

NES Snake

Generated by Doxygen 1.8.12



# Contents

<b>1</b>	<b>README</b>	<b>1</b>
<b>2</b>	<b>Data Structure Index</b>	<b>3</b>
2.1	Data Structures . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Data Structure Documentation</b>	<b>7</b>
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 . . . . .	9
4.2.2	Field Documentation . . . . .	9
4.2.2.1	body_element_coordinates . . . . .	9
4.2.2.2	grow_event_flag . . . . .	9
4.2.2.3	head_sprite . . . . .	9
4.2.2.4	head_sprite_attribute . . . . .	9
4.2.2.5	head_sprite_x . . . . .	9
4.2.2.6	head_sprite_y . . . . .	9
4.2.2.7	last_body_element_x . . . . .	10
4.2.2.8	last_body_element_y . . . . .	10
4.2.2.9	moving_direction . . . . .	10
4.2.2.10	size_index . . . . .	10
4.2.2.11	speed_counter . . . . .	10

<b>5</b>	<b>File Documentation</b>	<b>11</b>
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/level3_nam.h File Reference . . . . .	13
5.4.1	Detailed Description . . . . .	13
5.4.2	Variable Documentation . . . . .	14
5.4.2.1	level3_nam . . . . .	14
5.5	C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/level4_nam.h File Reference . . . . .	14
5.5.1	Detailed Description . . . . .	14
5.5.2	Variable Documentation . . . . .	15
5.5.2.1	level4_nam . . . . .	15
5.6	C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/level5_nam.h File Reference . . . . .	15
5.6.1	Detailed Description . . . . .	15
5.6.2	Variable Documentation . . . . .	15
5.6.2.1	level5_nam . . . . .	15
5.7	C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/levels_pal.h File Reference . . . . .	16
5.7.1	Detailed Description . . . . .	16
5.7.2	Variable Documentation . . . . .	16
5.7.2.1	levels_pal . . . . .	16
5.8	C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/menue_pal.h File Reference . . . . .	16

5.8.1	Detailed Description	17
5.8.2	Variable Documentation	17
5.8.2.1	menue_pal	17
5.9	C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/sprites_pal.h File Reference	17
5.9.1	Detailed Description	17
5.9.2	Variable Documentation	17
5.9.2.1	sprites_pal	17
5.10	C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/titlescreen_nam.h File Reference	18
5.10.1	Detailed Description	18
5.10.2	Variable Documentation	18
5.10.2.1	titlescreen_nam	18
5.11	C:/Users/Administrator/Documents/GitHub/NES-Snake/NESLibrary/neslib.h File Reference	19
5.11.1	Macro Definition Documentation	20
5.11.1.1	FALSE	20
5.11.1.2	MASK_BG	20
5.11.1.3	MASK_EDGE_BG	21
5.11.1.4	MASK_EDGE_SPR	21
5.11.1.5	MASK_SPR	21
5.11.1.6	MAX	21
5.11.1.7	MIN	21
5.11.1.8	MSB	21
5.11.1.9	NAMETABLE_A	21
5.11.1.10	NAMETABLE_B	21
5.11.1.11	NAMETABLE_C	21
5.11.1.12	NAMETABLE_D	22
5.11.1.13	NT_UPD_EOF	22
5.11.1.14	NT_UPD_HORZ	22
5.11.1.15	NT_UPD_VERT	22
5.11.1.16	NTADR_A	22
5.11.1.17	NTADR_B	22

5.11.1.18 NTADR_C . . . . .	22
5.11.1.19 NTADR_D . . . . .	22
5.11.1.20 NULL . . . . .	22
5.11.1.21 OAM_BEHIND . . . . .	23
5.11.1.22 OAM_FLIP_H . . . . .	23
5.11.1.23 OAM_FLIP_V . . . . .	23
5.11.1.24 PAD_A . . . . .	23
5.11.1.25 PAD_B . . . . .	23
5.11.1.26 PAD_DOWN . . . . .	23
5.11.1.27 PAD_LEFT . . . . .	23
5.11.1.28 PAD_RIGHT . . . . .	23
5.11.1.29 PAD_SELECT . . . . .	23
5.11.1.30 PAD_START . . . . .	23
5.11.1.31 PAD_UP . . . . .	24
5.11.1.32 TRUE . . . . .	24
5.11.2 Function Documentation . . . . .	24
5.11.2.1 bank_bg() . . . . .	24
5.11.2.2 bank_spr() . . . . .	24
5.11.2.3 delay() . . . . .	24
5.11.2.4 flush_vram_update() . . . . .	24
5.11.2.5 memcpy() . . . . .	24
5.11.2.6 memfill() . . . . .	25
5.11.2.7 music_pause() . . . . .	25
5.11.2.8 music_play() . . . . .	25
5.11.2.9 music_stop() . . . . .	25
5.11.2.10 oam_clear() . . . . .	25
5.11.2.11 oam_hide_rest() . . . . .	25
5.11.2.12 oam_meta_spr() . . . . .	26
5.11.2.13 oam_size() . . . . .	26
5.11.2.14 oam_spr() . . . . .	26

5.11.2.15	<code>pad_poll()</code>	26
5.11.2.16	<code>pad_state()</code>	26
5.11.2.17	<code>pad_trigger()</code>	27
5.11.2.18	<code>pal_all()</code>	27
5.11.2.19	<code>pal_bg()</code>	27
5.11.2.20	<code>pal_bg_bright()</code>	27
5.11.2.21	<code>pal_bright()</code>	27
5.11.2.22	<code>pal_clear()</code>	28
5.11.2.23	<code>pal_col()</code>	28
5.11.2.24	<code>pal_spr()</code>	28
5.11.2.25	<code>pal_spr_bright()</code>	28
5.11.2.26	<code>ppu_mask()</code>	28
5.11.2.27	<code>ppu_off()</code>	29
5.11.2.28	<code>ppu_on_all()</code>	29
5.11.2.29	<code>ppu_on_bg()</code>	29
5.11.2.30	<code>ppu_on_spr()</code>	30
5.11.2.31	<code>ppu_system()</code>	30
5.11.2.32	<code>ppu_wait_frame()</code>	30
5.11.2.33	<code>ppu_wait_nmi()</code>	30
5.11.2.34	<code>rand16()</code>	30
5.11.2.35	<code>rand8()</code>	30
5.11.2.36	<code>sample_play()</code>	31
5.11.2.37	<code>scroll()</code>	31
5.11.2.38	<code>set_rand()</code>	31
5.11.2.39	<code>set_vram_update()</code>	31
5.11.2.40	<code>sfx_play()</code>	31
5.11.2.41	<code>split()</code>	32
5.11.2.42	<code>vram_adr()</code>	32
5.11.2.43	<code>vram_fill()</code>	32
5.11.2.44	<code>vram_inc()</code>	32

5.11.2.45 vram_put()	32
5.11.2.46 vram_read()	33
5.11.2.47 vram_unrle()	33
5.11.2.48 vram_write()	33
5.12 C:/Users/Administrator/Documents/GitHub/NES-Snake/README.md File Reference	33
5.13 C:/Users/Administrator/Documents/GitHub/NES-Snake/src/globals.h File Reference	33
5.13.1 Detailed Description	35
5.13.2 Variable Documentation	35
5.13.2.1 coord_x	35
5.13.2.2 coord_y	35
5.13.2.3 current_level	35
5.13.2.4 gameover	35
5.13.2.5 gameover_loop	35
5.13.2.6 i	35
5.13.2.7 input	35
5.13.2.8 items	36
5.13.2.9 j	36
5.13.2.10 k	36
5.13.2.11 l	36
5.13.2.12 levelList	36
5.13.2.13 map	36
5.13.2.14 max_score	36
5.13.2.15 nameRow	37
5.13.2.16 nametable_fetch	37
5.13.2.17 pause	37
5.13.2.18 pause_loop	37
5.13.2.19 render_movement_flag	37
5.13.2.20 restart	37
5.13.2.21 snake	37
5.13.2.22 sprite_offset	37



5.13.2.23	titlescreen	38
5.13.2.24	ul	38
5.13.2.25	update_list	38
5.14	C:/Users/Administrator/Documents/GitHub/NES-Snake/src/init.c File Reference	38
5.14.1	Detailed Description	38
5.14.2	Function Documentation	39
5.14.2.1	calc_random_item_position()	39
5.14.2.2	init_items()	39
5.14.2.3	init_level_params()	40
5.14.2.4	load_map_data_into_array()	40
5.15	C:/Users/Administrator/Documents/GitHub/NES-Snake/src/input.c File Reference	41
5.15.1	Detailed Description	41
5.15.2	Function Documentation	42
5.15.2.1	input_btn_start()	42
5.15.2.2	mainloop_handle_input()	42
5.16	C:/Users/Administrator/Documents/GitHub/NES-Snake/src/macros.h File Reference	43
5.16.1	Detailed Description	44
5.16.2	Macro Definition Documentation	45
5.16.2.1	DIGIT_O_TILE	45
5.16.2.2	DIR_DOWN	45
5.16.2.3	DIR_LEFT	45
5.16.2.4	DIR_RIGHT	45
5.16.2.5	DIR_UP	45
5.16.2.6	EMPTY_TILE	45
5.16.2.7	ITEM_MAX_ON_SCREEN	45
5.16.2.8	LEVELS_ALL	45
5.16.2.9	LVL1_MAX_SCORE	46
5.16.2.10	LVL1_START_X	46
5.16.2.11	LVL1_START_Y	46
5.16.2.12	LVL2_MAX_SCORE	46

5.16.2.13 LVL2_START_X . . . . .	46
5.16.2.14 LVL2_START_Y . . . . .	46
5.16.2.15 LVL3_MAX_SCORE . . . . .	46
5.16.2.16 LVL3_START_X . . . . .	46
5.16.2.17 LVL3_START_Y . . . . .	46
5.16.2.18 LVL4_MAX_SCORE . . . . .	46
5.16.2.19 LVL4_START_X . . . . .	47
5.16.2.20 LVL4_START_Y . . . . .	47
5.16.2.21 LVL5_MAX_SCORE . . . . .	47
5.16.2.22 LVL5_START_X . . . . .	47
5.16.2.23 LVL5_START_Y . . . . .	47
5.16.2.24 MAP_HEIGHT . . . . .	47
5.16.2.25 MAP_WIDTH . . . . .	47
5.16.2.26 MAPARRAY_ADR . . . . .	47
5.16.2.27 NAMETABLE1_START . . . . .	47
5.16.2.28 NAMETABLE2_START . . . . .	48
5.16.2.29 SFX_ABSORB_ITEM . . . . .	48
5.16.2.30 SFX_CRASH . . . . .	48
5.16.2.31 SFX_NEXT_LEVEL . . . . .	48
5.16.2.32 SFX_PAUSE . . . . .	48
5.16.2.33 SNAKE_BODY_TILE . . . . .	48
5.16.2.34 SNAKE_HEAD_TILE_HORZ . . . . .	48
5.16.2.35 SNAKE_HEAD_TILE_VERT . . . . .	48
5.16.2.36 SNAKE_MAX_SIZE . . . . .	48
5.16.2.37 SPIDER_TILE . . . . .	49
5.16.2.38 SPRITE_ATTR . . . . .	49
5.16.2.39 WALL_TILE_1 . . . . .	49
5.16.2.40 WALL_TILE_2 . . . . .	49
5.17 C:/Users/Administrator/Documents/GitHub/NES-Snake/src/render.c File Reference . . . . .	49
5.17.1 Detailed Description . . . . .	50

5.17.2	Function Documentation	50
5.17.2.1	center_score_when_gameover()	50
5.17.2.2	draw_game_over_screen()	50
5.17.2.3	draw_items()	51
5.17.2.4	draw_level_screen()	52
5.17.2.5	draw_pause_screen()	53
5.17.2.6	draw_score()	54
5.17.2.7	draw_snake()	54
5.17.2.8	draw_title_screen()	55
5.17.2.9	init_updateList()	56
5.17.2.10	mainloop_render()	56
5.18	C:/Users/Administrator/Documents/GitHub/NES-Snake/src/snake.c File Reference	57
5.18.1	Detailed Description	57
5.18.2	Function Documentation	58
5.18.2.1	main()	58
5.19	C:/Users/Administrator/Documents/GitHub/NES-Snake/src/structures.h File Reference	59
5.19.1	Detailed Description	60
5.20	C:/Users/Administrator/Documents/GitHub/NES-Snake/src/update.c File Reference	60
5.20.1	Detailed Description	60
5.20.2	Function Documentation	61
5.20.2.1	add_snake_body_element()	61
5.20.2.2	check_collision_body()	61
5.20.2.3	check_collision_wall()	62
5.20.2.4	check_collision_with_items()	62
5.20.2.5	check_next_level()	63
5.20.2.6	mainloop_update()	63
5.20.2.7	respawn_items()	64
5.20.2.8	update_snake_body()	65



# Chapter 1

## README

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



## Chapter 2

# Data Structure Index

### 2.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">items_struct</a>	This structure contains all elements required to interact with and display items . . . . .	<a href="#">7</a>
<a href="#">snake_struct</a>	This structure contains all elements required to interact and display the snake . . . . .	<a href="#">8</a>





## 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/ <a href="#">game_over_nam.h</a> This header file contains the nametable (background) of the gameover screen. Created with NES Screen Tool 2.04 (Option Nametable -> Save nametable and attributes -> RLE packed as C header (.h) . . . . .	11
C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/ <a href="#">level1_nam.h</a> This header file contains the nametable (background) of level map 1. Created with NES Screen Tool 2.04 (Option Nametable -> Save nametable and attributes -> RLE packed as C header (.h) . . . . .	12
C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/ <a href="#">level2_nam.h</a> This header file contains the nametable (background) of level map 2. Created with NES Screen Tool 2.04 (Option Nametable -> Save nametable and attributes -> RLE packed as C header (.h) . . . . .	12
C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/ <a href="#">level3_nam.h</a> This header file contains the nametable (background) of level map 3. Created with NES Screen Tool 2.04 (Option Nametable -> Save nametable and attributes -> RLE packed as C header (.h) . . . . .	13
C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/ <a href="#">level4_nam.h</a> This header file contains the nametable (background) of level map 4. Created with NES Screen Tool 2.04 (Option Nametable -> Save nametable and attributes -> RLE packed as C header (.h) . . . . .	14
C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/ <a href="#">level5_nam.h</a> This header file contains the nametable (background) of level map 5. 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/gfx/ <a href="#">levels_pal.h</a> 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 . . . . .	16
C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/ <a href="#">menue_pal.h</a> This header file contains the color palette for menus (titlescreen, gameover screen). Created with NES Screen Tool 2.04 (Option Palettes -> Put C data to clipboard . . . . .	16
C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/ <a href="#">sprites_pal.h</a> This header file contains the color palette for sprites . . . . .	17
C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/ <a href="#">titlescreen_nam.h</a> This header file contains the nametable (background) of the titlescreen. Created with NES Screen Tool 2.04 (Option Nametable -> Save nametable and attributes -> RLE packed as C header (.h) . . . . .	18
C:/Users/Administrator/Documents/GitHub/NES-Snake/NESLibrary/ <a href="#">neslib.h</a> . . . . .	19
C:/Users/Administrator/Documents/GitHub/NES-Snake/src/ <a href="#">globals.h</a> This header file defines all global variables of the game . . . . .	33
C:/Users/Administrator/Documents/GitHub/NES-Snake/src/ <a href="#">init.c</a> This file contains functions for initializing game elements . . . . .	38

C:/Users/Administrator/Documents/GitHub/NES-Snake/src/ <a href="#">input.c</a>	
This file contains functions for input handling from a controller . . . . .	41
C:/Users/Administrator/Documents/GitHub/NES-Snake/src/ <a href="#">macros.h</a>	
This header file defines object-like macros (constants) and function-like macros for more efficient calculations . . . . .	43
C:/Users/Administrator/Documents/GitHub/NES-Snake/src/ <a href="#">render.c</a>	
This file contains all functionality to draw onto the screen, either as sprites or as background tiles	49
C:/Users/Administrator/Documents/GitHub/NES-Snake/src/ <a href="#">snake.c</a>	
Maingame file, containing the main game loop . . . . .	57
C:/Users/Administrator/Documents/GitHub/NES-Snake/src/ <a href="#">structures.h</a>	
This header file contains the definition of structures, created for the purpose of the game . . .	59
C:/Users/Administrator/Documents/GitHub/NES-Snake/src/ <a href="#">update.c</a>	
This file contains all ingame logic functionalities and utility functionalities . . . . .	60

## 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 [grow\\_event\\_flag](#)
- 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 grow\_event\_flag

```
unsigned char grow_event_flag
```

Variable for indication, that the snake has been grown

#### 4.2.2.3 head\_sprite

```
unsigned char head_sprite
```

tbd.

#### 4.2.2.4 head\_sprite\_attribute

```
unsigned char head_sprite_attribute
```

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

#### 4.2.2.5 head\_sprite\_x

```
unsigned char head_sprite_x
```

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

#### 4.2.2.6 head\_sprite\_y

```
unsigned char head_sprite_y
```

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

#### 4.2.2.7 last\_body\_element\_x

```
unsigned char last_body_element_x
```

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

#### 4.2.2.8 last\_body\_element\_y

```
unsigned char last_body_element_y
```

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

#### 4.2.2.9 moving\_direction

```
unsigned char moving_direction
```

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

#### 4.2.2.10 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.

#### 4.2.2.11 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](#) [178]

### 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[178]
```

#### Initial value:

```
= {
0x01, 0x00, 0x01, 0x20, 0x33, 0x23, 0x2f, 0x32, 0x25, 0x1a, 0x10, 0x01, 0x02, 0x0f, 0x10, 0x11,
0x10, 0x00, 0x01, 0x31, 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](#) [274]



### 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[274]
```

#### Initial value:

```
= {
0x01, 0x00, 0x01, 0x20, 0x33, 0x23, 0x2f, 0x32, 0x25, 0x1a, 0x00, 0x01, 0x02, 0x0f, 0x10, 0x12,
0x10, 0x00, 0x01, 0x31, 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, 0x00, 0x01, 0x3f,
0x01, 0x00
}
```

## 5.4 C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/level3\_nam.h File Reference

This header file contains the nametable (background) of level map 3. Created with NES Screen Tool 2.04 (Option Nametable -> Save nametable and attributes -> RLE packed as C header (.h)).

#### Variables

- const unsigned char [level3\\_nam](#) [363]

### 5.4.1 Detailed Description

This header file contains the nametable (background) of level map 3. Created with NES Screen Tool 2.04 (Option Nametable -> Save nametable and attributes -> RLE packed as C header (.h)).

#### Author

Sebastian Dine

## 5.4.2 Variable Documentation

### 5.4.2.1 level3\_nam

```
const unsigned char level3_nam[363]
```

**Initial value:**

```
= {
0x01, 0x00, 0x01, 0x20, 0x33, 0x23, 0x2f, 0x32, 0x25, 0x1a, 0x10, 0x01, 0x02, 0x0f, 0x10, 0x13,
0x10, 0x00, 0x01, 0x31, 0x43, 0x01, 0x3d, 0x44, 0x44, 0x43, 0x43, 0x00, 0x01, 0x07, 0x43, 0x00,
0x01, 0x12, 0x43, 0x01, 0x03, 0x00, 0x01, 0x07, 0x43, 0x00, 0x01, 0x12, 0x43, 0x01, 0x03, 0x00,
0x01, 0x07, 0x43, 0x00, 0x01, 0x0a, 0x43, 0x00, 0x01, 0x06, 0x43, 0x01, 0x03, 0x00, 0x01, 0x07,
0x43, 0x00, 0x01, 0x0a, 0x43, 0x00, 0x01, 0x06, 0x43, 0x01, 0x03, 0x00, 0x01, 0x07, 0x43, 0x00,
0x01, 0x0a, 0x43, 0x00, 0x01, 0x06, 0x43, 0x01, 0x03, 0x00, 0x01, 0x07, 0x43, 0x00, 0x01, 0x0a,
0x43, 0x00, 0x01, 0x06, 0x43, 0x01, 0x03, 0x00, 0x01, 0x07, 0x43, 0x00, 0x01, 0x0a, 0x43, 0x00,
0x01, 0x06, 0x43, 0x01, 0x03, 0x00, 0x01, 0x07, 0x43, 0x00, 0x01, 0x0a, 0x43, 0x00, 0x01, 0x06,
0x43, 0x01, 0x03, 0x00, 0x01, 0x13, 0x43, 0x00, 0x01, 0x06, 0x43, 0x01, 0x03, 0x00, 0x01, 0x0c,
0x43, 0x01, 0x03, 0x00, 0x01, 0x02, 0x43, 0x00, 0x01, 0x06, 0x43, 0x01, 0x03, 0x00, 0x01, 0x07,
0x43, 0x00, 0x01, 0x03, 0x43, 0x00, 0x00, 0x43, 0x00, 0x01, 0x02, 0x43, 0x00, 0x01, 0x06, 0x43,
0x01, 0x03, 0x00, 0x01, 0x07, 0x43, 0x00, 0x01, 0x03, 0x43, 0x00, 0x00, 0x43, 0x00, 0x01, 0x02,
0x43, 0x00, 0x01, 0x06, 0x43, 0x01, 0x03, 0x00, 0x01, 0x07, 0x43, 0x00, 0x01, 0x03, 0x43, 0x00,
0x00, 0x43, 0x00, 0x01, 0x02, 0x43, 0x00, 0x01, 0x06, 0x43, 0x01, 0x03, 0x00, 0x01, 0x07, 0x43,
0x00, 0x01, 0x03, 0x43, 0x01, 0x03, 0x00, 0x01, 0x02, 0x43, 0x00, 0x01, 0x06, 0x43, 0x01, 0x03,
0x00, 0x01, 0x07, 0x43, 0x00, 0x01, 0x0a, 0x43, 0x00, 0x01, 0x06, 0x43, 0x01, 0x03, 0x00, 0x01,
0x13, 0x43, 0x00, 0x01, 0x06, 0x43, 0x01, 0x03, 0x00, 0x01, 0x13, 0x43, 0x00, 0x01, 0x06, 0x43,
0x01, 0x03, 0x00, 0x01, 0x07, 0x43, 0x00, 0x01, 0x0a, 0x43, 0x00, 0x01, 0x06, 0x43, 0x01, 0x03,
0x00, 0x01, 0x07, 0x43, 0x00, 0x01, 0x0a, 0x43, 0x00, 0x01, 0x06, 0x43, 0x01, 0x03, 0x00, 0x01,
0x07, 0x43, 0x00, 0x01, 0x0a, 0x43, 0x00, 0x01, 0x06, 0x43, 0x01, 0x03, 0x00, 0x01, 0x07, 0x43,
0x00, 0x01, 0x0a, 0x43, 0x00, 0x01, 0x06, 0x43, 0x01, 0x03, 0x00, 0x01, 0x07, 0x43, 0x00, 0x01,
0x12, 0x43, 0x01, 0x03, 0x00, 0x01, 0x07, 0x43, 0x00, 0x01, 0x12, 0x43, 0x01, 0x2e, 0x44, 0x43,
0x01, 0x05, 0x44, 0x43, 0x01, 0x0a, 0x00, 0x01, 0x3f, 0x01, 0x00
}
```

## 5.5 C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/level4\_nam.h File Reference

This header file contains the nametable (background) of level map 4. Created with NES Screen Tool 2.04 (Option Nametable -> Save nametable and attributes -> RLE packed as C header (.h)).

### Variables

- const unsigned char [level4\\_nam](#) [383]

### 5.5.1 Detailed Description

This header file contains the nametable (background) of level map 4. Created with NES Screen Tool 2.04 (Option Nametable -> Save nametable and attributes -> RLE packed as C header (.h)).

### Author

Sebastian Dine

## 5.5.2 Variable Documentation

### 5.5.2.1 level4\_nam

```
const unsigned char level4_nam[383]
```

**Initial value:**

```
= {
0x01, 0x00, 0x01, 0x20, 0x33, 0x23, 0x2f, 0x32, 0x25, 0x1a, 0x10, 0x01, 0x02, 0x0f, 0x10, 0x14,
0x10, 0x00, 0x01, 0x31, 0x43, 0x01, 0x3d, 0x44, 0x44, 0x43, 0x43, 0x00, 0x01, 0x0c, 0x43, 0x43,
0x00, 0x01, 0x0c, 0x43, 0x01, 0x03, 0x00, 0x01, 0x0c, 0x43, 0x43, 0x00, 0x01, 0x0c, 0x43, 0x01,
0x03, 0x00, 0x01, 0x0c, 0x43, 0x43, 0x00, 0x01, 0x0c, 0x43, 0x01, 0x03, 0x00, 0x01, 0x0c, 0x43,
0x43, 0x00, 0x01, 0x0c, 0x43, 0x01, 0x0a, 0x00, 0x00, 0x43, 0x01, 0x09, 0x00, 0x00, 0x43, 0x44,
0x43, 0x01, 0x02, 0x44, 0x44, 0x43, 0x01, 0x0a, 0x00, 0x00, 0x43, 0x01, 0x09, 0x00, 0x00, 0x43,
0x44, 0x01, 0x05, 0x43, 0x01, 0x03, 0x00, 0x01, 0x04, 0x43, 0x43, 0x00, 0x01, 0x0d, 0x43, 0x43,
0x00, 0x01, 0x04, 0x43, 0x01, 0x03, 0x00, 0x01, 0x04, 0x43, 0x43, 0x00, 0x01, 0x0d, 0x43, 0x43,
0x00, 0x01, 0x04, 0x43, 0x01, 0x03, 0x00, 0x01, 0x04, 0x43, 0x43, 0x00, 0x01, 0x0d, 0x43, 0x43,
0x00, 0x01, 0x04, 0x43, 0x01, 0x03, 0x00, 0x01, 0x04, 0x43, 0x43, 0x00, 0x01, 0x0d, 0x43, 0x43,
0x02, 0x44, 0x00, 0x01, 0x04, 0x43, 0x43, 0x00, 0x01, 0x04, 0x43, 0x43, 0x01, 0x03, 0x00, 0x01, 0x04,
0x43, 0x43, 0x00, 0x01, 0x04, 0x43, 0x00, 0x00, 0x43, 0x00, 0x01, 0x04, 0x43, 0x43, 0x00, 0x01,
0x04, 0x43, 0x01, 0x03, 0x00, 0x01, 0x04, 0x43, 0x43, 0x00, 0x01, 0x04, 0x43, 0x00, 0x00, 0x43,
0x00, 0x01, 0x04, 0x43, 0x43, 0x00, 0x01, 0x04, 0x43, 0x43, 0x00, 0x01, 0x03, 0x00, 0x01, 0x0b, 0x43, 0x00,
0x00, 0x43, 0x00, 0x01, 0x0b, 0x43, 0x01, 0x03, 0x00, 0x01, 0x04, 0x43, 0x43, 0x00, 0x01, 0x04, 0x43, 0x00, 0x00,
0x43, 0x00, 0x01, 0x04, 0x43, 0x43, 0x00, 0x01, 0x04, 0x43, 0x43, 0x01, 0x03, 0x00, 0x01, 0x0b, 0x43, 0x00,
0x00, 0x43, 0x00, 0x01, 0x0b, 0x43, 0x01, 0x03, 0x00, 0x01, 0x04, 0x43, 0x43, 0x00, 0x01, 0x04, 0x43, 0x00, 0x00,
0x43, 0x00, 0x01, 0x04, 0x43, 0x43, 0x00, 0x01, 0x0d, 0x43, 0x43, 0x00, 0x01, 0x04, 0x43,
0x01, 0x03, 0x00, 0x01, 0x04, 0x43, 0x43, 0x00, 0x01, 0x0d, 0x43, 0x43, 0x00, 0x01, 0x04, 0x43,
0x01, 0x03, 0x00, 0x01, 0x04, 0x43, 0x43, 0x00, 0x01, 0x0d, 0x43, 0x43, 0x00, 0x01, 0x04, 0x43,
0x01, 0x0a, 0x00, 0x00, 0x43, 0x01, 0x09, 0x00, 0x00, 0x43, 0x01, 0x11, 0x00, 0x00, 0x43, 0x01,
0x09, 0x00, 0x00, 0x43, 0x01, 0x0a, 0x00, 0x01, 0x0c, 0x44, 0x44, 0x00, 0x01, 0x0c, 0x43, 0x01,
0x03, 0x00, 0x01, 0x0c, 0x44, 0x44, 0x00, 0x01, 0x0c, 0x43, 0x01, 0x03, 0x00, 0x01, 0x0c, 0x44,
0x44, 0x00, 0x01, 0x0c, 0x43, 0x01, 0x03, 0x00, 0x01, 0x0c, 0x44, 0x44, 0x00, 0x01, 0x0c, 0x43,
0x01, 0x2e, 0x44, 0x43, 0x01, 0x05, 0x44, 0x43, 0x01, 0x0a, 0x00, 0x01, 0x3f, 0x01, 0x00
}
```

## 5.6 C:/Users/Administrator/Documents/GitHub/NES-Snake/gfx/level5\_nam.h File Reference

This header file contains the nametable (background) of level map 5. Created with NES Screen Tool 2.04 (Option Nametable -> Save nametable and attributes -> RLE packed as C header (.h)).

### Variables

- const unsigned char [level5\\_nam](#) [511]

### 5.6.1 Detailed Description

This header file contains the nametable (background) of level map 5. Created with NES Screen Tool 2.04 (Option Nametable -> Save nametable and attributes -> RLE packed as C header (.h)).

### Author

Sebastian Dine

## 5.6.2 Variable Documentation

### 5.6.2.1 level5\_nam

```
const unsigned char level5_nam[511]
```

## 5.7 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]

### 5.7.1 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

### 5.7.2 Variable Documentation

#### 5.7.2.1 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 }  
}
```

## 5.8 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]

### 5.8.1 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

### 5.8.2 Variable Documentation

#### 5.8.2.1 menu\_pal

```
const unsigned char menu_pal[16]
```

#### Initial value:

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

## 5.9 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]

### 5.9.1 Detailed Description

This header file contains the color palette for sprites.

#### Author

Sebastian Dine

### 5.9.2 Variable Documentation

#### 5.9.2.1 sprites\_pal

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

#### Initial value:

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

## 5.10 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` [293]

#### 5.10.1 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

#### 5.10.2 Variable Documentation

##### 5.10.2.1 titlescreen\_nam

```
const unsigned char titlescreen_nam[293]
```

### Initial value:

```
= {
0x01, 0x43, 0x01, 0x3f, 0x44, 0x44, 0x00, 0x01, 0x1b, 0x44, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x44,
0x01, 0x03, 0x00, 0x01, 0x0e, 0x50, 0x51, 0x52, 0x53, 0x54, 0x55, 0x00, 0x01, 0x06, 0x44, 0x01,
0x03, 0x00, 0x01, 0x06, 0x6a, 0x6b, 0x6c, 0x6d, 0x6e, 0x6f, 0x4a, 0x4b, 0x60, 0x61, 0x62, 0x63,
0x64, 0x65, 0x00, 0x01, 0x06, 0x44, 0x01, 0x03, 0x00, 0x01, 0x06, 0x7a, 0x7b, 0x7c, 0x7d, 0x7e,
0x7f, 0x5a, 0x5b, 0x70, 0x71, 0x72, 0x73, 0x74, 0x75, 0x00, 0x01, 0x06, 0x44, 0x01, 0x03, 0x00,
0x01, 0x08, 0x8c, 0x00, 0x01, 0x11, 0x44, 0x01, 0x03, 0x00, 0x01, 0x1b, 0x44, 0x01, 0x03, 0x00,
0x01, 0x08, 0x54, 0x55, 0x50, 0x51, 0x56, 0x57, 0x58, 0x59, 0x52, 0x53, 0x00, 0x01, 0x08, 0x44,
0x01, 0x03, 0x00, 0x01, 0x08, 0x64, 0x65, 0x60, 0x61, 0x66, 0x67, 0x68, 0x69, 0x62, 0x63, 0x00,
0x01, 0x08, 0x44, 0x01, 0x03, 0x00, 0x01, 0x08, 0x74, 0x75, 0x70, 0x71, 0x76, 0x77, 0x78, 0x79,
0x72, 0x73, 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, 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,
0x33, 0x25, 0x22, 0x21, 0x33, 0x34, 0x29, 0x21, 0x2e, 0x00, 0x24, 0x29, 0x2e, 0x25, 0x0c, 0x12,
0x10, 0x11, 0x16, 0x0d, 0x12, 0x10, 0x11, 0x17, 0x00, 0x01, 0x03, 0x44, 0x44, 0x43, 0x01, 0x3f,
0x00, 0x01, 0x3f, 0x01, 0x00
}
```

## 5.11 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)

## 5.11.1 Macro Definition Documentation

### 5.11.1.1 FALSE

```
#define FALSE 0
```

### 5.11.1.2 MASK\_BG

```
#define MASK_BG 0x08
```



#### 5.11.1.3 MASK\_EDGE\_BG

```
#define MASK_EDGE_BG 0x02
```

#### 5.11.1.4 MASK\_EDGE\_SPR

```
#define MASK_EDGE_SPR 0x04
```

#### 5.11.1.5 MASK\_SPR

```
#define MASK_SPR 0x10
```

#### 5.11.1.6 MAX

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

#### 5.11.1.7 MIN

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

#### 5.11.1.8 MSB

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

#### 5.11.1.9 NAMETABLE\_A

```
#define NAMETABLE_A 0x2000
```

#### 5.11.1.10 NAMETABLE\_B

```
#define NAMETABLE_B 0x2400
```

#### 5.11.1.11 NAMETABLE\_C

```
#define NAMETABLE_C 0x2800
```

**5.11.1.12 NAMETABLE\_D**

```
#define NAMETABLE_D 0x2c00
```

**5.11.1.13 NT\_UPD\_EOF**

```
#define NT_UPD_EOF 0xff
```

**5.11.1.14 NT\_UPD\_HORZ**

```
#define NT_UPD_HORZ 0x40
```

**5.11.1.15 NT\_UPD\_VERT**

```
#define NT_UPD_VERT 0x80
```

**5.11.1.16 NTADR\_A**

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

**5.11.1.17 NTADR\_B**

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

**5.11.1.18 NTADR\_C**

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

**5.11.1.19 NTADR\_D**

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

**5.11.1.20 NULL**

```
#define NULL 0
```

**5.11.1.21 OAM\_BEHIND**

```
#define OAM_BEHIND 0x20
```

**5.11.1.22 OAM\_FLIP\_H**

```
#define OAM_FLIP_H 0x40
```

**5.11.1.23 OAM\_FLIP\_V**

```
#define OAM_FLIP_V 0x80
```

**5.11.1.24 PAD\_A**

```
#define PAD_A 0x01
```

**5.11.1.25 PAD\_B**

```
#define PAD_B 0x02
```

**5.11.1.26 PAD\_DOWN**

```
#define PAD_DOWN 0x20
```

**5.11.1.27 PAD\_LEFT**

```
#define PAD_LEFT 0x40
```

**5.11.1.28 PAD\_RIGHT**

```
#define PAD_RIGHT 0x80
```

**5.11.1.29 PAD\_SELECT**

```
#define PAD_SELECT 0x04
```

**5.11.1.30 PAD\_START**

```
#define PAD_START 0x08
```

#### 5.11.1.31 PAD\_UP

```
#define PAD_UP 0x10
```

#### 5.11.1.32 TRUE

```
#define TRUE 1
```

### 5.11.2 Function Documentation

#### 5.11.2.1 bank\_bg()

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

#### 5.11.2.2 bank\_spr()

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

#### 5.11.2.3 delay()

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

Here is the caller graph for this function:



#### 5.11.2.4 flush\_vram\_update()

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

#### 5.11.2.5 memcpy()

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

#### 5.11.2.6 memfill()

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

#### 5.11.2.7 music\_pause()

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

#### 5.11.2.8 music\_play()

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

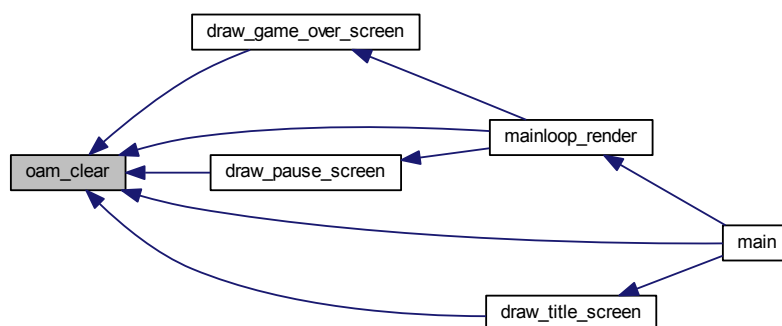
#### 5.11.2.9 music\_stop()

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

#### 5.11.2.10 oam\_clear()

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

Here is the caller graph for this function:



#### 5.11.2.11 oam\_hide\_rest()

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

#### 5.11.2.12 oam\_meta\_spr()

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

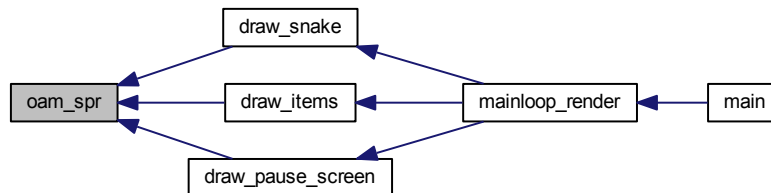
#### 5.11.2.13 oam\_size()

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

#### 5.11.2.14 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:



#### 5.11.2.15 pad\_poll()

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

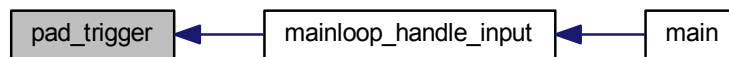
#### 5.11.2.16 pad\_state()

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

#### 5.11.2.17 pad\_trigger()

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

Here is the caller graph for this function:



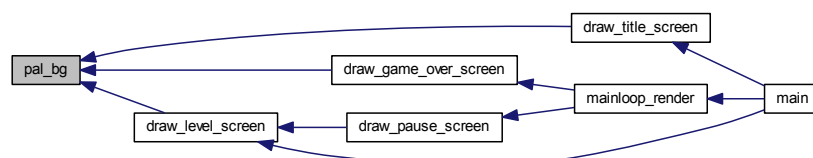
#### 5.11.2.18 pal\_all()

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

#### 5.11.2.19 pal\_bg()

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

Here is the caller graph for this function:



#### 5.11.2.20 pal\_bg\_bright()

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

#### 5.11.2.21 pal\_bright()

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

#### 5.11.2.22 pal\_clear()

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

#### 5.11.2.23 pal\_col()

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

#### 5.11.2.24 pal\_spr()

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

Here is the caller graph for this function:



#### 5.11.2.25 pal\_spr\_bright()

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

#### 5.11.2.26 ppu\_mask()

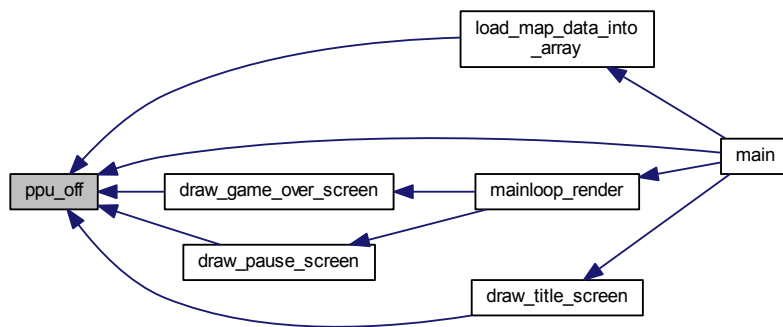
```
void __fastcall__ ppu_mask (
    unsigned char mask )
```



## 5.11.2.27 ppu\_off()

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

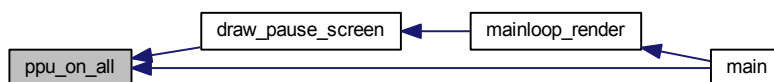
Here is the caller graph for this function:



## 5.11.2.28 ppu\_on\_all()

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

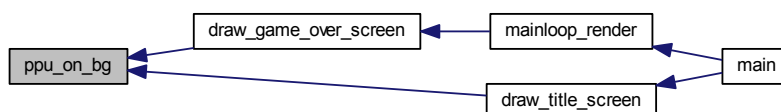
Here is the caller graph for this function:



## 5.11.2.29 ppu\_on\_bg()

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

Here is the caller graph for this function:



**5.11.2.30 ppu\_on\_spr()**

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

**5.11.2.31 ppu\_system()**

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

**5.11.2.32 ppu\_wait\_frame()**

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

**5.11.2.33 ppu\_wait\_nmi()**

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

Here is the caller graph for this function:

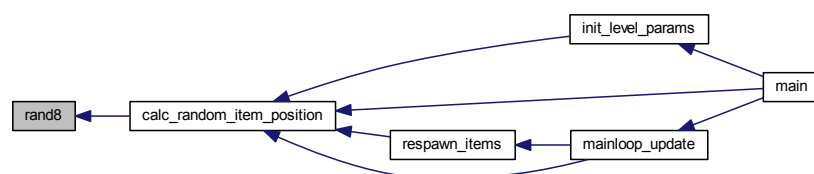
**5.11.2.34 rand16()**

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

**5.11.2.35 rand8()**

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

Here is the caller graph for this function:



#### 5.11.2.36 sample\_play()

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

#### 5.11.2.37 scroll()

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

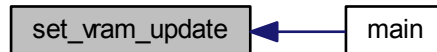
#### 5.11.2.38 set\_rand()

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

#### 5.11.2.39 set\_vram\_update()

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

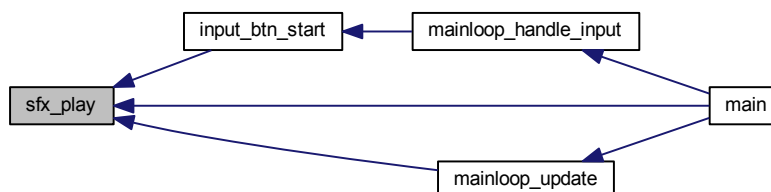
Here is the caller graph for this function:



#### 5.11.2.40 sfx\_play()

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

Here is the caller graph for this function:



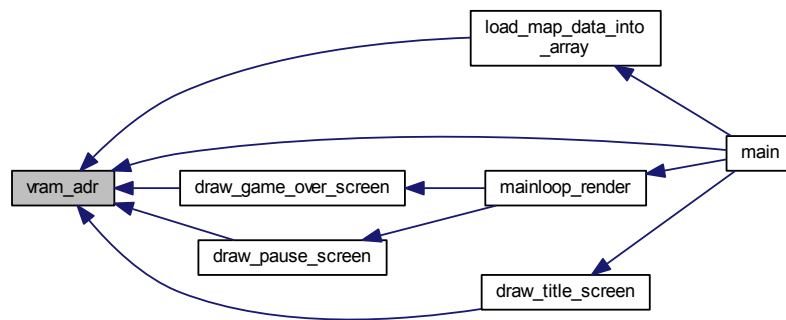
## 5.11.2.41 split()

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

## 5.11.2.42 vram\_adr()

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

Here is the caller graph for this function:



## 5.11.2.43 vram\_fill()

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

## 5.11.2.44 vram\_inc()

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

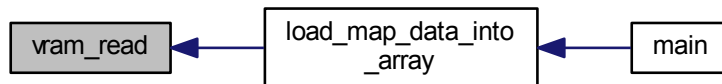
## 5.11.2.45 vram\_put()

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

## 5.11.2.46 vram\_read()

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

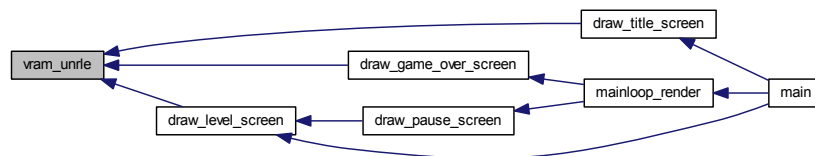
Here is the caller graph for this function:



## 5.11.2.47 vram\_unrle()

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

Here is the caller graph for this function:



## 5.11.2.48 vram\_write()

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

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

## 5.13 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, 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`
- static unsigned char `render_movement_flag`

**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, menus, and pointer to the associated palette.**

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

### 5.13.1 Detailed Description

This header file defines all global variables of the game.

#### Author

Sebastian Dine

### 5.13.2 Variable Documentation

#### 5.13.2.1 coord\_x

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

#### 5.13.2.2 coord\_y

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

#### 5.13.2.3 current\_level

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

Global variable, indicating the current level.

#### 5.13.2.4 gameover

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

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

#### 5.13.2.5 gameover\_loop

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

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

#### 5.13.2.6 i

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

#### 5.13.2.7 input

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

Global variable, holding the controller input of the current frame

#### 5.13.2.8 items

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

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

#### 5.13.2.9 j

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

#### 5.13.2.10 k

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

#### 5.13.2.11 l

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

#### 5.13.2.12 levelList

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

**Initial value:**

```
={  
    level1_nam, level2_nam, level3_nam, level4_nam,  
    level5_nam,  
    game_over_nam, titlescreen_nam,  
    levels_pal, menue_pal  
}
```

#### 5.13.2.13 map

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

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

#### 5.13.2.14 max\_score

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

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



#### 5.13.2.15 nameRow

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

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

#### 5.13.2.16 nametable\_fetch

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

Variable for fetching through nametable.

#### 5.13.2.17 pause

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

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

#### 5.13.2.18 pause\_loop

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

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

#### 5.13.2.19 render\_movement\_flag

```
unsigned char render_movement_flag [static]
```

Global variable, indicating when the movement of the snake can be rendered

#### 5.13.2.20 restart

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

Global variable, for handling the restart input

#### 5.13.2.21 snake

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

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

#### 5.13.2.22 sprite\_offset

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

#### 5.13.2.23 titlescreen

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

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

#### 5.13.2.24 ul

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

Pointer to array 'update\_list' to enable better handling of the list

#### 5.13.2.25 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

## 5.14 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)

### 5.14.1 Detailed Description

This file contains functions for initializing game elements.

#### Author

Sebastian Dine

## 5.14.2 Function Documentation

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

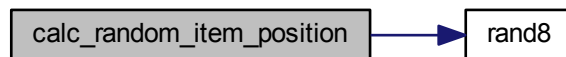
```
void calc_random_item_position (  
    void )
```

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'](#).

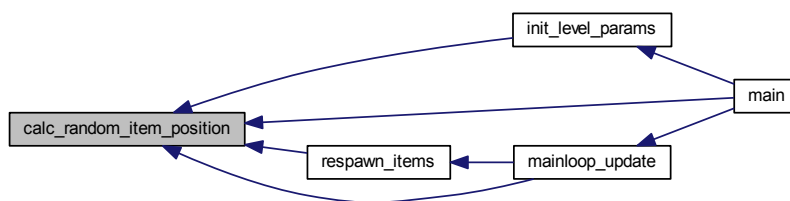
#### Author

Sebastian Dine

Here is the call graph for this function:



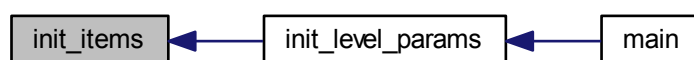
Here is the caller graph for this function:



### 5.14.2.2 init\_items()

```
void init_items (  
    void )
```

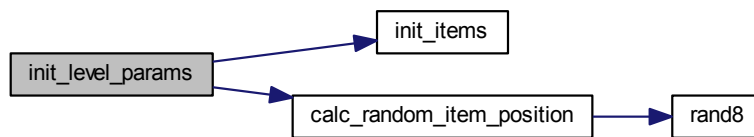
Here is the caller graph for this function:



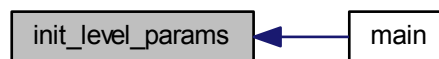
#### 5.14.2.3 init\_level\_params()

```
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:



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

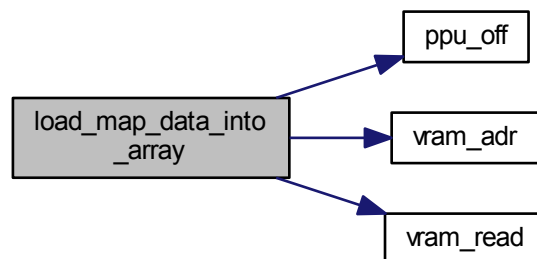
```
void load_map_data_into_array (  
    void )
```

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

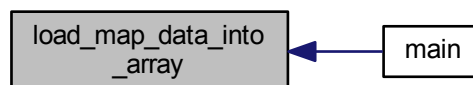
**Author**

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



## 5.15 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)

### 5.15.1 Detailed Description

This file contains functions for input handling from a controller.

**Author**

Sebastian Dine

## 5.15.2 Function Documentation

### 5.15.2.1 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.

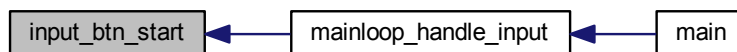
#### Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



### 5.15.2.2 mainloop\_handle\_input()

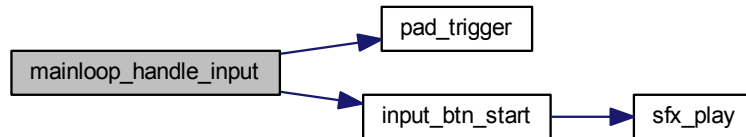
```
void mainloop_handle_input (  
    void )
```

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.

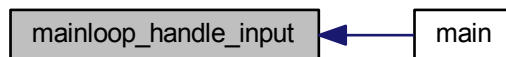
**Author**

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



## 5.16 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 10`
- `#define LVL2_START_X 56`
- `#define LVL2_START_Y 120`
- `#define LVL2_MAX_SCORE 20`
- `#define LVL3_START_X 56`
- `#define LVL3_START_Y 160`
- `#define LVL3_MAX_SCORE 30`
- `#define LVL4_START_X 40`
- `#define LVL4_START_Y 160`
- `#define LVL4_MAX_SCORE 40`
- `#define LVL5_START_X 24`

- #define LVL5\_START\_Y 160
- #define LVL5\_MAX\_SCORE 50
- #define NAMETABLE1\_START 0x2000
- #define NAMETABLE2\_START 0x2400

#### 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 constant 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 caluculations

- #define MAPARRAY\_ADR(x, y) ((y<<2)|(x>>3))
- #define SPRITE\_ATTR(flip\_horz, flip\_vert, prio, pal) ((flip\_horz<<7) + (flip\_vert<<6) + (prio<<5) + pal)

### 5.16.1 Detailed Description

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

#### Author

Sebastian Dine



## 5.16.2 Macro Definition Documentation

### 5.16.2.1 DIGIT\_O\_TILE

```
#define DIGIT_O_TILE 0x10
```

Tile of digit 0 (zero)

### 5.16.2.2 DIR\_DOWN

```
#define DIR_DOWN 2
```

### 5.16.2.3 DIR\_LEFT

```
#define DIR_LEFT 3
```

### 5.16.2.4 DIR\_RIGHT

```
#define DIR_RIGHT 4
```

### 5.16.2.5 DIR\_UP

```
#define DIR_UP 1
```

### 5.16.2.6 EMPTY\_TILE

```
#define EMPTY_TILE 0x00
```

Tile of empty space

### 5.16.2.7 ITEM\_MAX\_ON\_SCREEN

```
#define ITEM_MAX_ON_SCREEN 4
```

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

### 5.16.2.8 LEVELS\_ALL

```
#define LEVELS_ALL 5
```

Total number of level maps (ingame background nametables)

**5.16.2.9 LVL1\_MAX\_SCORE**

```
#define LVL1_MAX_SCORE 10
```

**5.16.2.10 LVL1\_START\_X**

```
#define LVL1_START_X 120
```

**5.16.2.11 LVL1\_START\_Y**

```
#define LVL1_START_Y 120
```

**5.16.2.12 LVL2\_MAX\_SCORE**

```
#define LVL2_MAX_SCORE 20
```

**5.16.2.13 LVL2\_START\_X**

```
#define LVL2_START_X 56
```

**5.16.2.14 LVL2\_START\_Y**

```
#define LVL2_START_Y 120
```

**5.16.2.15 LVL3\_MAX\_SCORE**

```
#define LVL3_MAX_SCORE 30
```

**5.16.2.16 LVL3\_START\_X**

```
#define LVL3_START_X 56
```

**5.16.2.17 LVL3\_START\_Y**

```
#define LVL3_START_Y 160
```

**5.16.2.18 LVL4\_MAX\_SCORE**

```
#define LVL4_MAX_SCORE 40
```

**5.16.2.19 LVL4\_START\_X**

```
#define LVL4_START_X 40
```

**5.16.2.20 LVL4\_START\_Y**

```
#define LVL4_START_Y 160
```

**5.16.2.21 LVL5\_MAX\_SCORE**

```
#define LVL5_MAX_SCORE 50
```

**5.16.2.22 LVL5\_START\_X**

```
#define LVL5_START_X 24
```

**5.16.2.23 LVL5\_START\_Y**

```
#define LVL5_START_Y 160
```

**5.16.2.24 MAP\_HEIGHT**

```
#define MAP_HEIGHT 30
```

**5.16.2.25 MAP\_WIDTH**

```
#define MAP_WIDTH 32
```

**5.16.2.26 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).

**5.16.2.27 NAMETABLE1\_START**

```
#define NAMETABLE1_START 0x2000
```

Start address in VRAM for first nametable

#### 5.16.2.28 NAMETABLE2\_START

```
#define NAMETABLE2_START 0x2400
```

#### 5.16.2.29 SFX\_ABSORB\_ITEM

```
#define SFX_ABSORB_ITEM 0
```

#### 5.16.2.30 SFX\_CRASH

```
#define SFX_CRASH 1
```

#### 5.16.2.31 SFX\_NEXT\_LEVEL

```
#define SFX_NEXT_LEVEL 3
```

#### 5.16.2.32 SFX\_PAUSE

```
#define SFX_PAUSE 2
```

#### 5.16.2.33 SNAKE\_BODY\_TILE

```
#define SNAKE_BODY_TILE 0x40
```

Tile of snake body element

#### 5.16.2.34 SNAKE\_HEAD\_TILE\_HORZ

```
#define SNAKE_HEAD_TILE_HORZ 0x42
```

Tile of horizontal snake head element

#### 5.16.2.35 SNAKE\_HEAD\_TILE\_VERT

```
#define SNAKE_HEAD_TILE_VERT 0x41
```

Tile of vertical snake head element

#### 5.16.2.36 SNAKE\_MAX\_SIZE

```
#define SNAKE_MAX_SIZE 100
```

Maximum of body elements, the snake can get.

### 5.16.2.37 SPIDER\_TILE

```
#define SPIDER_TILE 0x45
```

Tile of spider item

### 5.16.2.38 SPRITE\_ATTR

```
#define SPRITE_ATTR(  
    flip_horz,  
    flip_vert,  
    prio,  
    pal ) ((flip_horz<<7) + (flip_vert<<6) + (prio<<5) + pal)
```

Macro for calculating the attribute byte of a sprite. The bits of this byte store data as followed (taken from Nerdy↔ Nights tutorial):

76543210 ||| || || ++- Color Palette of sprite. Choose which set of 4 from the 16 colors to use ||| ||+----- Priority (0: in front of background; 1: behind background) |+----- Flip sprite horizontally +----- Flip sprite vertically

### 5.16.2.39 WALL\_TILE\_1

```
#define WALL_TILE_1 0x43
```

Tile of horiontal wall element

### 5.16.2.40 WALL\_TILE\_2

```
#define WALL_TILE_2 0x44
```

Tile of vertical wall element

## 5.17 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\\_level\\_screen](#) (void)
- void [draw\\_pause\\_screen](#) (void)
- void [mainloop\\_render](#) (void)

### 5.17.1 Detailed Description

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

#### Author

Sebastian Dine

### 5.17.2 Function Documentation

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

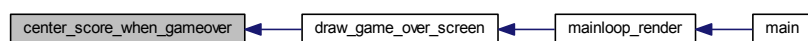
```
void center_score_when_gameover (
    void )
```

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:



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

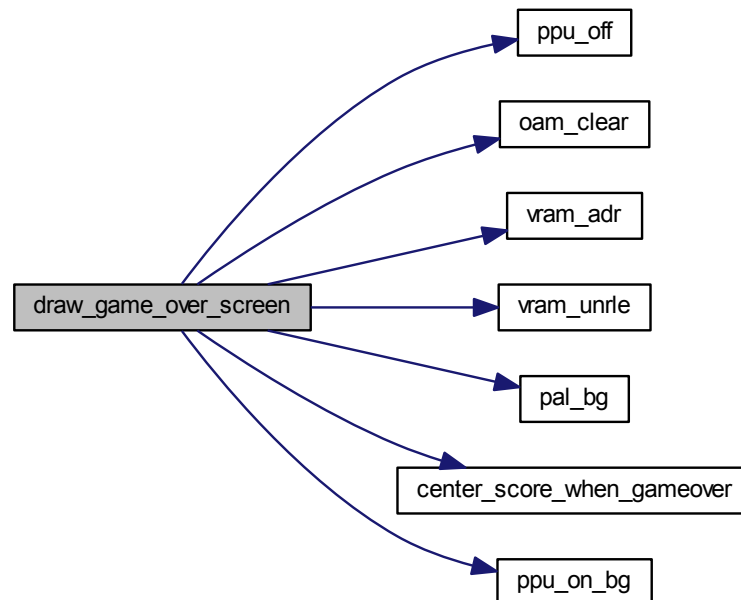
```
void draw_game_over_screen (
    void )
```

This function draws the gameover screen.

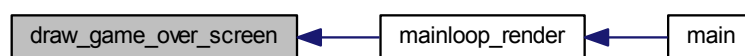
## Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



### 5.17.2.3 draw\_items()

```
void draw_items (
    void )
```

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

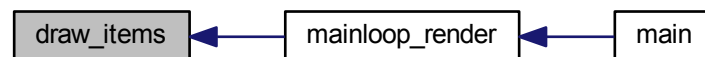
**Author**

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:

**5.17.2.4 draw\_level\_screen()**

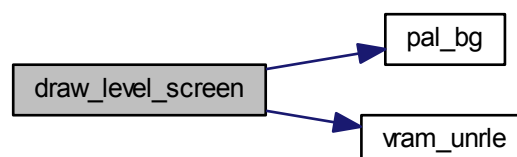
```
void draw_level_screen (  
    void )
```

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

**Author**

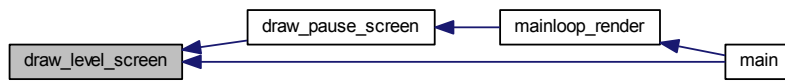
Sebastian Dine

Here is the call graph for this function:





Here is the caller graph for this function:



#### 5.17.2.5 draw\_pause\_screen()

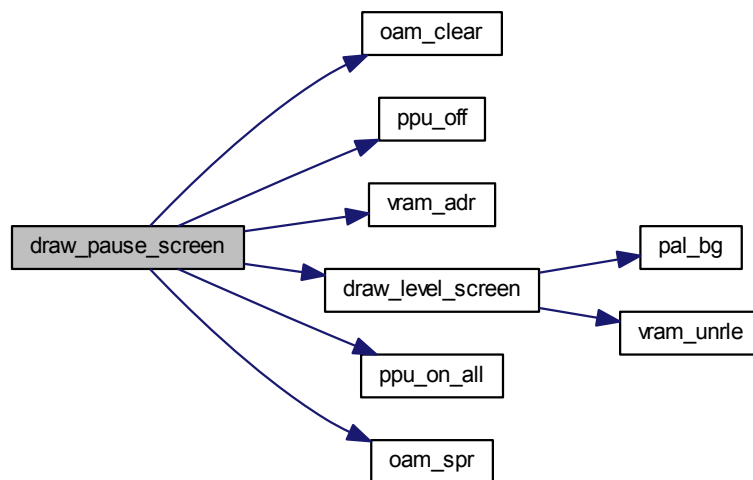
```
void draw_pause_screen (
    void )
```

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

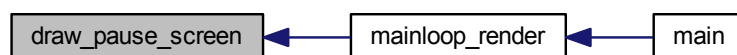
**Author**

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



#### 5.17.2.6 draw\_score()

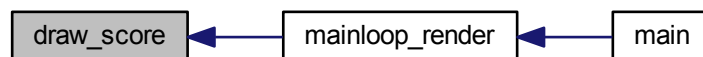
```
void draw_score (
    void )
```

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

##### Author

Sebastian Dine

Here is the caller graph for this function:



#### 5.17.2.7 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.

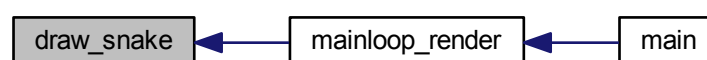
##### Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



## 5.17.2.8 draw\_title\_screen()

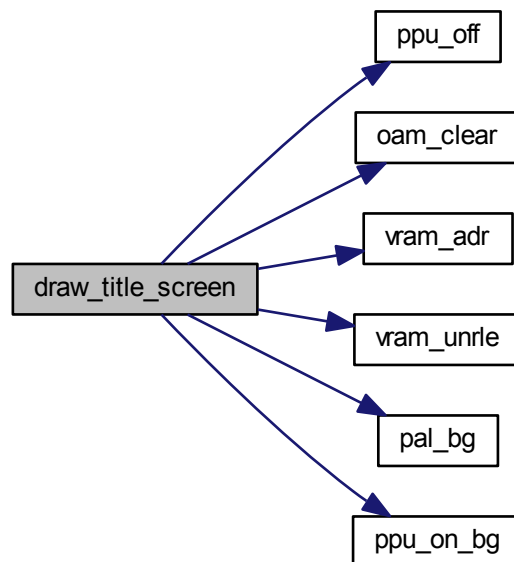
```
void draw_title_screen (  
    void )
```

This function draws the title screen.

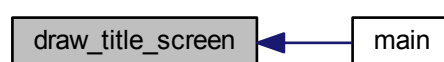
**Author**

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



### 5.17.2.9 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:



### 5.17.2.10 mainloop\_render()

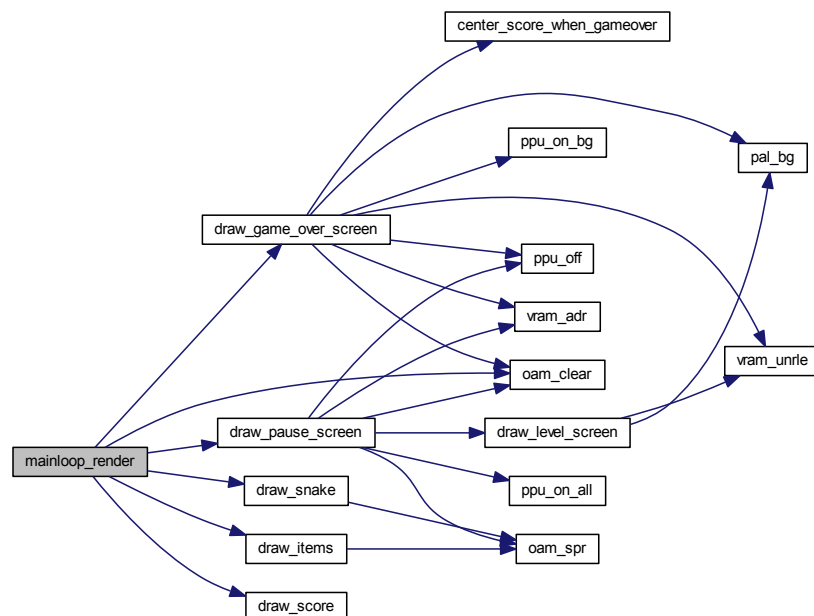
```
void mainloop_render (  
    void )
```

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

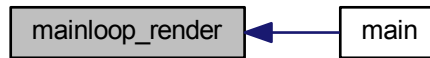
#### Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



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

Maingame file, containing the main game loop.

```
#include "level1_nam.h"
#include "level2_nam.h"
#include "level3_nam.h"
#include "level4_nam.h"
#include "level5_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.*

#### 5.18.1 Detailed Description

Maingame file, containing the main game loop.

#### Author

Sebastian Dine.

## 5.18.2 Function Documentation

### 5.18.2.1 main()

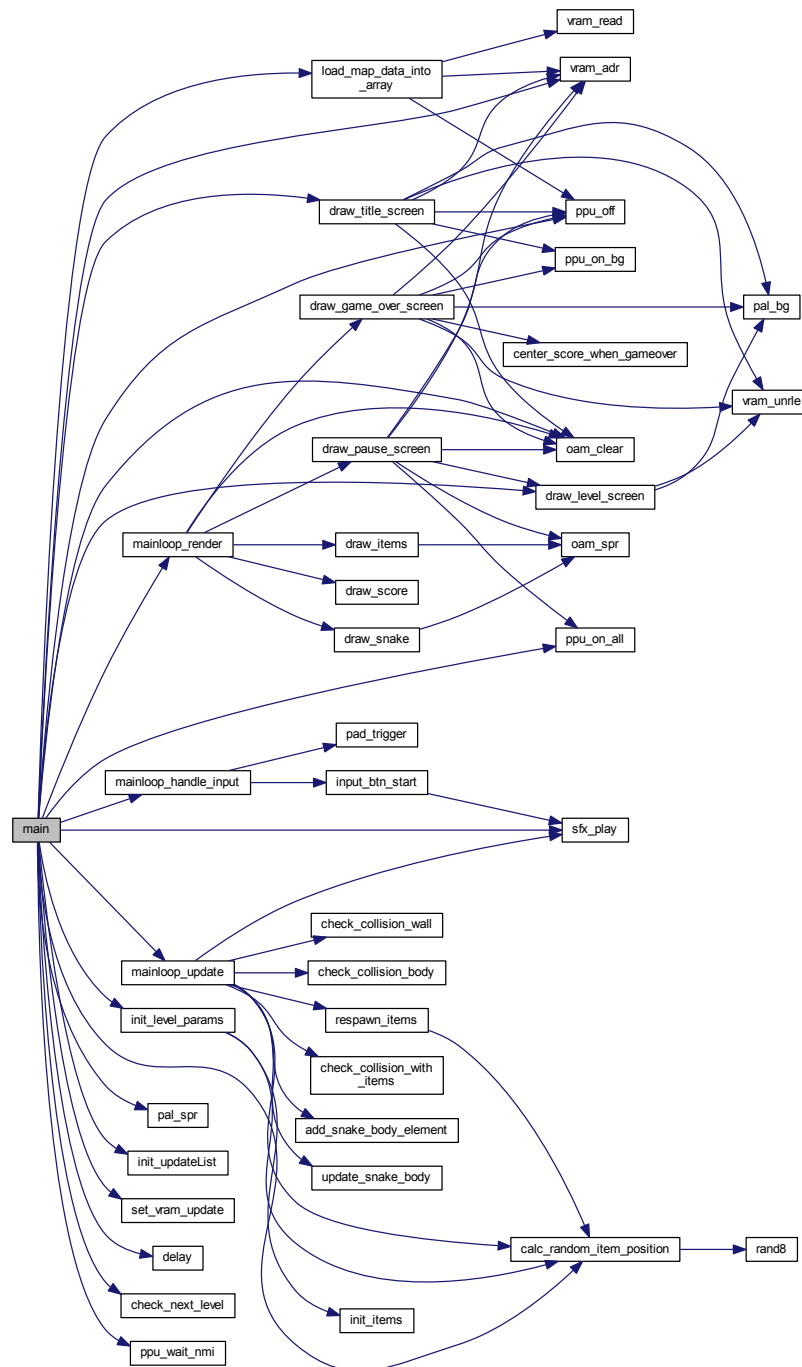
```
void main (
    void )
```

Main game loop.

**Author**

Sebastian Dine

Here is the call graph for this function:



## 5.19 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.*

### 5.19.1 Detailed Description

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

#### Author

Sebastian Dine

## 5.20 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)

### 5.20.1 Detailed Description

This file contains all ingame logic functionalities and utility functionalities.

#### Author

Sebastian Dine



## 5.20.2 Function Documentation

### 5.20.2.1 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:



### 5.20.2.2 check\_collision\_body()

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

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

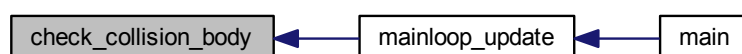
#### Returns

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

#### Author

Sebastian Dine

Here is the caller graph for this function:



### 5.20.2.3 check\_collision\_wall()

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

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

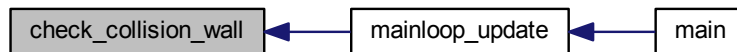
#### Returns

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

#### Author

Sebastian Dine

Here is the caller graph for this function:



### 5.20.2.4 check\_collision\_with\_items()

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

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

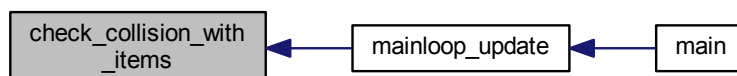
#### Returns

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

#### Author

Sebastian Dine

Here is the caller graph for this function:



#### 5.20.2.5 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:



#### 5.20.2.6 mainloop\_update()

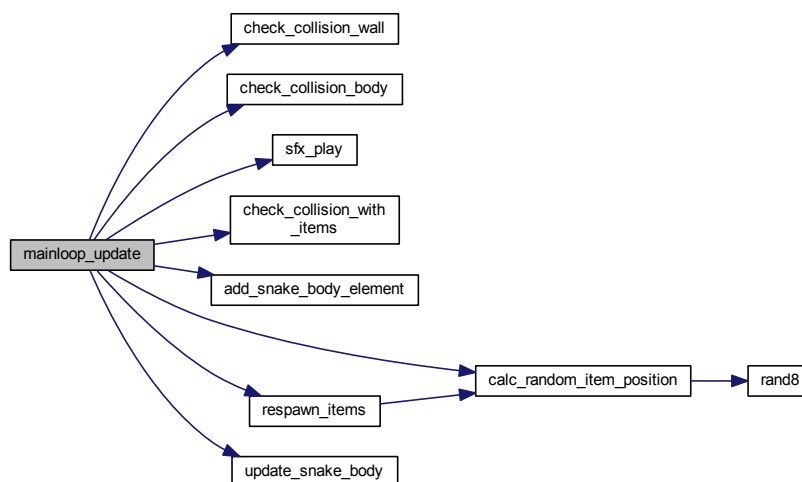
```
void mainloop_update (  
    void )
```

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

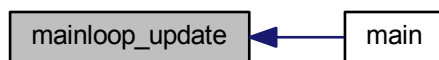
##### Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



#### 5.20.2.7 respawn\_items()

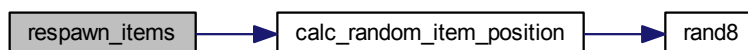
```
void respawn_items (  
    void )
```

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

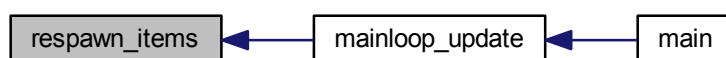
#### Author

Sebastian Dine

Here is the call graph for this function:



Here is the caller graph for this function:



#### 5.20.2.8 update\_snake\_body()

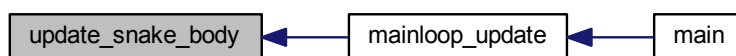
```
void update_snake_body ( )
```

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, [61](#)

bank\_bg  
neslib.h, [24](#)

bank\_spr  
neslib.h, [24](#)

body\_element\_coordinates  
snake\_struct, [9](#)

C:/Users/Administrator/Documents/GitHub/NES-  
Snake/NESLibrary/neslib.h, [19](#)

C:/Users/Administrator/Documents/GitHub/NES-  
Snake/README.md, [33](#)

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/level3\_nam.h, [13](#)

C:/Users/Administrator/Documents/GitHub/NES-  
Snake/gfx/level4\_nam.h, [14](#)

C:/Users/Administrator/Documents/GitHub/NES-  
Snake/gfx/level5\_nam.h, [15](#)

C:/Users/Administrator/Documents/GitHub/NES-  
Snake/gfx/levels\_pal.h, [16](#)

C:/Users/Administrator/Documents/GitHub/NES-  
Snake/gfx/menue\_pal.h, [16](#)

C:/Users/Administrator/Documents/GitHub/NES-  
Snake/gfx/sprites\_pal.h, [17](#)

C:/Users/Administrator/Documents/GitHub/NES-  
Snake/gfx/titlescreen\_nam.h, [18](#)

C:/Users/Administrator/Documents/GitHub/NES-  
Snake/src/globals.h, [33](#)

C:/Users/Administrator/Documents/GitHub/NES-  
Snake/src/init.c, [38](#)

C:/Users/Administrator/Documents/GitHub/NES-  
Snake/src/input.c, [41](#)

C:/Users/Administrator/Documents/GitHub/NES-  
Snake/src/macros.h, [43](#)

C:/Users/Administrator/Documents/GitHub/NES-  
Snake/src/render.c, [49](#)

C:/Users/Administrator/Documents/GitHub/NES-  
Snake/src/snake.c, [57](#)

C:/Users/Administrator/Documents/GitHub/NES-  
Snake/src/structures.h, [59](#)

C:/Users/Administrator/Documents/GitHub/NES-  
Snake/src/update.c, [60](#)

calc\_random\_item\_position  
init.c, [39](#)

center\_score\_when\_gameover  
render.c, [50](#)

check\_collision\_body  
update.c, [61](#)

check\_collision\_wall  
update.c, [61](#)

check\_collision\_with\_items  
update.c, [62](#)

check\_next\_level  
update.c, [62](#)

coord\_x  
globals.h, [35](#)

coord\_y  
globals.h, [35](#)

current\_level  
globals.h, [35](#)

DIGIT\_O\_TILE  
macros.h, [45](#)

DIR\_DOWN  
macros.h, [45](#)

DIR\_LEFT  
macros.h, [45](#)

DIR\_RIGHT  
macros.h, [45](#)

DIR\_UP  
macros.h, [45](#)

delay  
neslib.h, [24](#)

draw\_game\_over\_screen  
render.c, [50](#)

draw\_items  
render.c, [51](#)

draw\_level\_screen  
render.c, [52](#)

draw\_pause\_screen  
render.c, [53](#)

draw\_score  
render.c, [53](#)

draw\_snake  
render.c, [54](#)

draw\_title\_screen  
render.c, [54](#)

EMPTY\_TILE  
macros.h, [45](#)

FALSE

- neslib.h, [20](#)
- flush\_vram\_update
  - neslib.h, [24](#)
- game\_over\_nam
  - game\_over\_nam.h, [11](#)
- game\_over\_nam.h
  - game\_over\_nam, [11](#)
- gameover
  - globals.h, [35](#)
- gameover\_loop
  - globals.h, [35](#)
- globals.h
  - coord\_x, [35](#)
  - coord\_y, [35](#)
  - current\_level, [35](#)
  - gameover, [35](#)
  - gameover\_loop, [35](#)
  - i, [35](#)
  - input, [35](#)
  - items, [35](#)
  - j, [36](#)
  - k, [36](#)
  - l, [36](#)
  - levelList, [36](#)
  - map, [36](#)
  - max\_score, [36](#)
  - nameRow, [36](#)
  - nametable\_fetch, [37](#)
  - pause, [37](#)
  - pause\_loop, [37](#)
  - render\_movement\_flag, [37](#)
  - restart, [37](#)
  - snake, [37](#)
  - sprite\_offset, [37](#)
  - titlescreen, [37](#)
  - ul, [38](#)
  - update\_list, [38](#)
- grow\_event\_flag
  - snake\_struct, [9](#)
- 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, [35](#)
- ITEM\_MAX\_ON\_SCREEN
  - macros.h, [45](#)
- init.c
  - calc\_random\_item\_position, [39](#)
  - init\_items, [39](#)
  - init\_level\_params, [39](#)
  - load\_map\_data\_into\_array, [40](#)
- init\_items
  - init.c, [39](#)
- init\_level\_params
  - init.c, [39](#)
- init\_updateList
  - render.c, [55](#)
- input
  - globals.h, [35](#)
- input.c
  - input\_btn\_start, [42](#)
  - mainloop\_handle\_input, [42](#)
- input\_btn\_start
  - input.c, [42](#)
- 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, [35](#)
- items\_struct, [7](#)
  - item\_collision\_flags, [7](#)
  - item\_coordinates, [7](#)
  - item\_respawn\_count, [8](#)
  - item\_respawn\_frm\_rate, [8](#)
- j
  - globals.h, [36](#)
- k
  - globals.h, [36](#)
- l
  - globals.h, [36](#)
- LEVELS\_ALL
  - macros.h, [45](#)
- LVL1\_MAX\_SCORE
  - macros.h, [45](#)
- LVL1\_START\_X
  - macros.h, [46](#)
- LVL1\_START\_Y
  - macros.h, [46](#)
- LVL2\_MAX\_SCORE
  - macros.h, [46](#)
- LVL2\_START\_X
  - macros.h, [46](#)
- LVL2\_START\_Y
  - macros.h, [46](#)
- LVL3\_MAX\_SCORE
  - macros.h, [46](#)
- LVL3\_START\_X
  - macros.h, [46](#)
- LVL3\_START\_Y
  - macros.h, [46](#)
- LVL4\_MAX\_SCORE
  - macros.h, [46](#)



LVL4\_START\_X  
  macros.h, [46](#)

LVL4\_START\_Y  
  macros.h, [47](#)

LVL5\_MAX\_SCORE  
  macros.h, [47](#)

LVL5\_START\_X  
  macros.h, [47](#)

LVL5\_START\_Y  
  macros.h, [47](#)

last\_body\_element\_x  
  snake\_struct, [9](#)

last\_body\_element\_y  
  snake\_struct, [10](#)

level1\_nam  
  level1\_nam.h, [12](#)

level1\_nam.h  
  level1\_nam, [12](#)

level2\_nam  
  level2\_nam.h, [13](#)

level2\_nam.h  
  level2\_nam, [13](#)

level3\_nam  
  level3\_nam.h, [14](#)

level3\_nam.h  
  level3\_nam, [14](#)

level4\_nam  
  level4\_nam.h, [15](#)

level4\_nam.h  
  level4\_nam, [15](#)

level5\_nam  
  level5\_nam.h, [15](#)

level5\_nam.h  
  level5\_nam, [15](#)

levelList  
  globals.h, [36](#)

levels\_pal  
  levels\_pal.h, [16](#)

levels\_pal.h  
  levels\_pal, [16](#)

load\_map\_data\_into\_array  
  init.c, [40](#)

MAP\_HEIGHT  
  macros.h, [47](#)

MAP\_WIDTH  
  macros.h, [47](#)

MAPARRAY\_ADR  
  macros.h, [47](#)

MASK\_BG  
  neslib.h, [20](#)

MASK\_EDGE\_BG  
  neslib.h, [20](#)

MASK\_EDGE\_SPR  
  neslib.h, [21](#)

MASK\_SPR  
  neslib.h, [21](#)

MAX  
  neslib.h, [21](#)

MIN  
  neslib.h, [21](#)

MSB  
  neslib.h, [21](#)

macros.h  
  DIGIT\_O\_TILE, [45](#)  
  DIR\_DOWN, [45](#)  
  DIR\_LEFT, [45](#)  
  DIR\_RIGHT, [45](#)  
  DIR\_UP, [45](#)  
  EMPTY\_TILE, [45](#)  
  ITEM\_MAX\_ON\_SCREEN, [45](#)  
  LEVELS\_ALL, [45](#)  
  LVL1\_MAX\_SCORE, [45](#)  
  LVL1\_START\_X, [46](#)  
  LVL1\_START\_Y, [46](#)  
  LVL2\_MAX\_SCORE, [46](#)  
  LVL2\_START\_X, [46](#)  
  LVL2\_START\_Y, [46](#)  
  LVL3\_MAX\_SCORE, [46](#)  
  LVL3\_START\_X, [46](#)  
  LVL3\_START\_Y, [46](#)  
  LVL4\_MAX\_SCORE, [46](#)  
  LVL4\_START\_X, [46](#)  
  LVL4\_START\_Y, [47](#)  
  LVL5\_MAX\_SCORE, [47](#)  
  LVL5\_START\_X, [47](#)  
  LVL5\_START\_Y, [47](#)  
  MAP\_HEIGHT, [47](#)  
  MAP\_WIDTH, [47](#)  
  MAPARRAY\_ADR, [47](#)  
  NAMETABLE1\_START, [47](#)  
  NAMETABLE2\_START, [47](#)  
  SFX\_ABSORB\_ITEM, [48](#)  
  SFX\_CRASH, [48](#)  
  SFX\_NEXT\_LEVEL, [48](#)  
  SFX\_PAUSE, [48](#)  
  SNAKE\_BODY\_TILE, [48](#)  
  SNAKE\_HEAD\_TILE\_HORZ, [48](#)  
  SNAKE\_HEAD\_TILE\_VERT, [48](#)  
  SNAKE\_MAX\_SIZE, [48](#)  
  SPIDER\_TILE, [48](#)  
  SPRITE\_ATTR, [49](#)  
  WALL\_TILE\_1, [49](#)  
  WALL\_TILE\_2, [49](#)

main  
  snake.c, [58](#)

mainloop\_handle\_input  
  input.c, [42](#)

mainloop\_render  
  render.c, [56](#)

mainloop\_update  
  update.c, [63](#)

map  
  globals.h, [36](#)

max\_score  
  globals.h, [36](#)

memcpy

- neslib.h, 24
- memfill
  - neslib.h, 24
- menue\_pal
  - menue\_pal.h, 17
- menue\_pal.h
  - menue\_pal, 17
- moving\_direction
  - snake\_struct, 10
- music\_pause
  - neslib.h, 25
- music\_play
  - neslib.h, 25
- music\_stop
  - neslib.h, 25
- NAMETABLE1\_START
  - macros.h, 47
- NAMETABLE2\_START
  - macros.h, 47
- NAMETABLE\_A
  - neslib.h, 21
- NAMETABLE\_B
  - neslib.h, 21
- NAMETABLE\_C
  - neslib.h, 21
- NAMETABLE\_D
  - neslib.h, 21
- NT\_UPD\_EOF
  - neslib.h, 22
- NT\_UPD\_HORZ
  - neslib.h, 22
- NT\_UPD\_VERT
  - neslib.h, 22
- NTADR\_A
  - neslib.h, 22
- NTADR\_B
  - neslib.h, 22
- NTADR\_C
  - neslib.h, 22
- NTADR\_D
  - neslib.h, 22
- NULL
  - neslib.h, 22
- nameRow
  - globals.h, 36
- nametable\_fetch
  - globals.h, 37
- neslib.h
  - bank\_bg, 24
  - bank\_spr, 24
  - delay, 24
  - FALSE, 20
  - flush\_vram\_update, 24
  - MASK\_BG, 20
  - MASK\_EDGE\_BG, 20
  - MASK\_EDGE\_SPR, 21
  - MASK\_SPR, 21
  - MAX, 21
  - MIN, 21
  - MSB, 21
  - memcpy, 24
  - memfill, 24
  - music\_pause, 25
  - music\_play, 25
  - music\_stop, 25
  - NAMETABLE\_A, 21
  - NAMETABLE\_B, 21
  - NAMETABLE\_C, 21
  - NAMETABLE\_D, 21
  - NT\_UPD\_EOF, 22
  - NT\_UPD\_HORZ, 22
  - NT\_UPD\_VERT, 22
  - NTADR\_A, 22
  - NTADR\_B, 22
  - NTADR\_C, 22
  - NTADR\_D, 22
  - NULL, 22
  - OAM\_BEHIND, 22
  - OAM\_FLIP\_H, 23
  - OAM\_FLIP\_V, 23
  - oam\_clear, 25
  - oam\_hide\_rest, 25
  - oam\_meta\_spr, 25
  - oam\_size, 26
  - oam\_spr, 26
  - PAD\_DOWN, 23
  - PAD\_LEFT, 23
  - PAD\_RIGHT, 23
  - PAD\_SELECT, 23
  - PAD\_START, 23
  - PAD\_UP, 23
  - PAD\_A, 23
  - PAD\_B, 23
  - pad\_poll, 26
  - pad\_state, 26
  - pad\_trigger, 26
  - pal\_all, 27
  - pal\_bg, 27
  - pal\_bg\_bright, 27
  - pal\_bright, 27
  - pal\_clear, 27
  - pal\_col, 28
  - pal\_spr, 28
  - pal\_spr\_bright, 28
  - ppu\_mask, 28
  - ppu\_off, 28
  - ppu\_on\_all, 29
  - ppu\_on\_bg, 29
  - ppu\_on\_spr, 29
  - ppu\_system, 30
  - ppu\_wait\_frame, 30
  - ppu\_wait\_nmi, 30
  - rand16, 30
  - rand8, 30
  - sample\_play, 30
  - scroll, 31

- set\_rand, 31
- set\_vram\_update, 31
- sfx\_play, 31
- split, 31
- TRUE, 24
- vram\_adr, 32
- vram\_fill, 32
- vram\_inc, 32
- vram\_put, 32
- vram\_read, 32
- vram\_unrle, 33
- vram\_write, 33
- OAM\_BEHIND
  - neslib.h, 22
- OAM\_FLIP\_H
  - neslib.h, 23
- OAM\_FLIP\_V
  - neslib.h, 23
- oam\_clear
  - neslib.h, 25
- oam\_hide\_rest
  - neslib.h, 25
- oam\_meta\_spr
  - neslib.h, 25
- oam\_size
  - neslib.h, 26
- oam\_spr
  - neslib.h, 26
- PAD\_DOWN
  - neslib.h, 23
- PAD\_LEFT
  - neslib.h, 23
- PAD\_RIGHT
  - neslib.h, 23
- PAD\_SELECT
  - neslib.h, 23
- PAD\_START
  - neslib.h, 23
- PAD\_UP
  - neslib.h, 23
- PAD\_A
  - neslib.h, 23
- PAD\_B
  - neslib.h, 23
- pad\_poll
  - neslib.h, 26
- pad\_state
  - neslib.h, 26
- pad\_trigger
  - neslib.h, 26
- pal\_all
  - neslib.h, 27
- pal\_bg
  - neslib.h, 27
- pal\_bg\_bright
  - neslib.h, 27
- pal\_bright
  - neslib.h, 27
- pal\_clear
  - neslib.h, 27
- pal\_col
  - neslib.h, 28
- pal\_spr
  - neslib.h, 28
- pal\_spr\_bright
  - neslib.h, 28
- pause
  - globals.h, 37
- pause\_loop
  - globals.h, 37
- ppu\_mask
  - neslib.h, 28
- ppu\_off
  - neslib.h, 28
- ppu\_on\_all
  - neslib.h, 29
- ppu\_on\_bg
  - neslib.h, 29
- ppu\_on\_spr
  - neslib.h, 29
- ppu\_system
  - neslib.h, 30
- ppu\_wait\_frame
  - neslib.h, 30
- ppu\_wait\_nmi
  - neslib.h, 30
- rand16
  - neslib.h, 30
- rand8
  - neslib.h, 30
- render.c
  - center\_score\_when\_gameover, 50
  - draw\_game\_over\_screen, 50
  - draw\_items, 51
  - draw\_level\_screen, 52
  - draw\_pause\_screen, 53
  - draw\_score, 53
  - draw\_snake, 54
  - draw\_title\_screen, 54
  - init\_updateList, 55
  - mainloop\_render, 56
- render\_movement\_flag
  - globals.h, 37
- respawn\_items
  - update.c, 64
- restart
  - globals.h, 37
- SFX\_ABSORB\_ITEM
  - macros.h, 48
- SFX\_CRASH
  - macros.h, 48
- SFX\_NEXT\_LEVEL
  - macros.h, 48
- SFX\_PAUSE

- macros.h, 48
- SNAKE\_BODY\_TILE
  - macros.h, 48
- SNAKE\_HEAD\_TILE\_HORZ
  - macros.h, 48
- SNAKE\_HEAD\_TILE\_VERT
  - macros.h, 48
- SNAKE\_MAX\_SIZE
  - macros.h, 48
- SPIDER\_TILE
  - macros.h, 48
- SPRITE\_ATTR
  - macros.h, 49
- sample\_play
  - neslib.h, 30
- scroll
  - neslib.h, 31
- set\_rand
  - neslib.h, 31
- set\_vram\_update
  - neslib.h, 31
- sfx\_play
  - neslib.h, 31
- size\_index
  - snake\_struct, 10
- snake
  - globals.h, 37
- snake.c
  - main, 58
- snake\_struct, 8
  - body\_element\_coordinates, 9
  - grow\_event\_flag, 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, 10
  - moving\_direction, 10
  - size\_index, 10
  - speed\_counter, 10
- speed\_counter
  - snake\_struct, 10
- split
  - neslib.h, 31
- sprite\_offset
  - globals.h, 37
- sprites\_pal
  - sprites\_pal.h, 17
- sprites\_pal.h
  - sprites\_pal, 17
- TRUE
  - neslib.h, 24
- titlescreen
  - globals.h, 37
- titlescreen\_nam
  - titlescreen\_nam.h, 18
- titlescreen\_nam.h
  - titlescreen\_nam, 18
- ul
  - globals.h, 38
- update.c
  - add\_snake\_body\_element, 61
  - check\_collision\_body, 61
  - check\_collision\_wall, 61
  - check\_collision\_with\_items, 62
  - check\_next\_level, 62
  - mainloop\_update, 63
  - respawn\_items, 64
  - update\_snake\_body, 64
- update\_list
  - globals.h, 38
- update\_snake\_body
  - update.c, 64
- vram\_adr
  - neslib.h, 32
- vram\_fill
  - neslib.h, 32
- vram\_inc
  - neslib.h, 32
- vram\_put
  - neslib.h, 32
- vram\_read
  - neslib.h, 32
- vram\_unrle
  - neslib.h, 33
- vram\_write
  - neslib.h, 33
- WALL\_TILE\_1
  - macros.h, 49
- WALL\_TILE\_2
  - macros.h, 49