

NES Snake

Generated by Doxygen 1.8.12



# Contents

<b>1</b>	<b>README</b>	<b>1</b>
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>File Documentation</b>	<b>5</b>
3.1	C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/game_over_nam.h File Reference . . .	5
3.1.1	Detailed Description . . . . .	5
3.1.2	Variable Documentation . . . . .	5
3.1.2.1	game_over_nam . . . . .	5
3.2	C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/level1_nam.h File Reference . . . . .	6
3.2.1	Detailed Description . . . . .	6
3.2.2	Variable Documentation . . . . .	6
3.2.2.1	level1_nam . . . . .	6
3.3	C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/level2_nam.h File Reference . . . . .	6
3.3.1	Detailed Description . . . . .	7
3.3.2	Variable Documentation . . . . .	7
3.3.2.1	level2_nam . . . . .	7
3.4	C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/levels_pal.h File Reference . . . . .	7
3.4.1	Variable Documentation . . . . .	7
3.4.1.1	levels_pal . . . . .	7
3.5	C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/menue_pal.h File Reference . . . . .	8
3.5.1	Detailed Description . . . . .	8
3.5.2	Variable Documentation . . . . .	8

3.5.2.1	menue_pal . . . . .	8
3.6	C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/sprites_pal.h File Reference . . . . .	8
3.6.1	Detailed Description . . . . .	8
3.6.2	Variable Documentation . . . . .	9
3.6.2.1	sprites_pal . . . . .	9
3.7	C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/titlescreen_nam.h File Reference . . . . .	9
3.7.1	Detailed Description . . . . .	9
3.7.2	Variable Documentation . . . . .	9
3.7.2.1	titlescreen_nam . . . . .	9
3.8	C:/Users/Administrator/Documents/GitHub/NES-Snake/NESLibrary/bgsplit_nam.h File Reference . . . . .	10
3.8.1	Variable Documentation . . . . .	10
3.8.1.1	bgsplit_nam . . . . .	10
3.9	C:/Users/Administrator/Documents/GitHub/NES-Snake/NESLibrary/neslib.h File Reference . . . . .	10
3.9.1	Macro Definition Documentation . . . . .	12
3.9.1.1	FALSE . . . . .	12
3.9.1.2	MASK_BG . . . . .	12
3.9.1.3	MASK_EDGE_BG . . . . .	12
3.9.1.4	MASK_EDGE_SPR . . . . .	12
3.9.1.5	MASK_SPR . . . . .	12
3.9.1.6	MAX . . . . .	12
3.9.1.7	MIN . . . . .	13
3.9.1.8	MSB . . . . .	13
3.9.1.9	NAMETABLE_A . . . . .	13
3.9.1.10	NAMETABLE_B . . . . .	13
3.9.1.11	NAMETABLE_C . . . . .	13
3.9.1.12	NAMETABLE_D . . . . .	13
3.9.1.13	NT_UPD_EOF . . . . .	13
3.9.1.14	NT_UPD_HORZ . . . . .	13
3.9.1.15	NT_UPD_VERT . . . . .	13
3.9.1.16	NTADR_A . . . . .	14

3.9.1.17	NTADR_B	14
3.9.1.18	NTADR_C	14
3.9.1.19	NTADR_D	14
3.9.1.20	NULL	14
3.9.1.21	OAM_BEHIND	14
3.9.1.22	OAM_FLIP_H	14
3.9.1.23	OAM_FLIP_V	14
3.9.1.24	PAD_A	14
3.9.1.25	PAD_B	15
3.9.1.26	PAD_DOWN	15
3.9.1.27	PAD_LEFT	15
3.9.1.28	PAD_RIGHT	15
3.9.1.29	PAD_SELECT	15
3.9.1.30	PAD_START	15
3.9.1.31	PAD_UP	15
3.9.1.32	TRUE	15
3.9.2	Function Documentation	15
3.9.2.1	bank_bg()	15
3.9.2.2	bank_spr()	16
3.9.2.3	delay()	16
3.9.2.4	flush_vram_update()	16
3.9.2.5	memcpy()	16
3.9.2.6	memfill()	16
3.9.2.7	music_pause()	16
3.9.2.8	music_play()	17
3.9.2.9	music_stop()	17
3.9.2.10	oam_clear()	17
3.9.2.11	oam_hide_rest()	17
3.9.2.12	oam_meta_spr()	17
3.9.2.13	oam_size()	18

3.9.2.14	<code>oam_spr()</code>	18
3.9.2.15	<code>pad_poll()</code>	18
3.9.2.16	<code>pad_state()</code>	18
3.9.2.17	<code>pad_trigger()</code>	18
3.9.2.18	<code>pal_all()</code>	19
3.9.2.19	<code>pal_bg()</code>	19
3.9.2.20	<code>pal_bg_bright()</code>	19
3.9.2.21	<code>pal_bright()</code>	19
3.9.2.22	<code>pal_clear()</code>	19
3.9.2.23	<code>pal_col()</code>	19
3.9.2.24	<code>pal_spr()</code>	20
3.9.2.25	<code>pal_spr_bright()</code>	20
3.9.2.26	<code>ppu_mask()</code>	20
3.9.2.27	<code>ppu_off()</code>	20
3.9.2.28	<code>ppu_on_all()</code>	21
3.9.2.29	<code>ppu_on_bg()</code>	21
3.9.2.30	<code>ppu_on_spr()</code>	21
3.9.2.31	<code>ppu_system()</code>	21
3.9.2.32	<code>ppu_wait_frame()</code>	21
3.9.2.33	<code>ppu_wait_nmi()</code>	22
3.9.2.34	<code>rand16()</code>	22
3.9.2.35	<code>rand8()</code>	22
3.9.2.36	<code>sample_play()</code>	22
3.9.2.37	<code>scroll()</code>	22
3.9.2.38	<code>set_rand()</code>	23
3.9.2.39	<code>set_vram_update()</code>	23
3.9.2.40	<code>sfx_play()</code>	23
3.9.2.41	<code>split()</code>	23
3.9.2.42	<code>vram_adr()</code>	23
3.9.2.43	<code>vram_fill()</code>	24

3.9.2.44	<a href="#">vram_inc()</a>	24
3.9.2.45	<a href="#">vram_put()</a>	24
3.9.2.46	<a href="#">vram_read()</a>	24
3.9.2.47	<a href="#">vram_unrle()</a>	24
3.9.2.48	<a href="#">vram_write()</a>	25
3.10	<a href="#">C:/Users/Administrator/Documents/GitHub/NES-Snake/NESLibrary/test_nam.h File Reference</a>	25
3.10.1	<a href="#">Variable Documentation</a>	25
3.10.1.1	<a href="#">test_nam</a>	25
3.11	<a href="#">C:/Users/Administrator/Documents/GitHub/NES-Snake/README.md File Reference</a>	25
3.12	<a href="#">C:/Users/Administrator/Documents/GitHub/NES-Snake/src/definitions.h File Reference</a>	25
3.12.1	<a href="#">Detailed Description</a>	27
3.12.2	<a href="#">Macro Definition Documentation</a>	28
3.12.2.1	<a href="#">DIGIT_O_TILE</a>	28
3.12.2.2	<a href="#">DIR_DOWN</a>	28
3.12.2.3	<a href="#">DIR_LEFT</a>	28
3.12.2.4	<a href="#">DIR_RIGHT</a>	28
3.12.2.5	<a href="#">DIR_UP</a>	28
3.12.2.6	<a href="#">EMPTY_TILE</a>	28
3.12.2.7	<a href="#">LEVELS_ALL</a>	28
3.12.2.8	<a href="#">MAP_HEIGHT</a>	28
3.12.2.9	<a href="#">MAP_WIDTH</a>	28
3.12.2.10	<a href="#">MAPARRAY_ADR</a>	29
3.12.2.11	<a href="#">NAMETABLE1_START</a>	29
3.12.2.12	<a href="#">SNAKE_BODY_TILE</a>	29
3.12.2.13	<a href="#">SNAKE_HEAD_TILE_HORZ</a>	29
3.12.2.14	<a href="#">SNAKE_HEAD_TILE_VERT</a>	29
3.12.2.15	<a href="#">SNAKE_MAX_SIZE</a>	29
3.12.2.16	<a href="#">SPIDER_TILE</a>	29
3.12.2.17	<a href="#">WALL_TILE_1</a>	30
3.12.2.18	<a href="#">WALL_TILE_2</a>	30

3.12.3 Variable Documentation . . . . .	30
3.12.3.1 body_coordinates . . . . .	30
3.12.3.2 body_tile_x . . . . .	30
3.12.3.3 body_tile_y . . . . .	30
3.12.3.4 current_level . . . . .	30
3.12.3.5 direction . . . . .	30
3.12.3.6 gameover . . . . .	30
3.12.3.7 gameover_loop . . . . .	31
3.12.3.8 i . . . . .	31
3.12.3.9 input . . . . .	31
3.12.3.10 item_x . . . . .	31
3.12.3.11 item_y . . . . .	31
3.12.3.12 j . . . . .	31
3.12.3.13 k . . . . .	31
3.12.3.14 l . . . . .	31
3.12.3.15 last_body_pixel_x . . . . .	31
3.12.3.16 last_body_pixel_y . . . . .	32
3.12.3.17 levelList . . . . .	32
3.12.3.18 map . . . . .	32
3.12.3.19 nameRow . . . . .	32
3.12.3.20 nametable_fetch . . . . .	32
3.12.3.21 pause . . . . .	32
3.12.3.22 pause_loop . . . . .	32
3.12.3.23 restart . . . . .	33
3.12.3.24 size_index . . . . .	33
3.12.3.25 snake_head_attribute . . . . .	33
3.12.3.26 snake_head_tile . . . . .	33
3.12.3.27 snake_x . . . . .	33
3.12.3.28 snake_y . . . . .	33
3.12.3.29 speed_counter . . . . .	33



3.12.3.30	sprite_offset	33
3.12.3.31	titlescreen	33
3.12.3.32	ul	34
3.12.3.33	update_list	34
3.13	C:/Users/Administrator/Documents/GitHub/NES-Snake/src/input.c File Reference	34
3.13.1	Detailed Description	34
3.13.2	Function Documentation	34
3.13.2.1	input_btn_start()	34
3.13.2.2	mainloop_handle_input()	35
3.14	C:/Users/Administrator/Documents/GitHub/NES-Snake/src/render.c File Reference	35
3.14.1	Detailed Description	36
3.14.2	Function Documentation	36
3.14.2.1	center_score_when_gameover()	36
3.14.2.2	draw_game_over_screen()	36
3.14.2.3	draw_item()	37
3.14.2.4	draw_level_screen()	38
3.14.2.5	draw_pause_screen()	39
3.14.2.6	draw_score()	40
3.14.2.7	draw_snake()	40
3.14.2.8	draw_title_screen()	41
3.14.2.9	init_updateList()	42
3.14.2.10	mainloop_render()	42
3.15	C:/Users/Administrator/Documents/GitHub/NES-Snake/src/snake.c File Reference	43
3.15.1	Detailed Description	44
3.15.2	Function Documentation	44
3.15.2.1	main()	44
3.16	C:/Users/Administrator/Documents/GitHub/NES-Snake/src/update.c File Reference	45
3.16.1	Detailed Description	46
3.16.2	Function Documentation	46
3.16.2.1	add_snake_body_element()	46
3.16.2.2	calc_random_item_position()	47
3.16.2.3	check_collision_body()	47
3.16.2.4	check_collision_item()	48
3.16.2.5	check_collision_wall()	48
3.16.2.6	check_next_level()	49
3.16.2.7	load_map_data_into_array()	50
3.16.2.8	mainloop_update()	50
3.16.2.9	update_snake_body()	51



# Chapter 1

## README

NES Snake This project is my first attempt to write a simple NES Snake game using Shiru's NESLibrary, based on the CC65 project. You can find out more about Shiru's NESLibrary here: [http://shiru.undergrund.net/articles/programming\\_nes\\_games\\_in\\_c.htm](http://shiru.undergrund.net/articles/programming_nes_games_in_c.htm) Also, if you are interested in the general CC65 project, you can find it here: <http://www.cc65.org/> Or just visit the project directly on GitHub: <https://github.com/cc65/cc65>



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/ <a href="#">game_over_nam.h</a> This header file contains the nametable (background) of the gameover screen. Created with NES Screen Tool 2.04 (Option Nametable -> Save nametable and attributes -> RLE packed as C header (.h)) . . . . .	5
C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/ <a href="#">level1_nam.h</a> This header file contains the nametable (background) of level map 1. Created with NES Screen Tool 2.04 (Option Nametable -> Save nametable and attributes -> RLE packed as C header (.h)) . . . . .	6
C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/ <a href="#">level2_nam.h</a> This header file contains the nametable (background) of level map 2. Created with NES Screen Tool 2.04 (Option Nametable -> Save nametable and attributes -> RLE packed as C header (.h)) . . . . .	6
C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/ <a href="#">levels_pal.h</a> . . . . .	7
C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/ <a href="#">menue_pal.h</a> This header file contains the color palette for menus (titlescreen, gameover screen). Created with NES Screen Tool 2.04 (Option Palettes -> Put C data to clipboard) . . . . .	8
C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/ <a href="#">sprites_pal.h</a> This header file contains the color palette for sprites . . . . .	8
C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/ <a href="#">titlescreen_nam.h</a> This header file contains the nametable (background) of the titlescreen. Created with NES Screen Tool 2.04 (Option Nametable -> Save nametable and attributes -> RLE packed as C header (.h)) . . . . .	9
C:/Users/Administrator/Documents/GitHub/NES-Snake/NESLibrary/ <a href="#">bgsplit_nam.h</a> . . . . .	10
C:/Users/Administrator/Documents/GitHub/NES-Snake/NESLibrary/ <a href="#">neslib.h</a> . . . . .	10
C:/Users/Administrator/Documents/GitHub/NES-Snake/NESLibrary/ <a href="#">test_nam.h</a> . . . . .	25
C:/Users/Administrator/Documents/GitHub/NES-Snake/src/ <a href="#">definitions.h</a> This header files contains defines all global variables and constants, macros and includes of nametable and palette definition . . . . .	25
C:/Users/Administrator/Documents/GitHub/NES-Snake/src/ <a href="#">input.c</a> This file contains functions for input handling from a controller . . . . .	34
C:/Users/Administrator/Documents/GitHub/NES-Snake/src/ <a href="#">render.c</a> This file contains all functionality to draw onto the screen, either as sprites or as background tiles . . . . .	35
C:/Users/Administrator/Documents/GitHub/NES-Snake/src/ <a href="#">snake.c</a> Maingame file, containing the main game loop . . . . .	43
C:/Users/Administrator/Documents/GitHub/NES-Snake/src/ <a href="#">update.c</a> This file contains all ingame logic functionalities and utility functionalities . . . . .	45



## Chapter 3

# File Documentation

### 3.1 C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/game\_over\_nam.h File Reference

This header file contains the nametable (background) of the gameover screen. Created with NES Screen Tool 2.04 (Option Nametable -> Save nametable and attributes -> RLE packed as C header (.h)).

#### Variables

- const unsigned char [game\\_over\\_nam](#) [59]

#### 3.1.1 Detailed Description

This header file contains the nametable (background) of the gameover screen. Created with NES Screen Tool 2.04 (Option Nametable -> Save nametable and attributes -> RLE packed as C header (.h)).

#### Author

Sebastian Dine

#### 3.1.2 Variable Documentation

##### 3.1.2.1 game\_over\_nam

```
const unsigned char game_over_nam[59]
```

#### Initial value:

```
={
0x01,0x00,0x01,0xe9,0x27,0x21,0x2d,0x25,0x00,0x2f,0x36,0x25,0x32,0x00,0x01,0x56,
0x33,0x23,0x2f,0x32,0x25,0x1a,0x00,0x01,0x54,0x30,0x32,0x25,0x33,0x33,0x00,0x33,
0x34,0x21,0x32,0x34,0x00,0x34,0x2f,0x00,0x23,0x2f,0x2e,0x34,0x29,0x2e,0x35,0x25,
0x00,0x01,0xfe,0x00,0x01,0xfe,0x00,0x01,0x45,0x01,0x00
}
```

## 3.2 C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/level1\_nam.h File Reference

This header file contains the nametable (background) of level map 1. Created with NES Screen Tool 2.04 (Option Nametable -> Save nametable and attributes -> RLE packed as C header (.h)).

### Variables

- const unsigned char [level1\\_nam](#) [171]

#### 3.2.1 Detailed Description

This header file contains the nametable (background) of level map 1. Created with NES Screen Tool 2.04 (Option Nametable -> Save nametable and attributes -> RLE packed as C header (.h)).

### Author

Sebastian Dine

#### 3.2.2 Variable Documentation

##### 3.2.2.1 level1\_nam

```
const unsigned char level1_nam[171]
```

### Initial value:

```
= {
0x01, 0x00, 0x01, 0x20, 0x33, 0x23, 0x2f, 0x32, 0x25, 0x1a, 0x00, 0x01, 0x38, 0x43, 0x01, 0x3d,
0x44, 0x44, 0x43, 0x43, 0x00, 0x01, 0x1b, 0x43, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x43, 0x01, 0x03,
0x00, 0x01, 0x1b, 0x43, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x43, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x43,
0x01, 0x03, 0x00, 0x01, 0x1b, 0x43, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x43, 0x01, 0x03, 0x00, 0x01,
0x1b, 0x43, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x43, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x43, 0x01, 0x03,
0x00, 0x01, 0x1b, 0x43, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x43, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x43,
0x01, 0x03, 0x00, 0x01, 0x1b, 0x43, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x43, 0x01, 0x03, 0x00, 0x01,
0x1b, 0x43, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x43, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x43, 0x01, 0x03,
0x00, 0x01, 0x1b, 0x43, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x43, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x43,
0x01, 0x03, 0x00, 0x01, 0x1b, 0x43, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x43, 0x01, 0x2e, 0x44, 0x43,
0x01, 0x05, 0x44, 0x43, 0x01, 0x0a, 0x00, 0x01, 0x3f, 0x01, 0x00
}
```

## 3.3 C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/level2\_nam.h File Reference

This header file contains the nametable (background) of level map 2. Created with NES Screen Tool 2.04 (Option Nametable -> Save nametable and attributes -> RLE packed as C header (.h)).

### Variables

- const unsigned char [level2\\_nam](#) [264]



### 3.3.1 Detailed Description

This header file contains the nametable (background) of level map 2. Created with NES Screen Tool 2.04 (Option Nametable -> Save nametable and attributes -> RLE packed as C header (.h)).

#### Author

Sebastian Dine

### 3.3.2 Variable Documentation

#### 3.3.2.1 level2\_nam

```
const unsigned char level2_nam[264]
```

#### Initial value:

```
={
0x01,0x00,0x01,0x20,0x33,0x23,0x2f,0x32,0x25,0x1a,0x00,0x01,0x38,0x43,0x01,0x3d,
0x44,0x44,0x43,0x43,0x00,0x01,0x0c,0x44,0x43,0x44,0x00,0x01,0x0b,0x43,0x01,0x03,
0x00,0x01,0x0c,0x44,0x43,0x44,0x00,0x01,0x0b,0x43,0x01,0x03,0x00,0x01,0x0c,0x44,
0x43,0x44,0x00,0x01,0x0b,0x43,0x01,0x03,0x00,0x01,0x0c,0x44,0x43,0x44,0x00,0x01,
0x0b,0x43,0x01,0x03,0x00,0x01,0x0c,0x44,0x43,0x44,0x00,0x01,0x0b,0x43,0x01,0x03,
0x00,0x01,0x0c,0x44,0x43,0x44,0x00,0x01,0x0b,0x43,0x01,0x03,0x00,0x01,0x0c,0x44,
0x43,0x44,0x00,0x01,0x0b,0x43,0x01,0x03,0x00,0x01,0x0c,0x44,0x43,0x44,0x00,0x01,
0x0b,0x43,0x01,0x03,0x00,0x01,0x1b,0x43,0x01,0x03,0x00,0x01,0x1b,0x43,0x01,0x03,
0x00,0x01,0x1b,0x43,0x01,0x03,0x00,0x01,0x1b,0x43,0x01,0x03,0x00,0x01,0x1b,0x43,
0x01,0x03,0x00,0x01,0x1b,0x43,0x01,0x03,0x00,0x01,0x1b,0x43,0x01,0x03,0x00,0x01,
0x0c,0x44,0x43,0x44,0x00,0x01,0x0b,0x43,0x01,0x03,0x00,0x01,0x0c,0x44,0x43,0x44,
0x00,0x01,0x0b,0x43,0x01,0x03,0x00,0x01,0x0c,0x44,0x43,0x44,0x00,0x01,0x0b,0x43,
0x01,0x03,0x00,0x01,0x0c,0x44,0x43,0x44,0x00,0x01,0x0b,0x43,0x01,0x03,0x00,0x01,
0x0c,0x44,0x43,0x44,0x00,0x01,0x0b,0x43,0x01,0x03,0x00,0x01,0x0c,0x44,0x43,0x44,
0x00,0x01,0x0b,0x43,0x01,0x03,0x00,0x01,0x0c,0x44,0x43,0x44,0x00,0x01,0x0b,0x43,
0x01,0x03,0x00,0x01,0x0c,0x44,0x43,0x44,0x00,0x01,0x0b,0x43,0x01,0x2e,0x44,0x43,
0x01,0x05,0x44,0x43,0x01,0x0a,0x01,0x00
}
```

## 3.4 C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/levels\_pal.h File Reference

### Variables

- const unsigned char [levels\\_pal](#) [16]

### 3.4.1 Variable Documentation

#### 3.4.1.1 levels\_pal

```
const unsigned char levels_pal[16]
```

#### Initial value:

```
={
    0x0f,0x00,0x10,0x2a,
    0x0f,0x01,0x21,0x31,
    0x0f,0x06,0x16,0x26,
    0x0f,0x09,0x19,0x29 }
}
```

### 3.5 C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/menue\_pal.h File Reference

This header file contains the color palette for menus (titlescreen, gameover screen). Created with NES Screen Tool 2.04 (Option Palettes -> Put C data to clipboard).

#### Variables

- const unsigned char `menue_pal` [16]

#### 3.5.1 Detailed Description

This header file contains the color palette for menus (titlescreen, gameover screen). Created with NES Screen Tool 2.04 (Option Palettes -> Put C data to clipboard).

#### Author

Sebastian Dine

#### 3.5.2 Variable Documentation

##### 3.5.2.1 `menue_pal`

```
const unsigned char menue_pal[16]
```

#### Initial value:

```
={  
    0x0f, 0x2a, 0x10, 0x20,  
    0x0f, 0x01, 0x21, 0x31,  
    0x0f, 0x06, 0x16, 0x26,  
    0x0f, 0x09, 0x19, 0x29 }  
}
```

### 3.6 C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/sprites\_pal.h File Reference

This header file contains the color palette for sprites.

#### Variables

- const unsigned char `sprites_pal` [16]

#### 3.6.1 Detailed Description

This header file contains the color palette for sprites.

#### Author

Sebastian Dine

### 3.6.2 Variable Documentation

### 3.6.2.1 sprites\_pal

```
const unsigned char sprites_pal[16]
```

**Initial value:**

```
= {
    0x0f, 0x17, 0x27, 0x37,
    0x0f, 0x11, 0x21, 0x31,
    0x0f, 0x15, 0x25, 0x35,
    0x0f, 0x19, 0x29, 0x2a };
```

### 3.7 C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/titlescreen\_nam.h File Reference

This header file contains the nametable (background) of the titlescreen. Created with NES Screen Tool 2.04 (Option Nametable -> Save nametable and attributes -> RLE packed as C header (.h)).

## Variables

- `const unsigned char titlescreen_nam [253]`

### 3.7.1 Detailed Description

This header file contains the nametable (background) of the titlescreen. Created with NES Screen Tool 2.04 (Option Nametable -> Save nametable and attributes -> RLE packed as C header (.h)).

**Author**

Sebastian Dine

### 3.7.2 Variable Documentation

### 3.7.2.1 titlescreen nam

```
const unsigned char titlescreen_nam[253]
```

**Initial value:**

```
= {
0x01, 0x43, 0x01, 0x3f, 0x44, 0x44, 0x00, 0x01, 0x1b, 0x44, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x44,
0x01, 0x03, 0x00, 0x01, 0x1b, 0x44, 0x01, 0x03, 0x00, 0x01, 0x06, 0x50, 0x51, 0x52, 0x53, 0x54,
0x55, 0x50, 0x51, 0x56, 0x57, 0x58, 0x59, 0x52, 0x53, 0x00, 0x01, 0x06, 0x44, 0x01, 0x03, 0x00,
0x01, 0x06, 0x60, 0x61, 0x62, 0x63, 0x64, 0x65, 0x60, 0x61, 0x66, 0x67, 0x68, 0x69, 0x62, 0x63,
0x00, 0x01, 0x06, 0x44, 0x01, 0x03, 0x00, 0x01, 0x06, 0x70, 0x71, 0x72, 0x73, 0x74, 0x75, 0x70,
0x71, 0x76, 0x77, 0x78, 0x79, 0x72, 0x73, 0x00, 0x01, 0x06, 0x44, 0x01, 0x03, 0x00, 0x01, 0x1b,
0x44, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x44, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x44, 0x01, 0x03, 0x00,
0x01, 0x1b, 0x44, 0x01, 0x03, 0x00, 0x01, 0x07, 0x30, 0x32, 0x25, 0x33, 0x33, 0x00, 0x33, 0x34,
0x21, 0x32, 0x34, 0x00, 0x01, 0x08, 0x44, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x44, 0x01, 0x03, 0x00,
0x01, 0x1b, 0x44, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x44, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x44, 0x01,
0x03, 0x00, 0x01, 0x1b, 0x44, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x44, 0x01, 0x03, 0x00, 0x01, 0x1b,
0x44, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x44, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x44, 0x01, 0x03, 0x00,
0x01, 0x1b, 0x44, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x44, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x44, 0x01,
0x03, 0x00, 0x01, 0x1b, 0x44, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x44, 0x01, 0x03, 0x33, 0x33, 0x22,
0x21, 0x33, 0x34, 0x29, 0x21, 0x2e, 0x00, 0x24, 0x29, 0x2e, 0x25, 0x0c, 0x12, 0x10, 0x11, 0x16,
0x00, 0x01, 0x08, 0x44, 0x44, 0x43, 0x01, 0x3f, 0x00, 0x01, 0x3f, 0x01, 0x00
}
```

### 3.8 C:/Users/Administrator/Documents/GitHub/NES-Snake/NESLibrary/bgsplit\_nam.h File Reference

#### Variables

- const unsigned char [bgsplit\\_nam](#) [267]

#### 3.8.1 Variable Documentation

##### 3.8.1.1 bgsplit\_nam

```
const unsigned char bgsplit_nam[267]
```

#### Initial value:

```
= {
0x01, 0x00, 0x01, 0xa3, 0x40, 0x01, 0x06, 0x00, 0x40, 0x01, 0x06, 0x00, 0x40, 0x01, 0x06, 0x00,
0x01, 0x08, 0x40, 0x01, 0x06, 0x00, 0x40, 0x01, 0x02, 0x00, 0x40, 0x01, 0x02, 0x00, 0x40, 0x01,
0x02, 0x00, 0x40, 0x01, 0x02, 0x00, 0x01, 0x0a, 0x40, 0x01, 0x02, 0x00, 0x01, 0x02, 0x40, 0x01,
0x02, 0x00, 0x40, 0x01, 0x02, 0x00, 0x40, 0x01, 0x02, 0x00, 0x40, 0x01, 0x02, 0x00, 0x01, 0x0a,
0x40, 0x01, 0x02, 0x00, 0x01, 0x02, 0x40, 0x01, 0x02, 0x00, 0x40, 0x01, 0x02, 0x00, 0x40, 0x01,
0x06, 0x00, 0x01, 0x0a, 0x40, 0x01, 0x02, 0x00, 0x01, 0x02, 0x40, 0x01, 0x06, 0x00, 0x40, 0x01,
0x06, 0x00, 0x01, 0x0a, 0x40, 0x01, 0x02, 0x00, 0x01, 0x02, 0x40, 0x01, 0x06, 0x00, 0x40, 0x01,
0x02, 0x00, 0x01, 0x68, 0x42, 0x01, 0x1f, 0x00, 0x01, 0x62, 0x40, 0x00, 0x01, 0x06, 0x40, 0x00,
0x01, 0x02, 0x40, 0x00, 0x01, 0x12, 0x40, 0x00, 0x01, 0x06, 0x40, 0x00, 0x01, 0x02, 0x40, 0x00,
0x01, 0x12, 0x40, 0x01, 0x02, 0x00, 0x40, 0x01, 0x02, 0x00, 0x40, 0x40, 0x00, 0x00, 0x40, 0x40,
0x00, 0x00, 0x40, 0x01, 0x02, 0x00, 0x40, 0x01, 0x04, 0x00, 0x01, 0x06, 0x40, 0x00, 0x40, 0x00,
0x40, 0x00, 0x40, 0x00, 0x40, 0x00, 0x01, 0x02, 0x40, 0x00, 0x01, 0x02, 0x40, 0x00, 0x40, 0x00,
0x40, 0x00, 0x40, 0x00, 0x40, 0x00, 0x01, 0x06, 0x40, 0x00, 0x40, 0x00, 0x40, 0x00, 0x40, 0x00,
0x40, 0x00, 0x01, 0x02, 0x40, 0x00, 0x01, 0x02, 0x40, 0x00, 0x40, 0x00, 0x40, 0x00, 0x40, 0x00,
0x40, 0x00, 0x01, 0x06, 0x40, 0x01, 0x02, 0x00, 0x40, 0x01, 0x02, 0x00, 0x40, 0x01, 0x02, 0x00,
0x40, 0x01, 0x02, 0x00, 0x40, 0x01, 0x02, 0x00, 0x40, 0x00, 0x40, 0x00, 0x40, 0x00, 0x01, 0xdb,
0x50, 0x01, 0x07, 0xaa, 0x01, 0x17, 0x0a, 0x01, 0x07, 0x01, 0x00
}
```

### 3.9 C:/Users/Administrator/Documents/GitHub/NES-Snake/NESLibrary/neslib.h File Reference

#### Macros

- #define [PAD\\_A](#) 0x01
- #define [PAD\\_B](#) 0x02
- #define [PAD\\_SELECT](#) 0x04
- #define [PAD\\_START](#) 0x08
- #define [PAD\\_UP](#) 0x10
- #define [PAD\\_DOWN](#) 0x20
- #define [PAD\\_LEFT](#) 0x40
- #define [PAD\\_RIGHT](#) 0x80
- #define [OAM\\_FLIP\\_V](#) 0x80
- #define [OAM\\_FLIP\\_H](#) 0x40
- #define [OAM\\_BEHIND](#) 0x20
- #define [MAX](#)(x1, x2) ((x1)<(x2)?(x2):(x1))
- #define [MIN](#)(x1, x2) ((x1)<(x2)?(x1):(x2))
- #define [MASK\\_SPR](#) 0x10
- #define [MASK\\_BG](#) 0x08
- #define [MASK\\_EDGE\\_SPR](#) 0x04

- #define `MASK_EDGE_BG` 0x02
- #define `NAMETABLE_A` 0x2000
- #define `NAMETABLE_B` 0x2400
- #define `NAMETABLE_C` 0x2800
- #define `NAMETABLE_D` 0x2c00
- #define `NULL` 0
- #define `TRUE` 1
- #define `FALSE` 0
- #define `NT_UPD_HORZ` 0x40
- #define `NT_UPD_VERT` 0x80
- #define `NT_UPD_EOF` 0xff
- #define `NTADR_A`(x, y) (`NAMETABLE_A`|(((y)<<5)|(x)))
- #define `NTADR_B`(x, y) (`NAMETABLE_B`|(((y)<<5)|(x)))
- #define `NTADR_C`(x, y) (`NAMETABLE_C`|(((y)<<5)|(x)))
- #define `NTADR_D`(x, y) (`NAMETABLE_D`|(((y)<<5)|(x)))
- #define `MSB`(x) (((x)>>8))

## Functions

- void \_\_fastcall\_\_ `pal_all` (const char \*data)
- void \_\_fastcall\_\_ `pal_bg` (const char \*data)
- void \_\_fastcall\_\_ `pal_spr` (const char \*data)
- void \_\_fastcall\_\_ `pal_col` (unsigned char index, unsigned char color)
- void \_\_fastcall\_\_ `pal_clear` (void)
- void \_\_fastcall\_\_ `pal_bright` (unsigned char bright)
- void \_\_fastcall\_\_ `pal_spr_bright` (unsigned char bright)
- void \_\_fastcall\_\_ `pal_bg_bright` (unsigned char bright)
- void \_\_fastcall\_\_ `ppu_wait_nmi` (void)
- void \_\_fastcall\_\_ `ppu_wait_frame` (void)
- void \_\_fastcall\_\_ `ppu_off` (void)
- void \_\_fastcall\_\_ `ppu_on_all` (void)
- void \_\_fastcall\_\_ `ppu_on_bg` (void)
- void \_\_fastcall\_\_ `ppu_on_spr` (void)
- void \_\_fastcall\_\_ `ppu_mask` (unsigned char mask)
- unsigned char \_\_fastcall\_\_ `ppu_system` (void)
- void \_\_fastcall\_\_ `oam_clear` (void)
- void \_\_fastcall\_\_ `oam_size` (unsigned char size)
- unsigned char \_\_fastcall\_\_ `oam_spr` (unsigned char x, unsigned char y, unsigned char chrnum, unsigned char attr, unsigned char sprid)
- unsigned char \_\_fastcall\_\_ `oam_meta_spr` (unsigned char x, unsigned char y, unsigned char sprid, const unsigned char \*data)
- void \_\_fastcall\_\_ `oam_hide_rest` (unsigned char sprid)
- void \_\_fastcall\_\_ `music_play` (unsigned char song)
- void \_\_fastcall\_\_ `music_stop` (void)
- void \_\_fastcall\_\_ `music_pause` (unsigned char pause)
- void \_\_fastcall\_\_ `sfx_play` (unsigned char sound, unsigned char channel)
- void \_\_fastcall\_\_ `sample_play` (unsigned char sample)
- unsigned char \_\_fastcall\_\_ `pad_poll` (unsigned char pad)
- unsigned char \_\_fastcall\_\_ `pad_trigger` (unsigned char pad)
- unsigned char \_\_fastcall\_\_ `pad_state` (unsigned char pad)
- void \_\_fastcall\_\_ `scroll` (unsigned int x, unsigned int y)
- void \_\_fastcall\_\_ `split` (unsigned int x, unsigned int y)
- void \_\_fastcall\_\_ `bank_spr` (unsigned char n)
- void \_\_fastcall\_\_ `bank_bg` (unsigned char n)

- unsigned char \_\_fastcall\_\_ [rand8](#) (void)
- unsigned int \_\_fastcall\_\_ [rand16](#) (void)
- void \_\_fastcall\_\_ [set\\_rand](#) (unsigned int seed)
- void \_\_fastcall\_\_ [set\\_vram\\_update](#) (unsigned char \*buf)
- void \_\_fastcall\_\_ [flush\\_vram\\_update](#) (unsigned char \*buf)
- void \_\_fastcall\_\_ [vram\\_adr](#) (unsigned int adr)
- void \_\_fastcall\_\_ [vram\\_put](#) (unsigned char n)
- void \_\_fastcall\_\_ [vram\\_fill](#) (unsigned char n, unsigned int len)
- void \_\_fastcall\_\_ [vram\\_inc](#) (unsigned char n)
- void \_\_fastcall\_\_ [vram\\_read](#) (unsigned char \*dst, unsigned int size)
- void \_\_fastcall\_\_ [vram\\_write](#) (unsigned char \*src, unsigned int size)
- void \_\_fastcall\_\_ [vram\\_unrle](#) (const unsigned char \*data)
- void \_\_fastcall\_\_ [memcpy](#) (void \*dst, void \*src, unsigned int len)
- void \_\_fastcall\_\_ [memfill](#) (void \*dst, unsigned char value, unsigned int len)
- void \_\_fastcall\_\_ [delay](#) (unsigned char frames)

### 3.9.1 Macro Definition Documentation

#### 3.9.1.1 FALSE

```
#define FALSE 0
```

#### 3.9.1.2 MASK\_BG

```
#define MASK_BG 0x08
```

#### 3.9.1.3 MASK\_EDGE\_BG

```
#define MASK_EDGE_BG 0x02
```

#### 3.9.1.4 MASK\_EDGE\_SPR

```
#define MASK_EDGE_SPR 0x04
```

#### 3.9.1.5 MASK\_SPR

```
#define MASK_SPR 0x10
```

#### 3.9.1.6 MAX

```
#define MAX(  
    x1,  
    x2 ) ((x1) < (x2) ? (x2) : (x1))
```

### 3.9.1.7 MIN

```
#define MIN(  
    x1,  
    x2 ) ((x1)<(x2)?(x1):(x2))
```

### 3.9.1.8 MSB

```
#define MSB(  
    x ) (((x)>>8))
```

### 3.9.1.9 NAMETABLE\_A

```
#define NAMETABLE_A 0x2000
```

### 3.9.1.10 NAMETABLE\_B

```
#define NAMETABLE_B 0x2400
```

### 3.9.1.11 NAMETABLE\_C

```
#define NAMETABLE_C 0x2800
```

### 3.9.1.12 NAMETABLE\_D

```
#define NAMETABLE_D 0x2c00
```

### 3.9.1.13 NT\_UPD\_EOF

```
#define NT_UPD_EOF 0xff
```

### 3.9.1.14 NT\_UPD\_HORZ

```
#define NT_UPD_HORZ 0x40
```

### 3.9.1.15 NT\_UPD\_VERT

```
#define NT_UPD_VERT 0x80
```

### 3.9.1.16 NTADR\_A

```
#define NTADR_A(  
    x,  
    y ) (NAMETABLE_A|(((y)<<5)|(x)))
```

### 3.9.1.17 NTADR\_B

```
#define NTADR_B(  
    x,  
    y ) (NAMETABLE_B|(((y)<<5)|(x)))
```

### 3.9.1.18 NTADR\_C

```
#define NTADR_C(  
    x,  
    y ) (NAMETABLE_C|(((y)<<5)|(x)))
```

### 3.9.1.19 NTADR\_D

```
#define NTADR_D(  
    x,  
    y ) (NAMETABLE_D|(((y)<<5)|(x)))
```

### 3.9.1.20 NULL

```
#define NULL 0
```

### 3.9.1.21 OAM\_BEHIND

```
#define OAM_BEHIND 0x20
```

### 3.9.1.22 OAM\_FLIP\_H

```
#define OAM_FLIP_H 0x40
```

### 3.9.1.23 OAM\_FLIP\_V

```
#define OAM_FLIP_V 0x80
```

### 3.9.1.24 PAD\_A

```
#define PAD_A 0x01
```



### 3.9.1.25 PAD\_B

```
#define PAD_B 0x02
```

### 3.9.1.26 PAD\_DOWN

```
#define PAD_DOWN 0x20
```

### 3.9.1.27 PAD\_LEFT

```
#define PAD_LEFT 0x40
```

### 3.9.1.28 PAD\_RIGHT

```
#define PAD_RIGHT 0x80
```

### 3.9.1.29 PAD\_SELECT

```
#define PAD_SELECT 0x04
```

### 3.9.1.30 PAD\_START

```
#define PAD_START 0x08
```

### 3.9.1.31 PAD\_UP

```
#define PAD_UP 0x10
```

### 3.9.1.32 TRUE

```
#define TRUE 1
```

## 3.9.2 Function Documentation

### 3.9.2.1 bank\_bg()

```
void __fastcall__ bank_bg (  
    unsigned char n )
```

### 3.9.2.2 bank\_spr()

```
void __fastcall__ bank_spr (
    unsigned char n )
```

### 3.9.2.3 delay()

```
void __fastcall__ delay (
    unsigned char frames )
```

Here is the caller graph for this function:



### 3.9.2.4 flush\_vram\_update()

```
void __fastcall__ flush_vram_update (
    unsigned char * buf )
```

### 3.9.2.5 memcpy()

```
void __fastcall__ memcpy (
    void * dst,
    void * src,
    unsigned int len )
```

### 3.9.2.6 memfill()

```
void __fastcall__ memfill (
    void * dst,
    unsigned char value,
    unsigned int len )
```

### 3.9.2.7 music\_pause()

```
void __fastcall__ music_pause (
    unsigned char pause )
```

### 3.9.2.8 music\_play()

```
void __fastcall__ music_play (  
    unsigned char song )
```

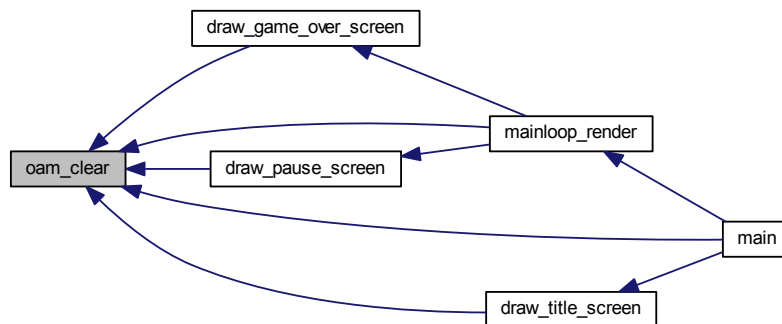
### 3.9.2.9 music\_stop()

```
void __fastcall__ music_stop (  
    void )
```

### 3.9.2.10 oam\_clear()

```
void __fastcall__ oam_clear (  
    void )
```

Here is the caller graph for this function:



### 3.9.2.11 oam\_hide\_rest()

```
void __fastcall__ oam_hide_rest (  
    unsigned char sprid )
```

### 3.9.2.12 oam\_meta\_spr()

```
unsigned char __fastcall__ oam_meta_spr (  
    unsigned char x,  
    unsigned char y,  
    unsigned char sprid,  
    const unsigned char * data )
```

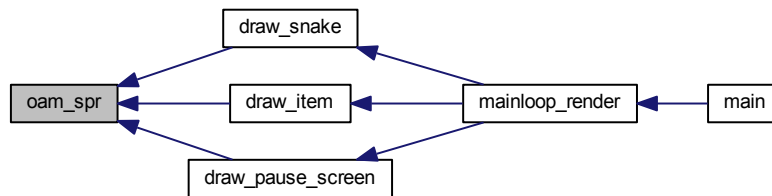
**3.9.2.13 oam\_size()**

```
void __fastcall__ oam_size (
    unsigned char size )
```

**3.9.2.14 oam\_spr()**

```
unsigned char __fastcall__ oam_spr (
    unsigned char x,
    unsigned char y,
    unsigned char chrnum,
    unsigned char attr,
    unsigned char sprid )
```

Here is the caller graph for this function:

**3.9.2.15 pad\_poll()**

```
unsigned char __fastcall__ pad_poll (
    unsigned char pad )
```

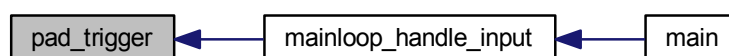
**3.9.2.16 pad\_state()**

```
unsigned char __fastcall__ pad_state (
    unsigned char pad )
```

**3.9.2.17 pad\_trigger()**

```
unsigned char __fastcall__ pad_trigger (
    unsigned char pad )
```

Here is the caller graph for this function:



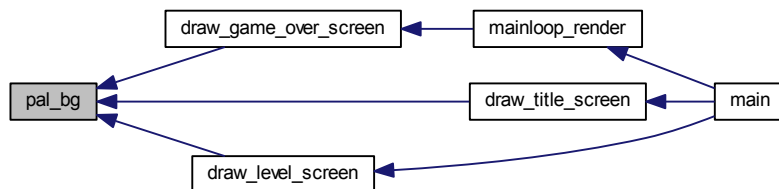
### 3.9.2.18 pal\_all()

```
void __fastcall__ pal_all (
    const char * data )
```

### 3.9.2.19 pal\_bg()

```
void __fastcall__ pal_bg (
    const char * data )
```

Here is the caller graph for this function:



### 3.9.2.20 pal\_bg\_bright()

```
void __fastcall__ pal_bg_bright (
    unsigned char bright )
```

### 3.9.2.21 pal\_bright()

```
void __fastcall__ pal_bright (
    unsigned char bright )
```

### 3.9.2.22 pal\_clear()

```
void __fastcall__ pal_clear (
    void )
```

### 3.9.2.23 pal\_col()

```
void __fastcall__ pal_col (
    unsigned char index,
    unsigned char color )
```

### 3.9.2.24 pal\_spr()

```
void __fastcall__ pal_spr (
    const char * data )
```

Here is the caller graph for this function:



### 3.9.2.25 pal\_spr\_bright()

```
void __fastcall__ pal_spr_bright (
    unsigned char bright )
```

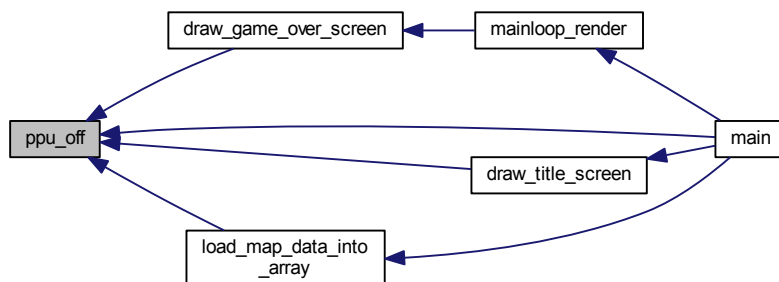
### 3.9.2.26 ppu\_mask()

```
void __fastcall__ ppu_mask (
    unsigned char mask )
```

### 3.9.2.27 ppu\_off()

```
void __fastcall__ ppu_off (
    void )
```

Here is the caller graph for this function:



### 3.9.2.28 ppu\_on\_all()

```
void __fastcall__ ppu_on_all (  
    void )
```

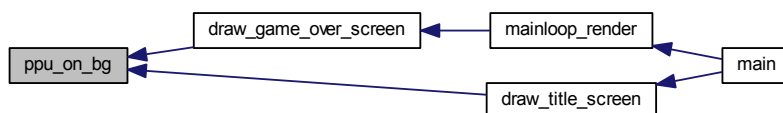
Here is the caller graph for this function:



### 3.9.2.29 ppu\_on\_bg()

```
void __fastcall__ ppu_on_bg (  
    void )
```

Here is the caller graph for this function:



### 3.9.2.30 ppu\_on\_spr()

```
void __fastcall__ ppu_on_spr (  
    void )
```

### 3.9.2.31 ppu\_system()

```
unsigned char __fastcall__ ppu_system (  
    void )
```

### 3.9.2.32 ppu\_wait\_frame()

```
void __fastcall__ ppu_wait_frame (  
    void )
```

### 3.9.2.33 ppu\_wait\_nmi()

```
void __fastcall__ ppu_wait_nmi (
    void )
```

Here is the caller graph for this function:



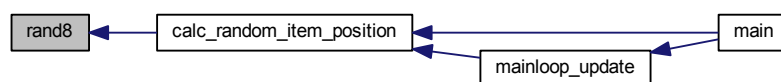
### 3.9.2.34 rand16()

```
unsigned int __fastcall__ rand16 (
    void )
```

### 3.9.2.35 rand8()

```
unsigned char __fastcall__ rand8 (
    void )
```

Here is the caller graph for this function:



### 3.9.2.36 sample\_play()

```
void __fastcall__ sample_play (
    unsigned char sample )
```

### 3.9.2.37 scroll()

```
void __fastcall__ scroll (
    unsigned int x,
    unsigned int y )
```



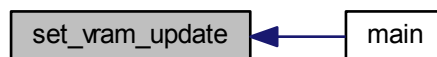
### 3.9.2.38 set\_rand()

```
void __fastcall__ set_rand (
    unsigned int seed )
```

### 3.9.2.39 set\_vram\_update()

```
void __fastcall__ set_vram_update (
    unsigned char * buf )
```

Here is the caller graph for this function:



### 3.9.2.40 sfx\_play()

```
void __fastcall__ sfx_play (
    unsigned char sound,
    unsigned char channel )
```

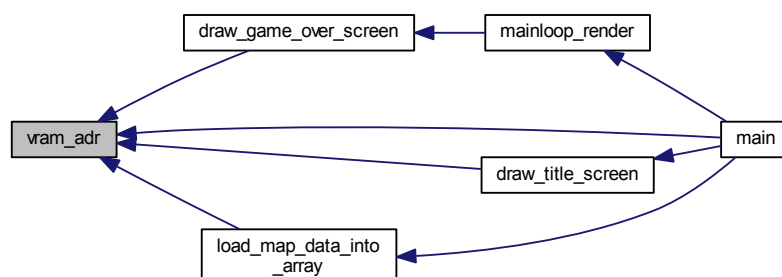
### 3.9.2.41 split()

```
void __fastcall__ split (
    unsigned int x,
    unsigned int y )
```

### 3.9.2.42 vram\_adr()

```
void __fastcall__ vram_adr (
    unsigned int adr )
```

Here is the caller graph for this function:



**3.9.2.43 vram\_fill()**

```
void __fastcall__ vram_fill (
    unsigned char n,
    unsigned int len )
```

**3.9.2.44 vram\_inc()**

```
void __fastcall__ vram_inc (
    unsigned char n )
```

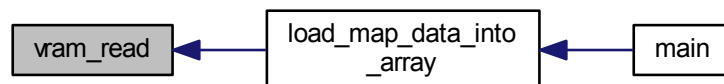
**3.9.2.45 vram\_put()**

```
void __fastcall__ vram_put (
    unsigned char n )
```

**3.9.2.46 vram\_read()**

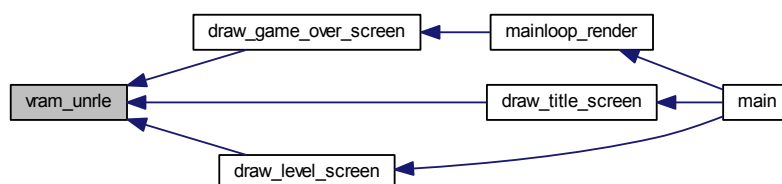
```
void __fastcall__ vram_read (
    unsigned char * dst,
    unsigned int size )
```

Here is the caller graph for this function:

**3.9.2.47 vram\_unrle()**

```
void __fastcall__ vram_unrle (
    const unsigned char * data )
```

Here is the caller graph for this function:



## 3.9.2.48 vram\_write()

```
void __fastcall__ vram_write (
    unsigned char * src,
    unsigned int size )
```

## 3.10 C:/Users/Administrator/Documents/GitHub/NES-Snake/NESLibrary/test\_nam.h File Reference

### Variables

- const unsigned char [test\\_nam](#) [308]

### 3.10.1 Variable Documentation

#### 3.10.1.1 test\_nam

```
const unsigned char test_nam[308]
```

#### Initial value:

```
= {
    0x01, 0x00, 0x01, 0xa3, 0x10, 0x01, 0x04, 0x00, 0x10, 0x01, 0x04, 0x00, 0x10, 0x01, 0x04, 0x00,
    0x10, 0x01, 0x04, 0x00, 0x01, 0x0a, 0x10, 0x00, 0x01, 0x02, 0x10, 0x00, 0x01, 0x04, 0x10, 0x00,
    0x01, 0x06, 0x10, 0x00, 0x01, 0x0c, 0x10, 0x00, 0x01, 0x02, 0x10, 0x01, 0x02, 0x00, 0x01, 0x02,
    0x10, 0x01, 0x04, 0x00, 0x01, 0x02, 0x10, 0x00, 0x01, 0x0c, 0x10, 0x00, 0x01, 0x02, 0x10, 0x00,
    0x01, 0x08, 0x10, 0x00, 0x01, 0x02, 0x10, 0x00, 0x01, 0x0c, 0x10, 0x00, 0x01, 0x02, 0x10, 0x01,
    0x04, 0x00, 0x10, 0x01, 0x04, 0x00, 0x01, 0x02, 0x10, 0x00, 0x01, 0x42, 0x10, 0x00, 0x01, 0x06,
    0x10, 0x01, 0x04, 0x00, 0x10, 0x00, 0x01, 0x04, 0x10, 0x01, 0x04, 0x00, 0x10, 0x00, 0x01, 0x04,
    0x10, 0x00, 0x01, 0x06, 0x10, 0x00, 0x01, 0x02, 0x10, 0x00, 0x10, 0x00, 0x01, 0x04, 0x10, 0x00,
    0x01, 0x06, 0x10, 0x01, 0x04, 0x00, 0x01, 0x06, 0x10, 0x00, 0x01, 0x02, 0x10, 0x00, 0x10, 0x00,
    0x01, 0x04, 0x10, 0x01, 0x02, 0x00, 0x01, 0x04, 0x10, 0x00, 0x01, 0x02, 0x10, 0x00, 0x01, 0x06,
    0x10, 0x01, 0x03, 0x00, 0x00, 0x10, 0x00, 0x01, 0x04, 0x10, 0x00, 0x01, 0x06, 0x10, 0x00, 0x01,
    0x02, 0x10, 0x00, 0x01, 0x06, 0x10, 0x00, 0x01, 0x02, 0x10, 0x00, 0x10, 0x01, 0x04, 0x00, 0x10,
    0x01, 0x04, 0x00, 0x01, 0x02, 0x10, 0x01, 0x04, 0x00, 0x01, 0x46, 0x10, 0x00, 0x01, 0x02, 0x10,
    0x00, 0x10, 0x01, 0x04, 0x00, 0x10, 0x01, 0x04, 0x00, 0x01, 0x0e, 0x10, 0x10, 0x00, 0x10, 0x10,
    0x00, 0x10, 0x00, 0x01, 0x02, 0x10, 0x00, 0x10, 0x00, 0x01, 0x02, 0x10, 0x00, 0x01, 0x0e, 0x10,
    0x00, 0x10, 0x00, 0x10, 0x00, 0x00, 0x01, 0x02, 0x10, 0x00, 0x10, 0x00, 0x01, 0x02, 0x10,
    0x00, 0x01, 0x0e, 0x10, 0x00, 0x01, 0x02, 0x10, 0x00, 0x10, 0x01, 0x04, 0x00, 0x10, 0x01, 0x04,
    0x00, 0x01, 0x0e, 0x10, 0x00, 0x01, 0x02, 0x10, 0x00, 0x10, 0x00, 0x01, 0x02, 0x10, 0x00, 0x10,
    0x00, 0x01, 0xde, 0x50, 0x01, 0x07, 0x55, 0x01, 0x07, 0xaa, 0x01, 0x07, 0xaa, 0x01, 0x0f, 0x0a,
    0x01, 0x07, 0x01, 0x00
}
```

## 3.11 C:/Users/Administrator/Documents/GitHub/NES-Snake/README.md File Reference

## 3.12 C:/Users/Administrator/Documents/GitHub/NES-Snake/src/definitions.h File Reference

This header files contains defines all global variables and constants, macros and includes of nametable and palette definition.

```
#include "level1_nam.h"
#include "level2_nam.h"
#include "game_over_nam.h"
#include "titlescreen_nam.h"
#include "levels_pal.h"
#include "sprites_pal.h"
#include "menue_pal.h"
```

## Macros

- #define `LEVELS_ALL` 5
- #define `SNAKE_MAX_SIZE` 100
- #define `NAMETABLE1_START` 0x2000

### Tile-based width and height of the level map

- #define `MAP_WIDTH` 32
- #define `MAP_HEIGHT` 30

### Direction constants

- #define `DIR_UP` 1
- #define `DIR_DOWN` 2
- #define `DIR_LEFT` 3
- #define `DIR_RIGHT` 4

### Tile constants

- #define `WALL_TILE_1` 0x43
- #define `WALL_TILE_2` 0x44
- #define `SNAKE_HEAD_TILE_VERT` 0x41
- #define `SNAKE_HEAD_TILE_HORZ` 0x42
- #define `SNAKE_BODY_TILE` 0x40
- #define `EMPTY_TILE` 0x00
- #define `SPIDER_TILE` 0x45
- #define `DIGIT_O_TILE` 0x10

### Macros for more efficient calculations

- #define `MAPARRAY_ADR(x, y)` ((y<<2)|(x>>3))

## Variables

### Global variables, used to interact with the level map

- static unsigned char `map` [`MAP_WIDTH` \*`MAP_HEIGHT`]
- static unsigned char `nameRow` [`MAP_WIDTH`]
- static unsigned int `nametable_fetch`

### Global variables, used to interact with the snake

- static unsigned char `body_coordinates` [`SNAKE_MAX_SIZE`<< 1]
- static unsigned char `size_index`
- static unsigned char `snake_head_attribute`
- static unsigned char `snake_head_tile`
- static unsigned char `speed_counter`
- static unsigned char `direction`

### Pixel based coordinates of snake's head sprite.

- static unsigned char `snake_x`
- static unsigned char `snake_y`

**Global variables, which are used to calculate pixel based coordinates (of body elements) to tile based coordinates.**

- static unsigned char `body_tile_x`
- static unsigned char `body_tile_y`

**Pixel based coordinates of the last body element from last frame.**

- static unsigned char `last_body_pixel_x`
- static unsigned char `last_body_pixel_y`

**Global variables, used to modify the background ingame**

- static unsigned char `update_list` [5 \*3+1]
- static unsigned char \* `ul`

**Global variables, used for rendering sprites ingame**

- static unsigned char `sprite_offset`

**Global variables, used for universal purpose e.g loops**

- static unsigned char `i`
- static unsigned char `j`
- static unsigned int `k`
- static unsigned int `l`

**Global variables, used to interact with items**

- static unsigned char `item_x`
- static unsigned char `item_y`

**Global variables, used for game-states, menues, input**

- static unsigned char `current_level`
- static unsigned char `pause`
- static unsigned char `pause_loop`
- static unsigned char `gameover`
- static unsigned char `gameover_loop`
- static unsigned char `titlescreen`
- static unsigned char `restart`
- static unsigned char `input`

**List of the levels, include pointer to the packed nametable of the levels, menues, and pointer to the associated palette.**

- const unsigned char \*const `levelList` [LEVELS\_ALL+2+2]

### 3.12.1 Detailed Description

This header files contains defines all global variables and constants, macros and includes of nametable and palette definition.

Author

Sebastian Dine

### 3.12.2 Macro Definition Documentation

#### 3.12.2.1 DIGIT\_O\_TILE

```
#define DIGIT_O_TILE 0x10
```

Tile of digit 0 (zero)

#### 3.12.2.2 DIR\_DOWN

```
#define DIR_DOWN 2
```

#### 3.12.2.3 DIR\_LEFT

```
#define DIR_LEFT 3
```

#### 3.12.2.4 DIR\_RIGHT

```
#define DIR_RIGHT 4
```

#### 3.12.2.5 DIR\_UP

```
#define DIR_UP 1
```

#### 3.12.2.6 EMPTY\_TILE

```
#define EMPTY_TILE 0x00
```

Tile of empty space

#### 3.12.2.7 LEVELS\_ALL

```
#define LEVELS_ALL 5
```

Total number of level maps (ingame background nametables)

#### 3.12.2.8 MAP\_HEIGHT

```
#define MAP_HEIGHT 30
```

#### 3.12.2.9 MAP\_WIDTH

```
#define MAP_WIDTH 32
```

### 3.12.2.10 MAPARRAY\_ADR

```
#define MAPARRAY_ADR(  
    x,  
    y ) ((y<<2)|(x>>3))
```

Macro for calculating in which tile of the 32\*30 tiles the given position is placed. Optimized with bitshifting, arithmetic pendant is  $((y/8)*32)+(x/8)$ . x and y are assumed to be Sprite-coordinates (not Tile-coordinates).

### 3.12.2.11 NAMETABLE1\_START

```
#define NAMETABLE1_START 0x2000
```

Start address in VRAM for first nametable

### 3.12.2.12 SNAKE\_BODY\_TILE

```
#define SNAKE_BODY_TILE 0x40
```

Tile of snake body element

### 3.12.2.13 SNAKE\_HEAD\_TILE\_HORZ

```
#define SNAKE_HEAD_TILE_HORZ 0x42
```

Tile of horizontal snake head element

### 3.12.2.14 SNAKE\_HEAD\_TILE\_VERT

```
#define SNAKE_HEAD_TILE_VERT 0x41
```

Tile of vertical snake head element

### 3.12.2.15 SNAKE\_MAX\_SIZE

```
#define SNAKE_MAX_SIZE 100
```

Number of body elements, the snake can get.

### 3.12.2.16 SPIDER\_TILE

```
#define SPIDER_TILE 0x45
```

Tile of spider item

#### 3.12.2.17 WALL\_TILE\_1

```
#define WALL_TILE_1 0x43
```

Tile of horizontal wall element

#### 3.12.2.18 WALL\_TILE\_2

```
#define WALL_TILE_2 0x44
```

Tile of vertical wall element

### 3.12.3 Variable Documentation

#### 3.12.3.1 body\_coordinates

```
unsigned char body_coordinates[SNAKE_MAX_SIZE<< 1] [static]
```

Array of snakes body-coordinates (pixel-based), two elements are a coordinate set, eg. body[0] is the x-coordinate of the first body-element and body[1] its y-coordinate.

#### 3.12.3.2 body\_tile\_x

```
unsigned char body_tile_x [static]
```

#### 3.12.3.3 body\_tile\_y

```
unsigned char body_tile_y [static]
```

#### 3.12.3.4 current\_level

```
unsigned char current_level [static]
```

Global variable, indicating the current level.

#### 3.12.3.5 direction

```
unsigned char direction [static]
```

Global variable, indicating to which direction the snake is moving. 1=up,2=down,3=left,4=right.

#### 3.12.3.6 gameover

```
unsigned char gameover [static]
```

Global variable, indicating the game over mode (1= game over 0= no game over).



### 3.12.3.7 gameover\_loop

```
unsigned char gameover_loop [static]
```

identifier to check, if first gameover loop is passed (1= true, 0= false).

### 3.12.3.8 i

```
unsigned char i [static]
```

### 3.12.3.9 input

```
unsigned char input [static]
```

Global variable, holding the controller input of the current frame

### 3.12.3.10 item\_x

```
unsigned char item_x [static]
```

### 3.12.3.11 item\_y

```
unsigned char item_y [static]
```

### 3.12.3.12 j

```
unsigned char j [static]
```

### 3.12.3.13 k

```
unsigned int k [static]
```

### 3.12.3.14 l

```
unsigned int l [static]
```

### 3.12.3.15 last\_body\_pixel\_x

```
unsigned char last_body_pixel_x [static]
```

### 3.12.3.16 last\_body\_pixel\_y

```
unsigned char last_body_pixel_y [static]
```

### 3.12.3.17 levelList

```
const unsigned char* const levelList[LEVELS_ALL+2+2]
```

**Initial value:**

```
={  
    level1_nam, level2_nam,  
    game_over_nam, titlescreen_nam,  
    levels_pal, menue_pal  
}
```

### 3.12.3.18 map

```
unsigned char map[MAP_WIDTH *MAP_HEIGHT] [static]
```

Array of the complete game map (tile-based).

### 3.12.3.19 nameRow

```
unsigned char nameRow[MAP_WIDTH] [static]
```

Array for fetching nametable into array 'map', row by row.

### 3.12.3.20 nametable\_fetch

```
unsigned int nametable_fetch [static]
```

Variable for fetching through nametable.

### 3.12.3.21 pause

```
unsigned char pause [static]
```

Global variable, indicating the pause mode (1= pause, 0= no pause).

### 3.12.3.22 pause\_loop

```
unsigned char pause_loop [static]
```

Identifier to check, if first pause-loop is passed (1= true, 0= false).

#### 3.12.3.23 restart

```
unsigned char restart [static]
```

Global variable, for handling the restart input

#### 3.12.3.24 size\_index

```
unsigned char size_index [static]
```

Index for array 'body\_coordinates' which points to the space for the next body-element to add. It will be increased in +=2-steps so it always points to a free x-coordinate.

#### 3.12.3.25 snake\_head\_attribute

```
unsigned char snake_head_attribute [static]
```

Global variable for holding attributes of the head sprite of the snake

#### 3.12.3.26 snake\_head\_tile

```
unsigned char snake_head_tile [static]
```

#### 3.12.3.27 snake\_x

```
unsigned char snake_x [static]
```

#### 3.12.3.28 snake\_y

```
unsigned char snake_y [static]
```

#### 3.12.3.29 speed\_counter

```
unsigned char speed_counter [static]
```

#### 3.12.3.30 sprite\_offset

```
unsigned char sprite_offset [static]
```

#### 3.12.3.31 titlescreen

```
unsigned char titlescreen [static]
```

Global variable, indicating the titlescreen mode (1=titlescreen 0= no titlescreen).

### 3.12.3.32 ul

```
unsigned char* ul [static]
```

### 3.12.3.33 update\_list

```
unsigned char update_list[5 *3+1] [static]
```

## 3.13 C:/Users/Administrator/Documents/GitHub/NES-Snake/src/input.c File Reference

This file contains functions for input handling from a controller.

### Functions

- void [input\\_btn\\_start](#) (void)
- void [mainloop\\_handle\\_input](#) (void)

### 3.13.1 Detailed Description

This file contains functions for input handling from a controller.

#### Author

Sebastian Dine

### 3.13.2 Function Documentation

#### 3.13.2.1 input\_btn\_start()

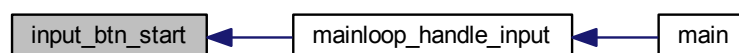
```
void input_btn_start (  
    void )
```

This function contains the logic for the START button according to different scenarios e.g. title screen, ingame, gameover.

#### Author

Sebastian Dine

Here is the caller graph for this function:



## 3.13.2.2 mainloop\_handle\_input()

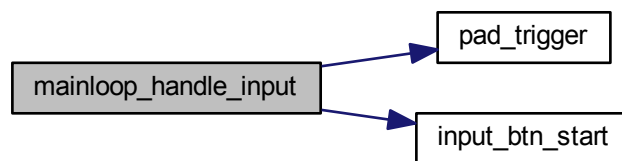
```
void mainloop_handle_input (
    void )
```

This function provides the main input handling functionalities for an controller on port 1. It contains logic for input of the following buttons: UP, DOWN, LEFT, RIGHT, START.

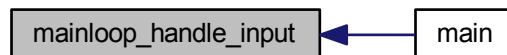
**Author**

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



## 3.14 C:/Users/Administrator/Documents/GitHub/NES-Snake/src/render.c File Reference

This file contains all functionality to draw onto the screen, either as sprites or as background tiles.

**Functions**

- void [draw\\_snake](#) (void)
- void [draw\\_item](#) (void)
- void [draw\\_score](#) (void)
- void [init\\_updateList](#) (void)
- void [center\\_score\\_when\\_gameover](#) (void)
- void [draw\\_game\\_over\\_screen](#) (void)
- void [draw\\_title\\_screen](#) (void)
- void [draw\\_pause\\_screen](#) (void)
- void [draw\\_level\\_screen](#) (void)
- void [mainloop\\_render](#) (void)

### 3.14.1 Detailed Description

This file contains all functionality to draw onto the screen, either as sprites or as background tiles.

#### Author

Sebastian Dine

### 3.14.2 Function Documentation

#### 3.14.2.1 center\_score\_when\_gameover()

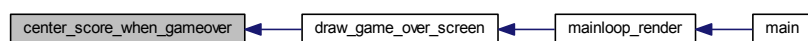
```
void center_score_when_gameover (
    void )
```

This function moves the rendering of the score from the upper left corner to the center of the screen.

#### Author

Sebastian Dine

Here is the caller graph for this function:



#### 3.14.2.2 draw\_game\_over\_screen()

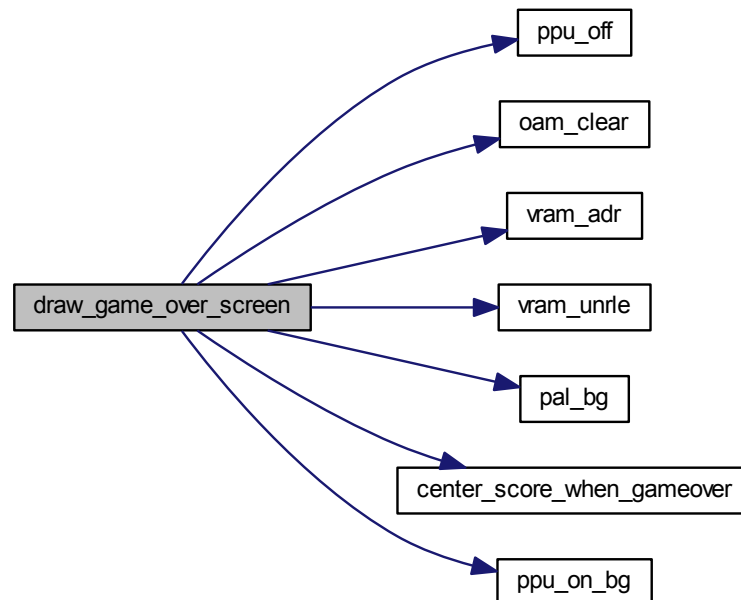
```
void draw_game_over_screen (
    void )
```

This function draws the gameover screen.

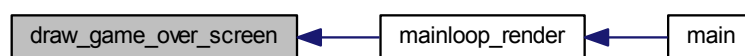
## Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



### 3.14.2.3 draw\_item()

```
void draw_item (
    void )
```

This function draws an element as a sprite to the screen.

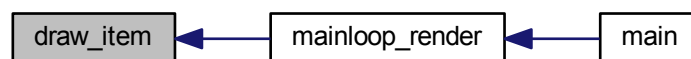
**Author**

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:

**3.14.2.4 draw\_level\_screen()**

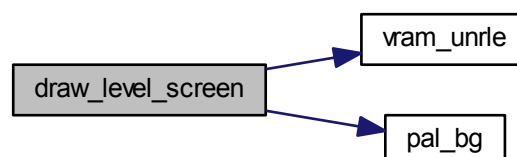
```
void draw_level_screen (  
    void )
```

This function draws the background of the current level to the screen.

**Author**

Sebastian Dine

Here is the call graph for this function:





Here is the caller graph for this function:



#### 3.14.2.5 draw\_pause\_screen()

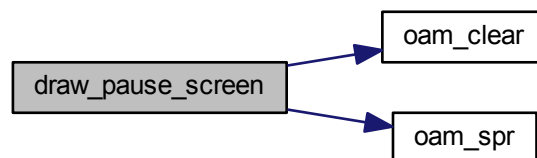
```
void draw_pause_screen (  
    void )
```

This function draws the letters PAUSE as sprites to the center of the screen, if the game is paused.

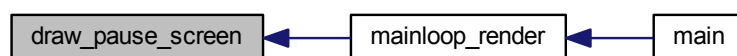
##### Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



### 3.14.2.6 draw\_score()

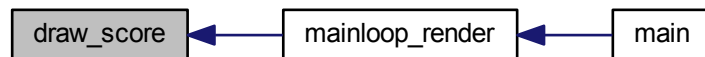
```
void draw_score (
    void )
```

This function draws the current score as background tiles to the screen.

#### Author

Sebastian Dine

Here is the caller graph for this function:



### 3.14.2.7 draw\_snake()

```
void draw_snake (
    void )
```

This function draws the whole snake. The head will be drawn as a sprite, the body elements as background tiles.

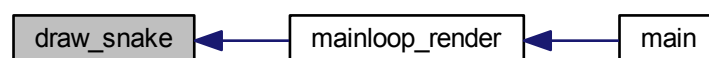
#### Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



## 3.14.2.8 draw\_title\_screen()

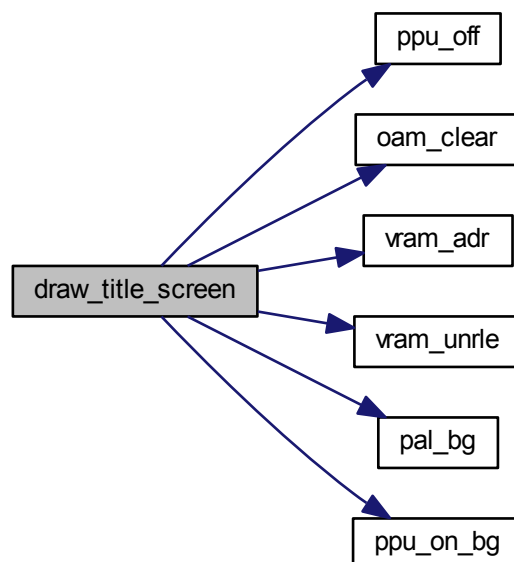
```
void draw_title_screen (  
    void )
```

This function draws the title screen.

**Author**

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



### 3.14.2.9 init\_updateList()

```
void init_updateList (  
    void )
```

This function initializes the (background tile) update-list with score-elements (zero-digits) and the EOF-indicator.

#### Author

Sebastian Dine

Here is the caller graph for this function:



### 3.14.2.10 mainloop\_render()

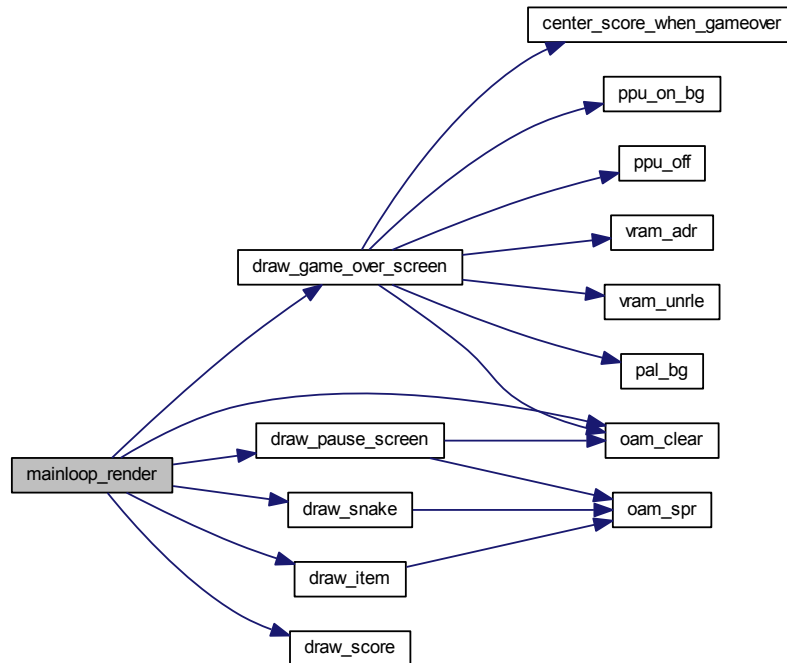
```
void mainloop_render (  
    void )
```

This function provides the coordination of all render routines according to the current status of the game, once per frame.

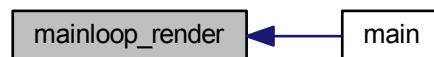
## Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



### 3.15 C:/Users/Administrator/Documents/GitHub/NES-Snake/src/snake.c File Reference

Maingame file, containing the main game loop.

```

#include "neslib.h"
#include "definitions.h"
#include "input.c"
#include "update.c"
#include "render.c"

```

## Functions

- void `main` (void)

*Main game loop.*

### 3.15.1 Detailed Description

Maingame file, containing the main game loop.

#### Author

Sebastian Dine.

### 3.15.2 Function Documentation

#### 3.15.2.1 `main()`

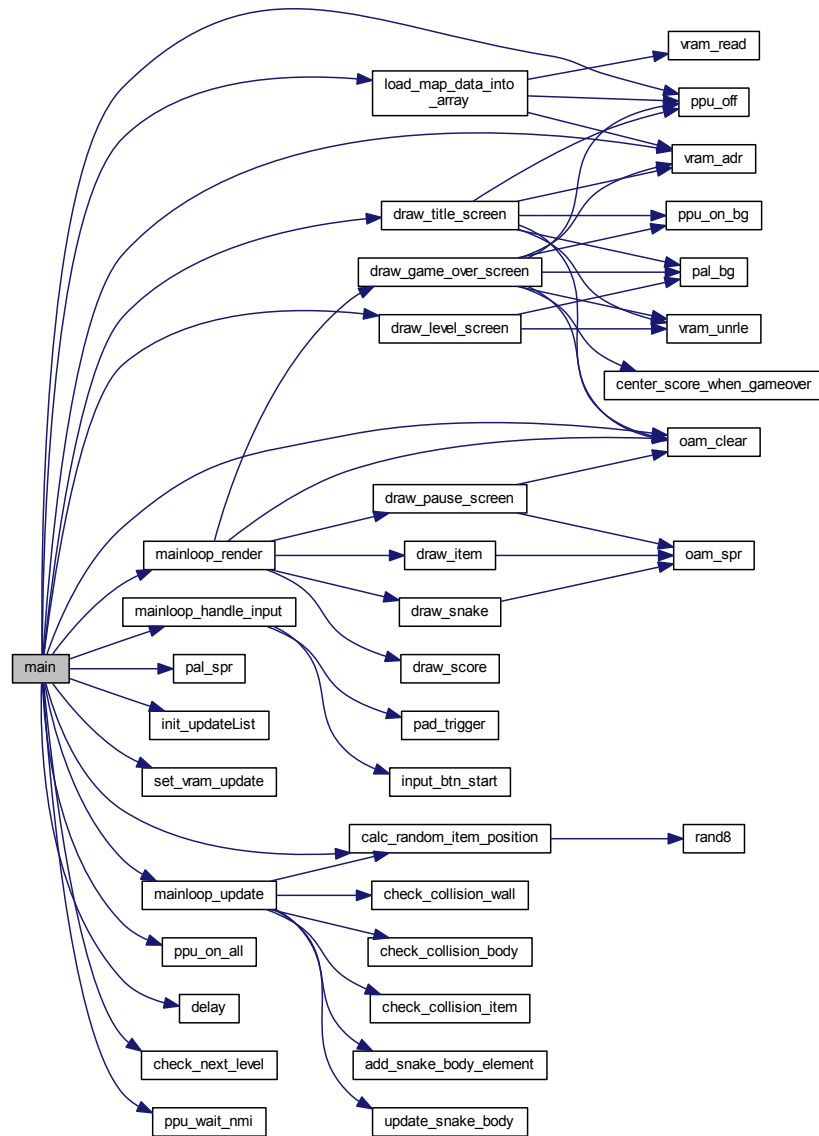
```
void main (
    void )
```

Main game loop.

## Author

Sebastian Dine

Here is the call graph for this function:



## 3.16 C:/Users/Administrator/Documents/GitHub/NES-Snake/src/update.c File Reference

This file contains all ingame logic functionalities and utility functionalities.

### Functions

- void [load\\_map\\_data\\_into\\_array](#) (void)

- void [calc\\_random\\_item\\_position](#) (void)
- void [update\\_snake\\_body](#) ()
- void [add\\_snake\\_body\\_element](#) ()
- unsigned char [check\\_collision\\_wall](#) (void)
- unsigned char [check\\_collision\\_body](#) (void)
- unsigned char [check\\_collision\\_item](#) (void)
- unsigned char [check\\_next\\_level](#) (void)
- void [mainloop\\_update](#) (void)

### 3.16.1 Detailed Description

This file contains all ingame logic functionalities and utility functionalities.

#### Author

Sebastian Dine

### 3.16.2 Function Documentation

#### 3.16.2.1 [add\\_snake\\_body\\_element\(\)](#)

```
void add_snake_body_element ( )
```

This function adds a new pair of body element coordinates to global array 'body\_coordinates'.

#### Author

Sebastian Dine

Here is the caller graph for this function:





### 3.16.2.2 calc\_random\_item\_position()

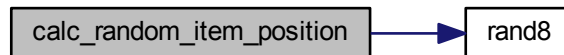
```
void calc_random_item_position (  
    void )
```

This function calculates the coordinates of an grow-item. It stores the calculated coordinates into global fields 'item\_x' and 'item\_y'.

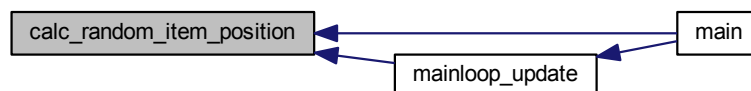
#### Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



### 3.16.2.3 check\_collision\_body()

```
unsigned char check_collision_body (  
    void )
```

Collision detection of snakes' head-sprite with body-tiles.

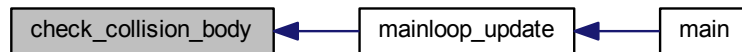
#### Returns

1 = collision with body element, 0 = no collision with body element

**Author**

Sebastian Dine

Here is the caller graph for this function:

**3.16.2.4 check\_collision\_item()**

```
unsigned char check_collision_item (  
    void )
```

Collision detection of snakes' head-sprite with an item-sprite.

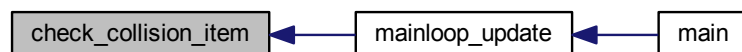
**Returns**

1 = collision with item sprite, 0 = no collision with item sprite

**Author**

Sebastian Dine

Here is the caller graph for this function:

**3.16.2.5 check\_collision\_wall()**

```
unsigned char check_collision_wall (  
    void )
```

Collision detection of snakes' head-sprite with wall-tiles.

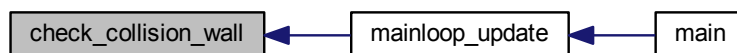
**Returns**

1 = collision with wall element, 0 = no collision with wall sprite

**Author**

Sebastian Dine

Here is the caller graph for this function:

**3.16.2.6 check\_next\_level()**

```
unsigned char check_next_level (  
    void )
```

Check, if the requirements for the next level are met.

**Returns**

1 = next level is reached, 0 = next level is not reached

**Author**

Sebastian Dine

Here is the caller graph for this function:



### 3.16.2.7 load\_map\_data\_into\_array()

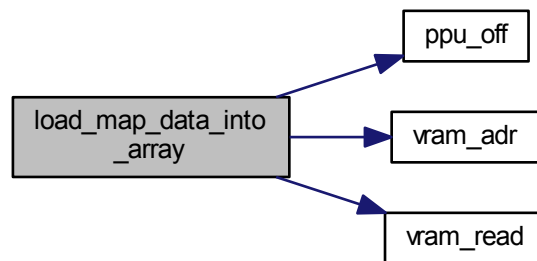
```
void load_map_data_into_array (  
    void )
```

This file reads the namespace into global array 'map', which is used for further calculations, e.g. collision detection.

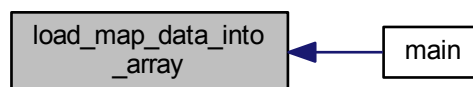
#### Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



### 3.16.2.8 mainloop\_update()

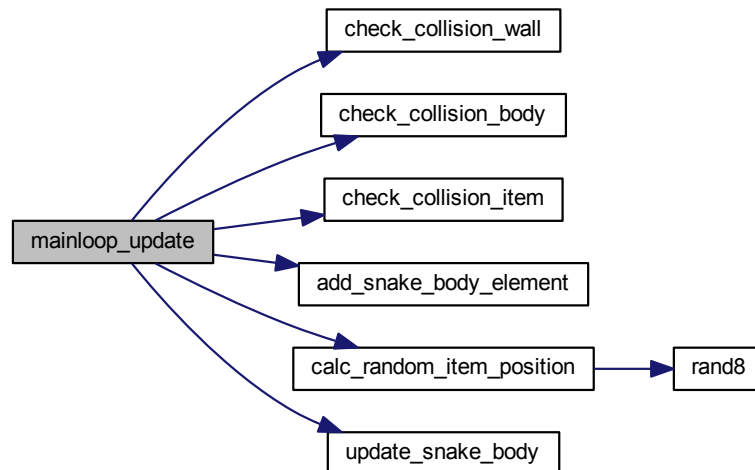
```
void mainloop_update (  
    void )
```

This function provides the coordination of all ingame logic routines, once per frame.

Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



### 3.16.2.9 update\_snake\_body()

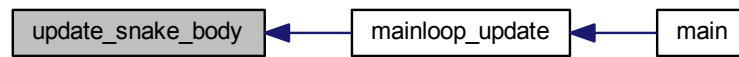
```
void update_snake_body ( )
```

This function updates the body coordinates of the snake in order to simulate its movement.

## Author

Sebastian Dine

Here is the caller graph for this function:



# Index

add\_snake\_body\_element  
update.c, [46](#)

bank\_bg  
neslib.h, [15](#)  
bank\_spr  
neslib.h, [15](#)  
bgsplit\_nam  
bgsplit\_nam.h, [10](#)  
bgsplit\_nam.h  
bgsplit\_nam, [10](#)  
body\_coordinates  
definitions.h, [30](#)  
body\_tile\_x  
definitions.h, [30](#)  
body\_tile\_y  
definitions.h, [30](#)

C:/Users/Administrator/Documents/GitHub/NES-  
Snake/NESLibrary/bgsplit\_nam.h, [10](#)  
C:/Users/Administrator/Documents/GitHub/NES-  
Snake/NESLibrary/neslib.h, [10](#)  
C:/Users/Administrator/Documents/GitHub/NES-  
Snake/NESLibrary/test\_nam.h, [25](#)  
C:/Users/Administrator/Documents/GitHub/NES-  
Snake/README.md, [25](#)  
C:/Users/Administrator/Documents/GitHub/NES-  
Snake/gfx/game\_over\_nam.h, [5](#)  
C:/Users/Administrator/Documents/GitHub/NES-  
Snake/gfx/level1\_nam.h, [6](#)  
C:/Users/Administrator/Documents/GitHub/NES-  
Snake/gfx/level2\_nam.h, [6](#)  
C:/Users/Administrator/Documents/GitHub/NES-  
Snake/gfx/levels\_pal.h, [7](#)  
C:/Users/Administrator/Documents/GitHub/NES-  
Snake/gfx/menue\_pal.h, [8](#)  
C:/Users/Administrator/Documents/GitHub/NES-  
Snake/gfx/sprites\_pal.h, [8](#)  
C:/Users/Administrator/Documents/GitHub/NES-  
Snake/gfx/titlescreen\_nam.h, [9](#)  
C:/Users/Administrator/Documents/GitHub/NES-  
Snake/src/definitions.h, [25](#)  
C:/Users/Administrator/Documents/GitHub/NES-  
Snake/src/input.c, [34](#)  
C:/Users/Administrator/Documents/GitHub/NES-  
Snake/src/render.c, [35](#)  
C:/Users/Administrator/Documents/GitHub/NES-  
Snake/src/snake.c, [43](#)  
C:/Users/Administrator/Documents/GitHub/NES-  
Snake/src/update.c, [45](#)

calc\_random\_item\_position  
update.c, [46](#)  
center\_score\_when\_gameover  
render.c, [36](#)  
check\_collision\_body  
update.c, [47](#)  
check\_collision\_item  
update.c, [48](#)  
check\_collision\_wall  
update.c, [48](#)  
check\_next\_level  
update.c, [49](#)  
current\_level  
definitions.h, [30](#)

DIGIT\_O\_TILE  
definitions.h, [28](#)  
DIR\_DOWN  
definitions.h, [28](#)  
DIR\_LEFT  
definitions.h, [28](#)  
DIR\_RIGHT  
definitions.h, [28](#)  
DIR\_UP  
definitions.h, [28](#)  
definitions.h  
body\_coordinates, [30](#)  
body\_tile\_x, [30](#)  
body\_tile\_y, [30](#)  
current\_level, [30](#)  
DIGIT\_O\_TILE, [28](#)  
DIR\_DOWN, [28](#)  
DIR\_LEFT, [28](#)  
DIR\_RIGHT, [28](#)  
DIR\_UP, [28](#)  
direction, [30](#)  
EMPTY\_TILE, [28](#)  
gameover, [30](#)  
gameover\_loop, [30](#)  
i, [31](#)  
input, [31](#)  
item\_x, [31](#)  
item\_y, [31](#)  
j, [31](#)  
k, [31](#)  
l, [31](#)  
LEVELS\_ALL, [28](#)  
last\_body\_pixel\_x, [31](#)  
last\_body\_pixel\_y, [31](#)  
levelList, [32](#)

- MAP\_HEIGHT, [28](#)
- MAP\_WIDTH, [28](#)
- MAPARRAY\_ADR, [28](#)
- map, [32](#)
- NAMETABLE1\_START, [29](#)
- nameRow, [32](#)
- nametable\_fetch, [32](#)
- pause, [32](#)
- pause\_loop, [32](#)
- restart, [32](#)
- SNAKE\_BODY\_TILE, [29](#)
- SNAKE\_HEAD\_TILE\_HORZ, [29](#)
- SNAKE\_HEAD\_TILE\_VERT, [29](#)
- SNAKE\_MAX\_SIZE, [29](#)
- SPIDER\_TILE, [29](#)
- size\_index, [33](#)
- snake\_head\_attribute, [33](#)
- snake\_head\_tile, [33](#)
- snake\_x, [33](#)
- snake\_y, [33](#)
- speed\_counter, [33](#)
- sprite\_offset, [33](#)
- titlescreen, [33](#)
- ul, [33](#)
- update\_list, [34](#)
- WALL\_TILE\_1, [29](#)
- WALL\_TILE\_2, [30](#)
- delay
  - neslib.h, [16](#)
- direction
  - definitions.h, [30](#)
- draw\_game\_over\_screen
  - render.c, [36](#)
- draw\_item
  - render.c, [37](#)
- draw\_level\_screen
  - render.c, [38](#)
- draw\_pause\_screen
  - render.c, [39](#)
- draw\_score
  - render.c, [39](#)
- draw\_snake
  - render.c, [40](#)
- draw\_title\_screen
  - render.c, [40](#)
- EMPTY\_TILE
  - definitions.h, [28](#)
- FALSE
  - neslib.h, [12](#)
- flush\_vram\_update
  - neslib.h, [16](#)
- game\_over\_nam
  - game\_over\_nam.h, [5](#)
- game\_over\_nam.h
  - game\_over\_nam, [5](#)
- gameover
  - definitions.h, [30](#)
- gameover\_loop
  - definitions.h, [30](#)
- i
  - definitions.h, [31](#)
- init\_updateList
  - render.c, [41](#)
- input
  - definitions.h, [31](#)
- input.c
  - input\_btn\_start, [34](#)
  - mainloop\_handle\_input, [34](#)
- input\_btn\_start
  - input.c, [34](#)
- item\_x
  - definitions.h, [31](#)
- item\_y
  - definitions.h, [31](#)
- j
  - definitions.h, [31](#)
- k
  - definitions.h, [31](#)
- l
  - definitions.h, [31](#)
- LEVELS\_ALL
  - definitions.h, [28](#)
- last\_body\_pixel\_x
  - definitions.h, [31](#)
- last\_body\_pixel\_y
  - definitions.h, [31](#)
- level1\_nam
  - level1\_nam.h, [6](#)
- level1\_nam.h
  - level1\_nam, [6](#)
- level2\_nam
  - level2\_nam.h, [7](#)
- level2\_nam.h
  - level2\_nam, [7](#)
- levelList
  - definitions.h, [32](#)
- levels\_pal
  - levels\_pal.h, [7](#)
- levels\_pal.h
  - levels\_pal, [7](#)
- load\_map\_data\_into\_array
  - update.c, [49](#)
- MAP\_HEIGHT
  - definitions.h, [28](#)
- MAP\_WIDTH
  - definitions.h, [28](#)
- MAPARRAY\_ADR
  - definitions.h, [28](#)
- MASK\_BG
  - neslib.h, [12](#)



MASK\_EDGE\_BG  
    neslib.h, 12  
MASK\_EDGE\_SPR  
    neslib.h, 12  
MASK\_SPR  
    neslib.h, 12  
MAX  
    neslib.h, 12  
MIN  
    neslib.h, 12  
MSB  
    neslib.h, 13  
main  
    snake.c, 44  
mainloop\_handle\_input  
    input.c, 34  
mainloop\_render  
    render.c, 42  
mainloop\_update  
    update.c, 50  
map  
    definitions.h, 32  
memcpy  
    neslib.h, 16  
memfill  
    neslib.h, 16  
menue\_pal  
    menue\_pal.h, 8  
menue\_pal.h  
    menue\_pal, 8  
music\_pause  
    neslib.h, 16  
music\_play  
    neslib.h, 16  
music\_stop  
    neslib.h, 17  
  
NAMETABLE1\_START  
    definitions.h, 29  
NAMETABLE\_A  
    neslib.h, 13  
NAMETABLE\_B  
    neslib.h, 13  
NAMETABLE\_C  
    neslib.h, 13  
NAMETABLE\_D  
    neslib.h, 13  
NT\_UPD\_EOF  
    neslib.h, 13  
NT\_UPD\_HORZ  
    neslib.h, 13  
NT\_UPD\_VERT  
    neslib.h, 13  
NTADR\_A  
    neslib.h, 13  
NTADR\_B  
    neslib.h, 14  
NTADR\_C  
    neslib.h, 14  
NTADR\_D  
    neslib.h, 14  
NULL  
    neslib.h, 14  
nameRow  
    definitions.h, 32  
nametable\_fetch  
    definitions.h, 32  
neslib.h  
    bank\_bg, 15  
    bank\_spr, 15  
    delay, 16  
    FALSE, 12  
    flush\_vram\_update, 16  
    MASK\_BG, 12  
    MASK\_EDGE\_BG, 12  
    MASK\_EDGE\_SPR, 12  
    MASK\_SPR, 12  
    MAX, 12  
    MIN, 12  
    MSB, 13  
    memcpy, 16  
    memfill, 16  
    music\_pause, 16  
    music\_play, 16  
    music\_stop, 17  
    NAMETABLE\_A, 13  
    NAMETABLE\_B, 13  
    NAMETABLE\_C, 13  
    NAMETABLE\_D, 13  
    NT\_UPD\_EOF, 13  
    NT\_UPD\_HORZ, 13  
    NT\_UPD\_VERT, 13  
    NTADR\_A, 13  
    NTADR\_B, 14  
    NTADR\_C, 14  
    NTADR\_D, 14  
    NULL, 14  
    OAM\_BEHIND, 14  
    OAM\_FLIP\_H, 14  
    OAM\_FLIP\_V, 14  
    oam\_clear, 17  
    oam\_hide\_rest, 17  
    oam\_meta\_spr, 17  
    oam\_size, 17  
    oam\_spr, 18  
    PAD\_DOWN, 15  
    PAD\_LEFT, 15  
    PAD\_RIGHT, 15  
    PAD\_SELECT, 15  
    PAD\_START, 15  
    PAD\_UP, 15  
    PAD\_A, 14  
    PAD\_B, 14  
    pad\_poll, 18  
    pad\_state, 18  
    pad\_trigger, 18  
    pal\_all, 18

- pal\_bg, [19](#)
- pal\_bg\_bright, [19](#)
- pal\_bright, [19](#)
- pal\_clear, [19](#)
- pal\_col, [19](#)
- pal\_spr, [19](#)
- pal\_spr\_bright, [20](#)
- ppu\_mask, [20](#)
- ppu\_off, [20](#)
- ppu\_on\_all, [20](#)
- ppu\_on\_bg, [21](#)
- ppu\_on\_spr, [21](#)
- ppu\_system, [21](#)
- ppu\_wait\_frame, [21](#)
- ppu\_wait\_nmi, [21](#)
- rand16, [22](#)
- rand8, [22](#)
- sample\_play, [22](#)
- scroll, [22](#)
- set\_rand, [22](#)
- set\_vram\_update, [23](#)
- sfx\_play, [23](#)
- split, [23](#)
- TRUE, [15](#)
- vram\_adr, [23](#)
- vram\_fill, [23](#)
- vram\_inc, [24](#)
- vram\_put, [24](#)
- vram\_read, [24](#)
- vram\_unrle, [24](#)
- vram\_write, [24](#)
- OAM\_BEHIND
  - neslib.h, [14](#)
- OAM\_FLIP\_H
  - neslib.h, [14](#)
- OAM\_FLIP\_V
  - neslib.h, [14](#)
- oam\_clear
  - neslib.h, [17](#)
- oam\_hide\_rest
  - neslib.h, [17](#)
- oam\_meta\_spr
  - neslib.h, [17](#)
- oam\_size
  - neslib.h, [17](#)
- oam\_spr
  - neslib.h, [18](#)
- PAD\_DOWN
  - neslib.h, [15](#)
- PAD\_LEFT
  - neslib.h, [15](#)
- PAD\_RIGHT
  - neslib.h, [15](#)
- PAD\_SELECT
  - neslib.h, [15](#)
- PAD\_START
  - neslib.h, [15](#)
- PAD\_UP
  - neslib.h, [15](#)
- PAD\_A
  - neslib.h, [14](#)
- PAD\_B
  - neslib.h, [14](#)
- pad\_poll
  - neslib.h, [18](#)
- pad\_state
  - neslib.h, [18](#)
- pad\_trigger
  - neslib.h, [18](#)
- pal\_all
  - neslib.h, [18](#)
- pal\_bg
  - neslib.h, [19](#)
- pal\_bg\_bright
  - neslib.h, [19](#)
- pal\_bright
  - neslib.h, [19](#)
- pal\_clear
  - neslib.h, [19](#)
- pal\_col
  - neslib.h, [19](#)
- pal\_spr
  - neslib.h, [19](#)
- pal\_spr\_bright
  - neslib.h, [20](#)
- pause
  - definitions.h, [32](#)
- pause\_loop
  - definitions.h, [32](#)
- ppu\_mask
  - neslib.h, [20](#)
- ppu\_off
  - neslib.h, [20](#)
- ppu\_on\_all
  - neslib.h, [20](#)
- ppu\_on\_bg
  - neslib.h, [21](#)
- ppu\_on\_spr
  - neslib.h, [21](#)
- ppu\_system
  - neslib.h, [21](#)
- ppu\_wait\_frame
  - neslib.h, [21](#)
- ppu\_wait\_nmi
  - neslib.h, [21](#)
- rand16
  - neslib.h, [22](#)
- rand8
  - neslib.h, [22](#)
- render.c
  - center\_score\_when\_gameover, [36](#)
  - draw\_game\_over\_screen, [36](#)
  - draw\_item, [37](#)
  - draw\_level\_screen, [38](#)
  - draw\_pause\_screen, [39](#)

- draw\_score, [39](#)
- draw\_snake, [40](#)
- draw\_title\_screen, [40](#)
- init\_updateList, [41](#)
- mainloop\_render, [42](#)
- restart
  - definitions.h, [32](#)
- SNAKE\_BODY\_TILE
  - definitions.h, [29](#)
- SNAKE\_HEAD\_TILE\_HORZ
  - definitions.h, [29](#)
- SNAKE\_HEAD\_TILE\_VERT
  - definitions.h, [29](#)
- SNAKE\_MAX\_SIZE
  - definitions.h, [29](#)
- SPIDER\_TILE
  - definitions.h, [29](#)
- sample\_play
  - neslib.h, [22](#)
- scroll
  - neslib.h, [22](#)
- set\_rand
  - neslib.h, [22](#)
- set\_vram\_update
  - neslib.h, [23](#)
- sfx\_play
  - neslib.h, [23](#)
- size\_index
  - definitions.h, [33](#)
- snake.c
  - main, [44](#)
- snake\_head\_attribute
  - definitions.h, [33](#)
- snake\_head\_tile
  - definitions.h, [33](#)
- snake\_x
  - definitions.h, [33](#)
- snake\_y
  - definitions.h, [33](#)
- speed\_counter
  - definitions.h, [33](#)
- split
  - neslib.h, [23](#)
- sprite\_offset
  - definitions.h, [33](#)
- sprites\_pal
  - sprites\_pal.h, [9](#)
- sprites\_pal.h
  - sprites\_pal, [9](#)
- TRUE
  - neslib.h, [15](#)
- test\_nam
  - test\_nam.h, [25](#)
- test\_nam.h
  - test\_nam, [25](#)
- titlescreen
  - definitions.h, [33](#)
- titlescreen\_nam
  - titlescreen\_nam.h, [9](#)
- titlescreen\_nam.h
  - titlescreen\_nam, [9](#)
- ul
  - definitions.h, [33](#)
- update.c
  - add\_snake\_body\_element, [46](#)
  - calc\_random\_item\_position, [46](#)
  - check\_collision\_body, [47](#)
  - check\_collision\_item, [48](#)
  - check\_collision\_wall, [48](#)
  - check\_next\_level, [49](#)
  - load\_map\_data\_into\_array, [49](#)
  - mainloop\_update, [50](#)
  - update\_snake\_body, [51](#)
- update\_list
  - definitions.h, [34](#)
- update\_snake\_body
  - update.c, [51](#)
- vram\_adr
  - neslib.h, [23](#)
- vram\_fill
  - neslib.h, [23](#)
- vram\_inc
  - neslib.h, [24](#)
- vram\_put
  - neslib.h, [24](#)
- vram\_read
  - neslib.h, [24](#)
- vram\_unrle
  - neslib.h, [24](#)
- vram\_write
  - neslib.h, [24](#)
- WALL\_TILE\_1
  - definitions.h, [29](#)
- WALL\_TILE\_2
  - definitions.h, [30](#)