

LIBJHI-SDL

2.0

Generated by Doxygen 1.8.6

Sat Nov 8 2014 02:20:15



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	JHI_Color_RGB Struct Reference . . . . .	5
3.1.1	Detailed Description . . . . .	5
3.1.2	Member Data Documentation . . . . .	5
3.1.2.1	b . . . . .	5
3.1.2.2	g . . . . .	5
3.1.2.3	r . . . . .	5
3.2	JHI_Effect Struct Reference . . . . .	5
3.2.1	Detailed Description . . . . .	6
3.2.2	Member Data Documentation . . . . .	6
3.2.2.1	mix_chunk . . . . .	6
3.3	JHI_Font Struct Reference . . . . .	6
3.3.1	Detailed Description . . . . .	6
3.3.2	Member Data Documentation . . . . .	6
3.3.2.1	font . . . . .	6
3.3.2.2	length . . . . .	6
3.4	JHI_Image Struct Reference . . . . .	7
3.4.1	Detailed Description . . . . .	7
3.4.2	Member Data Documentation . . . . .	7
3.4.2.1	pos . . . . .	7
3.4.2.2	sur . . . . .	7
3.5	JHI_JoystickSt Struct Reference . . . . .	7
3.5.1	Detailed Description . . . . .	7
3.5.2	Member Data Documentation . . . . .	8
3.5.2.1	axis_index . . . . .	8
3.5.2.2	axis_value . . . . .	8

3.5.2.3	button	8
3.5.2.4	button_state	8
3.5.2.5	joy_event	8
3.5.2.6	joy_index	8
3.6	JHI_KeyboardSt Struct Reference	8
3.6.1	Detailed Description	8
3.6.2	Member Data Documentation	8
3.6.2.1	key	8
3.6.2.2	key_event	9
3.7	JHI_MouseSt Struct Reference	9
3.7.1	Detailed Description	9
3.7.2	Member Data Documentation	9
3.7.2.1	mouse_event	9
3.7.2.2	x	9
3.7.2.3	y	9
3.8	JHI_Music Struct Reference	9
3.8.1	Detailed Description	10
3.8.2	Member Data Documentation	10
3.8.2.1	mix_music	10
3.9	JHI_Point2d Struct Reference	10
3.9.1	Detailed Description	10
3.9.2	Member Data Documentation	10
3.9.2.1	x	10
3.9.2.2	y	10
3.10	JHI_Text Struct Reference	10
3.10.1	Detailed Description	11
3.10.2	Member Data Documentation	11
3.10.2.1	pos	11
3.10.2.2	sur	11
3.11	JHI_Window Struct Reference	11
3.11.1	Detailed Description	11
3.11.2	Member Data Documentation	11
3.11.2.1	back_color	11
3.11.2.2	check_quit	12
3.11.2.3	event	12
3.11.2.4	height	12
3.11.2.5	joy	12
3.11.2.6	key	12
3.11.2.7	mouse	12
3.11.2.8	number_of_events	12

3.11.2.9	screen	12
3.11.2.10	width	12
<b>4</b>	<b>File Documentation</b>	<b>13</b>
4.1	jhi_colorsSt.h File Reference	13
4.1.1	Detailed Description	13
4.2	jhi_font.h File Reference	13
4.2.1	Detailed Description	14
4.2.2	Function Documentation	14
4.2.2.1	jhi_get_lenght_font	14
4.2.2.2	jhi_load_font	14
4.3	jhi_image.h File Reference	14
4.3.1	Detailed Description	15
4.3.2	Function Documentation	15
4.3.2.1	jhi_draw_image	15
4.3.2.2	jhi_draw_image_with_clip	15
4.3.2.3	jhi_get_image_height	15
4.3.2.4	jhi_get_image_width	16
4.3.2.5	jhi_load_image	16
4.3.2.6	jhi_load_image_with_transparent_color	16
4.3.2.7	jhi_resize_image	16
4.4	jhi_joystick.h File Reference	16
4.4.1	Detailed Description	17
4.4.2	Enumeration Type Documentation	18
4.4.2.1	JHI_JOYSTICK_EVENT	18
4.4.3	Function Documentation	18
4.4.3.1	jhi_free_joystick_index	18
4.4.3.2	jhi_get_joystick_dir	18
4.4.3.3	jhi_get_num_of_joystick	18
4.4.3.4	jhi_init_joystick	18
4.4.3.5	jhi_is_valid_joystick_index	19
4.4.3.6	jhi_open_joystick_index	19
4.5	jhi_keyboard.h File Reference	19
4.5.1	Detailed Description	20
4.5.2	Enumeration Type Documentation	20
4.5.2.1	JHI_KeyBoardEvents	20
4.5.2.2	JHI_Keys	20
4.5.3	Function Documentation	21
4.5.3.1	jhi_get_opposite_key_arrow	21
4.5.3.2	jhi_init_keyboard	21

4.5.3.3	<a href="#">jhi_is_key_arrow</a>	21
4.6	<a href="#">jhi_mouse.h File Reference</a>	22
4.6.1	<a href="#">Detailed Description</a>	22
4.6.2	<a href="#">Enumeration Type Documentation</a>	22
4.6.2.1	<a href="#">JHI_MouseEvents</a>	22
4.6.3	<a href="#">Function Documentation</a>	23
4.6.3.1	<a href="#">jhi_check_mouse_is_in</a>	23
4.6.3.2	<a href="#">jhi_init_mouse</a>	24
4.7	<a href="#">jhi_rand.h File Reference</a>	24
4.7.1	<a href="#">Detailed Description</a>	24
4.8	<a href="#">jhi_shapes.h File Reference</a>	24
4.8.1	<a href="#">Detailed Description</a>	25
4.8.2	<a href="#">Function Documentation</a>	25
4.8.2.1	<a href="#">jhi_draw_circle</a>	25
4.8.2.2	<a href="#">jhi_draw_fill_circle</a>	25
4.8.2.3	<a href="#">jhi_draw_fill_rect</a>	25
4.8.2.4	<a href="#">jhi_draw_line</a>	26
4.8.2.5	<a href="#">jhi_draw_point</a>	26
4.8.2.6	<a href="#">jhi_draw_polygon</a>	26
4.8.2.7	<a href="#">jhi_draw_rect</a>	26
4.8.2.8	<a href="#">jhi_get_central_pos</a>	26
4.8.2.9	<a href="#">jhi_get_point</a>	27
4.8.2.10	<a href="#">jhi_is_colid</a>	27
4.9	<a href="#">jhi_sound.h File Reference</a>	27
4.9.1	<a href="#">Detailed Description</a>	28
4.9.2	<a href="#">Function Documentation</a>	28
4.9.2.1	<a href="#">jhi_load_effect</a>	28
4.9.2.2	<a href="#">jhi_load_music</a>	28
4.9.2.3	<a href="#">jhi_play_effect</a>	28
4.9.2.4	<a href="#">jhi_play_music</a>	29
4.10	<a href="#">jhi_text.h File Reference</a>	29
4.10.1	<a href="#">Detailed Description</a>	29
4.10.2	<a href="#">Function Documentation</a>	29
4.10.2.1	<a href="#">jhi_draw_text</a>	29
4.10.2.2	<a href="#">jhi_get_text_height</a>	30
4.10.2.3	<a href="#">jhi_get_text_width</a>	30
4.10.2.4	<a href="#">jhi_init_text</a>	30
4.10.2.5	<a href="#">jhi_set_text</a>	30
4.11	<a href="#">jhi_timer.h File Reference</a>	30
4.11.1	<a href="#">Detailed Description</a>	31

4.11.2	Function Documentation	31
4.11.2.1	jhi_delay	31
4.11.2.2	jhi_set_fps_timer	31
4.12	jhi_window.h File Reference	31
4.12.1	Detailed Description	32
4.12.2	Function Documentation	33
4.12.2.1	jhi_choice_window_name	33
4.12.2.2	jhi_draw_object	34
4.12.2.3	jhi_draw_object_with_clip	34
4.12.2.4	jhi_get_close_window	34
4.12.2.5	jhi_get_height_window	34
4.12.2.6	jhi_get_joystick_status	34
4.12.2.7	jhi_get_keyboard_status	35
4.12.2.8	jhi_get_mouse_status	36
4.12.2.9	jhi_get_number_of_events	36
4.12.2.10	jhi_get_width_window	36
4.12.2.11	jhi_initialize_window	36
4.12.2.12	jhi_out_window	36
4.12.2.13	jhi_print_pixel	37
4.12.2.14	jhi_set_background_color	37
	<b>Index</b>	<b>38</b>





# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">JHI_Color_RGB</a>	
Structure of the RGB color . . . . .	5
<a href="#">JHI_Effect</a>	
Struct of the configuration of the Effect . . . . .	5
<a href="#">JHI_Font</a>	
Structure of configuration of the Font . . . . .	6
<a href="#">JHI_Image</a>	
Structure of configuration of the Image . . . . .	7
<a href="#">JHI_JoystickSt</a>	
Joystick structure . . . . .	7
<a href="#">JHI_KeyboardSt</a>	
Keyboard structure . . . . .	8
<a href="#">JHI_MouseSt</a>	
Mouse structure . . . . .	9
<a href="#">JHI_Music</a>	
Struct of configuration of the Music . . . . .	9
<a href="#">JHI_Point2d</a>	
Point that contain position (x,y) . . . . .	10
<a href="#">JHI_Text</a>	
Structure of configuration of the Text . . . . .	10
<a href="#">JHI_Window</a>	
Struct of configuration of the Window . . . . .	11



## Chapter 2

# File Index

### 2.1 File List

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

<a href="#">jhi_colorsSt.h</a>	This file contains structures and enumeration of the Colors . . . . .	13
<a href="#">jhi_font.h</a>	This file contains functions structure of the Font . . . . .	13
<a href="#">jhi_image.h</a>	This file contains functions structure of the Image . . . . .	14
<a href="#">jhi_joystick.h</a>	This file contains structures and enumeration of keyboard . . . . .	16
<a href="#">jhi_keyboard.h</a>	This file contains structures and enumeration of keyboard . . . . .	19
<a href="#">jhi_mouse.h</a>	This file contains structure of the mouse . . . . .	22
<a href="#">jhi_rand.h</a>	This file contains rand auxiliary MACROS . . . . .	24
<a href="#">jhi_shapes.h</a>	This file contains shapes functions of libjhi-sdl . . . . .	24
<a href="#">jhi_sound.h</a>	This file contains functions and structures of the Music and Effect . . . . .	27
<a href="#">jhi_text.h</a>	This file contains functions and structures of the Text . . . . .	29
<a href="#">jhi_timer.h</a>	This file contains functions structure of the Timer to control number of frames per second (fps)	30
<a href="#">jhi_window.h</a>	This file contains functions related to Window . . . . .	31



## Chapter 3

# Class Documentation

### 3.1 JHI\_Color\_RGB Struct Reference

Structure of the RGB color.

```
#include <jhi_colorsSt.h>
```

#### Public Attributes

- Uint8 [r](#)
- Uint8 [g](#)
- Uint8 [b](#)

#### 3.1.1 Detailed Description

Structure of the RGB color.

#### 3.1.2 Member Data Documentation

##### 3.1.2.1 Uint8 JHI\_Color\_RGB::b

Blue value to color

##### 3.1.2.2 Uint8 JHI\_Color\_RGB::g

Green value to color

##### 3.1.2.3 Uint8 JHI\_Color\_RGB::r

Red value to color

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

- [jhi\\_colorsSt.h](#)

### 3.2 JHI\_Effect Struct Reference

Struct of the configuration of the Effect.

```
#include <jhi_sound.h>
```

## Public Attributes

- Mix\_Chunk \* [mix\\_chunk](#)

### 3.2.1 Detailed Description

Struct of the configuration of the Effect.

### 3.2.2 Member Data Documentation

#### 3.2.2.1 Mix\_Chunk\* JHI\_Effect::mix\_chunk

SDL structure of Effect.

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

- [jhi\\_sound.h](#)

## 3.3 JHI\_Font Struct Reference

Structure of configuration of the Font.

```
#include <jhi_font.h>
```

## Public Attributes

- TTF\_Font \* [font](#)
- int [length](#)

### 3.3.1 Detailed Description

Structure of configuration of the Font.

### 3.3.2 Member Data Documentation

#### 3.3.2.1 TTF\_Font\* JHI\_Font::font

SDL structure of Font

#### 3.3.2.2 int JHI\_Font::length

Length of the Font

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

- [jhi\\_font.h](#)

## 3.4 JHI\_Image Struct Reference

Structure of configuration of the Image.

```
#include <jhi_image.h>
```

### Public Attributes

- [JHI\\_Point2d](#) pos
- [SDL\\_Surface](#) \* [sur](#)

#### 3.4.1 Detailed Description

Structure of configuration of the Image.

#### 3.4.2 Member Data Documentation

##### 3.4.2.1 JHI\_Point2d JHI\_Image::pos

Image's position

##### 3.4.2.2 [SDL\\_Surface](#)\* JHI\_Image::sur

SDL Surface of the Window.

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

- [jhi\\_image.h](#)

## 3.5 JHI\_JoystickSt Struct Reference

Joystick structure.

```
#include <jhi_joystick.h>
```

### Public Attributes

- [JHI\\_JOYSTICK\\_EVENT](#) joy\_event
- short int [joy\\_index](#)
- short int [axis\\_index](#)
- short int [axis\\_value](#) [JHI\_NUMBER\_AXES]
- short int [button](#)
- short int [button\\_state](#)

#### 3.5.1 Detailed Description

Joystick structure.

### 3.5.2 Member Data Documentation

#### 3.5.2.1 short int JHI\_JoystickSt::axis\_index

Joystick axis\_index that was captured

#### 3.5.2.2 short int JHI\_JoystickSt::axis\_value[JHI\_NUMBER\_AXES]

Joystick axis values

#### 3.5.2.3 short int JHI\_JoystickSt::button

Joystick value of button pressed

#### 3.5.2.4 short int JHI\_JoystickSt::button\_state

Joystick buttons is pressed or no

#### 3.5.2.5 JHI\_JOYSTICK\_EVENT JHI\_JoystickSt::joy\_event

Joystick event that was captured

#### 3.5.2.6 short int JHI\_JoystickSt::joy\_index

Joystick index that was captured

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

- [jhi\\_joystick.h](#)

## 3.6 JHI\_KeyboardSt Struct Reference

Keyboard structure.

```
#include <jhi_keyboard.h>
```

### Public Attributes

- Uint8 [key\\_event](#)
- [JHI\\_Keys](#) key

### 3.6.1 Detailed Description

Keyboard structure.

### 3.6.2 Member Data Documentation

#### 3.6.2.1 JHI\_Keys JHI\_KeyboardSt::key

Keyboard key that was pressed



### 3.6.2.2 Uint8 JHI\_KeyboardSt::key\_event

Keyboard event that was captured

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

- [jhi\\_keyboard.h](#)

## 3.7 JHI\_MouseSt Struct Reference

Mouse structure.

```
#include <jhi_mouse.h>
```

### Public Attributes

- int [x](#)
- int [y](#)
- [JHI\\_MouseEvents mouse\\_event](#)

### 3.7.1 Detailed Description

Mouse structure.

### 3.7.2 Member Data Documentation

#### 3.7.2.1 JHI\_MouseEvents JHI\_MouseSt::mouse\_event

Mouse event that was captured

#### 3.7.2.2 int JHI\_MouseSt::x

Coordinate x of the mouse

#### 3.7.2.3 int JHI\_MouseSt::y

Coordinate y of the mouse

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

- [jhi\\_mouse.h](#)

## 3.8 JHI\_Music Struct Reference

Struct of configuration of the Music.

```
#include <jhi_sound.h>
```

### Public Attributes

- [Mix\\_Music \\* mix\\_music](#)

### 3.8.1 Detailed Description

Struct of configuration of the Music.

### 3.8.2 Member Data Documentation

#### 3.8.2.1 Mix\_Music\* JHI\_Music::mix\_music

SDL structure of Music.

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

- [jhi\\_sound.h](#)

## 3.9 JHI\_Point2d Struct Reference

Point that contain position (x,y)

```
#include <jhi_shapes.h>
```

### Public Attributes

- int [x](#)
- int [y](#)

### 3.9.1 Detailed Description

Point that contain position (x,y)

### 3.9.2 Member Data Documentation

#### 3.9.2.1 int JHI\_Point2d::x

X position

#### 3.9.2.2 int JHI\_Point2d::y

Y position

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

- [jhi\\_shapes.h](#)

## 3.10 JHI\_Text Struct Reference

Structure of configuration of the Text.

```
#include <jhi_text.h>
```

## Public Attributes

- [JHI\\_Point2d](#) pos
- [SDL\\_Surface](#) \* [sur](#)

### 3.10.1 Detailed Description

Structure of configuration of the Text.

### 3.10.2 Member Data Documentation

#### 3.10.2.1 [JHI\\_Point2d](#) [JHI\\_Text::pos](#)

Position of text

#### 3.10.2.2 [SDL\\_Surface](#)\* [JHI\\_Text::sur](#)

SDL Surface of the Window

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

- [jhi\\_text.h](#)

## 3.11 JHI\_Window Struct Reference

Struct of configuration of the Window.

```
#include <jhi_window.h>
```

## Public Attributes

- [SDL\\_Surface](#) \* [screen](#)
- char [check\\_quit](#)
- [SDL\\_Event](#) [event](#)
- [JHI\\_Color](#) [back\\_color](#)
- [JHI\\_MouseSt](#) [mouse](#) [NUMBER\_MAX\_EVENTS]
- [JHI\\_KeyboardSt](#) [key](#) [NUMBER\_MAX\_EVENTS]
- [JHI\\_JoystickSt](#) [joy](#) [NUMBER\_MAX\_EVENTS]
- int [number\\_of\\_events](#)
- int [width](#)
- int [height](#)

### 3.11.1 Detailed Description

Struct of configuration of the Window.

### 3.11.2 Member Data Documentation

#### 3.11.2.1 [JHI\\_Color](#) [JHI\\_Window::back\\_color](#)

Background color of the Window

### 3.11.2.2 char JHI\_Window::check\_quit

Flag that indicate if the Window was close

### 3.11.2.3 SDL\_Event JHI\_Window::event

SDL event struct

### 3.11.2.4 int JHI\_Window::height

Height of the Window

### 3.11.2.5 JHI\_JoystickSt JHI\_Window::joy[NUMBER\_MAX\_EVENTS]

Vector to joystick status for each event captured

### 3.11.2.6 JHI\_KeyboardSt JHI\_Window::key[NUMBER\_MAX\_EVENTS]

Vector to keyboard status for each event captured

### 3.11.2.7 JHI\_MouseSt JHI\_Window::mouse[NUMBER\_MAX\_EVENTS]

Vector to mouse status for each event captured

### 3.11.2.8 int JHI\_Window::number\_of\_events

Number of events captured in the iteration

### 3.11.2.9 SDL\_Surface\* JHI\_Window::screen

SDL Surface of the Window.

### 3.11.2.10 int JHI\_Window::width

Width of the Window

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

- [jhi\\_window.h](#)

## Chapter 4

# File Documentation

### 4.1 jhi\_colorsSt.h File Reference

This file contains structures and enumeration of the Colors.

```
#include "SDL/SDL.h"
```

#### Classes

- struct [JHI\\_Color\\_RGB](#)  
*Structure of the RGB color.*

#### Enumerations

- enum [JHI\\_Color](#) {  
**RED, GREEN, BLUE, YELLOW,**  
**BLACK, WHITE, ORANGE** }  
*Enumeration of the possibles colors for this lib.*

#### 4.1.1 Detailed Description

This file contains structures and enumeration of the Colors.

### 4.2 jhi\_font.h File Reference

This file contains functions structure of the Font.

```
#include "SDL/SDL_ttf.h"  
#include "jhi_shapes.h"  
#include <stdio.h>
```

#### Classes

- struct [JHI\\_Font](#)  
*Structure of configuration of the Font.*

## Functions

- void [jhi\\_load\\_font](#) ([JHI\\_Font](#) \*font, const char \*font\_name, int length)  
*Load the music with the font\_name.*
- void [jhi\\_free\\_font](#) ([JHI\\_Font](#) \*font)  
*Free font structure.*
- int [jhi\\_get\\_lenght\\_font](#) ([JHI\\_Font](#) \*font)  
*Get the Font length.*

### 4.2.1 Detailed Description

This file contains functions structure of the Font.

### 4.2.2 Function Documentation

#### 4.2.2.1 int [jhi\\_get\\_lenght\\_font](#) ( [JHI\\_Font](#) \* font )

Get the Font length.

#### Returns

Font length

#### 4.2.2.2 void [jhi\\_load\\_font](#) ( [JHI\\_Font](#) \* font, const char \* font\_name, int length )

Load the music with the font\_name.

#### Parameters

<i>font</i>	Structure of the Font to load
<i>font_name</i>	Name of the Font file
<i>length</i>	Length of the Font

## 4.3 [jhi\\_image.h](#) File Reference

This file contains functions structure of the Image.

```
#include "SDL/SDL.h"
#include "SDL/SDL_image.h"
#include "SDL/SDL_rotozoom.h"
#include "jhi_colorsSt.h"
#include "jhi_shapes.h"
#include <stdio.h>
```

## Classes

- struct [JHI\\_Image](#)  
*Structure of configuration of the Image.*

## Functions

- void [jhi\\_load\\_image](#) ([JHI\\_Image](#) \*img, const char \*filename)  
*Load the Image with the filename.*
- void [jhi\\_load\\_image\\_with\\_transparent\\_color](#) ([JHI\\_Image](#) \*img, const char \*filename, [JHI\\_Color](#) col)  
*Load the Image with the filename and that will be transparent in the Image.*
- void [jhi\\_free\\_image](#) ([JHI\\_Image](#) \*img)  
*Free image structure.*
- int [jhi\\_get\\_image\\_width](#) ([JHI\\_Image](#) \*img)  
*Get the Image width.*
- int [jhi\\_get\\_image\\_height](#) ([JHI\\_Image](#) \*img)  
*Get the Image height.*
- void [jhi\\_resize\\_image](#) ([JHI\\_Image](#) \*img, int width, int height)  
*resize the image*
- void [jhi\\_draw\\_image](#) ([JHI\\_Image](#) \*img, [JHI\\_Point2d](#) point)  
*Draw image in the window.*
- void [jhi\\_draw\\_image\\_with\\_clip](#) ([JHI\\_Image](#) \*img, [JHI\\_Point2d](#) point\_position, [JHI\\_Point2d](#) point\_clip, int width, int height)  
*Draw clip image in the window.*

### 4.3.1 Detailed Description

This file contains functions structure of the Image.

### 4.3.2 Function Documentation

#### 4.3.2.1 void [jhi\\_draw\\_image](#) ( [JHI\\_Image](#) \* *img*, [JHI\\_Point2d](#) *point* )

Draw image in the window.

Parameters

<i>img</i>	Image Structure
<i>point</i>	new pos of the image

#### 4.3.2.2 void [jhi\\_draw\\_image\\_with\\_clip](#) ( [JHI\\_Image](#) \* *img*, [JHI\\_Point2d](#) *point\_position*, [JHI\\_Point2d](#) *point\_clip*, int *width*, int *height* )

Draw clip image in the window.

Parameters

<i>img</i>	Image Structure
<i>point</i>	new pos of the clip image
<i>width</i>	Width of this clip
<i>height</i>	Height of this clip

#### 4.3.2.3 int [jhi\\_get\\_image\\_height](#) ( [JHI\\_Image](#) \* *img* )

Get the Image height.

## Parameters

<i>img</i>	Image Structure
------------	-----------------

## Returns

Image height

#### 4.3.2.4 int jhi\_get\_image\_width ( JHI\_Image \* *img* )

Get the Image width.

## Parameters

<i>img</i>	Image Structure
------------	-----------------

## Returns

Image width

#### 4.3.2.5 void jhi\_load\_image ( JHI\_Image \* *img*, const char \* *filename* )

Load the Image with the filename.

## Parameters

<i>img</i>	Structure of Image to load
<i>filename</i>	Name of the music file

#### 4.3.2.6 void jhi\_load\_image\_with\_transparent\_color ( JHI\_Image \* *img*, const char \* *filename*, JHI\_Color *col* )

Load the Image with the filename and that will be transparent in the Image.

## Parameters

<i>img</i>	Structure of the image to load
<i>filename</i>	Name of the music file
<i>cor</i>	Transparent color in the image

#### 4.3.2.7 void jhi\_resize\_image ( JHI\_Image \* *img*, int *width*, int *height* )

resize the image

## Parameters

<i>img</i>	Image Structure
<i>width</i>	the new width for image
<i>height</i>	the new height for image

## 4.4 jhi\_joystick.h File Reference

This file contains structures and enumeration of keyboard.

```
#include "SDL/SDL.h"
```



## Classes

- struct [JHI\\_JoystickSt](#)  
*Joystick structure.*

## Macros

- #define **JHI\_NUMBER\_MAX\_JOYSTICKS** 2
- #define **JHI\_NUMBER\_AXES** 2
- #define **JHI\_JOY\_BUTTON\_PRESSED** 1
- #define **JHI\_JOY\_BUTTON\_RELEASED** 0
- #define **JHI\_MAX\_AXIS\_VALUE** 32767
- #define **JHI\_MIN\_AXIS\_VALUE** -32768

## Enumerations

- enum [JHI\\_JOYSTICK\\_EVENT](#) { [JHI\\_JOY\\_AXIS\\_MOTION](#) = 7, [JHI\\_JOY\\_BUTTON\\_DOWN](#) = 10, [JHI\\_JOY\\_BUTTON\\_UP](#) = 11, [JHI\\_JOY\\_NOT\\_EVENT](#) }  
*Enumeration of events of joystick.*
- enum [JHI\\_JOY\\_DIR](#) { [JOY\\_LEFT](#), [JOY\\_RIGHT](#), [JOY\\_UP](#), [JOY\\_DOWN](#), [JOY\\_NOT\\_DIR](#), [JOY\\_DIR\\_RELEASED](#) }  
*Enumeration of the possibles dir that will pressed in the joystick.*
- enum [JHI\\_JOY\\_BUTTONS](#) { [JOY\\_BUTTON\\_0](#), [JOY\\_BUTTON\\_1](#), [JOY\\_BUTTON\\_2](#), [JOY\\_BUTTON\\_3](#), [JOY\\_BUTTON\\_4](#), [JOY\\_BUTTON\\_5](#), [JOY\\_BUTTON\\_6](#), [JOY\\_BUTTON\\_7](#), [JOY\\_BUTTON\\_8](#), [JOY\\_BUTTON\\_9](#), [JOY\\_NOT\\_BUTTON](#) }  
*Enumeration of the possibles buttons that will pressed in the joystick.*

## Functions

- void [jhi\\_init\\_joystick](#) ([JHI\\_JoystickSt](#) \*joy)  
*Initialize the joystick structure.*
- int [jhi\\_open\\_joystick\\_index](#) (short int joy\_index)  
*Open Joystick with index joy\_index.*
- void [jhi\\_free\\_joystick\\_index](#) (short int joy\_index)  
*Free the joystick structure.*
- int [jhi\\_get\\_num\\_of\\_joystick](#) ()  
*Get number of joysticks.*
- int [jhi\\_is\\_valid\\_joystick\\_index](#) (int index)  
*Check if index is valid.*
- [JHI\\_JOY\\_DIR](#) [jhi\\_get\\_joystick\\_dir](#) ([JHI\\_JoystickSt](#) \*joy, int axe\_index)  
*Get dir of joystick.*

### 4.4.1 Detailed Description

This file contains structures and enumeration of keyboard.

## 4.4.2 Enumeration Type Documentation

### 4.4.2.1 enum JHI\_JOYSTICK\_EVENT

Enumeration of events of joystick.

Enumerator

**JHI\_JOY\_AXIS\_MOTION** Joystick axis motion  
**JHI\_JOY\_BUTTON\_DOWN** Joystick button pressed  
**JHI\_JOY\_BUTTON\_UP** Joystick button released

## 4.4.3 Function Documentation

### 4.4.3.1 void jhi\_free\_joystick\_index ( short int *joy\_index* )

Free the joystick structure.

Parameters

<i>joy</i>	joystick structure
------------	--------------------

Returns

1 success, 0 otherwise

### 4.4.3.2 JHI\_JOY\_DIR\_jhi\_get\_joystick\_dir ( JHI\_JoystickSt \* *joy*, int *axe\_index* )

Get dir of joystick.

Parameters

<i>joy</i>	joystick structure
<i>axe_index</i>	index of axe

Returns

1 ok, 0 otherwise

### 4.4.3.3 int jhi\_get\_num\_of\_joystick ( )

Get number of joysticks.

Returns

number of joysticks

### 4.4.3.4 void jhi\_init\_joystick ( JHI\_JoystickSt \* *joy* )

Initialize the joystick structure.

## Parameters

<i>joy</i>	joystick structure
------------	--------------------

4.4.3.5 int jhi\_is\_valid\_joystick\_index ( int *index* )

Check if index is valid.

## Returns

1 ok, 0 otherwise

4.4.3.6 int jhi\_open\_joystick\_index ( short int *joy\_index* )

Open Joystick with index *joy\_index*.

## Parameters

<i>joy_index</i>	
------------------	--

## Returns

1 success, 0 otherwise

## 4.5 jhi\_keyboard.h File Reference

This file contains structures and enumeration of keyboard.

```
#include "SDL/SDL.h"
```

### Classes

- struct [JHI\\_KeyboardSt](#)  
*Keyboard structure.*

### Enumerations

- enum [JHI\\_Keys](#) {  
[KEY\\_ENTER](#) = 13, [KEY\\_0](#) = 48, [KEY\\_1](#) = 49, [KEY\\_2](#) = 50,  
[KEY\\_3](#) = 51, [KEY\\_4](#) = 52, [KEY\\_5](#) = 53, [KEY\\_6](#) = 54,  
[KEY\\_7](#) = 55, [KEY\\_8](#) = 56, [KEY\\_9](#) = 57, [KEY\\_UP](#) = 273,  
[KEY\\_DOWN](#) = 274, [KEY\\_RIGHT](#) = 275, [KEY\\_LEFT](#) = 276, [KEY\\_A](#) = 97,  
[KEY\\_B](#) = 98, [KEY\\_C](#) = 99, [KEY\\_D](#) = 100, [KEY\\_E](#) = 101,  
[KEY\\_F](#) = 102, [KEY\\_G](#) = 103, [KEY\\_H](#) = 104, [KEY\\_I](#) = 105,  
[KEY\\_J](#) = 106, [KEY\\_K](#) = 107, [KEY\\_L](#) = 108, [KEY\\_M](#) = 109,  
[KEY\\_N](#) = 110, [KEY\\_O](#) = 111, [KEY\\_P](#) = 112, [KEY\\_Q](#) = 113,  
[KEY\\_R](#) = 114, [KEY\\_S](#) = 115, [KEY\\_T](#) = 116, [KEY\\_U](#) = 117,  
[KEY\\_V](#) = 118, [KEY\\_W](#) = 119, [KEY\\_X](#) = 120, [KEY\\_Y](#) = 121,  
[KEY\\_Z](#) = 122, [NO\\_KEY](#) = 123 }  
*Enumeration of the possibles keys that will pressed in the keyboard.*
- enum [JHI\\_KeyboardEvents](#) { [KEYBOARD\\_UP](#), [KEYBOARD\\_DOWN](#), [KEYBOARD\\_NOT\\_EVENT](#) }  
*Enumeration of events of keyboard.*

## Functions

- void [jhi\\_init\\_keyboard](#) ([JHI\\_KeyboardSt](#) \*key)  
*init the keyboard structure*
- int [jhi\\_is\\_key\\_arrow](#) ([JHI\\_Keys](#) key)  
*Check if the key is arrow key.*
- [JHI\\_Keys jhi\\_get\\_opposite\\_key\\_arrow](#) ([JHI\\_Keys](#) key)  
*Get the opposite dir key.*

### 4.5.1 Detailed Description

This file contains structures and enumeration of keyboard.

### 4.5.2 Enumeration Type Documentation

#### 4.5.2.1 enum [JHI\\_KeyBoardEvents](#)

Enumeration of events of keyboard.

##### Enumerator

**KEYBOARD\_UP** Keyboard was pressed  
**KEYBOARD\_DOWN** Keyboard was released  
**KEYBOARD\_NOT\_EVENT** Nothing

#### 4.5.2.2 enum [JHI\\_Keys](#)

Enumeration of the possibles keys that will pressed in the keyboard.

##### Enumerator

**KEY\_ENTER** Key Enter  
**KEY\_0** Key 0  
**KEY\_1** Key 1  
**KEY\_2** Key 2  
**KEY\_3** Key 3  
**KEY\_4** Key 4  
**KEY\_5** Key 5  
**KEY\_6** Key 6  
**KEY\_7** Key 7  
**KEY\_8** Key 8  
**KEY\_9** Key 9  
**KEY\_UP** Key UP  
**KEY\_DOWN** Key DOWN  
**KEY\_RIGHT** Key RIGHT  
**KEY\_LEFT** Key LEFT  
**KEY\_A** Key A  
**KEY\_B** Key B  
**KEY\_C** Key C

**KEY\_D** Key D  
**KEY\_E** Key E  
**KEY\_F** Key F  
**KEY\_G** Key G  
**KEY\_H** Key H  
**KEY\_I** Key I  
**KEY\_J** Key J  
**KEY\_K** Key K  
**KEY\_L** Key L  
**KEY\_M** Key M  
**KEY\_N** Key N  
**KEY\_O** Key O  
**KEY\_P** Key P  
**KEY\_Q** Key Q  
**KEY\_R** Key R  
**KEY\_S** Key S  
**KEY\_T** Key T  
**KEY\_U** Key U  
**KEY\_V** Key V  
**KEY\_W** Key W  
**KEY\_X** Key X  
**KEY\_Y** Key Y  
**KEY\_Z** Key Z  
**NO\_KEY** No Key

### 4.5.3 Function Documentation

#### 4.5.3.1 JHI\_Keys jhi\_get\_opposite\_key\_arrow ( JHI\_Keys key )

Get the opposite dir key.

Parameters

<i>key</i>	key dir
------------	---------

Returns

opposite dir key, NO\_KEY is return in error case

#### 4.5.3.2 void jhi\_init\_keyboard ( JHI\_KeyboardSt \* key )

init the keyboard structure

Parameters

<i>key</i>	keyboard that will init
------------	-------------------------

#### 4.5.3.3 int jhi\_is\_key\_arrow ( JHI\_Keys key )

Check if the key is arrow key.

## Parameters

<i>key</i>	key to check
------------	--------------

## Returns

1 yes, 0 no

## 4.6 jhi\_mouse.h File Reference

This file contains structure of the mouse.

```
#include "SDL/SDL.h"
```

## Classes

- struct [JHI\\_MouseSt](#)  
*Mouse structure.*

## Enumerations

- enum [JHI\\_MouseEvents](#) { [ON\\_CLICK\\_RIGHT](#), [ON\\_CLICK\\_LEFT](#), [MOUSE\\_MOTION](#), [MOUSE\\_NOT\\_EVENT](#) }  
*Enumeration of the events of the mouse.*

## Functions

- void [jhi\\_init\\_mouse](#) ([JHI\\_MouseSt](#) \*mouse)  
*Inititalize the mouse structure.*
- int [jhi\\_check\\_mouse\\_is\\_in](#) ([JHI\\_MouseSt](#) mouse, int xp, int yp, int w, int h)  
*Check if the mouse is in the space object.*

### 4.6.1 Detailed Description

This file contains structure of the mouse.

### 4.6.2 Enumeration Type Documentation

#### 4.6.2.1 enum JHI\_MouseEvents

Enumeration of the events of the mouse.

## Enumerator

***ON\_CLICK\_RIGHT*** Click of right button  
***ON\_CLICK\_LEFT*** Click of left button  
***MOUSE\_MOTION*** Mouse was moved  
***MOUSE\_NOT\_EVENT*** Nothing Event

### 4.6.3 Function Documentation

4.6.3.1 `int jhi_check_mouse_is_in ( JHI_MouseSt mouse, int xp, int yp, int w, int h )`

Check if the mouse is in the space object.

**Parameters**

<i>xp</i>	x of object
<i>yp</i>	y of object
<i>w</i>	width of the object
<i>h</i>	height of the object

**Returns**

1 ok, 0 otherwise

**4.6.3.2 void jhi\_init\_mouse ( JHI\_MouseSt \* mouse )**

Initialize the mouse structure.

**Parameters**

<i>mouse</i>	the structure mouse to configure
--------------	----------------------------------

**4.7 jhi\_rand.h File Reference**

This file contains rand auxiliary MACROS.

```
#include <stdlib.h>
```

**Macros**

- `#define RAND_INTERVAL(a, b) (rand() % ((b)-(a+1))) + (a)`  
*get a rand value between a,b*
- `#define RAND_01 ((double)(rand())/((double)(RAND_MAX)))`  
*get a double rand value between 0,1*

**4.7.1 Detailed Description**

This file contains rand auxiliary MACROS.

**4.8 jhi\_shapes.h File Reference**

This file contains shapes functions of libjhi-sdl.

```
#include "jhi_window.h"
#include "jhi_colorsSt.h"
#include <math.h>
```

**Classes**

- struct `JHI_Point2d`  
*Point that contain position (x,y)*



## Functions

- void [jhi\\_draw\\_point](#) ([JHI\\_Point2d](#) point, [JHI\\_Color](#) col)  
*draw point in the window*
- void [jhi\\_draw\\_line](#) ([JHI\\_Point2d](#) s\_point, [JHI\\_Point2d](#) d\_point, [JHI\\_Color](#) col)  
*Drawing a line in the screen.*
- void [jhi\\_draw\\_rect](#) ([JHI\\_Point2d](#) point, int height, int base, [JHI\\_Color](#) col)  
*Draw a rectangle or square in the screen.*
- void [jhi\\_draw\\_fill\\_rect](#) ([JHI\\_Point2d](#) point, int height, int base, [JHI\\_Color](#) col)  
*Draw a fill rectangle or square in the screen.*
- void [jhi\\_draw\\_circle](#) ([JHI\\_Point2d](#) center\_point, float radius, [JHI\\_Color](#) col)  
*Drawing a circle in the screen.*
- void [jhi\\_draw\\_fill\\_circle](#) ([JHI\\_Point2d](#) center\_point, int radius, [JHI\\_Color](#) col)  
*Draw a fill circle in the screen.*
- void [jhi\\_draw\\_polygon](#) ([JHI\\_Point2d](#) \*points, int num\_points, [JHI\\_Color](#) col)  
*Draw a Polygon in the screen.*
- int [jhi\\_is\\_colid](#) ([JHI\\_Point2d](#) p1, int h1, int w1, [JHI\\_Point2d](#) p2, int h2, int w2)  
*Checks whether the objects collide.*
- [JHI\\_Point2d](#) [jhi\\_get\\_central\\_pos](#) (int win\_w, int win\_h, int obj\_w, int obj\_h)  
*Get the central position of object in the window.*
- [JHI\\_Point2d](#) [jhi\\_get\\_point](#) (int x, int y)  
*Get the struct JHI\_Point2 by (x,y)*

### 4.8.1 Detailed Description

This file contains shapes functions of libjhi-sdl.

### 4.8.2 Function Documentation

#### 4.8.2.1 void [jhi\\_draw\\_circle](#) ( [JHI\\_Point2d](#) center\_point, float radius, [JHI\\_Color](#) col )

Drawing a circle in the screen.

##### Parameters

<i>center_point</i>	Center point of the circle center
<i>radius</i>	Circle's radius
<i>col</i>	Circle's color

#### 4.8.2.2 void [jhi\\_draw\\_fill\\_circle](#) ( [JHI\\_Point2d](#) center\_point, int radius, [JHI\\_Color](#) col )

Draw a fill circle in the screen.

##### Parameters

<i>center_point</i>	Center point of the circle center
<i>radius</i>	Circle's radius
<i>col</i>	Circle's color

#### 4.8.2.3 void [jhi\\_draw\\_fill\\_rect](#) ( [JHI\\_Point2d](#) point, int height, int base, [JHI\\_Color](#) col )

Draw a fill rectangle or square in the screen.

## Parameters

<i>point</i>	Upper left point of the rect
<i>height</i>	Rect's height
<i>base</i>	Rect's base
<i>col</i>	Rect's Color

4.8.2.4 void jhi\_draw\_line ( JHI\_Point2d *s\_point*, JHI\_Point2d *d\_point*, JHI\_Color *col* )

Drawing a line in the screen.

## Parameters

<i>s_point</i>	Source point of the line
<i>d_point</i>	Destination point of the line
<i>col</i>	Line Color

4.8.2.5 void jhi\_draw\_point ( JHI\_Point2d *point*, JHI\_Color *col* )

draw point in the window

## Parameters

<i>point</i>	point position
<i>col</i>	point color

4.8.2.6 void jhi\_draw\_polygon ( JHI\_Point2d \* *points*, int *num\_points*, JHI\_Color *col* )

Draw a Polygon in the screen.

## Parameters

<i>points</i>	Set of the Polygon's points
<i>num_points</i>	Number of points of the polygon color Color of the Polygon's lines

4.8.2.7 void jhi\_draw\_rect ( JHI\_Point2d *point*, int *height*, int *base*, JHI\_Color *col* )

Draw a rectangle or square in the screen.

## Parameters

<i>point</i>	Upper left point of the rect
<i>height</i>	Rect's height
<i>base</i>	Rect's base
<i>col</i>	Rect's Color

4.8.2.8 JHI\_Point2d jhi\_get\_central\_pos ( int *win\_w*, int *win\_h*, int *obj\_w*, int *obj\_h* )

Get the central position of object in the window.

## Parameters

<i>win_w</i>	Width of the window
<i>win_h</i>	Height of the windows
<i>obj_w</i>	Object's width
<i>obj_h</i>	Object's height

**Returns**

Central position for this object

**4.8.2.9 JHI\_Point2d jhi\_get\_point ( int x, int y )**

Get the struct JHI\_Point2 by (x,y)

**Parameters**

<i>x</i>	x pos
<i>y</i>	y pos
<i>point</i>	

**4.8.2.10 int jhi\_is\_colid ( JHI\_Point2d p1, int h1, int w1, JHI\_Point2d p2, int h2, int w2 )**

Checks whether the objects collide.

**Parameters**

<i>p1</i>	Point of object 1
<i>h1</i>	Height of object 1
<i>w1</i>	Weight of object 1
<i>p2</i>	Point of object 2
<i>h2</i>	Height of object 2
<i>w2</i>	Weight of object 2

**Returns**

1 colid, 0 otherwise

## 4.9 jhi\_sound.h File Reference

This file contains functions and structures of the Music and Effect.

```
#include "SDL/SDL.h"
#include "SDL/SDL_mixer.h"
#include <stdio.h>
```

**Classes**

- struct [JHI\\_Music](#)  
*Struct of configuration of the Music.*
- struct [JHI\\_Effect](#)  
*Struct of the configuration of the Effect.*

## Functions

- void `jhi_load_music` (`JHI_Music` \*music, const char \*filename)  
*Load the music with the filename.*
- void `jhi_play_music` (`JHI_Music` \*music, int loop)  
*Play the music in the background.*
- void `jhi_stop_music` ()  
*Stop the current music.*
- void `jhi_pause_music` ()  
*Pause the current music.*
- void `jhi_free_music` (`JHI_Music` \*music)  
*Free music structure.*
- void `jhi_load_effect` (`JHI_Effect` \*effect, const char \*filename)  
*Load the effect with the filename.*
- void `jhi_play_effect` (`JHI_Effect` \*effect, int delay)  
*Play the effect.*
- void `jhi_free_effect` (`JHI_Effect` \*effect)  
*Free effect structure.*

### 4.9.1 Detailed Description

This file contains functions and structures of the Music and Effect.

### 4.9.2 Function Documentation

#### 4.9.2.1 void `jhi_load_effect` ( `JHI_Effect` \* *effect*, const char \* *filename* )

Load the effect with the filename.

##### Parameters

<i>effect</i>	Structure of effect to load
<i>filename</i>	Name of the effect file

#### 4.9.2.2 void `jhi_load_music` ( `JHI_Music` \* *music*, const char \* *filename* )

Load the music with the filename.

##### Parameters

<i>music</i>	Structure of music to load
<i>filename</i>	Name of music file

#### 4.9.2.3 void `jhi_play_effect` ( `JHI_Effect` \* *effect*, int *delay* )

Play the effect.

##### Parameters

<i>effect</i>	Structure of effect to play
---------------	-----------------------------

<i>delay</i>	Delay to play the effect in Seconds
--------------	-------------------------------------

4.9.2.4 void jhi\_play\_music ( JHI\_Music \* music, int loop )

Play the music in the background.

Parameters

<i>music</i>	Structure of music to load
<i>loop</i>	Number of times that music will played. -1, it's infinite

## 4.10 jhi\_text.h File Reference

This file contains functions and structures of the Text.

```
#include "jhi_font.h"
#include "jhi_colorsSt.h"
#include "jhi_shapes.h"
#include "SDL/SDL.h"
```

### Classes

- struct [JHI\\_Text](#)  
*Structure of configuration of the Text.*

### Functions

- void [jhi\\_init\\_text](#) ([JHI\\_Text](#) \*text)  
*Init the Text structure.*
- void [jhi\\_set\\_text](#) ([JHI\\_Font](#) \*font, [JHI\\_Text](#) \*text, [JHI\\_Color](#) cor, const char \*txt)  
*Set the Text with the configurations.*
- void [jhi\\_free\\_text](#) ([JHI\\_Text](#) \*text)  
*Free memory of the Text strucure.*
- int [jhi\\_get\\_text\\_width](#) ([JHI\\_Text](#) \*text)  
*Get the Text width.*
- int [jhi\\_get\\_text\\_height](#) ([JHI\\_Text](#) \*text)  
*Get the Text height.*
- void [jhi\\_draw\\_text](#) ([JHI\\_Text](#) \*text, [JHI\\_Point2d](#) point)  
*Draw the text in the window.*

#### 4.10.1 Detailed Description

This file contains functions and structures of the Text.

#### 4.10.2 Function Documentation

4.10.2.1 void jhi\_draw\_text ( JHI\_Text \* text, JHI\_Point2d point )

Draw the text in the window.

## Parameters

<i>text</i>	Text Structure
<i>point</i>	New text position

4.10.2.2 int jhi\_get\_text\_height ( JHI\_Text \* *text* )

Get the Text height.

## Parameters

<i>text</i>	Text Structure
-------------	----------------

## Returns

Text height

4.10.2.3 int jhi\_get\_text\_width ( JHI\_Text \* *text* )

Get the Text width.

## Parameters

<i>text</i>	Text Structure
-------------	----------------

## Returns

Text width

4.10.2.4 void jhi\_init\_text ( JHI\_Text \* *text* )

Init the Text structure.

## Parameters

<i>text</i>	Text structure to be initialized
-------------	----------------------------------

4.10.2.5 void jhi\_set\_text ( JHI\_Font \* *font*, JHI\_Text \* *text*, JHI\_Color *cor*, const char \* *txt* )

Set the Text with the configurations.

## Parameters

<i>font</i>	Font of the Text
<i>text</i>	Text to be configured
<i>color</i>	Color of the Text
<i>txt</i>	String to the Text

## 4.11 jhi\_timer.h File Reference

This file contains functions structure of the Timer to control number of frames per second (fps).

```
#include "SDL/SDL.h"
```

## Functions

- void [jhi\\_set\\_fps\\_timer](#) (int fps)  
*Configure the fps timer.*
- void [jhi\\_timer\\_start](#) ()  
*Start the count to fps.*
- void [jhi\\_wait\\_time](#) ()  
*Wait the time necessary to control fps.*
- void [jhi\\_delay](#) (int seconds)  
*Wait the time in second.*
- void [jhi\\_delay\\_mili\\_seconds](#) (int ms)

### 4.11.1 Detailed Description

This file contains functions structure of the Timer to control number of frames per second (fps).

### 4.11.2 Function Documentation

#### 4.11.2.1 void [jhi\\_delay](#) ( int *seconds* )

Wait the time in second.

##### Parameters

<i>second</i>	Number of Seconds to wait
---------------	---------------------------

#### 4.11.2.2 void [jhi\\_set\\_fps\\_timer](#) ( int *fps* )

Configure the fps timer.

##### Parameters

<i>fps</i>	Frames per second to animation
------------	--------------------------------

## 4.12 jhi\_window.h File Reference

This file contains functions related to Window.

```
#include "SDL/SDL_mixer.h"
#include "jhi_colorsSt.h"
#include "jhi_keyboard.h"
#include "jhi_mouse.h"
#include "jhi_joystick.h"
#include <stdio.h>
```

## Classes

- struct [JHI\\_Window](#)  
*Struct of configuration of the Window.*

## Macros

- `#define JHI_CLOSE 1`
- `#define JHI_NOT_CLOSE 0`
- `#define NUMBER_MAX_EVENTS 100`

## Functions

- `int jhi_out_window (int x, int y)`  
*Check if (x,y) are in valid position inside of the Window.*
- `void jhi_initialize_window (int width, int height, int bitperpixel, JHI_Color back_color)`  
*Initialize configurations of the Window.*
- `void jhi_print_pixel (int x, int y, JHI_Color col)`  
*Draw a pixel in the screen.*
- `void jhi_draw_object (SDL_Surface *object, int x, int y)`  
*Draw object in the window.*
- `void jhi_draw_object_with_clip (SDL_Surface *object, int x, int y, int x_clip, int y_clip, int width, int height)`  
*Draw a cut/clip of an image in the screen.*
- `void jhi_choice_window_name (const char *win_name)`  
*Set the name of the Window.*
- `void jhi_init_mouse_keyboard_joystick_events ()`  
*Initialize structs of the mouse, keyboard and joystick.*
- `int jhi_get_number_of_events ()`  
*Return the number of events captured in the iteration.*
- `void jhi_update ()`  
*Capture the events of iteration and updates the Window.*
- `void jhi_set_background_color (JHI_Color back_color)`  
*Set the background color with the color especified.*
- `void jhi_clean ()`  
*Clean the Window with the background color.*
- `char jhi_get_close_window ()`  
*Get if close the Window.*
- `JHI_MouseSt jhi_get_mouse_status (int i)`  
*Get status of the mouse input.*
- `JHI_KeyboardSt jhi_get_keyboard_status (int i)`  
*Get status of the keyboard input.*
- `JHI_JoystickSt jhi_get_joystick_status (int i)`  
*Get status of the joystick input.*
- `void jhi_quit_and_free ()`  
*Free structure of the Window.*
- `int jhi_get_height_window ()`  
*Get the height of the window.*
- `int jhi_get_width_window ()`  
*Get the height of the window.*

### 4.12.1 Detailed Description

This file contains functions related to Window.



## 4.12.2 Function Documentation

### 4.12.2.1 void jhi\_choice\_window\_name ( const char \* *win\_name* )

Set the name of the Window.

## Parameters

<i>win_name</i>	Name of the Window
-----------------	--------------------

4.12.2.2 void jhi\_draw\_object ( SDL\_Surface \* *object*, int *x*, int *y* )

Draw object in the window.

## Parameters

<i>object</i>	Object to draw
---------------	----------------

4.12.2.3 void jhi\_draw\_object\_with\_clip ( SDL\_Surface \* *object*, int *x*, int *y*, int *x\_clip*, int *y\_clip*, int *width*, int *height* )

Draw a cut/clip of an image in the screen.

## Parameters

<i>object</i>	Object to draw
<i>x</i>	Coordinate x to clip object in the screen
<i>y</i>	Coordinate y to to clip object in the screen
<i>x_clip</i>	x position initial of clip
<i>y_clip</i>	y position initial of clip
<i>width</i>	Width of this clip
<i>height</i>	Height of this clip

## 4.12.2.4 char jhi\_get\_close\_window ( )

Get if close the Window.

## Returns

NOT\_CLOSE, case this Window wasn't closed, CLOSE otherwise

## 4.12.2.5 int jhi\_get\_height\_window ( )

Get the height of the window.

## Returns

height of the window

4.12.2.6 JHI\_JoystickSt jhi\_get\_joystick\_status ( int *i* )

Get status of the joystick input.

## Parameters

<i>i</i>	Number of event
----------	-----------------

## Returns

keyboard status

#### 4.12.2.7 JHI\_KeyboardSt jhi\_get\_keyboard\_status ( int *i* )

Get status of the keyboard input.

## Parameters

<i>i</i>	Number of event
----------	-----------------

## Returns

keyboard status

4.12.2.8 JHI\_MouseSt jhi\_get\_mouse\_status ( int *i* )

Get status of the mouse input.

## Parameters

<i>i</i>	Number of event
----------	-----------------

## Returns

Mouse status

## 4.12.2.9 int jhi\_get\_number\_of\_events ( )

Return the number of events captured in the iteration.

## Returns

Number of events captured in the iteration

## 4.12.2.10 int jhi\_get\_width\_window ( )

Get the height of the window.

## Returns

height of the window

4.12.2.11 void jhi\_initialize\_window ( int *width*, int *height*, int *bitperpixel*, JHI\_Color *back\_color* )

Initialize configurations of the Window.

## Parameters

<i>width</i>	Width of the Window
<i>height</i>	Height of the Window
<i>bitperpixel</i>	Number of the bits by pixel
<i>back_color</i>	Background color of the Window

4.12.2.12 int jhi\_out\_window ( int *x*, int *y* )

Check if (x,y) are in valid position inside of the Window.

## Parameters

<i>x</i>	Coordinate x
<i>y</i>	Coordinate y

## Returns

1 if is out, 0 otherwise

4.12.2.13 void jhi\_print\_pixel ( int *x*, int *y*, **JHI\_Color** *col* )

Draw a pixel in the screen.

## Parameters

<i>x</i>	Coordinate x of the pixel
<i>y</i>	Coordinate y of the pixel
<i>color</i>	Color of the pixel

4.12.2.14 void jhi\_set\_background\_color ( **JHI\_Color** *back\_color* )

Set the background color with the color specified.

## Parameters

<i>back_color</i>	Color to fill the background
-------------------	------------------------------

# Index

- axis\_index
  - JHI\_JoystickSt, 8
- axis\_value
  - JHI\_JoystickSt, 8
- b
  - JHI\_Color\_RGB, 5
- back\_color
  - JHI\_Window, 11
- button
  - JHI\_JoystickSt, 8
- button\_state
  - JHI\_JoystickSt, 8
- check\_quit
  - JHI\_Window, 11
- event
  - JHI\_Window, 12
- font
  - JHI\_Font, 6
- g
  - JHI\_Color\_RGB, 5
- height
  - JHI\_Window, 12
- JHI\_JOY\_AXIS\_MOTION
  - jhi\_joystick.h, 18
- JHI\_JOY\_BUTTON\_NUP
  - jhi\_joystick.h, 18
- JHI\_JOY\_BUTTON\_DOWN
  - jhi\_joystick.h, 18
- JHI\_Color\_RGB, 5
  - b, 5
  - g, 5
  - r, 5
- JHI\_Effect, 5
  - mix\_chunk, 6
- JHI\_Font, 6
  - font, 6
  - length, 6
- JHI\_Image, 7
  - pos, 7
  - sur, 7
- JHI\_JOYSTICK\_EVENT
  - jhi\_joystick.h, 18
- JHI\_JoystickSt, 7
  - axis\_index, 8
  - axis\_value, 8
  - button, 8
  - button\_state, 8
  - joy\_event, 8
  - joy\_index, 8
- JHI\_KeyBoardEvents
  - jhi\_keyboard.h, 20
- JHI\_KeyboardSt, 8
  - key, 8
  - key\_event, 8
- JHI\_Keys
  - jhi\_keyboard.h, 20
- JHI\_MouseEvents
  - jhi\_mouse.h, 22
- JHI\_MouseSt, 9
  - mouse\_event, 9
  - x, 9
  - y, 9
- JHI\_Music, 9
  - mix\_music, 10
- JHI\_Point2d, 10
  - x, 10
  - y, 10
- JHI\_Text, 10
  - pos, 11
  - sur, 11
- JHI\_Window, 11
  - back\_color, 11
  - check\_quit, 11
  - event, 12
  - height, 12
  - joy, 12
  - key, 12
  - mouse, 12
  - number\_of\_events, 12
  - screen, 12
  - width, 12
- jhi\_joystick.h
  - JHI\_JOY\_AXIS\_MOTION, 18
  - JHI\_JOY\_BUTTON\_NUP, 18
  - JHI\_JOY\_BUTTON\_DOWN, 18
- jhi\_keyboard.h
  - KEY\_0, 20
  - KEY\_1, 20
  - KEY\_2, 20
  - KEY\_3, 20
  - KEY\_4, 20
  - KEY\_5, 20
  - KEY\_6, 20

- KEY\_7, [20](#)
- KEY\_8, [20](#)
- KEY\_9, [20](#)
- KEY\_A, [20](#)
- KEY\_B, [20](#)
- KEY\_C, [20](#)
- KEY\_D, [20](#)
- KEY\_DOWN, [20](#)
- KEY\_E, [21](#)
- KEY\_ENTER, [20](#)
- KEY\_F, [21](#)
- KEY\_G, [21](#)
- KEY\_H, [21](#)
- KEY\_I, [21](#)
- KEY\_J, [21](#)
- KEY\_K, [21](#)
- KEY\_L, [21](#)
- KEY\_LEFT, [20](#)
- KEY\_M, [21](#)
- KEY\_N, [21](#)
- KEY\_O, [21](#)
- KEY\_P, [21](#)
- KEY\_Q, [21](#)
- KEY\_R, [21](#)
- KEY\_RIGHT, [20](#)
- KEY\_S, [21](#)
- KEY\_T, [21](#)
- KEY\_U, [21](#)
- KEY\_UP, [20](#)
- KEY\_V, [21](#)
- KEY\_W, [21](#)
- KEY\_X, [21](#)
- KEY\_Y, [21](#)
- KEY\_Z, [21](#)
- KEYBOARD\_DOWN, [20](#)
- KEYBOARD\_NOT\_EVENT, [20](#)
- KEYBOARD\_UP, [20](#)
- NO\_KEY, [21](#)
- jhi\_mouse.h
  - MOUSE\_MOTION, [22](#)
  - MOUSE\_NOT\_EVENT, [22](#)
  - ON\_CLICK\_LEFT, [22](#)
  - ON\_CLICK\_RIGHT, [22](#)
- jhi\_check\_mouse\_is\_in
  - jhi\_mouse.h, [23](#)
- jhi\_choice\_window\_name
  - jhi\_window.h, [33](#)
- jhi\_colorsSt.h, [13](#)
- jhi\_delay
  - jhi\_timer.h, [31](#)
- jhi\_draw\_circle
  - jhi\_shapes.h, [25](#)
- jhi\_draw\_fill\_circle
  - jhi\_shapes.h, [25](#)
- jhi\_draw\_fill\_rect
  - jhi\_shapes.h, [25](#)
- jhi\_draw\_image
  - jhi\_image.h, [15](#)
- jhi\_draw\_image\_with\_clip
  - jhi\_image.h, [15](#)
- jhi\_draw\_line
  - jhi\_shapes.h, [26](#)
- jhi\_draw\_object
  - jhi\_window.h, [34](#)
- jhi\_draw\_object\_with\_clip
  - jhi\_window.h, [34](#)
- jhi\_draw\_point
  - jhi\_shapes.h, [26](#)
- jhi\_draw\_polygon
  - jhi\_shapes.h, [26](#)
- jhi\_draw\_rect
  - jhi\_shapes.h, [26](#)
- jhi\_draw\_text
  - jhi\_text.h, [29](#)
- jhi\_font.h, [13](#)
  - jhi\_get\_lenght\_font, [14](#)
  - jhi\_load\_font, [14](#)
- jhi\_free\_joystick\_index
  - jhi\_joystick.h, [18](#)
- jhi\_get\_central\_pos
  - jhi\_shapes.h, [26](#)
- jhi\_get\_close\_window
  - jhi\_window.h, [34](#)
- jhi\_get\_height\_window
  - jhi\_window.h, [34](#)
- jhi\_get\_image\_height
  - jhi\_image.h, [15](#)
- jhi\_get\_image\_width
  - jhi\_image.h, [16](#)
- jhi\_get\_joystick\_dir
  - jhi\_joystick.h, [18](#)
- jhi\_get\_joystick\_status
  - jhi\_window.h, [34](#)
- jhi\_get\_keyboard\_status
  - jhi\_window.h, [34](#)
- jhi\_get\_lenght\_font
  - jhi\_font.h, [14](#)
- jhi\_get\_mouse\_status
  - jhi\_window.h, [36](#)
- jhi\_get\_num\_of\_joystick
  - jhi\_joystick.h, [18](#)
- jhi\_get\_number\_of\_events
  - jhi\_window.h, [36](#)
- jhi\_get\_opposite\_key\_arrow
  - jhi\_keyboard.h, [21](#)
- jhi\_get\_point
  - jhi\_shapes.h, [27](#)
- jhi\_get\_text\_height
  - jhi\_text.h, [30](#)
- jhi\_get\_text\_width
  - jhi\_text.h, [30](#)
- jhi\_get\_width\_window
  - jhi\_window.h, [36](#)
- jhi\_image.h, [14](#)
  - jhi\_draw\_image, [15](#)
  - jhi\_draw\_image\_with\_clip, [15](#)

- jhi\_get\_image\_height, 15
- jhi\_get\_image\_width, 16
- jhi\_load\_image, 16
- jhi\_load\_image\_with\_transparent\_color, 16
- jhi\_resize\_image, 16
- jhi\_init\_joystick
  - jhi\_joystick.h, 18
- jhi\_init\_keyboard
  - jhi\_keyboard.h, 21
- jhi\_init\_mouse
  - jhi\_mouse.h, 24
- jhi\_init\_text
  - jhi\_text.h, 30
- jhi\_initialize\_window
  - jhi\_window.h, 36
- jhi\_is\_colid
  - jhi\_shapes.h, 27
- jhi\_is\_key\_arrow
  - jhi\_keyboard.h, 21
- jhi\_is\_valid\_joystick\_index
  - jhi\_joystick.h, 19
- jhi\_joystick.h, 16
  - jhi\_free\_joystick\_index, 18
  - jhi\_get\_joystick\_dir, 18
  - jhi\_get\_num\_of\_joystick, 18
  - jhi\_init\_joystick, 18
  - jhi\_is\_valid\_joystick\_index, 19
  - jhi\_open\_joystick\_index, 19
- jhi\_keyboard.h, 19
  - JHI\_KeyBoardEvents, 20
  - JHI\_Keys, 20
  - jhi\_get\_opposite\_key\_arrow, 21
  - jhi\_init\_keyboard, 21
  - jhi\_is\_key\_arrow, 21
- jhi\_load\_effect
  - jhi\_sound.h, 28
- jhi\_load\_font
  - jhi\_font.h, 14
- jhi\_load\_image
  - jhi\_image.h, 16
- jhi\_load\_image\_with\_transparent\_color
  - jhi\_image.h, 16
- jhi\_load\_music
  - jhi\_sound.h, 28
- jhi\_mouse.h, 22
  - JHI\_MouseEvents, 22
  - jhi\_check\_mouse\_is\_in, 23
  - jhi\_init\_mouse, 24
- jhi\_open\_joystick\_index
  - jhi\_joystick.h, 19
- jhi\_out\_window
  - jhi\_window.h, 36
- jhi\_play\_effect
  - jhi\_sound.h, 28
- jhi\_play\_music
  - jhi\_sound.h, 29
- jhi\_print\_pixel
  - jhi\_window.h, 37
- jhi\_rand.h, 24
- jhi\_resize\_image
  - jhi\_image.h, 16
- jhi\_set\_background\_color
  - jhi\_window.h, 37
- jhi\_set\_fps\_timer
  - jhi\_timer.h, 31
- jhi\_set\_text
  - jhi\_text.h, 30
- jhi\_shapes.h, 24
  - jhi\_draw\_circle, 25
  - jhi\_draw\_fill\_circle, 25
  - jhi\_draw\_fill\_rect, 25
  - jhi\_draw\_line, 26
  - jhi\_draw\_point, 26
  - jhi\_draw\_polygon, 26
  - jhi\_draw\_rect, 26
  - jhi\_get\_central\_pos, 26
  - jhi\_get\_point, 27
  - jhi\_is\_colid, 27
- jhi\_sound.h, 27
  - jhi\_load\_effect, 28
  - jhi\_load\_music, 28
  - jhi\_play\_effect, 28
  - jhi\_play\_music, 29
- jhi\_text.h, 29
  - jhi\_draw\_text, 29
  - jhi\_get\_text\_height, 30
  - jhi\_get\_text\_width, 30
  - jhi\_init\_text, 30
  - jhi\_set\_text, 30
- jhi\_timer.h, 30
  - jhi\_delay, 31
  - jhi\_set\_fps\_timer, 31
- jhi\_window.h, 31
  - jhi\_choice\_window\_name, 33
  - jhi\_draw\_object, 34
  - jhi\_draw\_object\_with\_clip, 34
  - jhi\_get\_close\_window, 34
  - jhi\_get\_height\_window, 34
  - jhi\_get\_joystick\_status, 34
  - jhi\_get\_keyboard\_status, 34
  - jhi\_get\_mouse\_status, 36
  - jhi\_get\_number\_of\_events, 36
  - jhi\_get\_width\_window, 36
  - jhi\_initialize\_window, 36
  - jhi\_out\_window, 36
  - jhi\_print\_pixel, 37
  - jhi\_set\_background\_color, 37
- joy
  - JHI\_Window, 12
- joy\_event
  - JHI\_JoystickSt, 8
- joy\_index
  - JHI\_JoystickSt, 8
- KEY\_0
  - jhi\_keyboard.h, 20
- KEY\_1



jhi\_keyboard.h, [20](#)  
KEY\_2  
    jhi\_keyboard.h, [20](#)  
KEY\_3  
    jhi\_keyboard.h, [20](#)  
KEY\_4  
    jhi\_keyboard.h, [20](#)  
KEY\_5  
    jhi\_keyboard.h, [20](#)  
KEY\_6  
    jhi\_keyboard.h, [20](#)  
KEY\_7  
    jhi\_keyboard.h, [20](#)  
KEY\_8  
    jhi\_keyboard.h, [20](#)  
KEY\_9  
    jhi\_keyboard.h, [20](#)  
KEY\_A  
    jhi\_keyboard.h, [20](#)  
KEY\_B  
    jhi\_keyboard.h, [20](#)  
KEY\_C  
    jhi\_keyboard.h, [20](#)  
KEY\_D  
    jhi\_keyboard.h, [20](#)  
KEY\_DOWN  
    jhi\_keyboard.h, [20](#)  
KEY\_E  
    jhi\_keyboard.h, [21](#)  
KEY\_ENTER  
    jhi\_keyboard.h, [20](#)  
KEY\_F  
    jhi\_keyboard.h, [21](#)  
KEY\_G  
    jhi\_keyboard.h, [21](#)  
KEY\_H  
    jhi\_keyboard.h, [21](#)  
KEY\_I  
    jhi\_keyboard.h, [21](#)  
KEY\_J  
    jhi\_keyboard.h, [21](#)  
KEY\_K  
    jhi\_keyboard.h, [21](#)  
KEY\_L  
    jhi\_keyboard.h, [21](#)  
KEY\_LEFT  
    jhi\_keyboard.h, [20](#)  
KEY\_M  
    jhi\_keyboard.h, [21](#)  
KEY\_N  
    jhi\_keyboard.h, [21](#)  
KEY\_O  
    jhi\_keyboard.h, [21](#)  
KEY\_P  
    jhi\_keyboard.h, [21](#)  
KEY\_Q  
    jhi\_keyboard.h, [21](#)  
KEY\_R  
    jhi\_keyboard.h, [21](#)  
KEY\_RIGHT  
    jhi\_keyboard.h, [20](#)  
KEY\_S  
    jhi\_keyboard.h, [21](#)  
KEY\_T  
    jhi\_keyboard.h, [21](#)  
KEY\_U  
    jhi\_keyboard.h, [21](#)  
KEY\_UP  
    jhi\_keyboard.h, [20](#)  
KEY\_V  
    jhi\_keyboard.h, [21](#)  
KEY\_W  
    jhi\_keyboard.h, [21](#)  
KEY\_X  
    jhi\_keyboard.h, [21](#)  
KEY\_Y  
    jhi\_keyboard.h, [21](#)  
KEY\_Z  
    jhi\_keyboard.h, [21](#)  
KEYBOARD\_DOWN  
    jhi\_keyboard.h, [20](#)  
KEYBOARD\_NOT\_EVENT  
    jhi\_keyboard.h, [20](#)  
KEYBOARD\_UP  
    jhi\_keyboard.h, [20](#)  
key  
    JHI\_KeyboardSt, [8](#)  
    JHI\_Window, [12](#)  
key\_event  
    JHI\_KeyboardSt, [8](#)  
  
length  
    JHI\_Font, [6](#)  
  
MOUSE\_MOTION  
    jhi\_mouse.h, [22](#)  
MOUSE\_NOT\_EVENT  
    jhi\_mouse.h, [22](#)  
mix\_chunk  
    JHI\_Effect, [6](#)  
mix\_music  
    JHI\_Music, [10](#)  
mouse  
    JHI\_Window, [12](#)  
mouse\_event  
    JHI\_MouseSt, [9](#)  
  
NO\_KEY  
    jhi\_keyboard.h, [21](#)  
number\_of\_events  
    JHI\_Window, [12](#)  
  
ON\_CLICK\_LEFT  
    jhi\_mouse.h, [22](#)  
ON\_CLICK\_RIGHT  
    jhi\_mouse.h, [22](#)  
  
pos

JHI\_Image, [7](#)  
JHI\_Text, [11](#)

r

JHI\_Color\_RGB, [5](#)

screen

JHI\_Window, [12](#)

sur

JHI\_Image, [7](#)  
JHI\_Text, [11](#)

width

JHI\_Window, [12](#)

x

JHI\_MouseSt, [9](#)  
JHI\_Point2d, [10](#)

y

JHI\_MouseSt, [9](#)  
JHI\_Point2d, [10](#)