

LibJHI-SDL 2.0

Generated by Doxygen 1.8.6

Fri Nov 7 2014 11:30:58

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	3
2.1	File List	3
3	Class Documentation	5
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
4	File Documentation	13
4.1	jhi_font.h File Reference	13
4.1.1	Detailed Description	13
4.1.2	Function Documentation	13
4.1.2.1	jhi_get_lenght_font	13
4.1.2.2	jhi_load_font	14
4.2	jhi_image.h File Reference	15
4.2.1	Detailed Description	15
4.2.2	Function Documentation	15
4.2.2.1	jhi_draw_image	15
4.2.2.2	jhi_draw_image_with_clip	16
4.2.2.3	jhi_get_image_height	16
4.2.2.4	jhi_get_image_width	16
4.2.2.5	jhi_load_image	16
4.2.2.6	jhi_load_image_with_transparent_color	16
4.2.2.7	jhi_resize_image	17
4.3	jhi_joystick.h File Reference	17
4.3.1	Detailed Description	18
4.3.2	Enumeration Type Documentation	18
4.3.2.1	JHI_JOYSTICK_EVENT	18
4.3.3	Function Documentation	18
4.3.3.1	jhi_free_joystick_index	18
4.3.3.2	jhi_get_joystick_dir	18
4.3.3.3	jhi_get_num_of_joystick	19
4.3.3.4	jhi_init_joystick	19
4.3.3.5	jhi_is_valid_joystick_index	19
4.3.3.6	jhi_open_joystick_index	19
4.4	jhi_keyboard.h File Reference	19
4.4.1	Detailed Description	20
4.4.2	Enumeration Type Documentation	20
4.4.2.1	JHI_KeyBoardEvents	20
4.4.2.2	JHI_Keys	21
4.4.3	Function Documentation	22
4.4.3.1	jhi_get_opposite_key_arrow	22
4.4.3.2	jhi_init_keyboard	23
4.4.3.3	jhi_is_key_arrow	23
4.5	jhi_mouse.h File Reference	23

4.5.1	Detailed Description	23
4.5.2	Enumeration Type Documentation	24
4.5.2.1	JHI_MouseEvents	24
4.6	jhi_rand.h File Reference	24
4.6.1	Detailed Description	24
4.7	jhi_shapes.h File Reference	24
4.7.1	Detailed Description	25
4.7.2	Function Documentation	25
4.7.2.1	jhi_draw_circle	25
4.7.2.2	jhi_draw_fill_circle	25
4.7.2.3	jhi_draw_fill_rect	25
4.7.2.4	jhi_draw_line	26
4.7.2.5	jhi_draw_point	26
4.7.2.6	jhi_draw_polygon	26
4.7.2.7	jhi_draw_rect	26
4.7.2.8	jhi_get_central_pos	26
4.7.2.9	jhi_is_colid	27
4.8	jhi_sound.h File Reference	27
4.8.1	Detailed Description	28
4.8.2	Function Documentation	28
4.8.2.1	jhi_load_effect	28
4.8.2.2	jhi_load_music	28
4.8.2.3	jhi_play_effect	28
4.8.2.4	jhi_play_music	28
4.9	jhi_text.h File Reference	28
4.9.1	Detailed Description	29
4.9.2	Function Documentation	29
4.9.2.1	jhi_draw_text	29
4.9.2.2	jhi_get_text_height	29
4.9.2.3	jhi_get_text_width	30
4.9.2.4	jhi_init_text	31
4.9.2.5	jhi_set_text	31
4.10	jhi_window.h File Reference	31
4.10.1	Detailed Description	32
4.10.2	Function Documentation	32
4.10.2.1	jhi_choice_window_name	32
4.10.2.2	jhi_draw_object	33
4.10.2.3	jhi_draw_object_with_clip	33
4.10.2.4	jhi_get_close_window	33
4.10.2.5	jhi_get_height_window	33

4.10.2.6	jhi_get_joystick_status	33
4.10.2.7	jhi_get_keyboard_status	34
4.10.2.8	jhi_get_mouse_status	35
4.10.2.9	jhi_get_number_of_events	35
4.10.2.10	jhi_get_width_window	35
4.10.2.11	jhi_initialize_window	35
4.10.2.12	jhi_out_window	35
4.10.2.13	jhi_print_pixel	36
4.10.2.14	jhi_set_background_color	36
 Index		 37

Chapter 1

Class Index

1.1 Class List

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

JHI_Color_RGB	
Structure of the RGB color	5
JHI_Effect	
Struct of the configuration of the Effect	5
JHI_Font	
Structure of configuration of the Font	6
JHI_Image	
Structure of configuration of the Image	7
JHI_JoystickSt	
Joystick structure	7
JHI_KeyboardSt	
Keyboard structure	8
JHI_MouseSt	
Mouse structure	9
JHI_Music	
Struct of configuration of the Music	9
JHI_Point2d	
Point that contain position (x,y)	10
JHI_Text	
Structure of configuration of the Text	10
JHI_Window	
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:

jhi_colorsSt.h	??
jhi_font.h	
This file contains functions structure of the Font	13
jhi_image.h	
This file contains functions structure of the Image	15
jhi_joystick.h	
This file contains structures and enumeration of keyboard	17
jhi_keyboard.h	
This file contains structures and enumeration of keyboard	19
jhi_mouse.h	
This file contains structure of the mouse	23
jhi_rand.h	
This file contains rand auxiliary MACROS	24
jhi_shapes.h	
This file contains shapes functions of libjhi-sdl	24
jhi_sound.h	
This file contains functions and structures of the Music and Effect	27
jhi_text.h	
This file contains functions and structures of the Text	28
jhi_timer.h	??
jhi_window.h	
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_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.

4.1.1 Detailed Description

This file contains functions structure of the Font.

4.1.2 Function Documentation

4.1.2.1 int jhi_get_lenght_font ([JHI_Font](#) * font)

Get the Font length.

Returns

Font length

4.1.2.2 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

4.2 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.

4.2.1 Detailed Description

This file contains functions structure of the Image.

4.2.2 Function Documentation

4.2.2.1 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

4.2.2.2 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

4.2.2.3 int `jhi_get_image_height (JHI_Image * img)`

Get the Image height.

Parameters

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

Returns

Image height

4.2.2.4 int `jhi_get_image_width (JHI_Image * img)`

Get the Image width.

Parameters

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

Returns

Image width

4.2.2.5 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

4.2.2.6 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

4.2.2.7 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

4.3 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.

4.3.1 Detailed Description

This file contains structures and enumeration of keyboard.

4.3.2 Enumeration Type Documentation

4.3.2.1 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

4.3.3 Function Documentation

4.3.3.1 void [jhi_free_joystick_index](#) (short int *joy_index*)

Free the joystick structure.

Parameters

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

Returns

1 success, 0 otherwise

4.3.3.2 [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

4.3.3.3 int jhi_get_num_of_joystick ()

Get number of joysticks.

Returns

number of joysticks

4.3.3.4 void jhi_init_joystick (JHI_JoystickSt * *joy*)

Initialize the joystick structure.

Parameters

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

4.3.3.5 int jhi_is_valid_joystick_index (int *index*)

Check if index is valid.

Returns

1 ok, 0 otherwise

4.3.3.6 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

4.4 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.

4.4.1 Detailed Description

This file contains structures and enumeration of keyboard.

4.4.2 Enumeration Type Documentation

4.4.2.1 enum [JHI_KeyBoardEvents](#)

Enumeration of events of keyboard.

Enumerator

[KEYBOARD_UP](#) Keyboard was pressed

[KEYBOARD_DOWN](#) Keyboard was released

[KEYBOARD_NOT_EVENT](#) Nothing

4.4.2.2 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

4.4.3 Function Documentation

4.4.3.1 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

4.4.3.2 void jhi_init_keyboard (JHI_KeyboardSt * key)

init the keyboard structure

Parameters

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

4.4.3.3 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

4.5 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)

4.5.1 Detailed Description

This file contains structure of the mouse.

4.5.2 Enumeration Type Documentation

4.5.2.1 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 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

4.6.1 Detailed Description

This file contains rand auxiliary MACROS.

4.7 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 cirle in the screen.
- void [jhi_draw_fill_circle](#) ([JHI_Point2d](#) center_point, int radius, [JHI_Color](#) col)
Draw a fill cirle 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.

4.7.1 Detailed Description

This file contains shapes functions of libjhi-sdl.

4.7.2 Function Documentation

4.7.2.1 void [jhi_draw_circle](#) ([JHI_Point2d](#) center_point, float radius, [JHI_Color](#) col)

Drawing a cirle 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

4.7.2.2 void [jhi_draw_fill_circle](#) ([JHI_Point2d](#) center_point, int radius, [JHI_Color](#) col)

Draw a fill cirle 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

4.7.2.3 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

4.7.2.4 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

4.7.2.5 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

4.7.2.6 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

4.7.2.7 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

4.7.2.8 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

4.7.2.9 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

4.8 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.

4.8.1 Detailed Description

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

4.8.2 Function Documentation

4.8.2.1 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

4.8.2.2 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

4.8.2.3 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

4.8.2.4 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

4.9 `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.

4.9.1 Detailed Description

This file contains functions and structures of the Text.

4.9.2 Function Documentation

4.9.2.1 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

4.9.2.2 int [jhi_get_text_height](#) ([JHI_Text](#) * *text*)

Get the Text height.

Parameters

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

Returns

Text height

4.9.2.3 `int jhi_get_text_width (JHI_Text * text)`

Get the Text width.

Parameters

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

Returns

Text width

4.9.2.4 void jhi_init_text (JHI_Text * *text*)

Init the Text structure.

Parameters

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

4.9.2.5 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

4.10 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 especificied.
- 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.

4.10.1 Detailed Description

This file contains functions related to Window.

4.10.2 Function Documentation

4.10.2.1 void [jhi_choice_window_name](#) (const char * win_name)

Set the name of the Window.

Parameters

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

4.10.2.2 void jhi_draw_object (SDL_Surface * *object*, int *x*, int *y*)

Draw object in the window.

Parameters

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

4.10.2.3 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

4.10.2.4 char jhi_get_close_window ()

Get if close the Window.

Returns

NOT_CLOSE, case this Window wasn't closed, CLOSE otherwise

4.10.2.5 int jhi_get_height_window ()

Get the height of the window.

Returns

height of the window

4.10.2.6 JHI_JoystickSt jhi_get_joystick_status (int *i*)

Get status of the joystick input.

Parameters

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

Returns

keyboard status

4.10.2.7 JHI_KeyboardSt jhi_get_keyboard_status (int i)

Get status of the keyboard input.

Parameters

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

Returns

keyboard status

4.10.2.8 JHI_MouseSt jhi_get_mouse_status (int *i*)

Get status of the mouse input.

Parameters

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

Returns

Mouse status

4.10.2.9 int jhi_get_number_of_events ()

Return the number of events captured in the iteration.

Returns

Number of events captured in the iteration

4.10.2.10 int jhi_get_width_window ()

Get the height of the window.

Returns

height of the window

4.10.2.11 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

4.10.2.12 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

4.10.2.13 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

4.10.2.14 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, 24
- 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, 21
 - KEY_1, 21
 - KEY_2, 21
 - KEY_3, 21
 - KEY_4, 21
 - KEY_5, 21
 - KEY_6, 21

- KEY_7, [21](#)
- KEY_8, [21](#)
- KEY_9, [21](#)
- KEY_A, [21](#)
- KEY_B, [21](#)
- KEY_C, [21](#)
- KEY_D, [21](#)
- KEY_DOWN, [21](#)
- KEY_E, [21](#)
- KEY_ENTER, [21](#)
- KEY_F, [21](#)
- KEY_G, [21](#)
- KEY_H, [21](#)
- KEY_I, [21](#)
- KEY_J, [21](#)
- KEY_K, [21](#)
- KEY_L, [21](#)
- KEY_LEFT, [21](#)
- KEY_M, [21](#)
- KEY_N, [21](#)
- KEY_O, [21](#)
- KEY_P, [21](#)
- KEY_Q, [21](#)
- KEY_R, [21](#)
- KEY_RIGHT, [21](#)
- KEY_S, [21](#)
- KEY_T, [21](#)
- KEY_U, [21](#)
- KEY_UP, [21](#)
- 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, [24](#)
 - MOUSE_NOT_EVENT, [24](#)
 - ON_CLICK_LEFT, [24](#)
 - ON_CLICK_RIGHT, [24](#)
- jhi_choice_window_name
 - jhi_window.h, [32](#)
- 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, [16](#)
- jhi_draw_line
 - jhi_shapes.h, [26](#)
- jhi_draw_object
 - jhi_window.h, [33](#)
- jhi_draw_object_with_clip
 - jhi_window.h, [33](#)
- 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, [13](#)
 - jhi_load_font, [13](#)
- jhi_free_joystick_index
 - jhi_joystick.h, [18](#)
- jhi_get_central_pos
 - jhi_shapes.h, [26](#)
- jhi_get_close_window
 - jhi_window.h, [33](#)
- jhi_get_height_window
 - jhi_window.h, [33](#)
- jhi_get_image_height
 - jhi_image.h, [16](#)
- jhi_get_image_width
 - jhi_image.h, [16](#)
- jhi_get_joystick_dir
 - jhi_joystick.h, [18](#)
- jhi_get_joystick_status
 - jhi_window.h, [33](#)
- jhi_get_keyboard_status
 - jhi_window.h, [33](#)
- jhi_get_lenght_font
 - jhi_font.h, [13](#)
- jhi_get_mouse_status
 - jhi_window.h, [35](#)
- jhi_get_num_of_joystick
 - jhi_joystick.h, [19](#)
- jhi_get_number_of_events
 - jhi_window.h, [35](#)
- jhi_get_opposite_key_arrow
 - jhi_keyboard.h, [22](#)
- jhi_get_text_height
 - jhi_text.h, [29](#)
- jhi_get_text_width
 - jhi_text.h, [29](#)
- jhi_get_width_window
 - jhi_window.h, [35](#)
- jhi_image.h, [15](#)
 - jhi_draw_image, [15](#)
 - jhi_draw_image_with_clip, [16](#)
 - jhi_get_image_height, [16](#)
 - jhi_get_image_width, [16](#)
 - jhi_load_image, [16](#)
 - jhi_load_image_with_transparent_color, [16](#)
 - jhi_resize_image, [17](#)
- jhi_init_joystick
 - jhi_joystick.h, [19](#)

- jhi_init_keyboard
 - jhi_keyboard.h, 23
- jhi_init_text
 - jhi_text.h, 31
- jhi_initialize_window
 - jhi_window.h, 35
- jhi_is_colid
 - jhi_shapes.h, 27
- jhi_is_key_arrow
 - jhi_keyboard.h, 23
- jhi_is_valid_joystick_index
 - jhi_joystick.h, 19
- jhi_joystick.h, 17
 - jhi_free_joystick_index, 18
 - jhi_get_joystick_dir, 18
 - jhi_get_num_of_joystick, 19
 - jhi_init_joystick, 19
 - 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, 22
 - jhi_init_keyboard, 23
 - jhi_is_key_arrow, 23
- jhi_load_effect
 - jhi_sound.h, 28
- jhi_load_font
 - jhi_font.h, 13
- 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, 23
 - JHI_MouseEvents, 24
- jhi_open_joystick_index
 - jhi_joystick.h, 19
- jhi_out_window
 - jhi_window.h, 35
- jhi_play_effect
 - jhi_sound.h, 28
- jhi_play_music
 - jhi_sound.h, 28
- jhi_print_pixel
 - jhi_window.h, 36
- jhi_rand.h, 24
- jhi_resize_image
 - jhi_image.h, 17
- jhi_set_background_color
 - jhi_window.h, 36
- jhi_set_text
 - jhi_text.h, 31
- 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_is_colid, 27
- jhi_sound.h, 27
 - jhi_load_effect, 28
 - jhi_load_music, 28
 - jhi_play_effect, 28
 - jhi_play_music, 28
- jhi_text.h, 28
 - jhi_draw_text, 29
 - jhi_get_text_height, 29
 - jhi_get_text_width, 29
 - jhi_init_text, 31
 - jhi_set_text, 31
- jhi_window.h, 31
 - jhi_choice_window_name, 32
 - jhi_draw_object, 33
 - jhi_draw_object_with_clip, 33
 - jhi_get_close_window, 33
 - jhi_get_height_window, 33
 - jhi_get_joystick_status, 33
 - jhi_get_keyboard_status, 33
 - jhi_get_mouse_status, 35
 - jhi_get_number_of_events, 35
 - jhi_get_width_window, 35
 - jhi_initialize_window, 35
 - jhi_out_window, 35
 - jhi_print_pixel, 36
 - jhi_set_background_color, 36
- joy
 - JHI_Window, 12
- joy_event
 - JHI_JoystickSt, 8
- joy_index
 - JHI_JoystickSt, 8
- KEY_0
 - jhi_keyboard.h, 21
- KEY_1
 - jhi_keyboard.h, 21
- KEY_2
 - jhi_keyboard.h, 21
- KEY_3
 - jhi_keyboard.h, 21
- KEY_4
 - jhi_keyboard.h, 21
- KEY_5
 - jhi_keyboard.h, 21
- KEY_6
 - jhi_keyboard.h, 21
- KEY_7
 - jhi_keyboard.h, 21
- KEY_8
 - jhi_keyboard.h, 21
- KEY_9
 - jhi_keyboard.h, 21

KEY_A
 jhi_keyboard.h, 21
 KEY_B
 jhi_keyboard.h, 21
 KEY_C
 jhi_keyboard.h, 21
 KEY_D
 jhi_keyboard.h, 21
 KEY_DOWN
 jhi_keyboard.h, 21
 KEY_E
 jhi_keyboard.h, 21
 KEY_ENTER
 jhi_keyboard.h, 21
 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, 21
 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, 21
 KEY_S
 jhi_keyboard.h, 21
 KEY_T
 jhi_keyboard.h, 21
 KEY_U
 jhi_keyboard.h, 21
 KEY_UP
 jhi_keyboard.h, 21
 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, 24
 MOUSE_NOT_EVENT
 jhi_mouse.h, 24
 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, 24
 ON_CLICK_RIGHT
 jhi_mouse.h, 24
 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](#)