

NES Snake

Generated by Doxygen 1.8.12

Contents

1	README	1
2	File Index	3
2.1	File List	3
3	File Documentation	5
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	Detailed Description	7
3.4.2	Variable Documentation	8
3.4.2.1	levels_pal	8
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	9
3.6.1	Detailed Description	9
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	10
3.7.2.1	titlescreen_nam	10
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	11
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	13
3.9.1.4	MASK_EDGE_SPR	13
3.9.1.5	MASK_SPR	13
3.9.1.6	MAX	13
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	14
3.9.1.13	NT_UPD_EOF	14
3.9.1.14	NT_UPD_HORZ	14
3.9.1.15	NT_UPD_VERT	14

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	15
3.9.1.22	OAM_FLIP_H	15
3.9.1.23	OAM_FLIP_V	15
3.9.1.24	PAD_A	15
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	16
3.9.1.32	TRUE	16
3.9.2	Function Documentation	16
3.9.2.1	bank_bg()	16
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()	17
3.9.2.7	music_pause()	17
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()	18

3.9.2.13	<code>oam_size()</code>	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>	19
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>	20
3.9.2.23	<code>pal_col()</code>	20
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>	21
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>	22
3.9.2.31	<code>ppu_system()</code>	22
3.9.2.32	<code>ppu_wait_frame()</code>	22
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>	23
3.9.2.37	<code>scroll()</code>	23
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>	24

3.9.2.43	vram_fill()	24
3.9.2.44	vram_inc()	24
3.9.2.45	vram_put()	24
3.9.2.46	vram_read()	24
3.9.2.47	vram_unrle()	25
3.9.2.48	vram_write()	25
3.10	C:/Users/Administrator/Documents/GitHub/NES-Snake/NESLibrary/test_nam.h File Reference	25
3.10.1	Variable Documentation	25
3.10.1.1	test_nam	25
3.11	C:/Users/Administrator/Documents/GitHub/NES-Snake/README.md File Reference	26
3.12	C:/Users/Administrator/Documents/GitHub/NES-Snake/src/definitions.h File Reference	26
3.12.1	Detailed Description	28
3.12.2	Macro Definition Documentation	28
3.12.2.1	DIGIT_O_TILE	28
3.12.2.2	DIR_DOWN	28
3.12.2.3	DIR_LEFT	28
3.12.2.4	DIR_RIGHT	28
3.12.2.5	DIR_UP	28
3.12.2.6	EMPTY_TILE	28
3.12.2.7	LEVELS_ALL	28
3.12.2.8	LVL1_MAX_SCORE	29
3.12.2.9	LVL1_START_X	29
3.12.2.10	LVL1_START_Y	29
3.12.2.11	LVL2_MAX_SCORE	29
3.12.2.12	LVL2_START_X	29
3.12.2.13	LVL2_START_Y	29
3.12.2.14	MAP_HEIGHT	29
3.12.2.15	MAP_WIDTH	29
3.12.2.16	MAPARRAY_ADR	29
3.12.2.17	NAMETABLE1_START	30

3.12.2.18 SNAKE_BODY_TILE	30
3.12.2.19 SNAKE_HEAD_TILE_HORZ	30
3.12.2.20 SNAKE_HEAD_TILE_VERT	30
3.12.2.21 SNAKE_MAX_SIZE	30
3.12.2.22 SPIDER_TILE	30
3.12.2.23 WALL_TILE_1	30
3.12.2.24 WALL_TILE_2	30
3.12.3 Variable Documentation	31
3.12.3.1 body_coordinates	31
3.12.3.2 body_tile_x	31
3.12.3.3 body_tile_y	31
3.12.3.4 current_level	31
3.12.3.5 direction	31
3.12.3.6 gameover	31
3.12.3.7 gameover_loop	31
3.12.3.8 i	31
3.12.3.9 input	32
3.12.3.10 item_x	32
3.12.3.11 item_y	32
3.12.3.12 j	32
3.12.3.13 k	32
3.12.3.14 l	32
3.12.3.15 last_body_pixel_x	32
3.12.3.16 last_body_pixel_y	32
3.12.3.17 levelList	32
3.12.3.18 map	33
3.12.3.19 max_score	33
3.12.3.20 nameRow	33
3.12.3.21 nametable_fetch	33
3.12.3.22 pause	33

3.12.3.23	pause_loop	33
3.12.3.24	restart	33
3.12.3.25	size_index	33
3.12.3.26	snake_head_attribute	34
3.12.3.27	snake_head_tile	34
3.12.3.28	snake_x	34
3.12.3.29	snake_y	34
3.12.3.30	speed_counter	34
3.12.3.31	sprite_offset	34
3.12.3.32	titlescreen	34
3.12.3.33	ul	34
3.12.3.34	update_list	35
3.13	C:/Users/Administrator/Documents/GitHub/NES-Snake/src/init.c File Reference	35
3.13.1	Detailed Description	35
3.13.2	Function Documentation	35
3.13.2.1	init_level_params()	35
3.13.2.2	load_map_data_into_array()	36
3.14	C:/Users/Administrator/Documents/GitHub/NES-Snake/src/input.c File Reference	36
3.14.1	Detailed Description	37
3.14.2	Function Documentation	37
3.14.2.1	input_btn_start()	37
3.14.2.2	mainloop_handle_input()	37
3.15	C:/Users/Administrator/Documents/GitHub/NES-Snake/src/render.c File Reference	38
3.15.1	Detailed Description	39
3.15.2	Function Documentation	39
3.15.2.1	center_score_when_gameover()	39
3.15.2.2	draw_game_over_screen()	39
3.15.2.3	draw_item()	40
3.15.2.4	draw_level_screen()	41
3.15.2.5	draw_pause_screen()	42

3.15.2.6	draw_score()	43
3.15.2.7	draw_snake()	43
3.15.2.8	draw_title_screen()	44
3.15.2.9	init_updateList()	45
3.15.2.10	mainloop_render()	45
3.16	C:/Users/Administrator/Documents/GitHub/NES-Snake/src/snake.c File Reference	46
3.16.1	Detailed Description	47
3.16.2	Function Documentation	47
3.16.2.1	main()	47
3.17	C:/Users/Administrator/Documents/GitHub/NES-Snake/src/update.c File Reference	48
3.17.1	Detailed Description	49
3.17.2	Function Documentation	49
3.17.2.1	add_snake_body_element()	49
3.17.2.2	calc_random_item_position()	49
3.17.2.3	check_collision_body()	50
3.17.2.4	check_collision_item()	51
3.17.2.5	check_collision_wall()	51
3.17.2.6	check_next_level()	52
3.17.2.7	mainloop_update()	52
3.17.2.8	update_snake_body()	53
Index		55

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 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/ game_over_nam.h 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/ level1_nam.h 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/ level2_nam.h 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/ levels_pal.h This header file contains the color palette for all level maps. Created with NES Screen Tool 2.04 (Option Palettes -> Put C data to clipboard	7
C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/ menue_pal.h 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/ sprites_pal.h This header file contains the color palette for sprites	9
C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/ titlescreen_nam.h 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/ bgsplit_nam.h	10
C:/Users/Administrator/Documents/GitHub/NES-Snake/NESLibrary/ neslib.h	11
C:/Users/Administrator/Documents/GitHub/NES-Snake/NESLibrary/ test_nam.h	25
C:/Users/Administrator/Documents/GitHub/NES-Snake/src/ definitions.h This header files contains defines all global variables and constants, macros and includes of nametable and palette definition	26
C:/Users/Administrator/Documents/GitHub/NES-Snake/src/ init.c This file contains functions for initializing game elements	35
C:/Users/Administrator/Documents/GitHub/NES-Snake/src/ input.c This file contains functions for input handling from a controller	36
C:/Users/Administrator/Documents/GitHub/NES-Snake/src/ render.c This file contains all functionality to draw onto the screen, either as sprites or as background tiles	38
C:/Users/Administrator/Documents/GitHub/NES-Snake/src/ snake.c Maingame file, containing the main game loop	46
C:/Users/Administrator/Documents/GitHub/NES-Snake/src/ update.c This file contains all ingame logic functionalities and utility functionalities	48

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

This header file contains the color palette for all level maps. Created with NES Screen Tool 2.04 (Option Palettes -> Put C data to clipboard).

Variables

- const unsigned char [levels_pal](#) [16]

3.4.1 Detailed Description

This header file contains the color palette for all level maps. Created with NES Screen Tool 2.04 (Option Palettes -> Put C data to clipboard).

Author

Sebastian Dine

3.4.2 Variable Documentation

3.4.2.1 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,0x44,0x01,0x03,0x33,0x25,0x22,
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,
0x01,0x12,0x40,0x01,0x02,0x00,0x40,0x01,0x02,0x00,0x40,0x00,0x00,0x40,0x40,
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,
0x40,0x01,0x02,0x00,0x40,0x01,0x02,0x00,0x40,0x00,0x40,0x00,0x40,0x01,0x02,
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 )
```

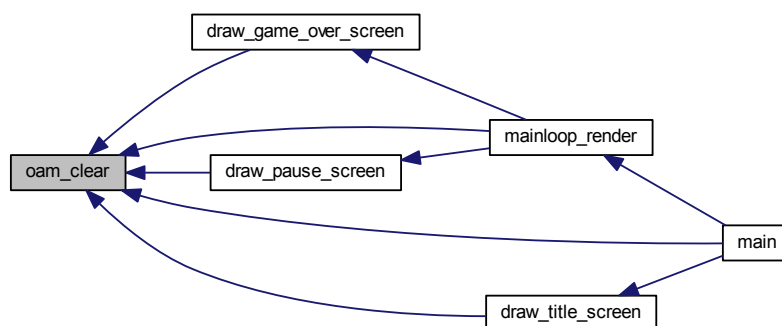
3.9.2.9 music_stop()

```
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 )
```

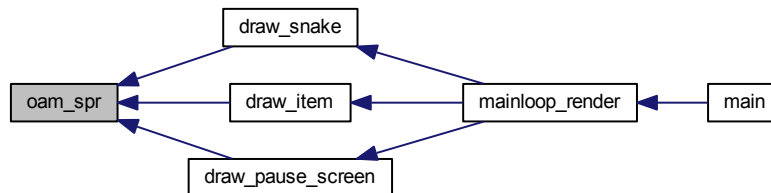
3.9.2.13 oam_size()

```
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 )
```

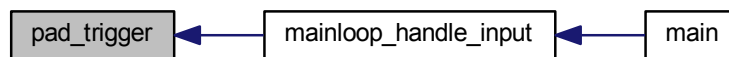
3.9.2.16 pad_state()

```
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:



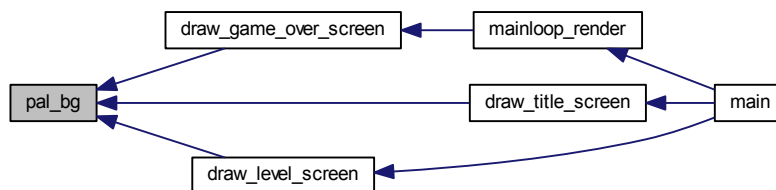
3.9.2.18 pal_all()

```
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 )
```

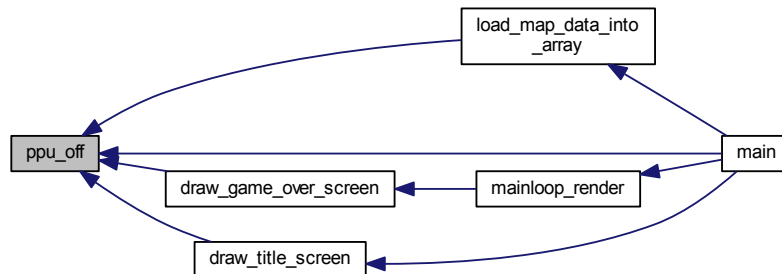
3.9.2.26 ppu_mask()

```
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 )
```

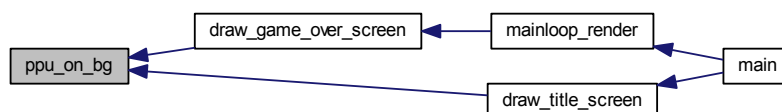
Here is the caller graph for this function:



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:

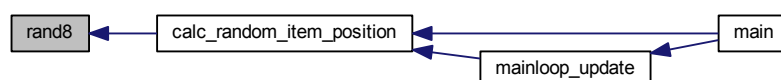
**3.9.2.34 rand16()**

```
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 )
```

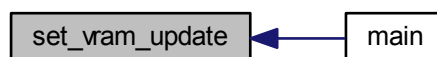
3.9.2.38 set_rand()

```
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 )
```

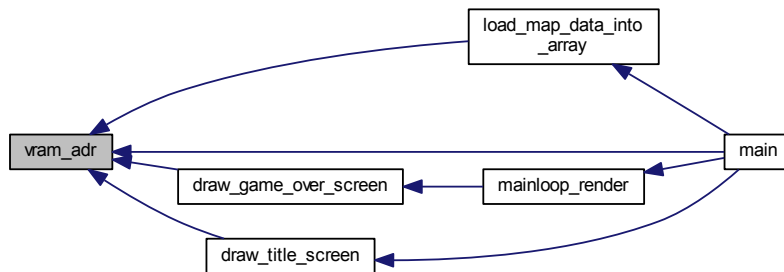
3.9.2.41 split()

```
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 )
```

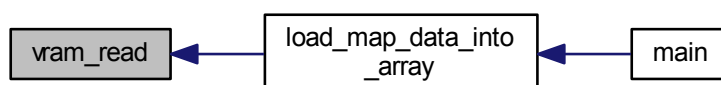
3.9.2.45 vram_put()

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

3.9.2.46 vram_read()

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

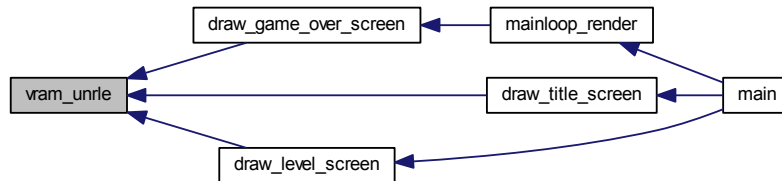
Here is the caller graph for this function:



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, 0x01, 0x02, 0x10, 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, 0xa5, 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 LVL1_START_X 120
- #define LVL1_START_Y 120
- #define LVL1_MAX_SCORE 4
- #define LVL2_START_X 56
- #define LVL2_START_Y 120
- #define LVL2_MAX_SCORE 8
- #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 caluclations

- #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 `speed_counter`
- static unsigned char `snake_head_attribute`
- static unsigned char `snake_head_tile`
- 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, menus, input

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

List of the levels, include pointer to the packed nametable of the levels, menus, 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 LVL1_MAX_SCORE

```
#define LVL1_MAX_SCORE 4
```

3.12.2.9 LVL1_START_X

```
#define LVL1_START_X 120
```

3.12.2.10 LVL1_START_Y

```
#define LVL1_START_Y 120
```

3.12.2.11 LVL2_MAX_SCORE

```
#define LVL2_MAX_SCORE 8
```

3.12.2.12 LVL2_START_X

```
#define LVL2_START_X 56
```

3.12.2.13 LVL2_START_Y

```
#define LVL2_START_Y 120
```

3.12.2.14 MAP_HEIGHT

```
#define MAP_HEIGHT 30
```

3.12.2.15 MAP_WIDTH

```
#define MAP_WIDTH 32
```

3.12.2.16 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).

3.12.2.17 NAMETABLE1_START

```
#define NAMETABLE1_START 0x2000
```

Start address in VRAM for first nametable

3.12.2.18 SNAKE_BODY_TILE

```
#define SNAKE_BODY_TILE 0x40
```

Tile of snake body element

3.12.2.19 SNAKE_HEAD_TILE_HORZ

```
#define SNAKE_HEAD_TILE_HORZ 0x42
```

Tile of horizontal snake head element

3.12.2.20 SNAKE_HEAD_TILE_VERT

```
#define SNAKE_HEAD_TILE_VERT 0x41
```

Tile of vertical snake head element

3.12.2.21 SNAKE_MAX_SIZE

```
#define SNAKE_MAX_SIZE 100
```

Number of body elements, the snake can get.

3.12.2.22 SPIDER_TILE

```
#define SPIDER_TILE 0x45
```

Tile of spider item

3.12.2.23 WALL_TILE_1

```
#define WALL_TILE_1 0x43
```

Tile of horizontal wall element

3.12.2.24 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 max_score

```
unsigned char max_score [static]
```

Global variable, indicating the maximum score of the current level.

3.12.3.20 nameRow

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

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

3.12.3.21 nametable_fetch

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

Variable for fetching through nametable.

3.12.3.22 pause

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

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

3.12.3.23 pause_loop

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

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

3.12.3.24 restart

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

Global variable, for handling the restart input

3.12.3.25 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.

3.12.3.26 snake_head_attribute

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

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

3.12.3.27 snake_head_tile

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

3.12.3.28 snake_x

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

3.12.3.29 snake_y

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

3.12.3.30 speed_counter

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

3.12.3.31 sprite_offset

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

3.12.3.32 titlescreen

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

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

3.12.3.33 ul

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

Pointer to array 'update_list' to enable better handling of the list

3.12.3.34 update_list

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

Array of bg-elements which will be used to update VRAM once per frame. Every 3 entries are describing one bg-element.

- the first 3 elements (9 array-elements) are assigned to the game score
- the 4. and 5. element are assigned to the first and last body element of the snake
- the last array-element needs to be the VRAM end-of-file-indicator NT_UPD_EOF.

Only two body elements need to be updated once per frame:

- The new first body element needs to be drawn
- The old last body element need to be disabled

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

This file contains functions for initializing game elements.

Functions

- void [load_map_data_into_array](#) (void)
- void [init_level_params](#) (void)

3.13.1 Detailed Description

This file contains functions for initializing game elements.

Author

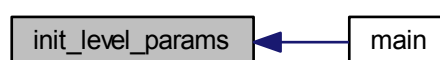
Sebastian Dine

3.13.2 Function Documentation

3.13.2.1 init_level_params()

```
void init_level_params (  
    void )
```

This function initializes game elements, which differ between levels. (e.g. score to reach for next level or start position of the snake) Here is the caller graph for this function:



3.13.2.2 load_map_data_into_array()

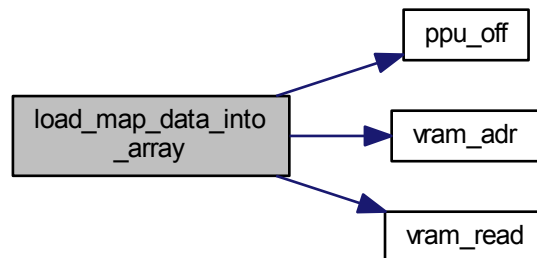
```
void load_map_data_into_array (  
    void )
```

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

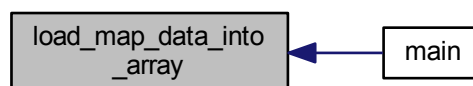
Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



3.14 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)

3.14.1 Detailed Description

This file contains functions for input handling from a controller.

Author

Sebastian Dine

3.14.2 Function Documentation

3.14.2.1 input_btn_start()

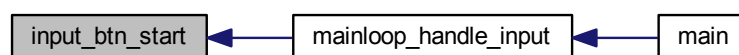
```
void input_btn_start (  
    void )
```

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:



3.14.2.2 mainloop_handle_input()

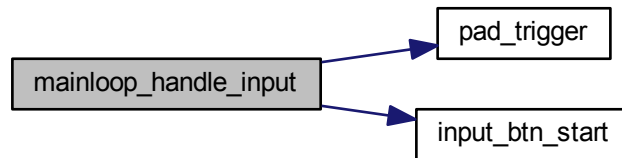
```
void mainloop_handle_input (  
    void )
```

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.

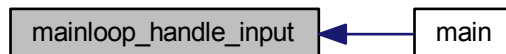
Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



3.15 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)

3.15.1 Detailed Description

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

Author

Sebastian Dine

3.15.2 Function Documentation

3.15.2.1 center_score_when_gameover()

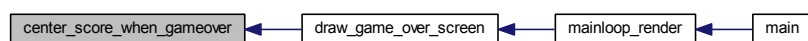
```
void center_score_when_gameover (  
    void )
```

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:



3.15.2.2 draw_game_over_screen()

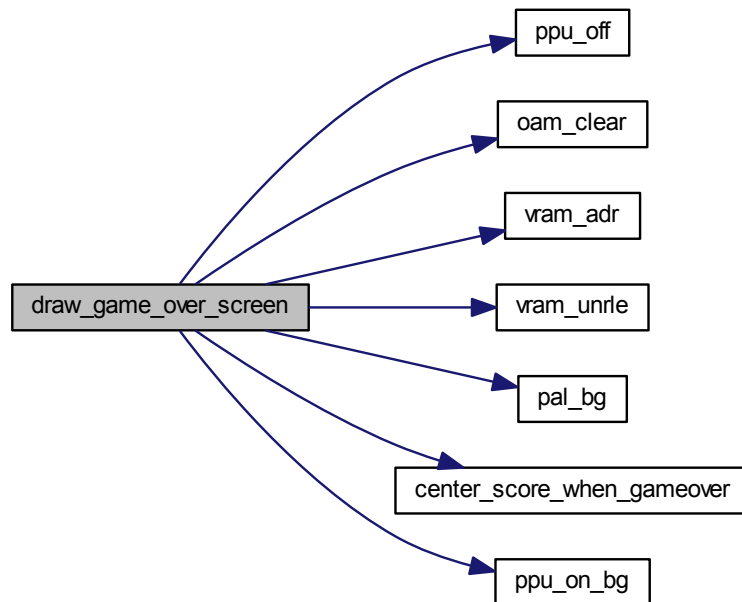
```
void draw_game_over_screen (  
    void )
```

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:



3.15.2.3 draw_item()

```
void draw_item (
    void )
```

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

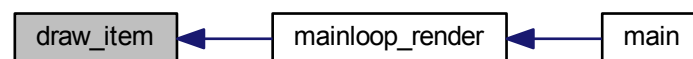
Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:

**3.15.2.4 draw_level_screen()**

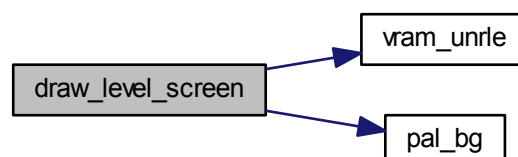
```
void draw_level_screen (  
    void )
```

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:



3.15.2.5 draw_pause_screen()

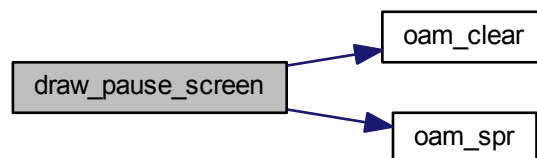
```
void draw_pause_screen (  
    void )
```

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

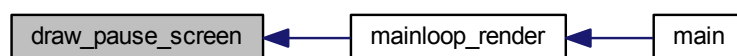
Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



3.15.2.6 draw_score()

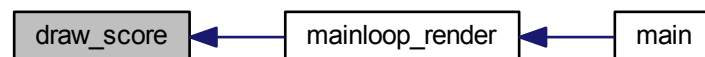
```
void draw_score (
    void )
```

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

Author

Sebastian Dine

Here is the caller graph for this function:



3.15.2.7 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.

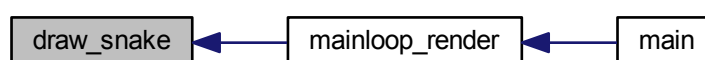
Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



3.15.2.8 draw_title_screen()

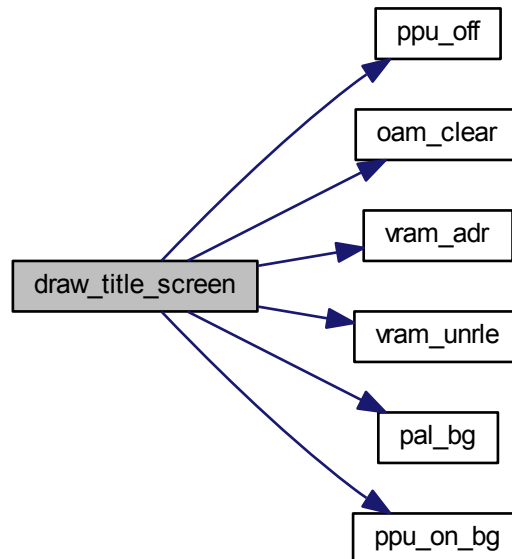
```
void draw_title_screen (  
    void )
```

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:



3.15.2.9 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:



3.15.2.10 mainloop_render()

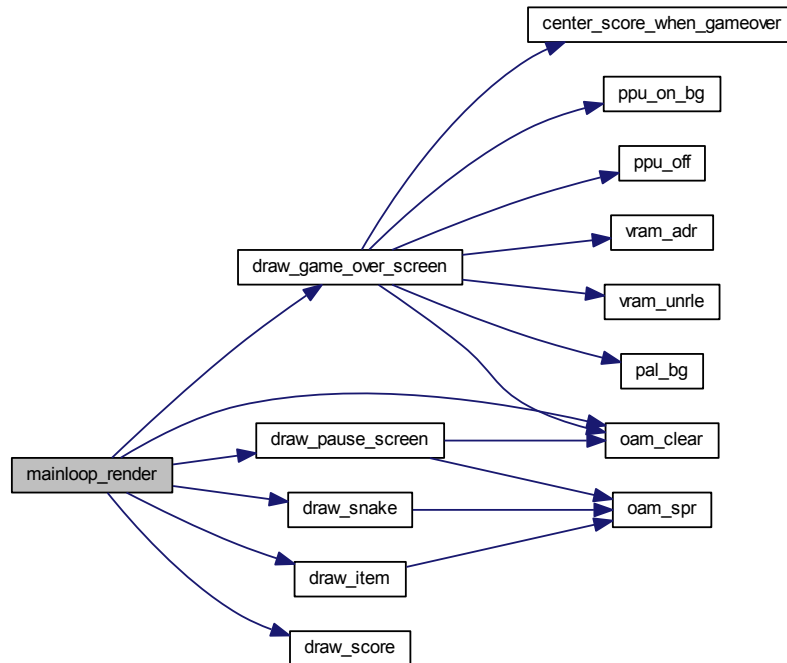
```
void mainloop_render (  
    void )
```

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

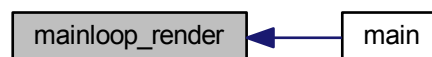
Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



3.16 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 "init.c"
#include "input.c"
#include "update.c"
#include "render.c"

```


Functions

- void `main` (void)

Main game loop.

3.16.1 Detailed Description

Maingame file, containing the main game loop.

Author

Sebastian Dine.

3.16.2 Function Documentation

3.16.2.1 `main()`

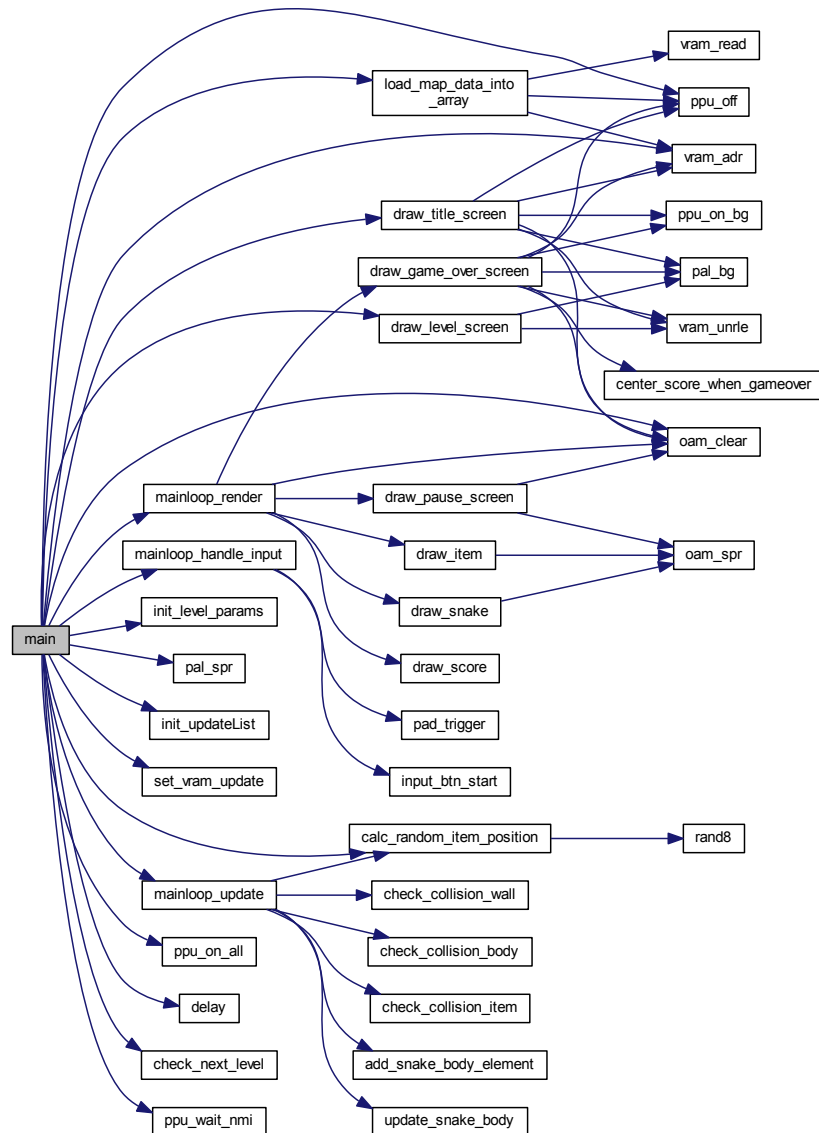
```
void main (  
    void )
```

Main game loop.

Author

Sebastian Dine

Here is the call graph for this function:



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

This file contains all ingame logic functionalities and utility functionalities.

Functions

- 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)

3.17.1 Detailed Description

This file contains all ingame logic functionalities and utility functionalities.

Author

Sebastian Dine

3.17.2 Function Documentation

3.17.2.1 `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:



3.17.2.2 `calc_random_item_position()`

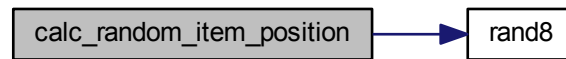
```
void calc_random_item_position (
    void )
```

This function calculates the coordinates of an grow-item. It stores the calculated coordinates into global fields 'item_x' and 'item_y'.

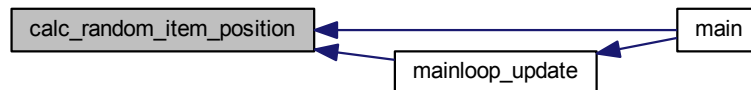
Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:

**3.17.2.3 check_collision_body()**

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

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

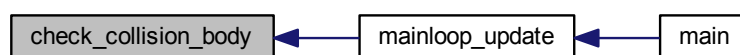
Returns

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

Author

Sebastian Dine

Here is the caller graph for this function:



3.17.2.4 check_collision_item()

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

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

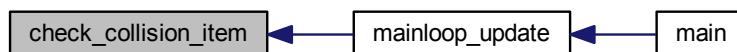
Returns

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

Author

Sebastian Dine

Here is the caller graph for this function:



3.17.2.5 check_collision_wall()

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

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

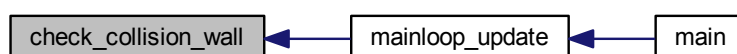
Returns

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

Author

Sebastian Dine

Here is the caller graph for this function:



3.17.2.6 check_next_level()

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

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

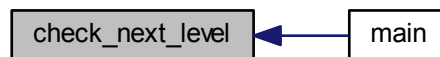
Returns

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

Author

Sebastian Dine

Here is the caller graph for this function:



3.17.2.7 mainloop_update()

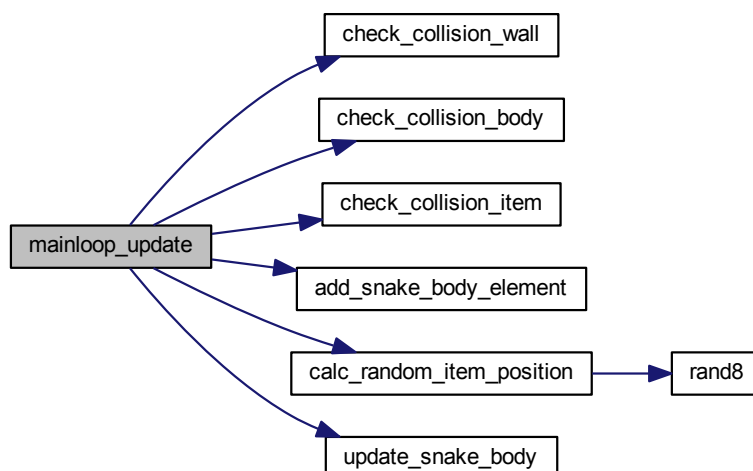
```
void mainloop_update (  
    void )
```

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:



3.17.2.8 update_snake_body()

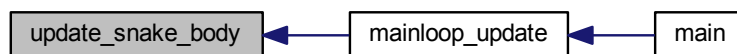
```
void update_snake_body ( )
```

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, 49

bank_bg
neslib.h, 16
bank_spr
neslib.h, 16
bgsplit_nam
bgsplit_nam.h, 10
bgsplit_nam.h
bgsplit_nam, 10
body_coordinates
definitions.h, 31
body_tile_x
definitions.h, 31
body_tile_y
definitions.h, 31

C:/Users/Administrator/Documents/GitHub/NES-
Snake/NESLibrary/bgsplit_nam.h, 10
C:/Users/Administrator/Documents/GitHub/NES-
Snake/NESLibrary/neslib.h, 11
C:/Users/Administrator/Documents/GitHub/NES-
Snake/NESLibrary/test_nam.h, 25
C:/Users/Administrator/Documents/GitHub/NES-
Snake/README.md, 26
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, 9
C:/Users/Administrator/Documents/GitHub/NES-
Snake/gfx/titlescreen_nam.h, 9
C:/Users/Administrator/Documents/GitHub/NES-
Snake/src/definitions.h, 26
C:/Users/Administrator/Documents/GitHub/NES-
Snake/src/init.c, 35
C:/Users/Administrator/Documents/GitHub/NES-
Snake/src/input.c, 36
C:/Users/Administrator/Documents/GitHub/NES-
Snake/src/render.c, 38
C:/Users/Administrator/Documents/GitHub/NES-
Snake/src/snake.c, 46

C:/Users/Administrator/Documents/GitHub/NES-
Snake/src/update.c, 48
calc_random_item_position
update.c, 49
center_score_when_gameover
render.c, 39
check_collision_body
update.c, 50
check_collision_item
update.c, 50
check_collision_wall
update.c, 51
check_next_level
update.c, 51
current_level
definitions.h, 31
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, 31
body_tile_x, 31
body_tile_y, 31
current_level, 31
DIGIT_O_TILE, 28
DIR_DOWN, 28
DIR_LEFT, 28
DIR_RIGHT, 28
DIR_UP, 28
direction, 31
EMPTY_TILE, 28
gameover, 31
gameover_loop, 31
i, 31
input, 31
item_x, 32
item_y, 32
j, 32
k, 32
l, 32
LEVELS_ALL, 28
LVL1_MAX_SCORE, 28

- LVL1_START_X, 29
- LVL1_START_Y, 29
- LVL2_MAX_SCORE, 29
- LVL2_START_X, 29
- LVL2_START_Y, 29
- last_body_pixel_x, 32
- last_body_pixel_y, 32
- levelList, 32
- MAP_HEIGHT, 29
- MAP_WIDTH, 29
- MAPARRAY_ADR, 29
- map, 32
- max_score, 33
- NAMETABLE1_START, 29
- nameRow, 33
- nametable_fetch, 33
- pause, 33
- pause_loop, 33
- restart, 33
- SNAKE_BODY_TILE, 30
- SNAKE_HEAD_TILE_HORZ, 30
- SNAKE_HEAD_TILE_VERT, 30
- SNAKE_MAX_SIZE, 30
- SPIDER_TILE, 30
- size_index, 33
- snake_head_attribute, 33
- snake_head_tile, 34
- snake_x, 34
- snake_y, 34
- speed_counter, 34
- sprite_offset, 34
- titlescreen, 34
- ul, 34
- update_list, 34
- WALL_TILE_1, 30
- WALL_TILE_2, 30
- delay
 - neslib.h, 16
- direction
 - definitions.h, 31
- draw_game_over_screen
 - render.c, 39
- draw_item
 - render.c, 40
- draw_level_screen
 - render.c, 41
- draw_pause_screen
 - render.c, 42
- draw_score
 - render.c, 42
- draw_snake
 - render.c, 43
- draw_title_screen
 - render.c, 43
- 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, 31
- gameover_loop
 - definitions.h, 31
- i
 - definitions.h, 31
- init.c
 - init_level_params, 35
 - load_map_data_into_array, 35
- init_level_params
 - init.c, 35
- init_updateList
 - render.c, 44
- input
 - definitions.h, 31
- input.c
 - input_btn_start, 37
 - mainloop_handle_input, 37
- input_btn_start
 - input.c, 37
- item_x
 - definitions.h, 32
- item_y
 - definitions.h, 32
- j
 - definitions.h, 32
- k
 - definitions.h, 32
- l
 - definitions.h, 32
- LEVELS_ALL
 - definitions.h, 28
- LVL1_MAX_SCORE
 - definitions.h, 28
- LVL1_START_X
 - definitions.h, 29
- LVL1_START_Y
 - definitions.h, 29
- LVL2_MAX_SCORE
 - definitions.h, 29
- LVL2_START_X
 - definitions.h, 29
- LVL2_START_Y
 - definitions.h, 29
- last_body_pixel_x
 - definitions.h, 32
- last_body_pixel_y

- definitions.h, [32](#)
- 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, [8](#)
- levels_pal.h
 - levels_pal, [8](#)
- load_map_data_into_array
 - init.c, [35](#)
- MAP_HEIGHT
 - definitions.h, [29](#)
- MAP_WIDTH
 - definitions.h, [29](#)
- MAPARRAY_ADR
 - definitions.h, [29](#)
- MASK_BG
 - neslib.h, [12](#)
- MASK_EDGE_BG
 - neslib.h, [12](#)
- MASK_EDGE_SPR
 - neslib.h, [13](#)
- MASK_SPR
 - neslib.h, [13](#)
- MAX
 - neslib.h, [13](#)
- MIN
 - neslib.h, [13](#)
- MSB
 - neslib.h, [13](#)
- main
 - snake.c, [47](#)
- mainloop_handle_input
 - input.c, [37](#)
- mainloop_render
 - render.c, [45](#)
- mainloop_update
 - update.c, [52](#)
- map
 - definitions.h, [32](#)
- max_score
 - definitions.h, [33](#)
- memcpy
 - neslib.h, [16](#)
- memfill
 - neslib.h, [16](#)
- menue_pal
 - menue_pal.h, [8](#)
- menue_pal.h
 - menue_pal, [8](#)
- music_pause
 - neslib.h, [17](#)
- music_play
 - neslib.h, [17](#)
- 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, [14](#)
- NT_UPD_HORZ
 - neslib.h, [14](#)
- NT_UPD_VERT
 - neslib.h, [14](#)
- NTADR_A
 - neslib.h, [14](#)
- NTADR_B
 - neslib.h, [14](#)
- NTADR_C
 - neslib.h, [14](#)
- NTADR_D
 - neslib.h, [14](#)
- NULL
 - neslib.h, [14](#)
- nameRow
 - definitions.h, [33](#)
- nametable_fetch
 - definitions.h, [33](#)
- neslib.h
 - bank_bg, [16](#)
 - bank_spr, [16](#)
 - delay, [16](#)
 - FALSE, [12](#)
 - flush_vram_update, [16](#)
 - MASK_BG, [12](#)
 - MASK_EDGE_BG, [12](#)
 - MASK_EDGE_SPR, [13](#)
 - MASK_SPR, [13](#)
 - MAX, [13](#)
 - MIN, [13](#)
 - MSB, [13](#)
 - memcpy, [16](#)
 - memfill, [16](#)
 - music_pause, [17](#)
 - music_play, [17](#)
 - music_stop, [17](#)
 - NAMETABLE_A, [13](#)
 - NAMETABLE_B, [13](#)
 - NAMETABLE_C, [13](#)
 - NAMETABLE_D, [13](#)
 - NT_UPD_EOF, [14](#)

NT_UPD_HORZ, [14](#)
 NT_UPD_VERT, [14](#)
 NTADR_A, [14](#)
 NTADR_B, [14](#)
 NTADR_C, [14](#)
 NTADR_D, [14](#)
 NULL, [14](#)
 OAM_BEHIND, [14](#)
 OAM_FLIP_H, [15](#)
 OAM_FLIP_V, [15](#)
 oam_clear, [17](#)
 oam_hide_rest, [17](#)
 oam_meta_spr, [17](#)
 oam_size, [18](#)
 oam_spr, [18](#)
 PAD_DOWN, [15](#)
 PAD_LEFT, [15](#)
 PAD_RIGHT, [15](#)
 PAD_SELECT, [15](#)
 PAD_START, [15](#)
 PAD_UP, [15](#)
 PAD_A, [15](#)
 PAD_B, [15](#)
 pad_poll, [18](#)
 pad_state, [18](#)
 pad_trigger, [18](#)
 pal_all, [19](#)
 pal_bg, [19](#)
 pal_bg_bright, [19](#)
 pal_bright, [19](#)
 pal_clear, [19](#)
 pal_col, [20](#)
 pal_spr, [20](#)
 pal_spr_bright, [20](#)
 ppu_mask, [20](#)
 ppu_off, [20](#)
 ppu_on_all, [21](#)
 ppu_on_bg, [21](#)
 ppu_on_spr, [21](#)
 ppu_system, [22](#)
 ppu_wait_frame, [22](#)
 ppu_wait_nmi, [22](#)
 rand16, [22](#)
 rand8, [22](#)
 sample_play, [22](#)
 scroll, [23](#)
 set_rand, [23](#)
 set_vram_update, [23](#)
 sfx_play, [23](#)
 split, [23](#)
 TRUE, [16](#)
 vram_adr, [23](#)
 vram_fill, [24](#)
 vram_inc, [24](#)
 vram_put, [24](#)
 vram_read, [24](#)
 vram_unrle, [24](#)
 vram_write, [25](#)
 OAM_BEHIND
 neslib.h, [14](#)
 OAM_FLIP_H
 neslib.h, [15](#)
 OAM_FLIP_V
 neslib.h, [15](#)
 oam_clear
 neslib.h, [17](#)
 oam_hide_rest
 neslib.h, [17](#)
 oam_meta_spr
 neslib.h, [17](#)
 oam_size
 neslib.h, [18](#)
 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, [15](#)
 PAD_B
 neslib.h, [15](#)
 pad_poll
 neslib.h, [18](#)
 pad_state
 neslib.h, [18](#)
 pad_trigger
 neslib.h, [18](#)
 pal_all
 neslib.h, [19](#)
 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, [20](#)
 pal_spr
 neslib.h, [20](#)
 pal_spr_bright
 neslib.h, [20](#)
 pause
 definitions.h, [33](#)
 pause_loop
 definitions.h, [33](#)

ppu_mask
 neslib.h, 20

ppu_off
 neslib.h, 20

ppu_on_all
 neslib.h, 21

ppu_on_bg
 neslib.h, 21

ppu_on_spr
 neslib.h, 21

ppu_system
 neslib.h, 22

ppu_wait_frame
 neslib.h, 22

ppu_wait_nmi
 neslib.h, 22

rand16
 neslib.h, 22

rand8
 neslib.h, 22

render.c
 center_score_when_gameover, 39
 draw_game_over_screen, 39
 draw_item, 40
 draw_level_screen, 41
 draw_pause_screen, 42
 draw_score, 42
 draw_snake, 43
 draw_title_screen, 43
 init_updateList, 44
 mainloop_render, 45

restart
 definitions.h, 33

SNAKE_BODY_TILE
 definitions.h, 30

SNAKE_HEAD_TILE_HORZ
 definitions.h, 30

SNAKE_HEAD_TILE_VERT
 definitions.h, 30

SNAKE_MAX_SIZE
 definitions.h, 30

SPIDER_TILE
 definitions.h, 30

sample_play
 neslib.h, 22

scroll
 neslib.h, 23

set_rand
 neslib.h, 23

set_vram_update
 neslib.h, 23

sfx_play
 neslib.h, 23

size_index
 definitions.h, 33

snake.c
 main, 47

snake_head_attribute
 definitions.h, 33

snake_head_tile
 definitions.h, 34

snake_x
 definitions.h, 34

snake_y
 definitions.h, 34

speed_counter
 definitions.h, 34

split
 neslib.h, 23

sprite_offset
 definitions.h, 34

sprites_pal
 sprites_pal.h, 9

sprites_pal.h
 sprites_pal, 9

TRUE
 neslib.h, 16

test_nam
 test_nam.h, 25

test_nam.h
 test_nam, 25

titlescreen
 definitions.h, 34

titlescreen_nam
 titlescreen_nam.h, 10

titlescreen_nam.h
 titlescreen_nam, 10

ul
 definitions.h, 34

update.c
 add_snake_body_element, 49
 calc_random_item_position, 49
 check_collision_body, 50
 check_collision_item, 50
 check_collision_wall, 51
 check_next_level, 51
 mainloop_update, 52
 update_snake_body, 53

update_list
 definitions.h, 34

update_snake_body
 update.c, 53

vram_adr
 neslib.h, 23

vram_fill
 neslib.h, 24

vram_inc
 neslib.h, 24

vram_put
 neslib.h, 24

vram_read
 neslib.h, 24

vram_unrle

neslib.h, [24](#)
vram_write
neslib.h, [25](#)

WALL_TILE_1
definitions.h, [30](#)
WALL_TILE_2
definitions.h, [30](#)