur : MinGW-g++ 5.3.0

4.1.2 Function Documentation

4.1.2.1 binary()

```
void binary (
             vector< uint8_t > & img_bw,
             const uint32_t w,
             const uint32_t h,
             const uint8_t mid_value )
```

Convert the ligth color in white and dark color in black.

Parameters

img_bw	image to convert in black and white
w	width of the image
h	higth of the image
mid_value	the value limit to change in white or in black

4.1.2.2 checker()

```
void checker (
             vector< uint8_t > & img_checker,
             const uint32_t w,
             const uint32_t h,
             const uint32_t size_square )
```

draw a checker on a image

Parameters

img_lines	image to draw on
w	width of the image
h	hight of the image
size_square	is the size of the squares

4.1.2.3 draw_diag()

```
void draw_diag (
             vector< uint8_t > & img_lines,
             const uint32_t w,
             const uint32_t h,
             const uint32_t w_center,
             const uint32_t h_center,
             const int direction,
             const uint8_t color )
```

Draw a diagonal on a image.

Parameters

img_lines	imgage to draw on.
w	width of the image
h	higth of the image
w_center	middle of the width
h_center	middle of the hight
direction	is the slope of the diagonal

4.1.2.4 draw_hline()

```
void draw_hline (
             vector< uint8_t > & img_lines,
             const uint32_t w,
             const uint32_t h,
             const uint32_t pos_vertical,
             const uint32_t thickness,
             const uint8_t color )
```

draw a horizontal line on a image

Parameters

img_lines	image to draw on.
w	width of the image
h	higth of the image
pos_vertical	is the position horizontal where start the line.
color	of the diagonal

4.1.2.5 draw_vline()

```
void draw_vline (
             vector< uint8_t > & img_lines,
             const uint32_t w,
             const uint32_t h,
             const uint32_t pos_horizontal,
             const uint32_t thickness,
             const uint8_t color )
```

draw a vertical line on a image

Parameters

img_lines	image to draw on.
w	width of the image
h	higth of the image
pos_horizontal	is the position vertical where start the line
thickness	of the line
color	of the line

4.1.2.6 invert()

```
void invert (
             vector< uint8_t > & img_bw,
             const uint32_t w,
             const uint32_t h )
```

Invert the color of a image.

Parameters

img_bw	a vector with all the pixels's color in 8 bits
w	width of the image
h	higth of the image

4.1.2.7 life_game()

```
void life_game (
             vector< uint8_t > & img_life,
             const uint32_t w_life,
             const uint32_t h_life,
             const uint32_t nb_iteration )
```

draw the result of a number of iteration of the game of life

Parameters

img_life	image to draw on
w_life	is the width of the image
h_life	is the higth of the image
nb_iteration	is the number of iteration to play to the game of life

Index