

NES Snake

Generated by Doxygen 1.8.12

Contents

1	README	1
2	Data Structure Index	3
2.1	Data Structures	3
3	File Index	5
3.1	File List	5
4	Data Structure Documentation	7
4.1	items_struct Struct Reference	7
4.1.1	Detailed Description	7
4.1.2	Field Documentation	7
4.1.2.1	item_collision_flags	7
4.1.2.2	item_coordinates	8
4.1.2.3	item_respawn_count	8
4.1.2.4	item_respawn_frm_rate	8
4.2	snake_struct Struct Reference	8
4.2.1	Detailed Description	8
4.2.2	Field Documentation	9
4.2.2.1	body_element_coordinates	9
4.2.2.2	head_sprite	9
4.2.2.3	head_sprite_attribute	9
4.2.2.4	head_sprite_x	9
4.2.2.5	head_sprite_y	9
4.2.2.6	last_body_element_x	9
4.2.2.7	last_body_element_y	9
4.2.2.8	moving_direction	10
4.2.2.9	size_index	10
4.2.2.10	speed_counter	10

5	File Documentation	11
5.1	C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/game_over_nam.h File Reference . . .	11
5.1.1	Detailed Description	11
5.1.2	Variable Documentation	11
5.1.2.1	game_over_nam	11
5.2	C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/level1_nam.h File Reference	12
5.2.1	Detailed Description	12
5.2.2	Variable Documentation	12
5.2.2.1	level1_nam	12
5.3	C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/level2_nam.h File Reference	12
5.3.1	Detailed Description	13
5.3.2	Variable Documentation	13
5.3.2.1	level2_nam	13
5.4	C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/levels_pal.h File Reference	13
5.4.1	Detailed Description	13
5.4.2	Variable Documentation	14
5.4.2.1	levels_pal	14
5.5	C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/menue_pal.h File Reference	14
5.5.1	Detailed Description	14
5.5.2	Variable Documentation	14
5.5.2.1	menue_pal	14
5.6	C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/sprites_pal.h File Reference	15
5.6.1	Detailed Description	15
5.6.2	Variable Documentation	15
5.6.2.1	sprites_pal	15
5.7	C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/titlescreen_nam.h File Reference	15
5.7.1	Detailed Description	15
5.7.2	Variable Documentation	16
5.7.2.1	titlescreen_nam	16
5.8	C:/Users/Administrator/Documents/GitHub/NES-Snake/NESLibrary/neslib.h File Reference	16

5.8.1	Macro Definition Documentation	18
5.8.1.1	FALSE	18
5.8.1.2	MASK_BG	18
5.8.1.3	MASK_EDGE_BG	18
5.8.1.4	MASK_EDGE_SPR	18
5.8.1.5	MASK_SPR	18
5.8.1.6	MAX	18
5.8.1.7	MIN	18
5.8.1.8	MSB	18
5.8.1.9	NAMETABLE_A	18
5.8.1.10	NAMETABLE_B	19
5.8.1.11	NAMETABLE_C	19
5.8.1.12	NAMETABLE_D	19
5.8.1.13	NT_UPD_EOF	19
5.8.1.14	NT_UPD_HORZ	19
5.8.1.15	NT_UPD_VERT	19
5.8.1.16	NTADR_A	19
5.8.1.17	NTADR_B	19
5.8.1.18	NTADR_C	19
5.8.1.19	NTADR_D	20
5.8.1.20	NULL	20
5.8.1.21	OAM_BEHIND	20
5.8.1.22	OAM_FLIP_H	20
5.8.1.23	OAM_FLIP_V	20
5.8.1.24	PAD_A	20
5.8.1.25	PAD_B	20
5.8.1.26	PAD_DOWN	20
5.8.1.27	PAD_LEFT	20
5.8.1.28	PAD_RIGHT	20
5.8.1.29	PAD_SELECT	21

5.8.1.30	PAD_START	21
5.8.1.31	PAD_UP	21
5.8.1.32	TRUE	21
5.8.2	Function Documentation	21
5.8.2.1	bank_bg()	21
5.8.2.2	bank_spr()	21
5.8.2.3	delay()	21
5.8.2.4	flush_vram_update()	22
5.8.2.5	memcpy()	22
5.8.2.6	memfill()	22
5.8.2.7	music_pause()	22
5.8.2.8	music_play()	22
5.8.2.9	music_stop()	22
5.8.2.10	oam_clear()	23
5.8.2.11	oam_hide_rest()	23
5.8.2.12	oam_meta_spr()	23
5.8.2.13	oam_size()	23
5.8.2.14	oam_spr()	24
5.8.2.15	pad_poll()	24
5.8.2.16	pad_state()	24
5.8.2.17	pad_trigger()	24
5.8.2.18	pal_all()	25
5.8.2.19	pal_bg()	25
5.8.2.20	pal_bg_bright()	25
5.8.2.21	pal_bright()	25
5.8.2.22	pal_clear()	25
5.8.2.23	pal_col()	25
5.8.2.24	pal_spr()	26
5.8.2.25	pal_spr_bright()	26
5.8.2.26	ppu_mask()	26

5.8.2.27	ppu_off()	26
5.8.2.28	ppu_on_all()	27
5.8.2.29	ppu_on_bg()	27
5.8.2.30	ppu_on_spr()	27
5.8.2.31	ppu_system()	27
5.8.2.32	ppu_wait_frame()	27
5.8.2.33	ppu_wait_nmi()	28
5.8.2.34	rand16()	28
5.8.2.35	rand8()	28
5.8.2.36	sample_play()	28
5.8.2.37	scroll()	28
5.8.2.38	set_rand()	29
5.8.2.39	set_vram_update()	29
5.8.2.40	sfx_play()	29
5.8.2.41	split()	29
5.8.2.42	vram_adr()	30
5.8.2.43	vram_fill()	30
5.8.2.44	vram_inc()	30
5.8.2.45	vram_put()	30
5.8.2.46	vram_read()	30
5.8.2.47	vram_unrle()	31
5.8.2.48	vram_write()	31
5.9	C:/Users/Administrator/Documents/GitHub/NES-Snake/README.md File Reference	31
5.10	C:/Users/Administrator/Documents/GitHub/NES-Snake/src/globals.h File Reference	31
5.10.1	Detailed Description	32
5.10.2	Variable Documentation	32
5.10.2.1	coord_x	32
5.10.2.2	coord_y	32
5.10.2.3	current_level	33
5.10.2.4	gameover	33

5.10.2.5	gameover_loop	33
5.10.2.6	i	33
5.10.2.7	input	33
5.10.2.8	items	33
5.10.2.9	j	33
5.10.2.10	k	33
5.10.2.11	l	34
5.10.2.12	levelList	34
5.10.2.13	map	34
5.10.2.14	max_score	34
5.10.2.15	nameRow	34
5.10.2.16	nametable_fetch	34
5.10.2.17	pause	34
5.10.2.18	pause_loop	35
5.10.2.19	restart	35
5.10.2.20	snake	35
5.10.2.21	sprite_offset	35
5.10.2.22	titlescreen	35
5.10.2.23	ul	35
5.10.2.24	update_list	35
5.11	C:/Users/Administrator/Documents/GitHub/NES-Snake/src/init.c File Reference	36
5.11.1	Detailed Description	36
5.11.2	Function Documentation	36
5.11.2.1	calc_random_item_position()	36
5.11.2.2	init_items()	37
5.11.2.3	init_level_params()	37
5.11.2.4	load_map_data_into_array()	38
5.12	C:/Users/Administrator/Documents/GitHub/NES-Snake/src/input.c File Reference	39
5.12.1	Detailed Description	39
5.12.2	Function Documentation	39

5.12.2.1	input_btn_start()	39
5.12.2.2	mainloop_handle_input()	40
5.13	C:/Users/Administrator/Documents/GitHub/NES-Snake/src/macros.h File Reference	40
5.13.1	Detailed Description	41
5.13.2	Macro Definition Documentation	41
5.13.2.1	DIGIT_O_TILE	41
5.13.2.2	DIR_DOWN	42
5.13.2.3	DIR_LEFT	42
5.13.2.4	DIR_RIGHT	42
5.13.2.5	DIR_UP	42
5.13.2.6	EMPTY_TILE	42
5.13.2.7	ITEM_MAX_ON_SCREEN	42
5.13.2.8	LEVELS_ALL	42
5.13.2.9	LVL1_MAX_SCORE	42
5.13.2.10	LVL1_START_X	42
5.13.2.11	LVL1_START_Y	43
5.13.2.12	LVL2_MAX_SCORE	43
5.13.2.13	LVL2_START_X	43
5.13.2.14	LVL2_START_Y	43
5.13.2.15	MAP_HEIGHT	43
5.13.2.16	MAP_WIDTH	43
5.13.2.17	MAPARRAY_ADR	43
5.13.2.18	NAMETABLE1_START	43
5.13.2.19	SFX_ABSORB_ITEM	43
5.13.2.20	SFX_CRASH	44
5.13.2.21	SFX_NEXT_LEVEL	44
5.13.2.22	SFX_PAUSE	44
5.13.2.23	SNAKE_BODY_TILE	44
5.13.2.24	SNAKE_HEAD_TILE_HORZ	44
5.13.2.25	SNAKE_HEAD_TILE_VERT	44

5.13.2.26 SNAKE_MAX_SIZE	44
5.13.2.27 SPIDER_TILE	44
5.13.2.28 WALL_TILE_1	45
5.13.2.29 WALL_TILE_2	45
5.14 C:/Users/Administrator/Documents/GitHub/NES-Snake/src/render.c File Reference	45
5.14.1 Detailed Description	45
5.14.2 Function Documentation	45
5.14.2.1 center_score_when_gameover()	45
5.14.2.2 draw_game_over_screen()	46
5.14.2.3 draw_items()	47
5.14.2.4 draw_level_screen()	47
5.14.2.5 draw_pause_screen()	48
5.14.2.6 draw_score()	49
5.14.2.7 draw_snake()	49
5.14.2.8 draw_title_screen()	50
5.14.2.9 init_updateList()	51
5.14.2.10 mainloop_render()	52
5.15 C:/Users/Administrator/Documents/GitHub/NES-Snake/src/snake.c File Reference	53
5.15.1 Detailed Description	53
5.15.2 Function Documentation	54
5.15.2.1 main()	54
5.16 C:/Users/Administrator/Documents/GitHub/NES-Snake/src/structures.h File Reference	55
5.16.1 Detailed Description	56
5.17 C:/Users/Administrator/Documents/GitHub/NES-Snake/src/update.c File Reference	56
5.17.1 Detailed Description	56
5.17.2 Function Documentation	57
5.17.2.1 add_snake_body_element()	57
5.17.2.2 check_collision_body()	57
5.17.2.3 check_collision_wall()	58
5.17.2.4 check_collision_with_items()	58
5.17.2.5 check_next_level()	59
5.17.2.6 mainloop_update()	59
5.17.2.7 respawn_items()	60
5.17.2.8 update_snake_body()	61

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

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

items_struct	This structure contains all elements required to interact with and display items	7
snake_struct	This structure contains all elements required to interact and display the snake	8

Chapter 3

File Index

3.1 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)	11
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)	12
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)	12
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	13
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	14
C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/ sprites_pal.h This header file contains the color palette for sprites	15
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)	15
C:/Users/Administrator/Documents/GitHub/NES-Snake/NESLibrary/ neslib.h	16
C:/Users/Administrator/Documents/GitHub/NES-Snake/src/ globals.h This header file defines all global variables of the game	31
C:/Users/Administrator/Documents/GitHub/NES-Snake/src/ init.c This file contains functions for initializing game elements	36
C:/Users/Administrator/Documents/GitHub/NES-Snake/src/ input.c This file contains functions for input handling from a controller	39
C:/Users/Administrator/Documents/GitHub/NES-Snake/src/ macros.h This header file defines object-like macros (constants) and function-like macros for more efficient calculations	40
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	45
C:/Users/Administrator/Documents/GitHub/NES-Snake/src/ snake.c Maingame file, containing the main game loop	53

C:/Users/Administrator/Documents/GitHub/NES-Snake/src/structures.h	
This header file contains the definition of structures, created for the purpose of the game	55
C:/Users/Administrator/Documents/GitHub/NES-Snake/src/update.c	
This file contains all ingame logic functionalities and utility functionalities	56

Chapter 4

Data Structure Documentation

4.1 items_struct Struct Reference

This structure contains all elements required to interact with and display items.

```
#include <structures.h>
```

Data Fields

- unsigned int [item_respawn_frm_rate](#)
- unsigned char [item_coordinates](#) [ITEM_MAX_ON_SCREEN<< 1]
- unsigned int [item_respawn_count](#) [ITEM_MAX_ON_SCREEN]
- unsigned char [item_collision_flags](#) [ITEM_MAX_ON_SCREEN]

4.1.1 Detailed Description

This structure contains all elements required to interact with and display items.

Author

Sebastian Dine

4.1.2 Field Documentation

4.1.2.1 item_collision_flags

```
unsigned char item_collision_flags[ITEM_MAX_ON_SCREEN]
```

Array which indicates collisions with items (0 = no collision, 1 = collision). E.g. [0]=1 means, that the snake collided with the first element.

4.1.2.2 item_coordinates

```
unsigned char item_coordinates[ITEM_MAX_ON_SCREEN<< 1]
```

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

4.1.2.3 item_respawn_count

```
unsigned int item_respawn_count[ITEM_MAX_ON_SCREEN]
```

Array which counts down the frames until an item respawn. 0 indicates a non active visible item. Due to this decision, items will respawn, once their frame countdown reaches 1.

4.1.2.4 item_respawn_frm_rate

```
unsigned int item_respawn_frm_rate
```

Variable, which contains the frame rate until an items should respawn. Remember NES is 60FPS.

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

- C:/Users/Administrator/Documents/GitHub/NES-Snake/src/[structures.h](#)

4.2 snake_struct Struct Reference

This structure contains all elements required to interact and display the snake.

```
#include <structures.h>
```

Data Fields

- unsigned char [size_index](#)
- unsigned char [speed_counter](#)
- unsigned char [moving_direction](#)
- unsigned char [head_sprite](#)
- unsigned char [head_sprite_attribute](#)
- unsigned char [head_sprite_x](#)
- unsigned char [head_sprite_y](#)
- unsigned char [last_body_element_x](#)
- unsigned char [last_body_element_y](#)
- unsigned char [body_element_coordinates](#) [SNAKE_MAX_SIZE<< 1]

4.2.1 Detailed Description

This structure contains all elements required to interact and display the snake.

Author

Sebastian Dine

4.2.2 Field Documentation

4.2.2.1 body_element_coordinates

```
unsigned char body_element_coordinates[SNAKE_MAX_SIZE<< 1]
```

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

4.2.2.2 head_sprite

```
unsigned char head_sprite
```

tbd.

4.2.2.3 head_sprite_attribute

```
unsigned char head_sprite_attribute
```

Variable for holding attributes of the head sprite of the snake.

4.2.2.4 head_sprite_x

```
unsigned char head_sprite_x
```

Pixel based X-coordinate of snake's head sprite.

4.2.2.5 head_sprite_y

```
unsigned char head_sprite_y
```

Pixel based Y-coordinate of snake's head sprite.

4.2.2.6 last_body_element_x

```
unsigned char last_body_element_x
```

Pixel based X-coordinate of the last body element from last frame.

4.2.2.7 last_body_element_y

```
unsigned char last_body_element_y
```

Pixel based Y-coordinate of the last body element from last frame.

4.2.2.8 moving_direction

```
unsigned char moving_direction
```

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

4.2.2.9 size_index

```
unsigned char size_index
```

Index for array 'body_element_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.

4.2.2.10 speed_counter

```
unsigned char speed_counter
```

tbd.

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

- C:/Users/Administrator/Documents/GitHub/NES-Snake/src/[structures.h](#)

Chapter 5

File Documentation

5.1 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]

5.1.1 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

5.1.2 Variable Documentation

5.1.2.1 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
}
```

5.2 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]

5.2.1 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

5.2.2 Variable Documentation

5.2.2.1 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
}
```

5.3 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]

5.3.1 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

5.3.2 Variable Documentation

5.3.2.1 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
}
```

5.4 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]

5.4.1 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

5.4.2 Variable Documentation

5.4.2.1 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 }  
}
```

5.5 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]

5.5.1 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

5.5.2 Variable Documentation

5.5.2.1 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 }  
}
```


5.6 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]

5.6.1 Detailed Description

This header file contains the color palette for sprites.

Author

Sebastian Dine

5.6.2 Variable Documentation

5.6.2.1 `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 }  
}
```

5.7 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]

5.7.1 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

5.7.2 Variable Documentation

5.7.2.1 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
}
```

5.8 C:/Users/Administrator/Documents/GitHub/NES-Snake/NESLibrary/nelib.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__ [memset](#) (void *dst, unsigned char value, unsigned int len)
- void __fastcall__ [delay](#) (unsigned char frames)

5.8.1 Macro Definition Documentation

5.8.1.1 FALSE

```
#define FALSE 0
```

5.8.1.2 MASK_BG

```
#define MASK_BG 0x08
```

5.8.1.3 MASK_EDGE_BG

```
#define MASK_EDGE_BG 0x02
```

5.8.1.4 MASK_EDGE_SPR

```
#define MASK_EDGE_SPR 0x04
```

5.8.1.5 MASK_SPR

```
#define MASK_SPR 0x10
```

5.8.1.6 MAX

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

5.8.1.7 MIN

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

5.8.1.8 MSB

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

5.8.1.9 NAMETABLE_A

```
#define NAMETABLE_A 0x2000
```

5.8.1.10 NAMETABLE_B

```
#define NAMETABLE_B 0x2400
```

5.8.1.11 NAMETABLE_C

```
#define NAMETABLE_C 0x2800
```

5.8.1.12 NAMETABLE_D

```
#define NAMETABLE_D 0x2c00
```

5.8.1.13 NT_UPD_EOF

```
#define NT_UPD_EOF 0xff
```

5.8.1.14 NT_UPD_HORZ

```
#define NT_UPD_HORZ 0x40
```

5.8.1.15 NT_UPD_VERT

```
#define NT_UPD_VERT 0x80
```

5.8.1.16 NTADR_A

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

5.8.1.17 NTADR_B

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

5.8.1.18 NTADR_C

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

5.8.1.19 NTADR_D

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

5.8.1.20 NULL

```
#define NULL 0
```

5.8.1.21 OAM_BEHIND

```
#define OAM_BEHIND 0x20
```

5.8.1.22 OAM_FLIP_H

```
#define OAM_FLIP_H 0x40
```

5.8.1.23 OAM_FLIP_V

```
#define OAM_FLIP_V 0x80
```

5.8.1.24 PAD_A

```
#define PAD_A 0x01
```

5.8.1.25 PAD_B

```
#define PAD_B 0x02
```

5.8.1.26 PAD_DOWN

```
#define PAD_DOWN 0x20
```

5.8.1.27 PAD_LEFT

```
#define PAD_LEFT 0x40
```

5.8.1.28 PAD_RIGHT

```
#define PAD_RIGHT 0x80
```

5.8.1.29 PAD_SELECT

```
#define PAD_SELECT 0x04
```

5.8.1.30 PAD_START

```
#define PAD_START 0x08
```

5.8.1.31 PAD_UP

```
#define PAD_UP 0x10
```

5.8.1.32 TRUE

```
#define TRUE 1
```

5.8.2 Function Documentation

5.8.2.1 bank_bg()

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

5.8.2.2 bank_spr()

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

5.8.2.3 delay()

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

Here is the caller graph for this function:



5.8.2.4 flush_vram_update()

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

5.8.2.5 memcpy()

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

5.8.2.6 memfill()

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

5.8.2.7 music_pause()

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

5.8.2.8 music_play()

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

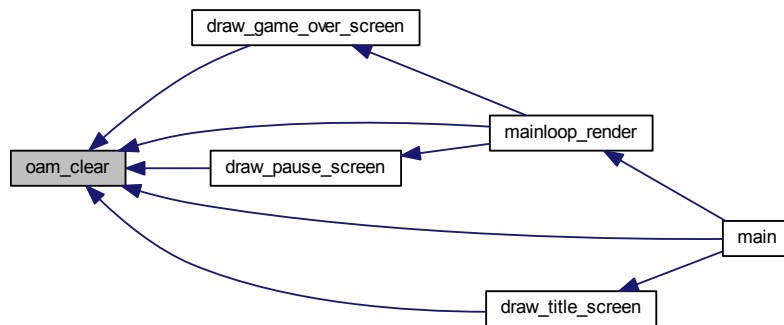
5.8.2.9 music_stop()

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


5.8.2.10 oam_clear()

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

Here is the caller graph for this function:



5.8.2.11 oam_hide_rest()

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

5.8.2.12 oam_meta_spr()

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

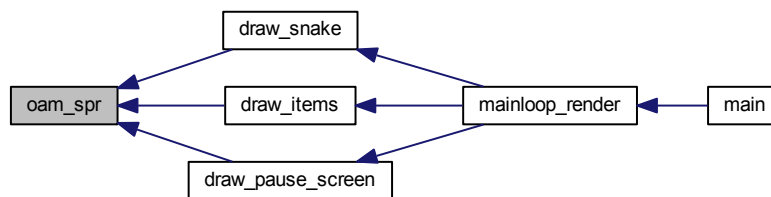
5.8.2.13 oam_size()

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

5.8.2.14 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:



5.8.2.15 pad_poll()

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

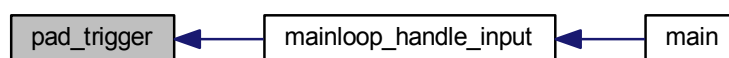
5.8.2.16 pad_state()

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

5.8.2.17 pad_trigger()

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

Here is the caller graph for this function:



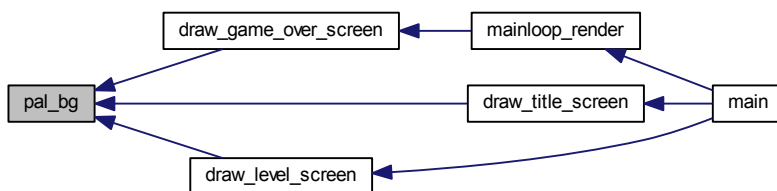
5.8.2.18 pal_all()

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

5.8.2.19 pal_bg()

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

Here is the caller graph for this function:



5.8.2.20 pal_bg_bright()

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

5.8.2.21 pal_bright()

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

5.8.2.22 pal_clear()

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

5.8.2.23 pal_col()

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

5.8.2.24 pal_spr()

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

Here is the caller graph for this function:



5.8.2.25 pal_spr_bright()

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

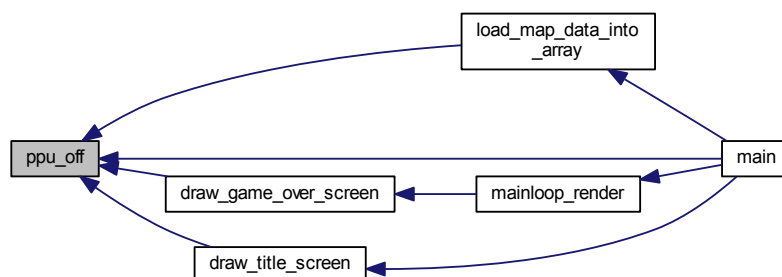
5.8.2.26 ppu_mask()

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

5.8.2.27 ppu_off()

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

Here is the caller graph for this function:



5.8.2.28 ppu_on_all()

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

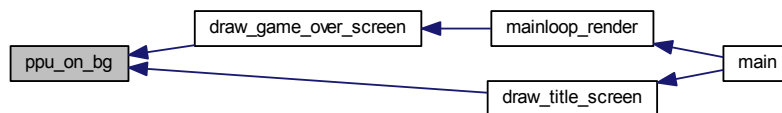
Here is the caller graph for this function:



5.8.2.29 ppu_on_bg()

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

Here is the caller graph for this function:



5.8.2.30 ppu_on_spr()

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

5.8.2.31 ppu_system()

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

5.8.2.32 ppu_wait_frame()

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

5.8.2.33 ppu_wait_nmi()

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

Here is the caller graph for this function:



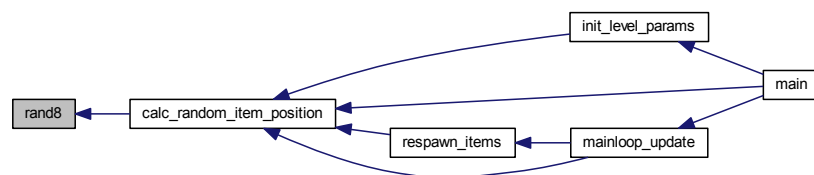
5.8.2.34 rand16()

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

5.8.2.35 rand8()

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

Here is the caller graph for this function:



5.8.2.36 sample_play()

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

5.8.2.37 scroll()

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

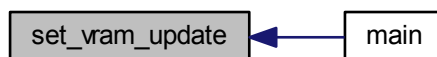
5.8.2.38 set_rand()

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

5.8.2.39 set_vram_update()

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

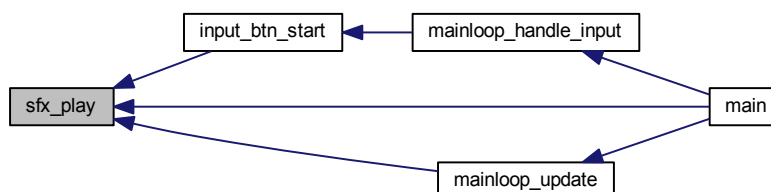
Here is the caller graph for this function:



5.8.2.40 sfx_play()

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

Here is the caller graph for this function:



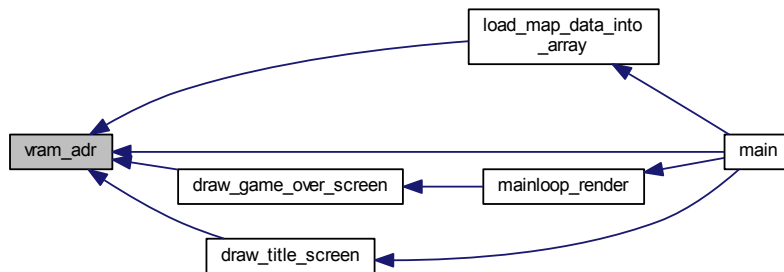
5.8.2.41 split()

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

5.8.2.42 vram_adr()

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

Here is the caller graph for this function:



5.8.2.43 vram_fill()

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

5.8.2.44 vram_inc()

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

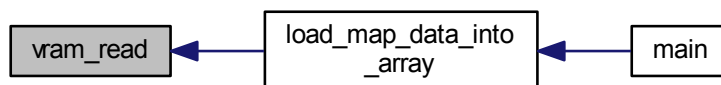
5.8.2.45 vram_put()

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

5.8.2.46 vram_read()

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

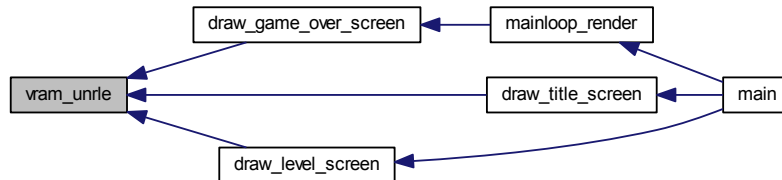
Here is the caller graph for this function:



5.8.2.47 vram_unrle()

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

Here is the caller graph for this function:



5.8.2.48 vram_write()

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

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

5.10 C:/Users/Administrator/Documents/GitHub/NES-Snake/src/globals.h File Reference

This header file defines all global variables of the game.

Variables

- static struct [snake_struct](#) snake
- static struct [items_struct](#) items

Global variables, which are used for several coordination calculations,

e.g. calculate pixel based coordinates (of body elements) to tile based coordinates.

- static unsigned char [coord_x](#)
- static unsigned char [coord_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 for game-states, menues, 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](#)

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](#)

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]

5.10.1 Detailed Description

This header file defines all global variables of the game.

Author

Sebastian Dine

5.10.2 Variable Documentation

5.10.2.1 coord_x

```
unsigned char coord_x [static]
```

5.10.2.2 coord_y

```
unsigned char coord_y [static]
```

5.10.2.3 `current_level`

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

Global variable, indicating the current level.

5.10.2.4 `gameover`

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

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

5.10.2.5 `gameover_loop`

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

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

5.10.2.6 `i`

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

5.10.2.7 `input`

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

Global variable, holding the controller input of the current frame

5.10.2.8 `items`

```
struct items_struct items [static]
```

Global variable, containing all elements used to interact with and display items

5.10.2.9 `j`

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

5.10.2.10 `k`

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

5.10.2.11 l

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

5.10.2.12 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  
}
```

5.10.2.13 map

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

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

5.10.2.14 max_score

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

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

5.10.2.15 nameRow

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

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

5.10.2.16 nametable_fetch

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

Variable for fetching through nametable.

5.10.2.17 pause

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

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

5.10.2.18 pause_loop

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

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

5.10.2.19 restart

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

Global variable, for handling the restart input

5.10.2.20 snake

```
struct snake_struct snake [static]
```

Global variable, containing all elements used to interact and display the snake

5.10.2.21 sprite_offset

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

5.10.2.22 titlescreen

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

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

5.10.2.23 ul

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

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

5.10.2.24 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

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

This file contains functions for initializing game elements.

Functions

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

5.11.1 Detailed Description

This file contains functions for initializing game elements.

Author

Sebastian Dine

5.11.2 Function Documentation

5.11.2.1 [calc_random_item_position\(\)](#)

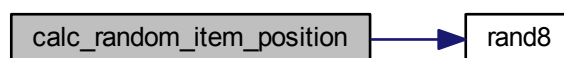
```
void calc_random_item_position (  
    void )
```

This function calculates the coordinates of an grow-item. It stores the calculated coordinates into global fields 'coord_x' and 'coord_y'. In terms of the game structure, this function should be placed in file '[update.c](#)'. But since I would like to spawn the initial items randomly as well, this function needs to placed in file '[init.c](#)'.

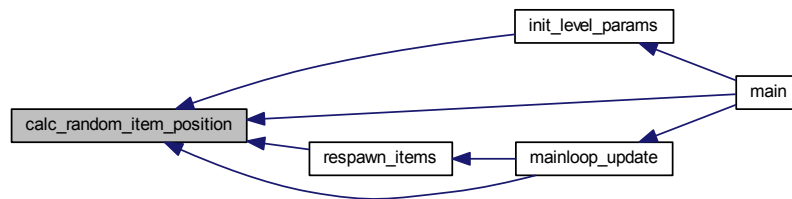
Author

Sebastian Dine

Here is the call graph for this function:



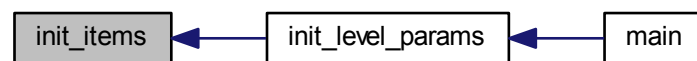
Here is the caller graph for this function:



5.11.2.2 `init_items()`

```
void init_items (
    void )
```

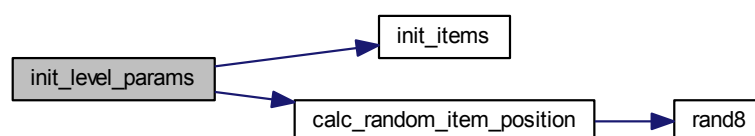
Here is the caller graph for this function:



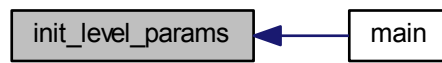
5.11.2.3 `init_level_params()`

```
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 call graph for this function:



Here is the caller graph for this function:



5.11.2.4 load_map_data_into_array()

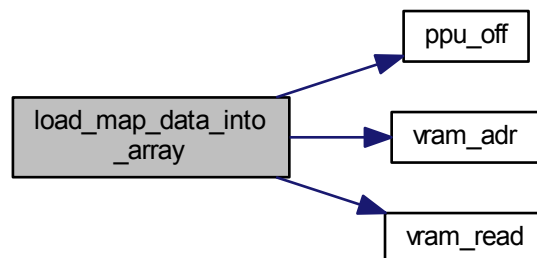
```
void load_map_data_into_array (  
    void )
```

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

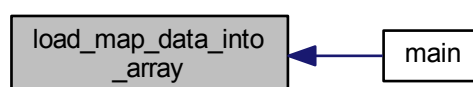
Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



5.12 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)

5.12.1 Detailed Description

This file contains functions for input handling from a controller.

Author

Sebastian Dine

5.12.2 Function Documentation

5.12.2.1 [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 call graph for this function:



Here is the caller graph for this function:



5.12.2.2 mainloop_handle_input()

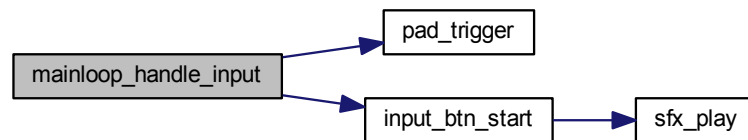
```
void mainloop_handle_input (  
    void )
```

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.

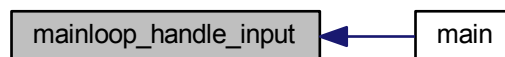
Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



5.13 C:/Users/Administrator/Documents/GitHub/NES-Snake/src/macros.h File Reference

This header file defines object-like macros (constants) and function-like macros for more efficient calculations.

Macros

- `#define LEVELS_ALL 5`
- `#define SNAKE_MAX_SIZE 100`
- `#define ITEM_MAX_ON_SCREEN 4`
- `#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

Sound effect constants.

Each constants represents the number of an sound effect within sfx/snake_sfx.s (connect between game and .s file is created in file NESLibrary/crt0.s).

- #define SFX_ABSORB_ITEM 0
- #define SFX_CRASH 1
- #define SFX_PAUSE 2
- #define SFX_NEXT_LEVEL 3

Macros for more efficient calucations

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

5.13.1 Detailed Description

This header file defines object-like macros (constants) and function-like macros for more efficient calculations.

Author

Sebastian Dine

5.13.2 Macro Definition Documentation

5.13.2.1 DIGIT_O_TILE

```
#define DIGIT_O_TILE 0x10
```

Tile of digit 0 (zero)

5.13.2.2 DIR_DOWN

```
#define DIR_DOWN 2
```

5.13.2.3 DIR_LEFT

```
#define DIR_LEFT 3
```

5.13.2.4 DIR_RIGHT

```
#define DIR_RIGHT 4
```

5.13.2.5 DIR_UP

```
#define DIR_UP 1
```

5.13.2.6 EMPTY_TILE

```
#define EMPTY_TILE 0x00
```

Tile of empty space

5.13.2.7 ITEM_MAX_ON_SCREEN

```
#define ITEM_MAX_ON_SCREEN 4
```

Maximum of items, that can be on the screen on the same time.

5.13.2.8 LEVELS_ALL

```
#define LEVELS_ALL 5
```

Total number of level maps (ingame background nametables)

5.13.2.9 LVL1_MAX_SCORE

```
#define LVL1_MAX_SCORE 4
```

5.13.2.10 LVL1_START_X

```
#define LVL1_START_X 120
```

5.13.2.11 LVL1_START_Y

```
#define LVL1_START_Y 120
```

5.13.2.12 LVL2_MAX_SCORE

```
#define LVL2_MAX_SCORE 8
```

5.13.2.13 LVL2_START_X

```
#define LVL2_START_X 56
```

5.13.2.14 LVL2_START_Y

```
#define LVL2_START_Y 120
```

5.13.2.15 MAP_HEIGHT

```
#define MAP_HEIGHT 30
```

5.13.2.16 MAP_WIDTH

```
#define MAP_WIDTH 32
```

5.13.2.17 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).

5.13.2.18 NAMETABLE1_START

```
#define NAMETABLE1_START 0x2000
```

Start address in VRAM for first nametable

5.13.2.19 SFX_ABSORB_ITEM

```
#define SFX_ABSORB_ITEM 0
```

5.13.2.20 SFX_CRASH

```
#define SFX_CRASH 1
```

5.13.2.21 SFX_NEXT_LEVEL

```
#define SFX_NEXT_LEVEL 3
```

5.13.2.22 SFX_PAUSE

```
#define SFX_PAUSE 2
```

5.13.2.23 SNAKE_BODY_TILE

```
#define SNAKE_BODY_TILE 0x40
```

Tile of snake body element

5.13.2.24 SNAKE_HEAD_TILE_HORZ

```
#define SNAKE_HEAD_TILE_HORZ 0x42
```

Tile of horizontal snake head element

5.13.2.25 SNAKE_HEAD_TILE_VERT

```
#define SNAKE_HEAD_TILE_VERT 0x41
```

Tile of vertical snake head element

5.13.2.26 SNAKE_MAX_SIZE

```
#define SNAKE_MAX_SIZE 100
```

Maximum of body elements, the snake can get.

5.13.2.27 SPIDER_TILE

```
#define SPIDER_TILE 0x45
```

Tile of spider item

5.13.2.28 WALL_TILE_1

```
#define WALL_TILE_1 0x43
```

Tile of horizontal wall element

5.13.2.29 WALL_TILE_2

```
#define WALL_TILE_2 0x44
```

Tile of vertical wall element

5.14 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_items](#) (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)

5.14.1 Detailed Description

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

Author

Sebastian Dine

5.14.2 Function Documentation

5.14.2.1 center_score_when_gameover()

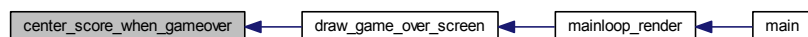
```
void center_score_when_gameover (
    void )
```

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:



5.14.2.2 draw_game_over_screen()

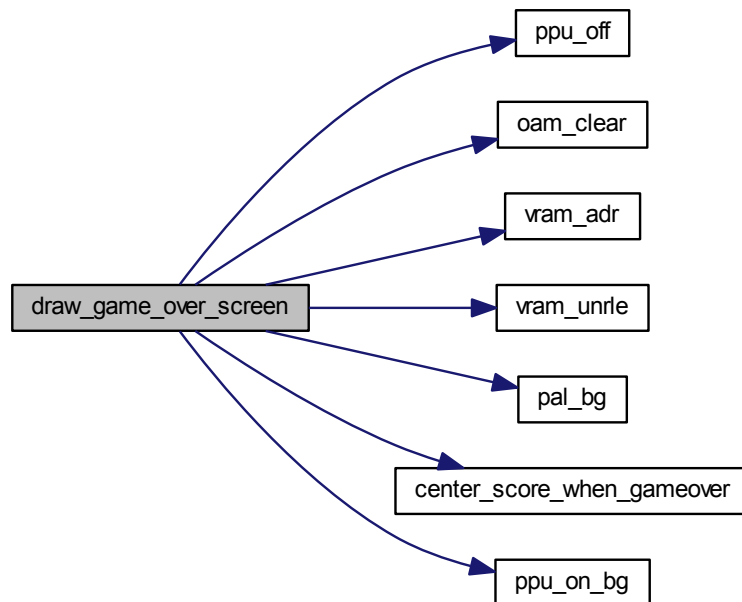
```
void draw_game_over_screen (  
    void )
```

This function draws the gameover screen.

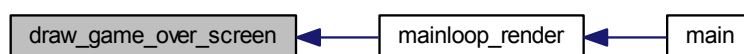
Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



5.14.2.3 draw_items()

```
void draw_items (
    void )
```

This function draws all item elements as sprites to the screen.

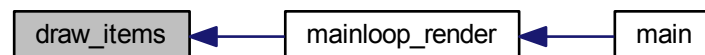
Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



5.14.2.4 draw_level_screen()

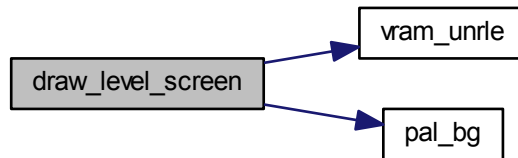
```
void draw_level_screen (
    void )
```

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

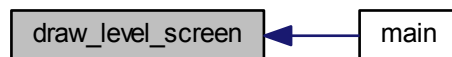
Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:

**5.14.2.5 draw_pause_screen()**

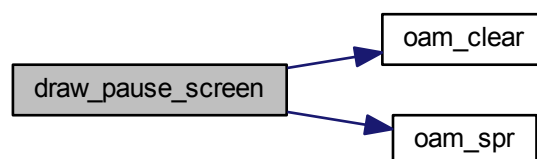
```
void draw_pause_screen (  
    void )
```

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

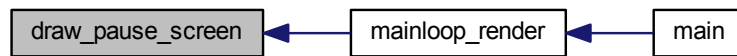
Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



5.14.2.6 `draw_score()`

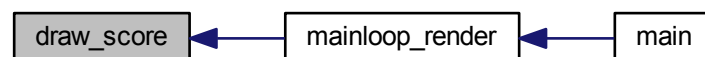
```
void draw_score (  
    void )
```

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

Author

Sebastian Dine

Here is the caller graph for this function:



5.14.2.7 `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.

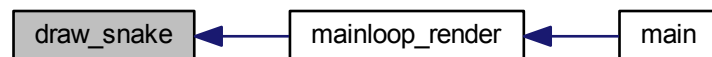
Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:

**5.14.2.8 draw_title_screen()**

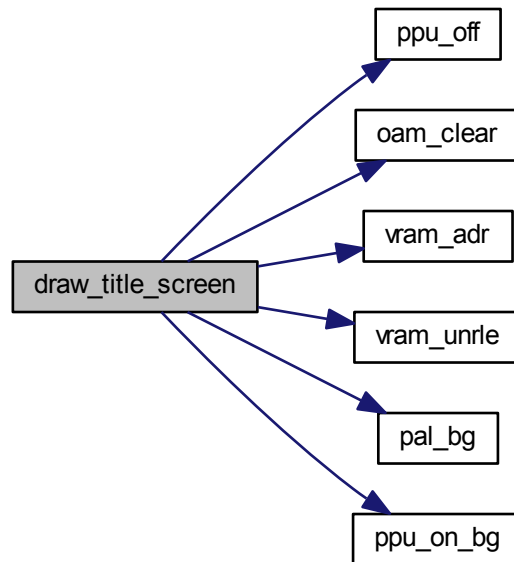
```
void draw_title_screen (
    void )
```

This function draws the title screen.

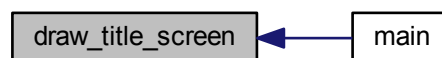
Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



5.14.2.9 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:

**5.14.2.10 mainloop_render()**

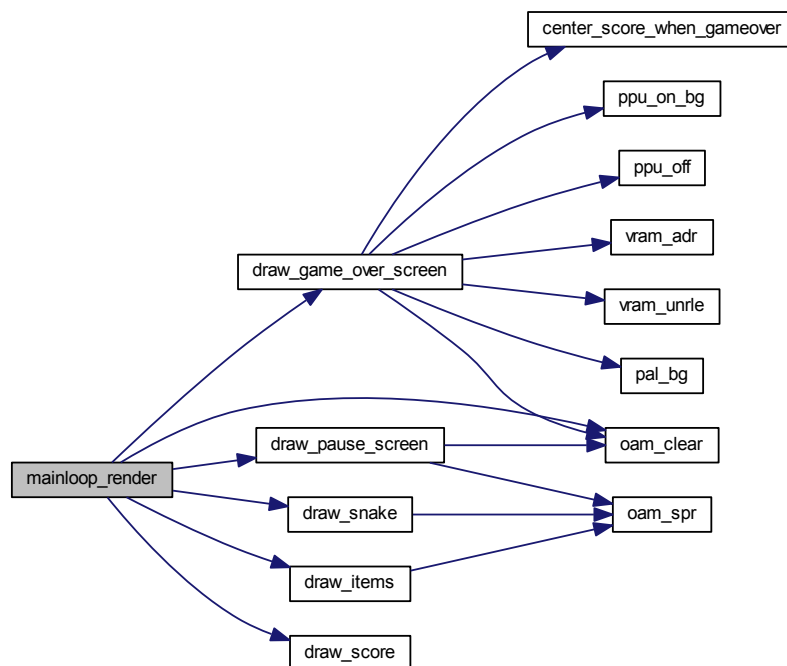
```
void mainloop_render (  
    void )
```

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:



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

Maingame file, containing the main game loop.

```
#include "levell_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"
#include "neslib.h"
#include "macros.h"
#include "structures.h"
#include "globals.h"
#include "init.c"
#include "input.c"
#include "update.c"
#include "render.c"
```

Functions

- void `main` (void)
Main game loop.

5.15.1 Detailed Description

Maingame file, containing the main game loop.

Author

Sebastian Dine.

5.15.2 Function Documentation

5.15.2.1 main()

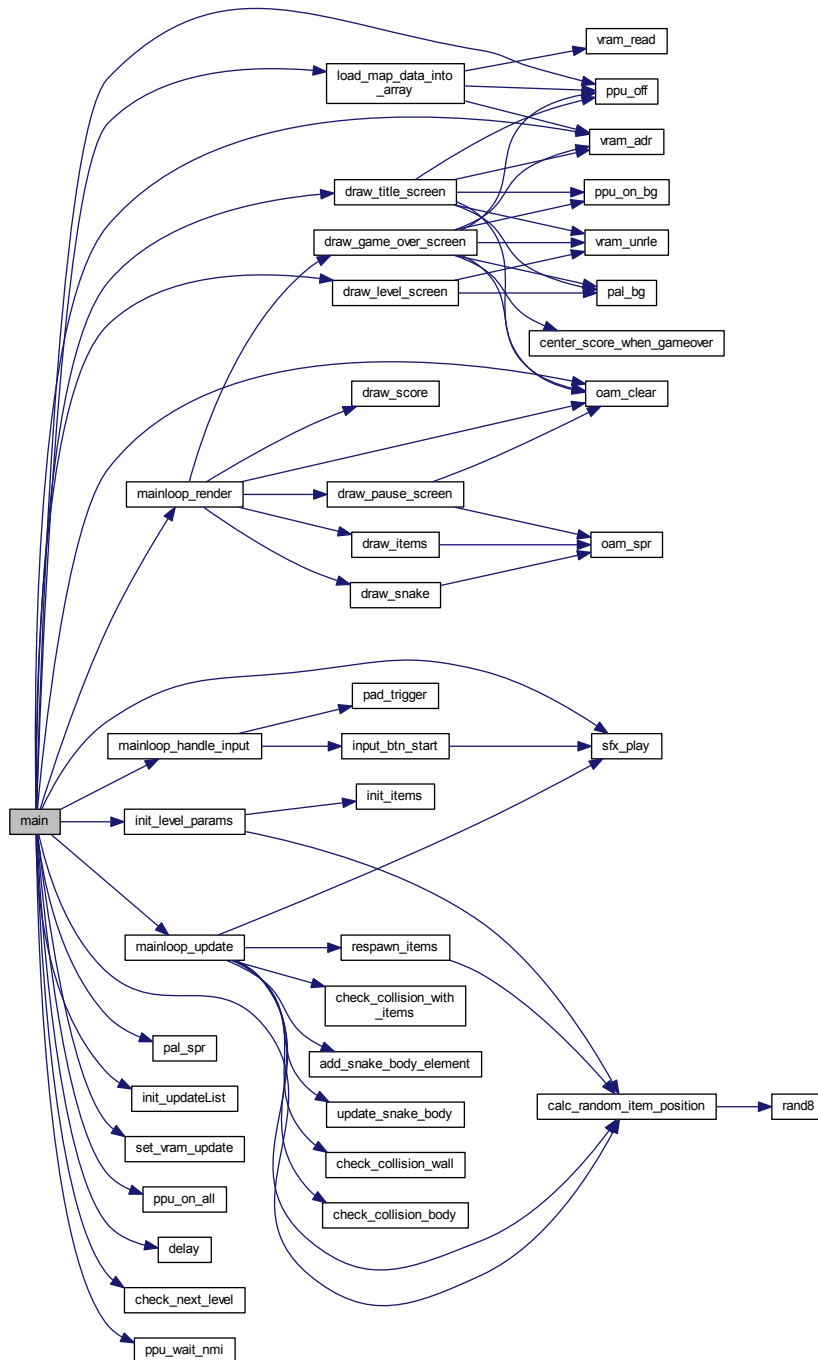
```
void main (
    void )
```

Main game loop.

Author

Sebastian Dine

Here is the call graph for this function:



5.16 C:/Users/Administrator/Documents/GitHub/NES-Snake/src/structures.h File Reference

This header file contains the definition of structures, created for the purpose of the game.

Data Structures

- struct [snake_struct](#)

This structure contains all elements required to interact and display the snake.

- struct [items_struct](#)

This structure contains all elements required to interact with and display items.

5.16.1 Detailed Description

This header file contains the definition of structures, created for the purpose of the game.

Author

Sebastian Dine

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

This file contains all ingame logic functionalities and utility functionalities.

Functions

- void [update_snake_body](#) ()
- void [add_snake_body_element](#) ()
- unsigned char [check_collision_wall](#) (void)
- unsigned char [check_collision_body](#) (void)
- unsigned char [check_collision_with_items](#) (void)
- void [respawn_items](#) (void)
- unsigned char [check_next_level](#) (void)
- void [mainloop_update](#) (void)

5.17.1 Detailed Description

This file contains all ingame logic functionalities and utility functionalities.

Author

Sebastian Dine

5.17.2 Function Documentation

5.17.2.1 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:



5.17.2.2 check_collision_body()

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

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

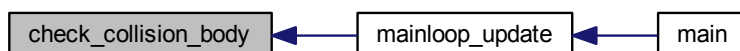
Returns

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

Author

Sebastian Dine

Here is the caller graph for this function:



5.17.2.3 check_collision_wall()

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

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

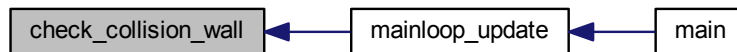
Returns

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

Author

Sebastian Dine

Here is the caller graph for this function:



5.17.2.4 check_collision_with_items()

```
unsigned char check_collision_with_items (  
    void )
```

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

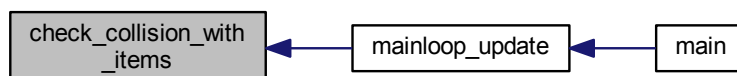
Returns

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

Author

Sebastian Dine

Here is the caller graph for this function:



5.17.2.5 check_next_level()

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

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

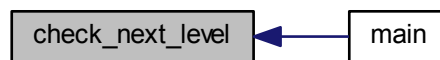
Returns

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

Author

Sebastian Dine

Here is the caller graph for this function:



5.17.2.6 mainloop_update()

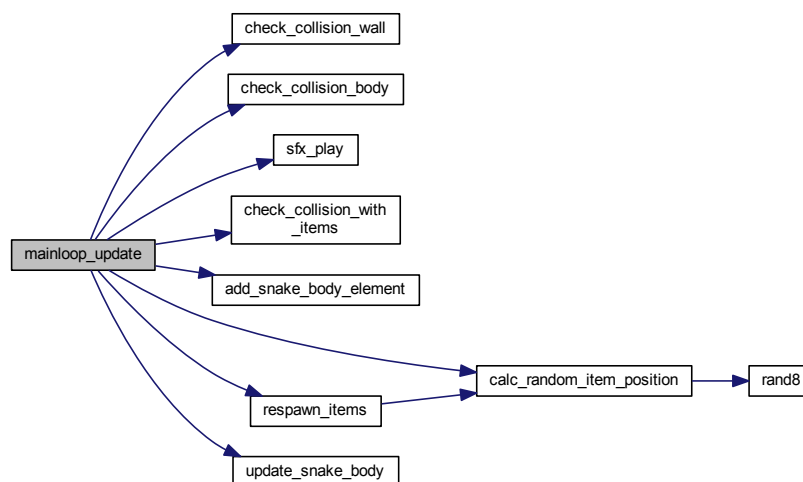
```
void mainloop_update (  
    void )
```

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

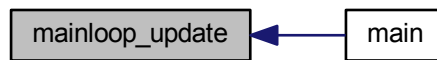
Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



5.17.2.7 respawn_items()

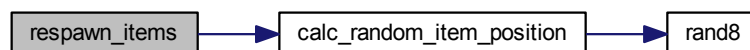
```
void respawn_items (  
    void )
```

This function counts down the frame rate for items until they respawn and calculates the new position in case of a respawn.

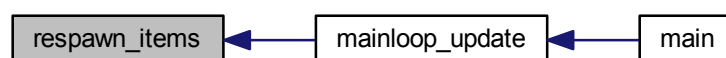
Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



5.17.2.8 update_snake_body()

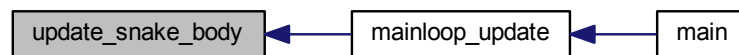
```
void update_snake_body ( )
```

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, [57](#)

bank_bg
 neslib.h, [21](#)

bank_spr
 neslib.h, [21](#)

body_element_coordinates
 snake_struct, [9](#)

C:/Users/Administrator/Documents/GitHub/NES-
 Snake/NESLibrary/neslib.h, [16](#)

C:/Users/Administrator/Documents/GitHub/NES-
 Snake/README.md, [31](#)

C:/Users/Administrator/Documents/GitHub/NES-
 Snake/gfx/game_over_nam.h, [11](#)

C:/Users/Administrator/Documents/GitHub/NES-
 Snake/gfx/level1_nam.h, [12](#)

C:/Users/Administrator/Documents/GitHub/NES-
 Snake/gfx/level2_nam.h, [12](#)

C:/Users/Administrator/Documents/GitHub/NES-
 Snake/gfx/levels_pal.h, [13](#)

C:/Users/Administrator/Documents/GitHub/NES-
 Snake/gfx/menue_pal.h, [14](#)

C:/Users/Administrator/Documents/GitHub/NES-
 Snake/gfx/sprites_pal.h, [15](#)

C:/Users/Administrator/Documents/GitHub/NES-
 Snake/gfx/titlescreen_nam.h, [15](#)

C:/Users/Administrator/Documents/GitHub/NES-
 Snake/src/globals.h, [31](#)

C:/Users/Administrator/Documents/GitHub/NES-
 Snake/src/init.c, [36](#)

C:/Users/Administrator/Documents/GitHub/NES-
 Snake/src/input.c, [39](#)

C:/Users/Administrator/Documents/GitHub/NES-
 Snake/src/macros.h, [40](#)

C:/Users/Administrator/Documents/GitHub/NES-
 Snake/src/render.c, [45](#)

C:/Users/Administrator/Documents/GitHub/NES-
 Snake/src/snake.c, [53](#)

C:/Users/Administrator/Documents/GitHub/NES-
 Snake/src/structures.h, [55](#)

C:/Users/Administrator/Documents/GitHub/NES-
 Snake/src/update.c, [56](#)

calc_random_item_position
 init.c, [36](#)

center_score_when_gameover
 render.c, [45](#)

check_collision_body
 update.c, [57](#)

check_collision_wall
 update.c, [57](#)

check_collision_with_items
 update.c, [58](#)

check_next_level
 update.c, [58](#)

coord_x
 globals.h, [32](#)

coord_y
 globals.h, [32](#)

current_level
 globals.h, [32](#)

DIGIT_O_TILE
 macros.h, [41](#)

DIR_DOWN
 macros.h, [41](#)

DIR_LEFT
 macros.h, [42](#)

DIR_RIGHT
 macros.h, [42](#)

DIR_UP
 macros.h, [42](#)

delay
 neslib.h, [21](#)

draw_game_over_screen
 render.c, [45](#)

draw_items
 render.c, [46](#)

draw_level_screen
 render.c, [47](#)

draw_pause_screen
 render.c, [48](#)

draw_score
 render.c, [49](#)

draw_snake
 render.c, [49](#)

draw_title_screen
 render.c, [50](#)

EMPTY_TILE
 macros.h, [42](#)

FALSE
 neslib.h, [18](#)

flush_vram_update
 neslib.h, [21](#)

game_over_nam
 game_over_nam.h, [11](#)

- game_over_nam.h
 - game_over_nam, 11
- gameover
 - globals.h, 33
- gameover_loop
 - globals.h, 33
- globals.h
 - coord_x, 32
 - coord_y, 32
 - current_level, 32
 - gameover, 33
 - gameover_loop, 33
 - i, 33
 - input, 33
 - items, 33
 - j, 33
 - k, 33
 - l, 33
 - levelList, 34
 - map, 34
 - max_score, 34
 - nameRow, 34
 - nametable_fetch, 34
 - pause, 34
 - pause_loop, 34
 - restart, 35
 - snake, 35
 - sprite_offset, 35
 - titlescreen, 35
 - ul, 35
 - update_list, 35
- head_sprite
 - snake_struct, 9
- head_sprite_attribute
 - snake_struct, 9
- head_sprite_x
 - snake_struct, 9
- head_sprite_y
 - snake_struct, 9
- i
 - globals.h, 33
- ITEM_MAX_ON_SCREEN
 - macros.h, 42
- init.c
 - calc_random_item_position, 36
 - init_items, 37
 - init_level_params, 37
 - load_map_data_into_array, 38
- init_items
 - init.c, 37
- init_level_params
 - init.c, 37
- init_updateList
 - render.c, 51
- input
 - globals.h, 33
- input.c
 - input_btn_start, 39
 - mainloop_handle_input, 39
- input_btn_start
 - input.c, 39
- item_collision_flags
 - items_struct, 7
- item_coordinates
 - items_struct, 7
- item_respawn_count
 - items_struct, 8
- item_respawn_frm_rate
 - items_struct, 8
- items
 - globals.h, 33
- items_struct, 7
 - item_collision_flags, 7
 - item_coordinates, 7
 - item_respawn_count, 8
 - item_respawn_frm_rate, 8
- j
 - globals.h, 33
- k
 - globals.h, 33
- l
 - globals.h, 33
- LEVELS_ALL
 - macros.h, 42
- LVL1_MAX_SCORE
 - macros.h, 42
- LVL1_START_X
 - macros.h, 42
- LVL1_START_Y
 - macros.h, 42
- LVL2_MAX_SCORE
 - macros.h, 43
- LVL2_START_X
 - macros.h, 43
- LVL2_START_Y
 - macros.h, 43
- last_body_element_x
 - snake_struct, 9
- last_body_element_y
 - snake_struct, 9
- level1_nam
 - level1_nam.h, 12
- level1_nam.h
 - level1_nam, 12
- level2_nam
 - level2_nam.h, 13
- level2_nam.h
 - level2_nam, 13
- levelList
 - globals.h, 34
- levels_pal
 - levels_pal.h, 14
- levels_pal.h

- levels_pal, [14](#)
- load_map_data_into_array
 - init.c, [38](#)
- MAP_HEIGHT
 - macros.h, [43](#)
- MAP_WIDTH
 - macros.h, [43](#)
- MAPARRAY_ADR
 - macros.h, [43](#)
- MASK_BG
 - neslib.h, [18](#)
- MASK_EDGE_BG
 - neslib.h, [18](#)
- MASK_EDGE_SPR
 - neslib.h, [18](#)
- MASK_SPR
 - neslib.h, [18](#)
- MAX
 - neslib.h, [18](#)
- MIN
 - neslib.h, [18](#)
- MSB
 - neslib.h, [18](#)
- macros.h
 - DIGIT_O_TILE, [41](#)
 - DIR_DOWN, [41](#)
 - DIR_LEFT, [42](#)
 - DIR_RIGHT, [42](#)
 - DIR_UP, [42](#)
 - EMPTY_TILE, [42](#)
 - ITEM_MAX_ON_SCREEN, [42](#)
 - LEVELS_ALL, [42](#)
 - LVL1_MAX_SCORE, [42](#)
 - LVL1_START_X, [42](#)
 - LVL1_START_Y, [42](#)
 - LVL2_MAX_SCORE, [43](#)
 - LVL2_START_X, [43](#)
 - LVL2_START_Y, [43](#)
 - MAP_HEIGHT, [43](#)
 - MAP_WIDTH, [43](#)
 - MAPARRAY_ADR, [43](#)
 - NAMETABLE1_START, [43](#)
 - SFX_ABSORB_ITEM, [43](#)
 - SFX_CRASH, [43](#)
 - SFX_NEXT_LEVEL, [44](#)
 - SFX_PAUSE, [44](#)
 - SNAKE_BODY_TILE, [44](#)
 - SNAKE_HEAD_TILE_HORZ, [44](#)
 - SNAKE_HEAD_TILE_VERT, [44](#)
 - SNAKE_MAX_SIZE, [44](#)
 - SPIDER_TILE, [44](#)
 - WALL_TILE_1, [44](#)
 - WALL_TILE_2, [44](#)
- main
 - snake.c, [54](#)
- mainloop_handle_input
 - input.c, [39](#)
- mainloop_render
 - render.c, [52](#)
- mainloop_update
 - update.c, [59](#)
- map
 - globals.h, [34](#)
- max_score
 - globals.h, [34](#)
- memcpy
 - neslib.h, [22](#)
- memfill
 - neslib.h, [22](#)
- menue_pal
 - menue_pal.h, [14](#)
- menue_pal.h
 - menue_pal, [14](#)
- moving_direction
 - snake_struct, [9](#)
- music_pause
 - neslib.h, [22](#)
- music_play
 - neslib.h, [22](#)
- music_stop
 - neslib.h, [22](#)
- NAMETABLE1_START
 - macros.h, [43](#)
- NAMETABLE_A
 - neslib.h, [18](#)
- NAMETABLE_B
 - neslib.h, [18](#)
- NAMETABLE_C
 - neslib.h, [19](#)
- NAMETABLE_D
 - neslib.h, [19](#)
- NT_UPD_EOF
 - neslib.h, [19](#)
- NT_UPD_HORZ
 - neslib.h, [19](#)
- NT_UPD_VERT
 - neslib.h, [19](#)
- NTADR_A
 - neslib.h, [19](#)
- NTADR_B
 - neslib.h, [19](#)
- NTADR_C
 - neslib.h, [19](#)
- NTADR_D
 - neslib.h, [19](#)
- NULL
 - neslib.h, [20](#)
- nameRow
 - globals.h, [34](#)
- nametable_fetch
 - globals.h, [34](#)
- neslib.h
 - bank_bg, [21](#)
 - bank_spr, [21](#)
 - delay, [21](#)
 - FALSE, [18](#)

flush_vram_update, 21
 MASK_BG, 18
 MASK_EDGE_BG, 18
 MASK_EDGE_SPR, 18
 MASK_SPR, 18
 MAX, 18
 MIN, 18
 MSB, 18
 memcpy, 22
 memfill, 22
 music_pause, 22
 music_play, 22
 music_stop, 22
 NAMETABLE_A, 18
 NAMETABLE_B, 18
 NAMETABLE_C, 19
 NAMETABLE_D, 19
 NT_UPD_EOF, 19
 NT_UPD_HORZ, 19
 NT_UPD_VERT, 19
 NTADR_A, 19
 NTADR_B, 19
 NTADR_C, 19
 NTADR_D, 19
 NULL, 20
 OAM_BEHIND, 20
 OAM_FLIP_H, 20
 OAM_FLIP_V, 20
 oam_clear, 22
 oam_hide_rest, 23
 oam_meta_spr, 23
 oam_size, 23
 oam_spr, 23
 PAD_DOWN, 20
 PAD_LEFT, 20
 PAD_RIGHT, 20
 PAD_SELECT, 20
 PAD_START, 21
 PAD_UP, 21
 PAD_A, 20
 PAD_B, 20
 pad_poll, 24
 pad_state, 24
 pad_trigger, 24
 pal_all, 24
 pal_bg, 25
 pal_bg_bright, 25
 pal_bright, 25
 pal_clear, 25
 pal_col, 25
 pal_spr, 25
 pal_spr_bright, 26
 ppu_mask, 26
 ppu_off, 26
 ppu_on_all, 26
 ppu_on_bg, 27
 ppu_on_spr, 27
 ppu_system, 27
 ppu_wait_frame, 27
 ppu_wait_nmi, 27
 rand16, 28
 rand8, 28
 sample_play, 28
 scroll, 28
 set_rand, 28
 set_vram_update, 29
 sfx_play, 29
 split, 29
 TRUE, 21
 vram_adr, 29
 vram_fill, 30
 vram_inc, 30
 vram_put, 30
 vram_read, 30
 vram_unrle, 30
 vram_write, 31
 OAM_BEHIND
 neslib.h, 20
 OAM_FLIP_H
 neslib.h, 20
 OAM_FLIP_V
 neslib.h, 20
 oam_clear
 neslib.h, 22
 oam_hide_rest
 neslib.h, 23
 oam_meta_spr
 neslib.h, 23
 oam_size
 neslib.h, 23
 oam_spr
 neslib.h, 23
 PAD_DOWN
 neslib.h, 20
 PAD_LEFT
 neslib.h, 20
 PAD_RIGHT
 neslib.h, 20
 PAD_SELECT
 neslib.h, 20
 PAD_START
 neslib.h, 21
 PAD_UP
 neslib.h, 21
 PAD_A
 neslib.h, 20
 PAD_B
 neslib.h, 20
 pad_poll
 neslib.h, 24
 pad_state
 neslib.h, 24
 pad_trigger
 neslib.h, 24
 pal_all

- neslib.h, [24](#)
- pal_bg
 - neslib.h, [25](#)
- pal_bg_bright
 - neslib.h, [25](#)
- pal_bright
 - neslib.h, [25](#)
- pal_clear
 - neslib.h, [25](#)
- pal_col
 - neslib.h, [25](#)
- pal_spr
 - neslib.h, [25](#)
- pal_spr_bright
 - neslib.h, [26](#)
- pause
 - globals.h, [34](#)
- pause_loop
 - globals.h, [34](#)
- ppu_mask
 - neslib.h, [26](#)
- ppu_off
 - neslib.h, [26](#)
- ppu_on_all
 - neslib.h, [26](#)
- ppu_on_bg
 - neslib.h, [27](#)
- ppu_on_spr
 - neslib.h, [27](#)
- ppu_system
 - neslib.h, [27](#)
- ppu_wait_frame
 - neslib.h, [27](#)
- ppu_wait_nmi
 - neslib.h, [27](#)
- rand16
 - neslib.h, [28](#)
- rand8
 - neslib.h, [28](#)
- render.c
 - center_score_when_gameover, [45](#)
 - draw_game_over_screen, [45](#)
 - draw_items, [46](#)
 - draw_level_screen, [47](#)
 - draw_pause_screen, [48](#)
 - draw_score, [49](#)
 - draw_snake, [49](#)
 - draw_title_screen, [50](#)
 - init_updateList, [51](#)
 - mainloop_render, [52](#)
- respawn_items
 - update.c, [60](#)
- restart
 - globals.h, [35](#)
- SFX_ABSORB_ITEM
 - macros.h, [43](#)
- SFX_CRASH
 - macros.h, [43](#)
- SFX_NEXT_LEVEL
 - macros.h, [44](#)
- SFX_PAUSE
 - macros.h, [44](#)
- SNAKE_BODY_TILE
 - macros.h, [44](#)
- SNAKE_HEAD_TILE_HORZ
 - macros.h, [44](#)
- SNAKE_HEAD_TILE_VERT
 - macros.h, [44](#)
- SNAKE_MAX_SIZE
 - macros.h, [44](#)
- SPIDER_TILE
 - macros.h, [44](#)
- sample_play
 - neslib.h, [28](#)
- scroll
 - neslib.h, [28](#)
- set_rand
 - neslib.h, [28](#)
- set_vram_update
 - neslib.h, [29](#)
- sfx_play
 - neslib.h, [29](#)
- size_index
 - snake_struct, [10](#)
- snake
 - globals.h, [35](#)
- snake.c
 - main, [54](#)
- snake_struct, [8](#)
 - body_element_coordinates, [9](#)
 - head_sprite, [9](#)
 - head_sprite_attribute, [9](#)
 - head_sprite_x, [9](#)
 - head_sprite_y, [9](#)
 - last_body_element_x, [9](#)
 - last_body_element_y, [9](#)
 - moving_direction, [9](#)
 - size_index, [10](#)
 - speed_counter, [10](#)
- speed_counter
 - snake_struct, [10](#)
- split
 - neslib.h, [29](#)
- sprite_offset
 - globals.h, [35](#)
- sprites_pal
 - sprites_pal.h, [15](#)
- sprites_pal.h
 - sprites_pal, [15](#)
- TRUE
 - neslib.h, [21](#)
- titlescreen
 - globals.h, [35](#)
- titlescreen_nam
 - titlescreen_nam.h, [16](#)

- titlescreen_nam.h
 - titlescreen_nam, [16](#)
- ul
 - globals.h, [35](#)
- update.c
 - add_snake_body_element, [57](#)
 - check_collision_body, [57](#)
 - check_collision_wall, [57](#)
 - check_collision_with_items, [58](#)
 - check_next_level, [58](#)
 - mainloop_update, [59](#)
 - respawn_items, [60](#)
 - update_snake_body, [60](#)
- update_list
 - globals.h, [35](#)
- update_snake_body
 - update.c, [60](#)
- vram_adr
 - neslib.h, [29](#)
- vram_fill
 - neslib.h, [30](#)
- vram_inc
 - neslib.h, [30](#)
- vram_put
 - neslib.h, [30](#)
- vram_read
 - neslib.h, [30](#)
- vram_unrle
 - neslib.h, [30](#)
- vram_write
 - neslib.h, [31](#)
- WALL_TILE_1
 - macros.h, [44](#)
- WALL_TILE_2
 - macros.h, [45](#)