

Word Asteroid

1

Generato da Doxygen 1.8.13

Indice

1	Indice dei tipi composti	1
1.1	Elenco dei tipi composti	1
2	Indice dei file	3
2.1	Elenco dei file	3
3	Documentazione delle classi	5
3.1	Riferimenti per la struct asteroid_t	5
3.1.1	Descrizione dettagliata	5
3.1.2	Documentazione dei membri dato	5
3.1.2.1	bmp	6
3.1.2.2	radius	6
3.1.2.3	slope	6
3.1.2.4	special	6
3.1.2.5	word	6
3.1.2.6	x	6
3.1.2.7	y	6
3.2	Riferimenti per la struct bullet_t	7
3.2.1	Descrizione dettagliata	7
3.2.2	Documentazione dei membri dato	7
3.2.2.1	radius	7
3.2.2.2	slope	7
3.2.2.3	x	7
3.2.2.4	y	8

3.3	Riferimenti per la struct <code>display_info_t</code>	8
3.3.1	Descrizione dettagliata	8
3.3.2	Documentazione dei membri dato	8
3.3.2.1	height	8
3.3.2.2	width	8
3.4	Riferimenti per la struct <code>elem_asteroid_t</code>	9
3.4.1	Descrizione dettagliata	9
3.4.2	Documentazione dei membri dato	9
3.4.2.1	asteroid	9
3.4.2.2	next	10
3.5	Riferimenti per la struct <code>file_t</code>	10
3.5.1	Descrizione dettagliata	10
3.5.2	Documentazione dei membri dato	10
3.5.2.1	filename	10
3.5.2.2	size	10
3.6	Riferimenti per la struct <code>general_gv_t</code>	11
3.6.1	Descrizione dettagliata	11
3.6.2	Documentazione dei membri dato	11
3.6.2.1	BACK_FONTSIZE	11
3.6.2.2	BACK_LEFT_OFFSET	11
3.6.2.3	BACK_TEXT_SHADOW_GAP	11
3.6.2.4	BACK_TOP_OFFSET	12
3.6.2.5	HEAD_FONTSIZE	12
3.6.2.6	HEAD_TEXT_SHADOW_GAP	12
3.6.2.7	HEAD_TOP_OFFSET	12
3.6.2.8	RESULT_TOP_OFFSET	12
3.7	Riferimenti per la struct <code>instr_gv_t</code>	12
3.7.1	Descrizione dettagliata	13
3.7.2	Documentazione dei membri dato	13
3.7.2.1	BONUS_CASE_WIDTH	13

3.7.2.2	BONUS_RADIUS	13
3.7.2.3	SUBTITLE_FONTSIZE	13
3.7.2.4	TEXT_FONTSIZE	13
3.8	Riferimenti per la struct look_into_t	13
3.8.1	Descrizione dettagliata	14
3.8.2	Documentazione dei membri dato	14
3.8.2.1	email	14
3.8.2.2	filename	14
3.8.2.3	is_searcher	14
3.9	Riferimenti per la struct ls_t	15
3.9.1	Descrizione dettagliata	15
3.9.2	Documentazione dei membri dato	15
3.9.2.1	files	15
3.9.2.2	nfile	16
3.10	Riferimenti per la struct match_vars_t	16
3.10.1	Descrizione dettagliata	17
3.10.2	Documentazione dei membri dato	17
3.10.2.1	ast_queues	17
3.10.2.2	asteroid_head	18
3.10.2.3	asteroid_to_bonus	18
3.10.2.4	asteroid_to_generate	18
3.10.2.5	asteroids_destroyed	18
3.10.2.6	current_asteroid	18
3.10.2.7	fps	18
3.10.2.8	gameover	18
3.10.2.9	insert_asteroid_index	18
3.10.2.10	instant_ast_dest	19
3.10.2.11	items	19
3.10.2.12	key_error	19
3.10.2.13	music_playwa	19

3.10.2.14 playing	19
3.10.2.15 score	19
3.10.2.16 sound_error	19
3.10.2.17 sound_exp	19
3.10.2.18 sound_fire	20
3.10.2.19 sound_gameover	20
3.10.2.20 spawn_interval	20
3.10.2.21 spawn_timer	20
3.10.2.22 timer_rallenty_bonus	20
3.10.2.23 timer_shield_bonus	20
3.11 Riferimenti per la struct menu_gv_t	20
3.11.1 Descrizione dettagliata	21
3.11.2 Documentazione dei membri dato	21
3.11.2.1 COPYRIGHT_FONTSIZE	21
3.11.2.2 COPYRIGHT_TOP_OFFSET	21
3.11.2.3 INTERNAL_OFFSET	21
3.11.2.4 OPTION_FONTSIZE	21
3.11.2.5 OPTION_H	22
3.11.2.6 OPTION_TEXT_OFFSET	22
3.11.2.7 OPTION_W	22
3.11.2.8 TITLE_H	22
3.11.2.9 TITLE_W	22
3.11.2.10 UPPER_OFFSET	22
3.12 Riferimenti per la struct playwa_gv_t	22
3.12.1 Descrizione dettagliata	23
3.12.2 Documentazione dei membri dato	23
3.12.2.1 ASTEROID_FONTSIZE	23
3.12.2.2 ASTEROID_SHADOW_GAP	23
3.12.2.3 ASTEROID_TEXT_SHADOW_GAP	23
3.12.2.4 BONUS_HINT_FONTSIZE	24

3.12.2.5	BONUS_LEFT_OFFSET	24
3.12.2.6	BONUS_NUM_FONTSIZE	24
3.12.2.7	BONUS_TOP_OFFSET	24
3.12.2.8	BULLET_RADIUS	24
3.12.2.9	BULLET_SCALE	24
3.12.2.10	DANGER_RADIUS	24
3.12.2.11	HEIGHT_BONUS	24
3.12.2.12	HINT_FONTSIZE	25
3.12.2.13	HINT_LEFT_OFFSET	25
3.12.2.14	HINT_TOP_OFFSET	25
3.12.2.15	SCORE_FONTSIZE	25
3.12.2.16	SCORE_LEFT_OFFSET	25
3.12.2.17	SCORE_TOP_OFFSET	25
3.12.2.18	SHOOTER_AREA_RAD	25
3.12.2.19	SHOOTER_SCALE	25
3.12.2.20	TEXT_INTERNAL_OFFSET	26
3.13	Riferimenti per la struct result_t	26
3.13.1	Descrizione dettagliata	26
3.13.2	Documentazione dei membri dato	26
3.13.2.1	count_result	26
3.13.2.2	pass_result	26
3.14	Riferimenti per la struct search_manager_t	27
3.14.1	Descrizione dettagliata	27
3.14.2	Documentazione dei membri dato	27
3.14.2.1	email	28
3.14.2.2	file_list	28
3.14.2.3	res	28
3.14.2.4	threads	28
3.15	Riferimenti per la struct settings_t	28
3.15.1	Descrizione dettagliata	28

3.15.2 Documentazione dei membri dato	29
3.15.2.1 music	29
3.15.2.2 sound	29
3.15.2.3 visible_hints	29
3.16 Riferimenti per la struct shooter_t	29
3.16.1 Descrizione dettagliata	29
3.16.2 Documentazione dei membri dato	30
3.16.2.1 radius	30
3.16.2.2 slope	30
3.16.2.3 x	30
3.16.2.4 y	30
3.17 Riferimenti per la struct thread_manager_t	30
3.17.1 Descrizione dettagliata	31
3.17.2 Documentazione dei membri dato	32
3.17.2.1 consumer_sem	32
3.17.2.2 gameover_cond	32
3.17.2.3 gameover_mutex	32
3.17.2.4 search_manager	32
3.17.2.5 search_result_mutex	32
3.17.2.6 searcher_sem	32
3.17.2.7 searcher_thread	32
3.17.2.8 spawn_cond	33
3.17.2.9 spawn_mutex	33
3.17.2.10 spawner_thread	33
3.17.2.11 words_buffer_mutex	33
3.18 Riferimenti per la struct usr_gv_t	33
3.18.1 Descrizione dettagliata	33
3.18.2 Documentazione dei membri dato	34
3.18.2.1 ENTER_FONTSIZE	34
3.18.2.2 ENTER_HEIGHT	34
3.18.2.3 ENTER_LEFT_OFFSET	34
3.18.2.4 ENTER_TOP_OFFSET	34
3.18.2.5 ENTER_WIDTH	34
3.18.2.6 STATS_FONTSIZE	34
3.18.2.7 USERNAME_BLOCK_TOP_OFFSET	34
3.18.2.8 USERNAME_FONTSIZE	35
3.19 Riferimenti per la struct words_buffer_t	35
3.19.1 Descrizione dettagliata	35
3.19.2 Documentazione dei membri dato	35
3.19.2.1 load	35
3.19.2.2 words	35

4 Documentazione dei file	37
4.1 Riferimenti per il file <code>archive_functions.cpp</code>	37
4.1.1 Documentazione delle funzioni	37
4.1.1.1 <code>start_searching()</code>	38
4.2 Riferimenti per il file <code>archive_functions.h</code>	38
4.2.1 Documentazione delle funzioni	39
4.2.1.1 <code>start_searching()</code>	39
4.2.2 Documentazione delle variabili	39
4.2.2.1 <code>thread_manager</code>	39
4.2.2.2 <code>words_buffer</code>	39
4.3 Riferimenti per il file <code>data.h</code>	40
4.3.1 Documentazione delle definizioni	43
4.3.1.1 <code>BG_REFRESH_TIME</code>	43
4.3.1.2 <code>GAMERES_H</code>	43
4.3.1.3 <code>GAMERES_W</code>	43
4.3.1.4 <code>MAX_ASTEROID_HEAD</code>	44
4.3.1.5 <code>MAX_DIGIT_SCORE</code>	44
4.3.1.6 <code>MAX_LENGTH_WORD</code>	44
4.3.1.7 <code>MAX_MATCHING_PASS</code>	44
4.3.1.8 <code>MAX_USERNAME_LENGTH</code>	44
4.3.1.9 <code>MENU_CPY_COLOR</code>	44
4.3.1.10 <code>MENU_FONT_COLOR</code>	45
4.3.1.11 <code>MIN_TIME_SPAWN</code>	45
4.3.1.12 <code>MONITOR_OFFSET_RATE</code>	45
4.3.1.13 <code>NUM_INSTR_BITMAPS</code>	45
4.3.1.14 <code>NUM_INSTR_FONTS</code>	45
4.3.1.15 <code>NUM_ITEMS</code>	45
4.3.1.16 <code>NUM_MENU_BITMAPS</code>	46
4.3.1.17 <code>NUM_MENU_FONTS</code>	46
4.3.1.18 <code>NUM_OPTIONS</code>	46

4.3.1.19	NUM_OPTIONS_SETTINGS	46
4.3.1.20	NUM_PLAYWA_BITMAPS	46
4.3.1.21	NUM_PLAYWA_FONTS	46
4.3.1.22	NUM_SETTINGS_FONTS	47
4.3.1.23	NUM_USER_FONTS	47
4.3.1.24	SCORE_INCREMENT	47
4.3.1.25	SHADOW_COLOR	47
4.3.1.26	TIMER_RALLENTY_BONUS	47
4.3.1.27	TIMER_SHIELD_BONUS	47
4.3.2	Documentazione delle ridefinizioni di tipo (typedef)	47
4.3.2.1	asteroid_list	48
4.3.3	Documentazione dei tipi enumerati	48
4.3.3.1	asteroid_special_t	48
4.3.3.2	enable_t	48
4.3.4	Documentazione delle variabili	48
4.3.4.1	BMP_PATH_BASE	48
4.3.4.2	FONT_PATH	49
4.3.4.3	MUSIC_GAMEOVER_PATH_WA	49
4.3.4.4	MUSIC_MENU_PATH_WA	49
4.3.4.5	MUSIC_PLAYWA_PATH_WA	49
4.3.4.6	N_QUEUES	49
4.3.4.7	SOUND_PATH_ERROR	49
4.3.4.8	SOUND_PATH_EXPLOSION	50
4.3.4.9	SOUND_PATH_FIRE	50
4.3.4.10	SOUND_PATH_GAMEOVER	50
4.4	Riferimenti per il file drawing_functions.cpp	50
4.4.1	Documentazione delle funzioni	52
4.4.1.1	destroy_arr_bitmaps()	52
4.4.1.2	destroy_arr_fonts()	52
4.4.1.3	draw_back()	53

4.4.1.4	draw_background()	53
4.4.1.5	draw_gameover_animation()	53
4.4.1.6	draw_head()	53
4.4.1.7	draw_instructions()	54
4.4.1.8	draw_intro_init()	54
4.4.1.9	draw_intro_menu_fadein()	54
4.4.1.10	draw_intro_title_slide()	54
4.4.1.11	draw_menu_copyright()	55
4.4.1.12	draw_menu_option()	55
4.4.1.13	draw_menu_title()	55
4.4.1.14	draw_playagain_button()	55
4.4.1.15	draw_playwa_asteroids_bullets()	56
4.4.1.16	draw_playwa_bonus_music()	56
4.4.1.17	draw_playwa_score_hint()	56
4.4.1.18	draw_playwa_shooter()	57
4.4.1.19	draw_results()	57
4.4.1.20	draw_stat()	57
4.4.1.21	draw_stats_blocks()	57
4.4.1.22	draw_username()	58
4.4.1.23	draw_username_block()	58
4.4.1.24	generate_instr_bitmaps()	58
4.4.1.25	generate_instr_fonts()	58
4.4.1.26	generate_menu_bitmaps()	58
4.4.1.27	generate_menu_fonts()	59
4.4.1.28	generate_playwa_bitmaps()	59
4.4.1.29	generate_playwa_fonts()	59
4.4.1.30	generate_settings_fonts()	59
4.4.1.31	generate_user_fonts()	59
4.5	Riferimenti per il file drawing_functions.h	60
4.5.1	Documentazione delle funzioni	62

4.5.1.1	<code>destroy_arr_bitmaps()</code>	62
4.5.1.2	<code>destroy_arr_fonts()</code>	62
4.5.1.3	<code>draw_back()</code>	63
4.5.1.4	<code>draw_background()</code>	63
4.5.1.5	<code>draw_gameover_animation()</code>	63
4.5.1.6	<code>draw_head()</code>	63
4.5.1.7	<code>draw_instructions()</code>	64
4.5.1.8	<code>draw_intro_init()</code>	64
4.5.1.9	<code>draw_intro_menu_fadein()</code>	64
4.5.1.10	<code>draw_intro_title_slide()</code>	64
4.5.1.11	<code>draw_menu_copyright()</code>	65
4.5.1.12	<code>draw_menu_option()</code>	65
4.5.1.13	<code>draw_menu_title()</code>	65
4.5.1.14	<code>draw_playagain_button()</code>	65
4.5.1.15	<code>draw_playwa_asteroids_bullets()</code>	66
4.5.1.16	<code>draw_playwa_bonus_music()</code>	66
4.5.1.17	<code>draw_playwa_score_hint()</code>	66
4.5.1.18	<code>draw_playwa_shooter()</code>	67
4.5.1.19	<code>draw_results()</code>	67
4.5.1.20	<code>draw_stat()</code>	67
4.5.1.21	<code>draw_stats_blocks()</code>	67
4.5.1.22	<code>draw_username()</code>	68
4.5.1.23	<code>draw_username_block()</code>	68
4.5.1.24	<code>generate_instr_bitmaps()</code>	68
4.5.1.25	<code>generate_instr_fonts()</code>	68
4.5.1.26	<code>generate_menu_bitmaps()</code>	68
4.5.1.27	<code>generate_menu_fonts()</code>	69
4.5.1.28	<code>generate_playwa_bitmaps()</code>	69
4.5.1.29	<code>generate_playwa_fonts()</code>	69
4.5.1.30	<code>generate_settings_fonts()</code>	69

4.5.1.31	<code>generate_user_fonts()</code>	69
4.5.2	Documentazione delle variabili	69
4.5.2.1	<code>display_info</code>	70
4.5.2.2	<code>GENERAL_GV</code>	70
4.5.2.3	<code>INSTR_GV</code>	70
4.5.2.4	<code>MENU_GV</code>	70
4.5.2.5	<code>PLAYWA_GV</code>	70
4.5.2.6	<code>USR_GV</code>	70
4.6	Riferimenti per il file <code>game_functions.cpp</code>	71
4.6.1	Documentazione delle funzioni	72
4.6.1.1	<code>change_music()</code>	72
4.6.1.2	<code>check_distance()</code>	72
4.6.1.3	<code>check_gameover()</code>	72
4.6.1.4	<code>check_shield()</code>	73
4.6.1.5	<code>generate_asteroid()</code>	73
4.6.1.6	<code>handler_key_pressed()</code>	73
4.6.1.7	<code>handler_timer()</code>	74
4.6.1.8	<code>move_asteroids()</code>	74
4.6.1.9	<code>move_bullets()</code>	74
4.6.1.10	<code>move_shooter()</code>	74
4.6.1.11	<code>music_wa_play()</code>	75
4.6.1.12	<code>sound_play()</code>	75
4.6.1.13	<code>spawn_asteroid_thread()</code>	75
4.7	Riferimenti per il file <code>game_functions.h</code>	75
4.7.1	Documentazione delle funzioni	77
4.7.1.1	<code>change_music()</code>	77
4.7.1.2	<code>check_distance()</code>	77
4.7.1.3	<code>check_gameover()</code>	77
4.7.1.4	<code>check_shield()</code>	78
4.7.1.5	<code>generate_asteroid()</code>	78

4.7.1.6	handler_key_pressed()	78
4.7.1.7	handler_timer()	79
4.7.1.8	move_asteroids()	79
4.7.1.9	move_bullets()	79
4.7.1.10	move_shooter()	79
4.7.1.11	music_wa_play()	80
4.7.1.12	sound_play()	80
4.7.1.13	spawn_asteroid_thread()	80
4.7.2	Documentazione delle variabili	80
4.7.2.1	display_info	80
4.7.2.2	music_wa	80
4.7.2.3	PLAYWA_GV	81
4.7.2.4	thread_manager	81
4.7.2.5	words_buffer	81
4.8	Riferimenti per il file graphical_values.cpp	81
4.8.1	Documentazione delle funzioni	82
4.8.1.1	update_graphical_values()	82
4.8.2	Documentazione delle variabili	82
4.8.2.1	GENERAL_GV	82
4.8.2.2	INSTR_GV	82
4.8.2.3	MENU_GV	82
4.8.2.4	PLAYWA_GV	82
4.8.2.5	USR_GV	83
4.9	Riferimenti per il file graphical_values.h	83
4.9.1	Documentazione delle funzioni	84
4.9.1.1	update_graphical_values()	84
4.9.2	Documentazione delle variabili	84
4.9.2.1	display_info	84
4.10	Riferimenti per il file init.cpp	84
4.10.1	Documentazione delle funzioni	85

4.10.1.1	init_arr_bullet()	85
4.10.1.2	init_bullet()	85
4.10.1.3	init_display()	86
4.10.1.4	init_display_res()	86
4.10.1.5	init_game_requirements()	86
4.10.1.6	init_match_vars()	86
4.10.1.7	init_queue_event()	86
4.10.1.8	init_settings()	87
4.10.1.9	init_shooter()	87
4.10.1.10	init_sound_music()	87
4.10.1.11	init_target_bonus()	87
4.10.1.12	init_thread_manager()	88
4.10.1.13	init_words_buffer()	88
4.10.1.14	sort_file_list()	88
4.11	Riferimenti per il file init.h	88
4.11.1	Documentazione delle funzioni	90
4.11.1.1	init_arr_bullet()	90
4.11.1.2	init_bullet()	90
4.11.1.3	init_display()	90
4.11.1.4	init_display_res()	91
4.11.1.5	init_game_requirements()	91
4.11.1.6	init_match_vars()	91
4.11.1.7	init_queue_event()	91
4.11.1.8	init_settings()	91
4.11.1.9	init_shooter()	92
4.11.1.10	init_sound_music()	92
4.11.1.11	init_target_bonus()	92
4.11.1.12	init_thread_manager()	92
4.11.1.13	init_words_buffer()	93
4.11.2	Documentazione delle variabili	93

4.11.2.1	display_info	93
4.11.2.2	music_wa	93
4.11.2.3	PLAYWA_GV	93
4.12	Riferimenti per il file main.cpp	93
4.12.1	Documentazione delle funzioni	94
4.12.1.1	main()	94
4.12.2	Documentazione delle variabili	94
4.12.2.1	display_info	94
4.12.2.2	music_wa	94
4.12.2.3	thread_manager	94
4.12.2.4	words_buffer	95
4.13	Riferimenti per il file sections.cpp	95
4.13.1	Documentazione delle funzioni	95
4.13.1.1	intro()	95
4.13.1.2	main_menu()	96
4.13.1.3	play_wa()	96
4.13.1.4	settings_menu()	96
4.14	Riferimenti per il file sections.h	97
4.14.1	Documentazione delle funzioni	98
4.14.1.1	intro()	98
4.14.1.2	main_menu()	98
4.14.1.3	play_wa()	98
4.14.1.4	settings_menu()	99
4.14.2	Documentazione delle variabili	99
4.14.2.1	display_info	99
4.14.2.2	MENU_GV	99
4.14.2.3	music_wa	99
4.14.2.4	thread_manager	99
4.14.2.5	USR_GV	100
4.14.2.6	words_buffer	100

4.15	Riferimenti per il file utilities.cpp	100
4.15.1	Documentazione delle funzioni	101
4.15.1.1	archives_exist()	101
4.15.1.2	change_enable()	101
4.15.1.3	delete_ast_queues()	101
4.15.1.4	delete_display_evqueue_music()	101
4.15.1.5	delete_sounds()	102
4.15.1.6	delete_timer()	102
4.15.1.7	exit()	102
4.15.1.8	extract_elem()	102
4.15.1.9	insert_elem_head()	103
4.15.1.10	int_to_str_score()	103
4.15.1.11	ls_tar()	103
4.15.1.12	which_quadrant()	103
4.16	Riferimenti per il file utilities.h	104
4.16.1	Documentazione delle funzioni	105
4.16.1.1	archives_exist()	105
4.16.1.2	change_enable()	105
4.16.1.3	delete_ast_queues()	106
4.16.1.4	delete_display_evqueue_music()	106
4.16.1.5	delete_sounds()	106
4.16.1.6	delete_timer()	106
4.16.1.7	exit()	107
4.16.1.8	extract_elem()	107
4.16.1.9	insert_elem_head()	107
4.16.1.10	int_to_str_score()	107
4.16.1.11	ls_tar()	108
4.16.1.12	which_quadrant()	108
4.16.2	Documentazione delle variabili	108
4.16.2.1	display_info	108

Capitolo 1

Indice dei tipi composti

1.1 Elenco dei tipi composti

Queste sono le classi, le struct, le union e le interfacce con una loro breve descrizione:

asteroid_t	Struct contenente le informazioni relative al singolo asteroide	5
bullet_t	Struct contenente le informazioni relative al singolo proiettile	7
display_info_t	Struct contenente le informazioni relative al display	8
elem_asteroid_t	Struct del singolo elemento di una lista di asteroidi	9
file_t	Struct relativa alla descrizione di un archivio che andrà esaminato	10
general_gv_t	Struct relativo alle proporzioni grafiche generali	11
instr_gv_t	Struct relativo alle proporzioni grafiche della schermata di pausa	12
look_into_t	Struct di appoggio utilizzata dai thread searcher/worker per ricevere informazioni riguardo: 1) filename - dentro quale archivio cercare; 2) email - quale email cercare; 3) is_searcher - se il thread si tratta di un searcher (fornisce le parole al gioco) o di un worker	13
ls_t	Struct che raccoglie tutti gli archivi	15
match_vars_t	Struct contenente le variabili di gioco	16
menu_gv_t	Struct relativo alle proporzioni grafiche dei menu	20
playwa_gv_t	Struct relativo alle proporzioni grafiche della schermata di gioco	22
result_t	Struct contenenti le password corrispondenti alla mail data in input dall'utente	26
search_manager_t	Struct che rappresenta il manager dei thread atti a ricercare le password all'interno degli archivi	27
settings_t	Struct contenente le impostazioni di gioco	28
shooter_t	Struct contenente le informazioni relative al singolo proiettile	29
thread_manager_t	Struct che contiene le variabili di sincronizzazione del gioco	30

usr_gv_t	Struct relativo alle proporzioni grafiche della schermata di fine partita	33
words_buffer_t	Struct buffer che implementa l'architettura producer/consumer	35

Capitolo 2

Indice dei file

2.1 Elenco dei file

Questo è un elenco di tutti i file con una loro breve descrizione:

archive_functions.cpp	37
archive_functions.h	38
data.h	40
drawing_functions.cpp	50
drawing_functions.h	60
game_functions.cpp	71
game_functions.h	75
graphical_values.cpp	81
graphical_values.h	83
init.cpp	84
init.h	88
main.cpp	93
sections.cpp	95
sections.h	97
utilities.cpp	100
utilities.h	104

Capitolo 3

Documentazione delle classi

3.1 Riferimenti per la struct asteroid_t

Struct contenente le informazioni relative al singolo asteroide.

```
#include <data.h>
```

Attributi pubblici

- float [x](#)
- float [y](#)
- float [slope](#)
- float [radius](#)
- ALLEGRO_BITMAP * [bmp](#)
- [asteroid_special_t](#) [special](#)
- char [word](#) [255]

3.1.1 Descrizione dettagliata

Struct contenente le informazioni relative al singolo asteroide.

Campi:

- 1) [x](#) - Ascissa dell'asteroide;
- 2) [y](#) - Ordinata dell'asteroide;
- 3) [slope](#) - Inclinazione della traiettoria dell'asteroide;
- 4) [type](#) - Tipo dell'asteroide (corrisponde alla lunghezza della parola; [3,14]);
- 5) [radius](#) - Raggio dell'asteroide;
- 6) [bmp](#) - Bitmap dell'asteroide;
- 7) [special](#) - Bonus contenuto nell'asteroide;
- 8) [word](#) - Parola relativa all'asteroide.

3.1.2 Documentazione dei membri dato

3.1.2.1 bmp

```
ALLEGRO_BITMAP* asteroid_t::bmp
```

3.1.2.2 radius

```
float asteroid_t::radius
```

3.1.2.3 slope

```
float asteroid_t::slope
```

3.1.2.4 special

```
asteroid_special_t asteroid_t::special
```

3.1.2.5 word

```
char asteroid_t::word[255]
```

3.1.2.6 x

```
float asteroid_t::x
```

3.1.2.7 y

```
float asteroid_t::y
```

La documentazione per questa struct è stata generata a partire dal seguente file:

- [data.h](#)

3.2 Riferimenti per la struct `bullet_t`

Struct contenente le informazioni relative al singolo proiettile.

```
#include <data.h>
```

Attributi pubblici

- float `x`
- float `y`
- float `radius`
- float `slope`

3.2.1 Descrizione dettagliata

Struct contenente le informazioni relative al singolo proiettile.

Campi:

- 1) `x` - Ascissa del proiettile;
- 2) `y` - Ordinata del proiettile;
- 3) `slope` - Inclinazione della traiettoria del proiettile;
- 4) `radius` - Raggio del proiettile.

3.2.2 Documentazione dei membri dato

3.2.2.1 `radius`

```
float bullet_t::radius
```

3.2.2.2 `slope`

```
float bullet_t::slope
```

3.2.2.3 `x`

```
float bullet_t::x
```

3.2.2.4 y

```
float bullet_t::y
```

La documentazione per questa struct è stata generata a partire dal seguente file:

- [data.h](#)

3.3 Riferimenti per la struct display_info_t

Struct contenente le informazioni relative al display.

```
#include <data.h>
```

Attributi pubblici

- float [width](#)
- float [height](#)

3.3.1 Descrizione dettagliata

Struct contenente le informazioni relative al display.

Campi:

- 1) width - Larghezza (in pixel) del display;
- 2) height - Altezza (in pixel) del display.

3.3.2 Documentazione dei membri dato

3.3.2.1 height

```
float display_info_t::height
```

3.3.2.2 width

```
float display_info_t::width
```

La documentazione per questa struct è stata generata a partire dal seguente file:

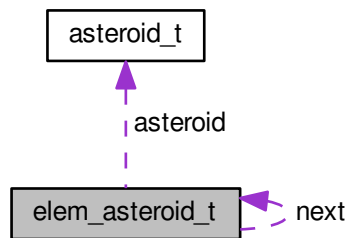
- [data.h](#)

3.4 Riferimenti per la struct elem_asteroid_t

Struct del singolo elemento di una lista di asteroidi.

```
#include <data.h>
```

Diagramma di collaborazione per elem_asteroid_t:



Attributi pubblici

- `asteroid_t asteroid`
- `elem_asteroid_t * next`

3.4.1 Descrizione dettagliata

Struct del singolo elemento di una lista di asteroidi.

Campi:

- 1) `asteroid` - Asteroide;
- 2) `next` - Puntatore al prossimo elemento.

3.4.2 Documentazione dei membri dato

3.4.2.1 `asteroid`

```
asteroid_t elem_asteroid_t::asteroid
```

3.4.2.2 next

```
elem_asteroid_t* elem_asteroid_t::next
```

La documentazione per questa struct è stata generata a partire dal seguente file:

- [data.h](#)

3.5 Riferimenti per la struct file_t

Struct relativa alla descrizione di un archivio che andrà esaminato.

```
#include <data.h>
```

Attributi pubblici

- char * [filename](#)
- long int [size](#)

3.5.1 Descrizione dettagliata

Struct relativa alla descrizione di un archivio che andrà esaminato.

Raccoglie il nome (l'intero percorso) e la sua dimensione per poterli ordinare.

3.5.2 Documentazione dei membri dato

3.5.2.1 filename

```
char* file_t::filename
```

3.5.2.2 size

```
long int file_t::size
```

La documentazione per questa struct è stata generata a partire dal seguente file:

- [data.h](#)

3.6 Riferimenti per la struct `general_gv_t`

Struct relativo alle proporzioni grafiche generali.

```
#include <data.h>
```

Attributi pubblici

- float `HEAD_TOP_OFFSET`
- float `HEAD_TEXT_SHADOW_GAP`
- float `BACK_TOP_OFFSET`
- float `BACK_LEFT_OFFSET`
- float `BACK_TEXT_SHADOW_GAP`
- float `RESULT_TOP_OFFSET`
- int `HEAD_FONTSIZE`
- int `BACK_FONTSIZE`

3.6.1 Descrizione dettagliata

Struct relativo alle proporzioni grafiche generali.

3.6.2 Documentazione dei membri dato

3.6.2.1 `BACK_FONTSIZE`

```
int general_gv_t::BACK_FONTSIZE
```

3.6.2.2 `BACK_LEFT_OFFSET`

```
float general_gv_t::BACK_LEFT_OFFSET
```

3.6.2.3 `BACK_TEXT_SHADOW_GAP`

```
float general_gv_t::BACK_TEXT_SHADOW_GAP
```

3.6.2.4 BACK_TOP_OFFSET

```
float general_gv_t::BACK_TOP_OFFSET
```

3.6.2.5 HEAD_FONTSIZE

```
int general_gv_t::HEAD_FONTSIZE
```

3.6.2.6 HEAD_TEXT_SHADOW_GAP

```
float general_gv_t::HEAD_TEXT_SHADOW_GAP
```

3.6.2.7 HEAD_TOP_OFFSET

```
float general_gv_t::HEAD_TOP_OFFSET
```

3.6.2.8 RESULT_TOP_OFFSET

```
float general_gv_t::RESULT_TOP_OFFSET
```

La documentazione per questa struct è stata generata a partire dal seguente file:

- [data.h](#)

3.7 Riferimenti per la struct `instr_gv_t`

Struct relativo alle proporzioni grafiche della schermata di pausa.

```
#include <data.h>
```

Attributi pubblici

- float [BONUS_RADIUS](#)
- float [BONUS_CASE_WIDTH](#)
- int [SUBTITLE_FONTSIZE](#)
- int [TEXT_FONTSIZE](#)

3.7.1 Descrizione dettagliata

Struct relativo alle proporzioni grafiche della schermata di pausa.

3.7.2 Documentazione dei membri dato

3.7.2.1 BONUS_CASE_WIDTH

```
float instr_gv_t::BONUS_CASE_WIDTH
```

3.7.2.2 BONUS_RADIUS

```
float instr_gv_t::BONUS_RADIUS
```

3.7.2.3 SUBTITLE_FONTSIZE

```
int instr_gv_t::SUBTITLE_FONTSIZE
```

3.7.2.4 TEXT_FONTSIZE

```
int instr_gv_t::TEXT_FONTSIZE
```

La documentazione per questa struct è stata generata a partire dal seguente file:

- [data.h](#)

3.8 Riferimenti per la struct look_into_t

Struct di appoggio utilizzata dai thread searcher/worker per ricevere informazioni riguardo: 1) filename - dentro quale archivio cercare; 2) email - quale email cercare; 3) is_searcher - se il thread si tratta di un searcher (fornisce le parole al gioco) o di un worker.

```
#include <data.h>
```

Attributi pubblici

- char * [filename](#)
- char * [email](#)
- bool [is_searcher](#)

3.8.1 Descrizione dettagliata

Struct di appoggio utilizzata dai thread searcher/worker per ricevere informazioni riguardo: 1) filename - dentro quale archivio cercare; 2) email - quale email cercare; 3) is_searcher - se il thread si tratta di un searcher (fornisce le parole al gioco) o di un worker.

3.8.2 Documentazione dei membri dato

3.8.2.1 email

```
char* look_into_t::email
```

3.8.2.2 filename

```
char* look_into_t::filename
```

3.8.2.3 is_searcher

```
bool look_into_t::is_searcher
```

La documentazione per questa struct è stata generata a partire dal seguente file:

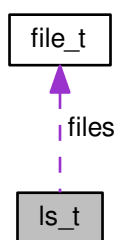
- [data.h](#)

3.9 Riferimenti per la struct ls_t

Struct che raccoglie tutti gli archivi.

```
#include <data.h>
```

Diagramma di collaborazione per ls_t:



Attributi pubblici

- `file_t * files`
- `int nfile`

3.9.1 Descrizione dettagliata

Struct che raccoglie tutti gli archivi.

Tiene anche conto di quelli letti attraverso un array di flag.

3.9.2 Documentazione dei membri dato

3.9.2.1 files

```
file_t* ls_t::files
```

3.9.2.2 nfile

```
int ls_t::nfile
```

La documentazione per questa struct è stata generata a partire dal seguente file:

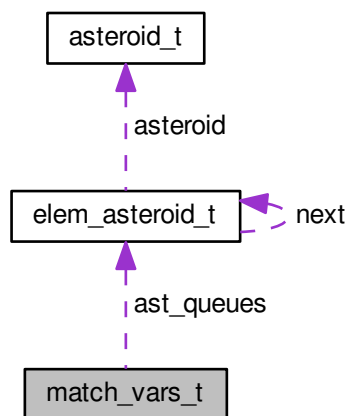
- [data.h](#)

3.10 Riferimenti per la struct match_vars_t

Struct contenente le variabili di gioco.

```
#include <data.h>
```

Diagramma di collaborazione per match_vars_t:



Attributi pubblici

- `asteroid_list ast_queues [N_QUEUES]`
- `int score`
- `int asteroids_destroyed`
- `ALLEGRO_TIMER * spawn_timer`
- `double spawn_interval`
- `ALLEGRO_TIMER * timer_shield_bonus`
- `ALLEGRO_TIMER * timer_rallenty_bonus`
- `float fps`
- `int insert_asteroid_index`
- `int asteroid_head`
- `asteroid_special_t asteroid_to_generate`
- `ALLEGRO_SAMPLE * sound_exp`
- `ALLEGRO_SAMPLE * sound_fire`

- ALLEGRO_SAMPLE * `sound_gameover`
- ALLEGRO_SAMPLE * `sound_error`
- ALLEGRO_SAMPLE_INSTANCE * `music_playwa`
- int `current_asteroid`
- int `key_error`
- `enable_t` `playing`
- bool `gameover`
- int `items` [NUM_ITEMS]
- int `asteroid_to_bonus`
- int `instant_ast_dest`

3.10.1 Descrizione dettagliata

Struct contenente le variabili di gioco.

Campi:

- 1) `ast_queues` - Array di liste di asteroidi;
- 2) `score` - Punteggio;
- 3) `asteroids_destroyed` - Numero di asteroidi distrutti;
- 4) `spawn_interval` - Tempo di attesa (espresso in secondi) tra la generazione di due asteroidi;
- 5) `spawn_timer` - Timer relativo alla generazione di asteroidi;
- 6) `timer_shield_bonus` - Timer relativo alla durata del bonus "Shield";
- 7) `timer_rallenty_bonus` - Timer relativo alla durata del bonus "Rallenty";
- 8) `fps` - Intervallo di refresh della schermata;
- 9) `diff_liv` - Livello di difficoltà del gioco ([3,14]);
- 10) `insert_asteroid_index` - Indice che dice se c'è un asteroide in coda che andrà fatto comparire nel gioco;
- 11) `asteroid_head` - Numero di asteroidi presenti in testa nel vettore delle liste ([0,NUM_QUEUES]);
- 12) `asteroid_to_generate` - Tipo dell'asteroide che verrà generato;
- 13) `sound_exp` - Suono dell'esplosione di un asteroide;
- 14) `sound_fire` - Suono dello sparo di un proiettile;
- 15) `sound_gameover` - Suono che viene eseguito quando si verifica il gameover;
- 16) `sound_rallenty` - Suono che sostituisce la musica durante il bonus "Rallenty";
- 17) `sound_error` - Suono che viene eseguito quando si digita una lettera sbagliata;
- 18) `current_asteroid` - Indice dell'asteroide che si sta cercando di distruggere.
Vale -1 se si deve ancora agganciare un asteroide da colpire;
- 19) `key_error` - Indice dell'asteroide del quale si è sbagliata la lettera. Vale -1 se non vi sono errori;
- 20) `playing` - Variabile che indica se il gioco è in pausa o si sta giocando;
- 21) `gameover` - Variabile che indica se si è verificato il gameover;
- 22) `items` - Array dei bonus: indica la numerosità di ognuno dei quattro bonus;
- 23) `asteroid_to_bonus` - Numero di asteroidi da distruggere per ottenere un bonus;
- 24) `instant_ast_dest` - Numero di asteroidi distrutti in un determinato istante, in modo da calcolare se si è arrivati al bonus.

3.10.2 Documentazione dei membri dato

3.10.2.1 `ast_queues`

```
asteroid_list match_vars_t::ast_queues[N_QUEUES]
```

3.10.2.2 asteroid_head

```
int match_vars_t::asteroid_head
```

3.10.2.3 asteroid_to_bonus

```
int match_vars_t::asteroid_to_bonus
```

3.10.2.4 asteroid_to_generate

```
asteroid_special_t match_vars_t::asteroid_to_generate
```

3.10.2.5 asteroids_destroyed

```
int match_vars_t::asteroids_destroyed
```

3.10.2.6 current_asteroid

```
int match_vars_t::current_asteroid
```

3.10.2.7 fps

```
float match_vars_t::fps
```

3.10.2.8 gameover

```
bool match_vars_t::gameover
```

3.10.2.9 insert_asteroid_index

```
int match_vars_t::insert_asteroid_index
```

3.10.2.10 `instant_ast_dest`

```
int match_vars_t::instant_ast_dest
```

3.10.2.11 `items`

```
int match_vars_t::items[NUM\_ITEMS]
```

3.10.2.12 `key_error`

```
int match_vars_t::key_error
```

3.10.2.13 `music_playwa`

```
ALLEGRO\_SAMPLE\_INSTANCE\* match_vars_t::music_playwa
```

3.10.2.14 `playing`

```
enable\_t match_vars_t::playing
```

3.10.2.15 `score`

```
int match_vars_t::score
```

3.10.2.16 `sound_error`

```
ALLEGRO\_SAMPLE\* match_vars_t::sound_error
```

3.10.2.17 `sound_exp`

```
ALLEGRO\_SAMPLE\* match_vars_t::sound_exp
```

3.10.2.18 sound_fire

```
ALLEGRO_SAMPLE* match_vars_t::sound_fire
```

3.10.2.19 sound_gameover

```
ALLEGRO_SAMPLE* match_vars_t::sound_gameover
```

3.10.2.20 spawn_interval

```
double match_vars_t::spawn_interval
```

3.10.2.21 spawn_timer

```
ALLEGRO_TIMER* match_vars_t::spawn_timer
```

3.10.2.22 timer_rallenty_bonus

```
ALLEGRO_TIMER* match_vars_t::timer_rallenty_bonus
```

3.10.2.23 timer_shield_bonus

```
ALLEGRO_TIMER* match_vars_t::timer_shield_bonus
```

La documentazione per questa struct è stata generata a partire dal seguente file:

- [data.h](#)

3.11 Riferimenti per la struct menu_gv_t

Struct relativo alle proporzioni grafiche dei menu.

```
#include <data.h>
```

Attributi pubblici

- float [TITLE_W](#)
- float [TITLE_H](#)
- float [OPTION_W](#)
- float [OPTION_H](#)
- float [OPTION_TEXT_OFFSET](#)
- float [UPPER_OFFSET](#)
- float [INTERNAL_OFFSET](#)
- float [COPYRIGHT_TOP_OFFSET](#)
- int [OPTION_FONTSIZE](#)
- int [COPYRIGHT_FONTSIZE](#)

3.11.1 Descrizione dettagliata

Struct relativo alle proporzioni grafiche dei menu.

3.11.2 Documentazione dei membri dato

3.11.2.1 COPYRIGHT_FONTSIZE

```
int menu_gv_t::COPYRIGHT_FONTSIZE
```

3.11.2.2 COPYRIGHT_TOP_OFFSET

```
float menu_gv_t::COPYRIGHT_TOP_OFFSET
```

3.11.2.3 INTERNAL_OFFSET

```
float menu_gv_t::INTERNAL_OFFSET
```

3.11.2.4 OPTION_FONTSIZE

```
int menu_gv_t::OPTION_FONTSIZE
```

3.11.2.5 OPTION_H

```
float menu_gv_t::OPTION_H
```

3.11.2.6 OPTION_TEXT_OFFSET

```
float menu_gv_t::OPTION_TEXT_OFFSET
```

3.11.2.7 OPTION_W

```
float menu_gv_t::OPTION_W
```

3.11.2.8 TITLE_H

```
float menu_gv_t::TITLE_H
```

3.11.2.9 TITLE_W

```
float menu_gv_t::TITLE_W
```

3.11.2.10 UPPER_OFFSET

```
float menu_gv_t::UPPER_OFFSET
```

La documentazione per questa struct è stata generata a partire dal seguente file:

- [data.h](#)

3.12 Riferimenti per la struct `playwa_gv_t`

Struct relativo alle proporzioni grafiche della schermata di gioco.

```
#include <data.h>
```


Attributi pubblici

- float `TEXT_INTERNAL_OFFSET`
- float `SCORE_TOP_OFFSET`
- float `SCORE_LEFT_OFFSET`
- float `HINT_TOP_OFFSET`
- float `HINT_LEFT_OFFSET`
- float `SHOOTER_AREA_RAD`
- float `SHOOTER_SCALE`
- float `ASTEROID_SHADOW_GAP`
- float `ASTEROID_TEXT_SHADOW_GAP`
- float `BULLET_RADIUS`
- float `HEIGHT_BONUS`
- float `BONUS_TOP_OFFSET`
- float `BONUS_LEFT_OFFSET`
- float `BULLET_SCALE`
- float `DANGER_RADIUS`
- int `SCORE_FONTSIZE`
- int `HINT_FONTSIZE`
- int `ASTEROID_FONTSIZE`
- int `BONUS_NUM_FONTSIZE`
- int `BONUS_HINT_FONTSIZE`

3.12.1 Descrizione dettagliata

Struct relativo alle proporzioni grafiche della schermata di gioco.

3.12.2 Documentazione dei membri dato

3.12.2.1 `ASTEROID_FONTSIZE`

```
int playwa_gv_t::ASTEROID_FONTSIZE
```

3.12.2.2 `ASTEROID_SHADOW_GAP`

```
float playwa_gv_t::ASTEROID_SHADOW_GAP
```

3.12.2.3 `ASTEROID_TEXT_SHADOW_GAP`

```
float playwa_gv_t::ASTEROID_TEXT_SHADOW_GAP
```

3.12.2.4 BONUS_HINT_FONTSIZE

```
int playwa_gv_t::BONUS_HINT_FONTSIZE
```

3.12.2.5 BONUS_LEFT_OFFSET

```
float playwa_gv_t::BONUS_LEFT_OFFSET
```

3.12.2.6 BONUS_NUM_FONTSIZE

```
int playwa_gv_t::BONUS_NUM_FONTSIZE
```

3.12.2.7 BONUS_TOP_OFFSET

```
float playwa_gv_t::BONUS_TOP_OFFSET
```

3.12.2.8 BULLET_RADIUS

```
float playwa_gv_t::BULLET_RADIUS
```

3.12.2.9 BULLET_SCALE

```
float playwa_gv_t::BULLET_SCALE
```

3.12.2.10 DANGER_RADIUS

```
float playwa_gv_t::DANGER_RADIUS
```

3.12.2.11 HEIGHT_BONUS

```
float playwa_gv_t::HEIGHT_BONUS
```

3.12.2.12 HINT_FONTSIZE

```
int playwa_gv_t::HINT_FONTSIZE
```

3.12.2.13 HINT_LEFT_OFFSET

```
float playwa_gv_t::HINT_LEFT_OFFSET
```

3.12.2.14 HINT_TOP_OFFSET

```
float playwa_gv_t::HINT_TOP_OFFSET
```

3.12.2.15 SCORE_FONTSIZE

```
int playwa_gv_t::SCORE_FONTSIZE
```

3.12.2.16 SCORE_LEFT_OFFSET

```
float playwa_gv_t::SCORE_LEFT_OFFSET
```

3.12.2.17 SCORE_TOP_OFFSET

```
float playwa_gv_t::SCORE_TOP_OFFSET
```

3.12.2.18 SHOOTER_AREA_RAD

```
float playwa_gv_t::SHOOTER_AREA_RAD
```

3.12.2.19 SHOOTER_SCALE

```
float playwa_gv_t::SHOOTER_SCALE
```

3.12.2.20 TEXT_INTERNAL_OFFSET

```
float playwa_gv_t::TEXT_INTERNAL_OFFSET
```

La documentazione per questa struct è stata generata a partire dal seguente file:

- [data.h](#)

3.13 Riferimenti per la struct result_t

Struct contenenti le password corrispondenti alla mail data in input dall'utente.

```
#include <data.h>
```

Attributi pubblici

- char * [pass_result](#) [[MAX_MATCHING_PASS](#)]
- int [count_result](#)

3.13.1 Descrizione dettagliata

Struct contenenti le password corrispondenti alla mail data in input dall'utente.

3.13.2 Documentazione dei membri dato

3.13.2.1 count_result

```
int result_t::count_result
```

3.13.2.2 pass_result

```
char* result_t::pass_result [MAX\_MATCHING\_PASS]
```

La documentazione per questa struct è stata generata a partire dal seguente file:

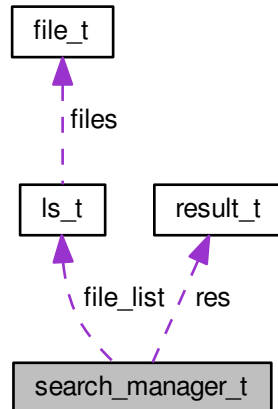
- [data.h](#)

3.14 Riferimenti per la struct search_manager_t

Struct che rappresenta il manager dei thread atti a ricercare le password all'interno degli archivi.

```
#include <data.h>
```

Diagramma di collaborazione per search_manager_t:



Attributi pubblici

- pthread_t * [threads](#)
- ls_t * [file_list](#)
- result_t [res](#)
- char * [email](#)

3.14.1 Descrizione dettagliata

Struct che rappresenta il manager dei thread atti a ricercare le password all'interno degli archivi.

Campi:

- 1) t - Array di thread searcher/worker;
- 2) file_list - Lista degli archivi in cui cercare;
- 3) res - Password che corrisponde all'email dell'utente;
- 4) email - Email inserita dall'utente.

3.14.2 Documentazione dei membri dato

3.14.2.1 email

```
char* search_manager_t::email
```

3.14.2.2 file_list

```
ls_t* search_manager_t::file_list
```

3.14.2.3 res

```
result_t search_manager_t::res
```

3.14.2.4 threads

```
pthread_t* search_manager_t::threads
```

La documentazione per questa struct è stata generata a partire dal seguente file:

- [data.h](#)

3.15 Riferimenti per la struct settings_t

Struct contenente le impostazioni di gioco.

```
#include <data.h>
```

Attributi pubblici

- [enable_t](#) sound
- [enable_t](#) music
- [bool](#) [visible_hints](#)

3.15.1 Descrizione dettagliata

Struct contenente le impostazioni di gioco.

Campi:

2) sound - Impostazione relativa agli effetti sonori ([OFF,ON]);

3) music - Impostazione relativa alla musica ([OFF,ON]);

4) visible_hints - Se "true", i suggerimenti presenti sulla schermata di gioco risultano essere visibili.

3.15.2 Documentazione dei membri dato

3.15.2.1 music

```
enable_t settings_t::music
```

3.15.2.2 sound

```
enable_t settings_t::sound
```

3.15.2.3 visible_hints

```
bool settings_t::visible_hints
```

La documentazione per questa struct è stata generata a partire dal seguente file:

- [data.h](#)

3.16 Riferimenti per la struct shooter_t

Struct contenente le informazioni relative al singolo proiettile.

```
#include <data.h>
```

Attributi pubblici

- float [x](#)
- float [y](#)
- float [slope](#)
- float [radius](#)

3.16.1 Descrizione dettagliata

Struct contenente le informazioni relative al singolo proiettile.

Campi:

- 1) x - Ascissa della navicella (centro del display);
- 2) y - Ordinata della navicella (centro del display);
- 3) slope - Angolo di rotazione della "testa" della navicella;
- 4) radius - Raggio della navicella.

3.16.2 Documentazione dei membri dato

3.16.2.1 radius

```
float shooter_t::radius
```

3.16.2.2 slope

```
float shooter_t::slope
```

3.16.2.3 x

```
float shooter_t::x
```

3.16.2.4 y

```
float shooter_t::y
```

La documentazione per questa struct è stata generata a partire dal seguente file:

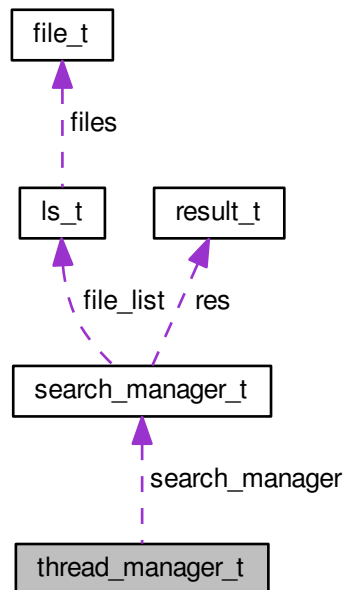
- [data.h](#)

3.17 Riferimenti per la struct `thread_manager_t`

Struct che contiene le variabili di sincronizzazione del gioco.

```
#include <data.h>
```


Diagramma di collaborazione per `thread_manager_t`:



Attributi pubblici

- `search_manager_t search_manager`
- `pthread_t searcher_thread`
- `pthread_t spawner_thread`
- `sem_t searcher_sem`
- `sem_t consumer_sem`
- `pthread_cond_t spawn_cond`
- `pthread_mutex_t spawn_mutex`
- `pthread_mutex_t words_buffer_mutex`
- `pthread_cond_t gameover_cond`
- `pthread_mutex_t gameover_mutex`
- `pthread_mutex_t search_result_mutex`

3.17.1 Descrizione dettagliata

Struct che contiene le variabili di sincronizzazione del gioco.

Campi: 1) `search_manager` - struct che contiene le informazioni per la ricerca dentro gli archivi; 2) `searcher_thread` - thread che si occupa della ricerca delle informazioni (producer); 3) `spawner_thread` - thread che si occupa di fornire le parole al gioco (consumer); 4) `searcher_sem` - semaforo privato per i producer; 5) `consumer_sem` - semaforo privato per i consumer; 6) `spawn_cond` - condition per la sincronizzazione dello spawner thread; 7) `spawn_mutex` - mutex per la sincronizzazione dello spawner thread; 8) `words_buffer_mutex` - mutex per gestire il buffer condiviso; 9) `gameover_cond` - condition per la sincronizzazione dei threads con il gioco principale; 10) `gameover_mutex` - mutex per la sincronizzazione dei threads con il gioco principale; 11) `search_result_mutex` - mutex per gestire la mutua esclusione sui risultati della ricerca;

3.17.2 Documentazione dei membri dato

3.17.2.1 consumer_sem

`sem_t thread_manager_t::consumer_sem`

3.17.2.2 gameover_cond

`pthread_cond_t thread_manager_t::gameover_cond`

3.17.2.3 gameover_mutex

`pthread_mutex_t thread_manager_t::gameover_mutex`

3.17.2.4 search_manager

`search_manager_t thread_manager_t::search_manager`

3.17.2.5 search_result_mutex

`pthread_mutex_t thread_manager_t::search_result_mutex`

3.17.2.6 searcher_sem

`sem_t thread_manager_t::searcher_sem`

3.17.2.7 searcher_thread

`pthread_t thread_manager_t::searcher_thread`

3.17.2.8 `spawn_cond`

```
pthread_cond_t thread_manager_t::spawn_cond
```

3.17.2.9 `spawn_mutex`

```
pthread_mutex_t thread_manager_t::spawn_mutex
```

3.17.2.10 `spawner_thread`

```
pthread_t thread_manager_t::spawner_thread
```

3.17.2.11 `words_buffer_mutex`

```
pthread_mutex_t thread_manager_t::words_buffer_mutex
```

La documentazione per questa struct è stata generata a partire dal seguente file:

- [data.h](#)

3.18 Riferimenti per la struct `usr_gv_t`

Struct relativo alle proporzioni grafiche della schermata di fine partita.

```
#include <data.h>
```

Attributi pubblici

- float [USERNAME_BLOCK_TOP_OFFSET](#)
- float [ENTER_LEFT_OFFSET](#)
- float [ENTER_TOP_OFFSET](#)
- float [ENTER_WIDTH](#)
- float [ENTER_HEIGHT](#)
- int [USERNAME_FONTSIZE](#)
- int [STATS_FONTSIZE](#)
- int [ENTER_FONTSIZE](#)

3.18.1 Descrizione dettagliata

Struct relativo alle proporzioni grafiche della schermata di fine partita.

3.18.2 Documentazione dei membri dato

3.18.2.1 ENTER_FONTSIZE

```
int usr_gv_t::ENTER_FONTSIZE
```

3.18.2.2 ENTER_HEIGHT

```
float usr_gv_t::ENTER_HEIGHT
```

3.18.2.3 ENTER_LEFT_OFFSET

```
float usr_gv_t::ENTER_LEFT_OFFSET
```

3.18.2.4 ENTER_TOP_OFFSET

```
float usr_gv_t::ENTER_TOP_OFFSET
```

3.18.2.5 ENTER_WIDTH

```
float usr_gv_t::ENTER_WIDTH
```

3.18.2.6 STATS_FONTSIZE

```
int usr_gv_t::STATS_FONTSIZE
```

3.18.2.7 USERNAME_BLOCK_TOP_OFFSET

```
float usr_gv_t::USERNAME_BLOCK_TOP_OFFSET
```

3.18.2.8 USERNAME_FONTSIZE

```
int usr_gv_t::USERNAME_FONTSIZE
```

La documentazione per questa struct è stata generata a partire dal seguente file:

- [data.h](#)

3.19 Riferimenti per la struct words_buffer_t

Struct buffer che implementa l'architettura producer/consumer.

```
#include <data.h>
```

Attributi pubblici

- char * [words](#) [[N_QUEUES](#)]
- int [load](#)

3.19.1 Descrizione dettagliata

Struct buffer che implementa l'architettura producer/consumer.

3.19.2 Documentazione dei membri dato

3.19.2.1 load

```
int words_buffer_t::load
```

3.19.2.2 words

```
char* words_buffer_t::words[N\_QUEUES]
```

La documentazione per questa struct è stata generata a partire dal seguente file:

- [data.h](#)

Capitolo 4

Documentazione dei file

4.1 Riferimenti per il file archive_functions.cpp

```
#include <stdio.h>
#include <string.h>
#include <dirent.h>
#include <malloc.h>
#include <archive.h>
#include <archive_entry.h>
#include <stdlib.h>
#include <pthread.h>
#include <semaphore.h>
#include <cctype>
#include "archive_functions.h"
```

Grafo delle dipendenze di inclusione per archive_functions.cpp:



Funzioni

- void * [start_searching](#) (void *search_manager)

Funzione che gestisce i thread searcher e worker.

4.1.1 Documentazione delle funzioni

4.1.1.1 start_searching()

```
void* start_searching (
    void * search_manager )
```

Funzione che gestisce i thread searcher e worker.

Questi thread gestiscono la ricerca dell'email dello username all'interno delle collection.

Parametri:

1) search_manager - Variabile contenente le strutture per la sincronizzazione la ricerca.

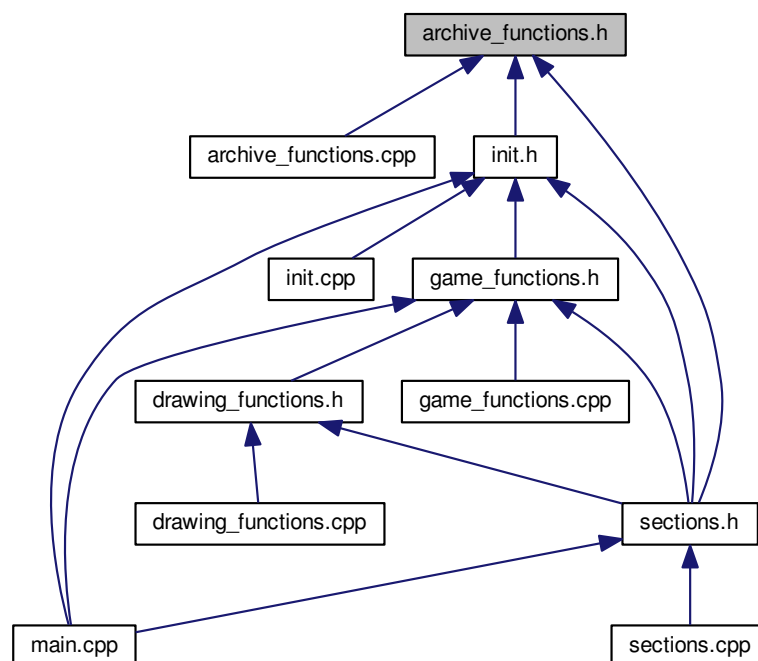
4.2 Riferimenti per il file archive_functions.h

```
#include "data.h"
```

Grafo delle dipendenze di inclusione per archive_functions.h:



Questo grafo mostra quali altri file includono direttamente o indirettamente questo file:



Funzioni

- void * [start_searching](#) (void *search_manager)
Funzione che gestisce i thread searcher e worker.

Variabili

- [thread_manager_t](#) thread_manager
- [words_buffer_t](#) words_buffer

4.2.1 Documentazione delle funzioni

4.2.1.1 start_searching()

```
void* start_searching (
    void * search_manager )
```

Funzione che gestisce i thread searcher e worker.

Questi thread gestiscono la ricerca dell'email dello username all'interno delle collection.

Parametri:

1) search_manager - Variabile contenente le strutture per la sincronizzazione la ricerca.

4.2.2 Documentazione delle variabili

4.2.2.1 thread_manager

```
thread\_manager\_t thread_manager
```

4.2.2.2 words_buffer

```
words\_buffer\_t words_buffer
```

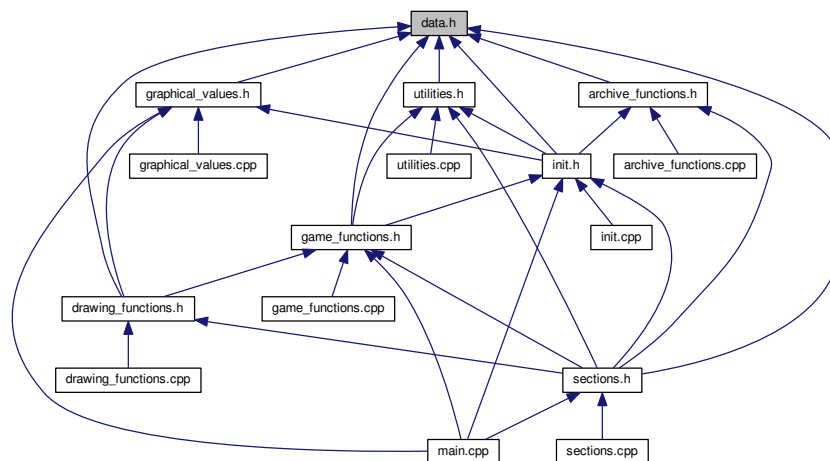
4.3 Riferimenti per il file data.h

```
#include <allegro5/allegro.h>
#include <allegro5/allegro_native_dialog.h>
#include <allegro5/allegro_image.h>
#include <allegro5/allegro_primitives.h>
#include <allegro5/allegro_font.h>
#include <allegro5/allegro_ttf.h>
#include <allegro5/allegro_audio.h>
#include <allegro5/allegro_acodec.h>
#include <semaphore.h>
```

Grafo delle dipendenze di inclusione per data.h:



Questo grafo mostra quali altri file includono direttamente o indirettamente questo file:



Composti

- struct [display_info_t](#)
Struct contenente le informazioni relative al display.
- struct [asteroid_t](#)
Struct contenente le informazioni relative al singolo asteroide.
- struct [elem_asteroid_t](#)
Struct del singolo elemento di una lista di asteroidi.
- struct [bullet_t](#)
Struct contenente le informazioni relative al singolo proiettile.
- struct [shooter_t](#)
Struct contenente le informazioni relative al singolo proiettile.
- struct [result_t](#)

- *Struct contenenti le password corrispondenti alla mail data in input dall'utente.*
- struct [match_vars_t](#)
 - *Struct contenente le variabili di gioco.*
- struct [settings_t](#)
 - *Struct contenente le impostazioni di gioco.*
- struct [general_gv_t](#)
 - *Struct relativo alle proporzioni grafiche generali.*
- struct [menu_gv_t](#)
 - *Struct relativo alle proporzioni grafiche dei menu.*
- struct [playwa_gv_t](#)
 - *Struct relativo alle proporzioni grafiche della schermata di gioco.*
- struct [instr_gv_t](#)
 - *Struct relativo alle proporzioni grafiche della schermata di pausa.*
- struct [usr_gv_t](#)
 - *Struct relativo alle proporzioni grafiche della schermata di fine partita.*
- struct [file_t](#)
 - *Struct relativa alla descrizione di un archivio che andrà esaminato.*
- struct [ls_t](#)
 - *Struct che raccoglie tutti gli archivi.*
- struct [search_manager_t](#)
 - *Struct che rappresenta il manager dei thread atti a ricercare le password all'interno degli archivi.*
- struct [thread_manager_t](#)
 - *Struct che contiene le variabili di sincronizzazione del gioco.*
- struct [words_buffer_t](#)
 - *Struct buffer che implementa l'architettura producer/consumer.*
- struct [look_into_t](#)
 - *Struct di appoggio utilizzata dai thread searcher/worker per ricevere informazioni riguardo: 1) filename - dentro quale archivio cercare; 2) email - quale email cercare; 3) is_searcher - se il thread si tratta di un searcher (fornisce le parole al gioco) o di un worker.*

Definizioni

- #define [TIMER_SHIELD_BONUS](#) 8.0
 - *Tempo (in secondi) di durata del bonus "Shield".*
- #define [TIMER_RALLENTY_BONUS](#) 5.0
 - *Tempo (in secondi) di durata del bonus "Rallenty".*
- #define [GAMERES_W](#) 1280
 - *Larghezza della risoluzione base.*
- #define [GAMERES_H](#) 800
 - *Altezza della risoluzione base.*
- #define [MONITOR_OFFSET_RATE](#) 5
 - *Frazione di schermo (in altezza) lasciata libera per la creazione della schermata di gioco.*
- #define [NUM_ITEMS](#) 4
 - *Numerosità dei bonus.*
- #define [MAX_USERNAME_LENGTH](#) 40
 - *Lunghezza massima dello username del giocatore.*
- #define [MAX_ASTEROID_HEAD](#) 9
 - *Numero massimo di asteroidi in campo (ovvero in testa alle code).*
- #define [MAX_DIGIT_SCORE](#) 8
 - *Numero massimo di caratteri relativi allo score.*

- #define `MAX_LENGTH_WORD` 8
Lunghezza massima delle parole che verranno presentate al giocatore.
- #define `MAX_MATCHING_PASS` 10
Massimo numero di password cercate negli archivi.
- #define `MIN_TIME_SPAWN` 0.5
Tempo minimo di spawn tra un asteroide e l'altro.
- #define `NUM_MENU_BITMAPS` 4
Numero di bitmap relativi ai menu.
- #define `NUM_PLAYWA_BITMAPS` 14
Numero di bitmap relativi alla schermata di gioco.
- #define `NUM_INSTR_BITMAPS` 6
Numero di bitmap relativi alla pausa.
- #define `NUM_MENU_FONTS` 3
Numero di font relativi ai menu.
- #define `NUM_INSTR_FONTS` 4
Numero di font relativi alla schermata di pausa.
- #define `NUM_SETTINGS_FONTS` 3
Numero di font relativi al menu delle impostazioni.
- #define `NUM_PLAYWA_FONTS` 5
Numero di font relativi alla schermata di gioco.
- #define `NUM_USER_FONTS` 4
Numero di font relativi alla schermata di fine partita.
- #define `BG_REFRESH_TIME` 0.0005
Tempo (espresso in secondi) di aggiornamento relativo allo scorrimento dello sfondo.
- #define `NUM_OPTIONS` 3
Numero di opzioni presenti nel menu principale.
- #define `NUM_OPTIONS_SETTINGS` 3
Numero di opzioni presenti nel menu delle impostazioni.
- #define `SCORE_INCREMENT` 50
Punteggio dato dalla distruzione di un asteroide.
- #define `SHADOW_COLOR` `al_map_rgba_f(0, 0, 0, 0.75)`
- #define `MENU_FONT_COLOR` `al_map_rgb(246, 242, 212)`
- #define `MENU_CPY_COLOR` `al_map_rgb(122, 124, 119)`

Ridefinizioni di tipo (typedef)

- typedef `elem_asteroid_t` * `asteroid_list`
Definizione del tipo "lista di asteroidi".

Tipi enumerati (enum)

- enum `enable_t` { `ON`, `OFF` }
Enumerato relativo alle impostazioni di gioco: attive o non attive.
- enum `asteroid_special_t` {
 `NORMAL`, `ATOMIC`, `RALLENTY`, `FIRE`,
 `SHIELD` }
Enumerato relativo al bonus contenuto nel singonlo asteroide: nessuno, Atomic, Rallenty, Fire o Shield.

Variabili

- const int `N_QUEUES` = 'Z' - 'A' + 1
Numerosita' delle code di asteroidi ('A'..'Z') e del buffer.
- const char `MUSIC_MENU_PATH_WA` [] = "media/sounds/menu.wav"
Percorso del file audio relativo alla musica dei menu.
- const char `MUSIC_GAMEOVER_PATH_WA` [] = "media/sounds/gameover.wav"
Percorso del file audio relativo alla musica della schermata di gameover.
- const char `SOUND_PATH_EXPLOSION` [] = "media/sounds/explosion.wav"
Percorso del file audio relativo all'effetto sonoro dell'esplosione di un asteroide.
- const char `SOUND_PATH_FIRE` [] = "media/sounds/fire.wav"
Percorso del file audio relativo all'effetto sonoro dello sparo di un proiettile.
- const char `SOUND_PATH_ERROR` [] = "media/sounds/error.wav"
Percorso del file audio relativo all'effetto sonoro che viene riprodotto quando si sbaglia una lettera.
- const char `SOUND_PATH_GAMEOVER` [] = "media/sounds/death.wav"
Percorso del file audio relativo all'effetto sonoro che viene riprodotto quando un asteroide tocca la navicella.
- const char `MUSIC_PLAYWA_PATH_WA` [] = "media/sounds/playwa.wav"
Percorso del file audio relativo alla musica della schermata di gioco.
- const char `FONT_PATH` [] = "media/font/visitor.ttf"
Percorso del font utilizzato in tutto il progetto.
- const char `BMP_PATH_BASE` [] = "media/images/"
Percorso della cartella contenente le immagini.

4.3.1 Documentazione delle definizioni

4.3.1.1 BG_REFRESH_TIME

```
#define BG_REFRESH_TIME 0.0005
```

Tempo (espresso in secondi) di aggiornamento relativo allo scorrimento dello sfondo.

4.3.1.2 GAMERES_H

```
#define GAMERES_H 800
```

Altezza della risoluzione base.

4.3.1.3 GAMERES_W

```
#define GAMERES_W 1280
```

Larghezza della risoluzione base.

4.3.1.4 MAX_ASTEROID_HEAD

```
#define MAX_ASTEROID_HEAD 9
```

Numero massimo di asteroidi in campo (ovvero in testa alle code).

Questo accorgimento garantisce un normale flusso nella generazione degli asteroidi: dal momento che le parole relative agli asteroidi vengono generate casualmente, troppe teste occupate (Es. 9 su 10) provocherebbero una situazione di "stallo" dovuto alla probabilità minima di individuare casualmente le teste libere (Es. La decima).

4.3.1.5 MAX_DIGIT_SCORE

```
#define MAX_DIGIT_SCORE 8
```

Numero massimo di caratteri relativi allo score.

4.3.1.6 MAX_LENGTH_WORD

```
#define MAX_LENGTH_WORD 8
```

Lunghezza massima delle parole che verranno presentate al giocatore.

4.3.1.7 MAX_MATCHING_PASS

```
#define MAX_MATCHING_PASS 10
```

Massimo numero di password cercate negli archivi.

4.3.1.8 MAX_USERNAME_LENGTH

```
#define MAX_USERNAME_LENGTH 40
```

Lunghezza massima dello username del giocatore.

4.3.1.9 MENU_CPY_COLOR

```
#define MENU_CPY_COLOR al_map_rgb(122, 124, 119)
```

4.3.1.10 MENU_FONT_COLOR

```
#define MENU_FONT_COLOR al_map_rgb(246, 242, 212)
```

4.3.1.11 MIN_TIME_SPAWN

```
#define MIN_TIME_SPAWN 0.5
```

Tempo minimo di spawn tra un asteroide e l'altro.

4.3.1.12 MONITOR_OFFSET_RATE

```
#define MONITOR_OFFSET_RATE 5
```

Frazione di schermo (in altezza) lasciata libera per la creazione della schermata di gioco.

4.3.1.13 NUM_INSTR_BITMAPS

```
#define NUM_INSTR_BITMAPS 6
```

Numero di bitmap relativi alla pausa.

4.3.1.14 NUM_INSTR_FONTS

```
#define NUM_INSTR_FONTS 4
```

Numero di font relativi alla schermata di pausa.

4.3.1.15 NUM_ITEMS

```
#define NUM_ITEMS 4
```

Numerosità dei bonus.

4.3.1.16 NUM_MENU_BITMAPS

```
#define NUM_MENU_BITMAPS 4
```

Numero di bitmap relativi ai menu.

4.3.1.17 NUM_MENU_FONTS

```
#define NUM_MENU_FONTS 3
```

Numero di font relativi ai menu.

4.3.1.18 NUM_OPTIONS

```
#define NUM_OPTIONS 3
```

Numero di opzioni presenti nel menu principale.

4.3.1.19 NUM_OPTIONS_SETTINGS

```
#define NUM_OPTIONS_SETTINGS 3
```

Numero di opzioni presenti nel menu delle impostazioni.

4.3.1.20 NUM_PLAYWA_BITMAPS

```
#define NUM_PLAYWA_BITMAPS 14
```

Numero di bitmap relativi alla schermata di gioco.

4.3.1.21 NUM_PLAYWA_FONTS

```
#define NUM_PLAYWA_FONTS 5
```

Numero di font relativi alla schermata di gioco.

4.3.1.22 NUM_SETTINGS_FONTS

```
#define NUM_SETTINGS_FONTS 3
```

Numero di font relativi al menu delle impostazioni.

4.3.1.23 NUM_USER_FONTS

```
#define NUM_USER_FONTS 4
```

Numero di font relativi alla schermata di fine partita.

4.3.1.24 SCORE_INCREMENT

```
#define SCORE_INCREMENT 50
```

Punteggio dato dalla distruzione di un asteroide.

4.3.1.25 SHADOW_COLOR

```
#define SHADOW_COLOR al_map_rgba_f(0, 0, 0, 0.75)
```

4.3.1.26 TIMER_RALLENTY_BONUS

```
#define TIMER_RALLENTY_BONUS 5.0
```

Tempo (in secondi) di durata del bonus "Rallenty".

4.3.1.27 TIMER_SHIELD_BONUS

```
#define TIMER_SHIELD_BONUS 8.0
```

Tempo (in secondi) di durata del bonus "Shield".

4.3.2 Documentazione delle ridefinizioni di tipo (typedef)

4.3.2.1 asteroid_list

```
typedef elem_asteroid_t* asteroid_list
```

Definizione del tipo "lista di asteroidi".

4.3.3 Documentazione dei tipi enumerati

4.3.3.1 asteroid_special_t

```
enum asteroid_special_t
```

Enumerato relativo al bonus contenuto nel singonlo asteroide: nessuno, Atomic, Rallenty, Fire o Shield.

Valori del tipo enumerato

NORMAL	
ATOMIC	
RALLENTY	
FIRE	
SHIELD	

4.3.3.2 enable_t

```
enum enable_t
```

Enumerato relativo alle impostazioni di gioco: attive o non attive.

Valori del tipo enumerato

ON	
OFF	

4.3.4 Documentazione delle variabili

4.3.4.1 BMP_PATH_BASE

```
const char BMP_PATH_BASE[] = "media/images/"
```

Percorso della cartella contenente le immagini.

4.3.4.2 FONT_PATH

```
const char FONT_PATH[] = "media/font/visitor.ttf"
```

Percorso del font utilizzato in tutto il progetto.

4.3.4.3 MUSIC_GAMEOVER_PATH_WA

```
const char MUSIC_GAMEOVER_PATH_WA[] = "media/sounds/gameover.wav"
```

Percorso del file audio relativo alla musica della schermata di gameover.

4.3.4.4 MUSIC_MENU_PATH_WA

```
const char MUSIC_MENU_PATH_WA[] = "media/sounds/menu.wav"
```

Percorso del file audio relativo alla musica dei menu.

4.3.4.5 MUSIC_PLAYWA_PATH_WA

```
const char MUSIC_PLAYWA_PATH_WA[] = "media/sounds/playwa.wav"
```

Percorso del file audio relativo alla musica della schermata di gioco.

4.3.4.6 N_QUEUES

```
const int N_QUEUES = 'Z' - 'A' + 1
```

Numerosita' delle code di asteroidi ('A'..'Z') e del buffer.

4.3.4.7 SOUND_PATH_ERROR

```
const char SOUND_PATH_ERROR[] = "media/sounds/error.wav"
```

Percorso del file audio relativo all'effetto sonoro che viene riprodotto quando si sbaglia una lettera.

4.3.4.8 SOUND_PATH_EXPLOSION

```
const char SOUND_PATH_EXPLOSION[] = "media/sounds/explosion.wav"
```

Percorso del file audio relativo all'effetto sonoro dell'esplosione di un asteroide.

4.3.4.9 SOUND_PATH_FIRE

```
const char SOUND_PATH_FIRE[] = "media/sounds/fire.wav"
```

Percorso del file audio relativo all'effetto sonoro dello sparo di un proiettile.

4.3.4.10 SOUND_PATH_GAMEOVER

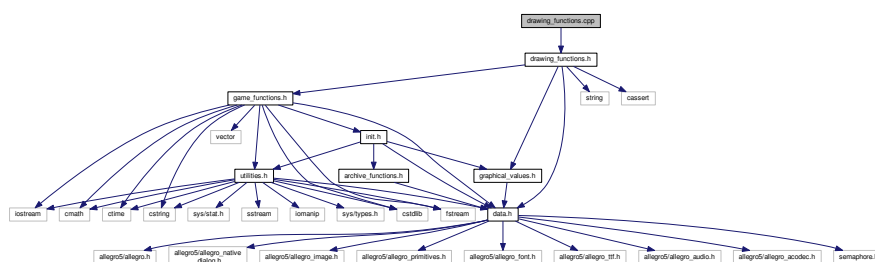
```
const char SOUND_PATH_GAMEOVER[] = "media/sounds/death.wav"
```

Percorso del file audio relativo all'effetto sonoro che viene riprodotto quando un asteroide tocca la navicella.

4.4 Riferimenti per il file drawing_functions.cpp

```
#include "drawing_functions.h"
```

Grafo delle dipendenze di inclusione per drawing_functions.cpp:



Funzioni

- `ALLEGRO_BITMAP ** generate_menu_bitmaps ()`
Funzione che genera l'array contenente i puntatori ai bitmap relativi alla schermata del menu principale.
- `ALLEGRO_BITMAP ** generate_playwa_bitmaps ()`
Funzione che genera l'array contenente i puntatori ai bitmap relativi alla schermata di gioco.
- `ALLEGRO_BITMAP ** generate_instr_bitmaps ()`
Funzione che genera l'array contenente i puntatori ai bitmap relativi alla schermata di pausa.
- `ALLEGRO_FONT ** generate_playwa_fonts ()`
Funzione che genera l'array contenente i puntatori ai font relativi alla schermata di gioco.
- `ALLEGRO_FONT ** generate_instr_fonts ()`
Funzione che genera l'array contenente i puntatori ai font relativi alla schermata di pausa.
- `ALLEGRO_FONT ** generate_menu_fonts ()`
Funzione che genera l'array contenente i puntatori ai font relativi ai menu.
- `ALLEGRO_FONT ** generate_user_fonts ()`
Funzione che genera l'array contenente i puntatori ai font relativi alla schermata di gameover.
- `ALLEGRO_FONT ** generate_settings_fonts ()`
Funzione che genera l'array contenente i puntatori ai font relativi alla schermata delle impostazioni.
- `void destroy_arr_bitmaps (ALLEGRO_BITMAP **arr, const int num)`
Funzione che dealloca un array di bitmap, oltre che tutti i bitmap in esso contenuti.
- `void destroy_arr_fonts (ALLEGRO_FONT **arr, const int num)`
Funzione che dealloca un array di font, oltre che tutti i font in esso contenuti.
- `void draw_background (ALLEGRO_BITMAP *bmp_bg, float &offset_bg, const bool playing)`
Funzione che stampa lo sfondo della schermata.
- `void draw_head (char *head_str, ALLEGRO_FONT *font)`
Funzione che stampa il titolo della schermata.
- `void draw_back (ALLEGRO_FONT *font, const bool pause)`
Funzione che stampa la stringa di ritorno al gioco, o al menu principale (in base alla variabile "pause").
- `void draw_intro_init (ALLEGRO_BITMAP **arr_bitmaps, float &bg_offset)`
Funzione che gestisce l'inizializzazione dell'animazione di introduzione.
- `void draw_intro_title_slide (ALLEGRO_BITMAP **arr_bitmaps, float &bg_offset)`
Funzione che gestisce lo slide del titolo nell'animazione di introduzione.
- `void draw_intro_menu_fadein (ALLEGRO_BITMAP **arr_bitmaps, ALLEGRO_FONT **arr_fonts, float &bg_offset)`
Funzione che gestisce il fadein del menu nell'animazione di introduzione.
- `void draw_menu_title (ALLEGRO_BITMAP *bmp_title)`
Funzione che stampa il bitmap relativo al titolo presente nel menu principale.
- `void draw_menu_option (const int &index, ALLEGRO_BITMAP *bmp_option, ALLEGRO_FONT *font, char *str_scelta, const bool settings)`
Funzione che stampa l'opzione indicata (menu).
- `void draw_menu_copyright (ALLEGRO_FONT *font)`
Funzione che stampa la stringa di copyright (menu).
- `void draw_playwa_score_hint (const match_vars_t &match_vars, ALLEGRO_FONT *font_score, ALLEGRO_FONT *font_hints, const bool visible_hints)`
Funzione che stampa il punteggio e i suggerimenti (schermata di gioco).
- `void draw_playwa_shooter (const match_vars_t &match_vars, const bullet_t *arr_bullet, const shooter_t &shooter, ALLEGRO_BITMAP **arr_bitmaps)`
Funzione che stampa lo shooter, la sua area (bianca normalmente, rossa in caso di pericolo o azzurra nel caso in cui il bonus "shield" venga attivato).
- `void draw_playwa_asteroids_bullets (const match_vars_t &match_vars, const bullet_t *arr_bullet, ALLEGRO_BITMAP **arr_bitmaps, ALLEGRO_FONT *font)`
Funzione che stampa lo gli asteroidi ed i proiettili presenti in gioco.

- void `draw_playwa_bonus_music` (const `match_vars_t` &match_vars, ALLEGRO_BITMAP **arr_bitmaps, ALLEGRO_FONT *font_score, ALLEGRO_FONT *font_bonus, ALLEGRO_FONT *font_bonus_hint, `settings_t` settings)
Funzione che stampa i bonus della partita (e le icone relative al suono).
- void `draw_stats_blocks` (ALLEGRO_FONT *font)
Funzione che disegna i blocchi di sfondo relativi ai risultati di fine partita.
- void `draw_stat` (const int &idx_stat, int stat_value, ALLEGRO_FONT *font)
Funzione che disegna un singolo risultato.
- void `draw_gameover_animation` (const int &score_goal, const int &dast_goal, ALLEGRO_BITMAP *bg_bmp, float &bg_offset, ALLEGRO_FONT **arr_fonts)
Funzione che gestisce l'animazione di fine partita.
- void `draw_playagain_button` (ALLEGRO_FONT *font)
Funzione che disegna il bottone "play again" nella schermata di Game Over.
- void `draw_results` (ALLEGRO_FONT *font, `result_t` res)
Funzione che disegna i risultati estratti dagli archivi corrispondenti alla password inserita dell'utente.
- void `draw_username_block` (const bool enter_active, ALLEGRO_FONT *font)
Funzione che disegna il blocco di inserimento dello username.
- void `draw_username` (char *username, const int &cur_pos, ALLEGRO_FONT *font, const bool bar_visible)
Funzione che disegna lo username.
- void `draw_instructions` (ALLEGRO_FONT **arr_fonts, ALLEGRO_BITMAP **arr_bitmaps)
Funzione che disegna le istruzioni durante la pausa.

4.4.1 Documentazione delle funzioni

4.4.1.1 `destroy_arr_bitmaps()`

```
void destroy_arr_bitmaps (
    ALLEGRO_BITMAP ** arr,
    const int num )
```

Funzione che dealloca un array di bitmap, oltre che tutti i bitmap in esso contenuti.

Parametri:

- 1) arr - Array di bitmap;
- 2) num - Dimensione dell'array di bitmap.

4.4.1.2 `destroy_arr_fonts()`

```
void destroy_arr_fonts (
    ALLEGRO_FONT ** arr,
    const int num )
```

Funzione che dealloca un array di font, oltre che tutti i font in esso contenuti.

Parametri:

- 1) arr - Array di font;
- 2) num - Dimensione dell'array di font.

4.4.1.3 `draw_back()`

```
void draw_back (
    ALLEGRO_FONT * font,
    const bool pause )
```

Funzione che stampa la stringa di ritorno al gioco, o al menu principale (in base alla variabile "pause").

Parametri:

- 1) font - Font relativo alla stringa di ritorno;
- 2) pause - In base al valore assunto da questa variabile viene stampata la stringa adatta.

4.4.1.4 `draw_background()`

```
void draw_background (
    ALLEGRO_BITMAP * bmp_bg,
    float & offset_bg,
    const bool playing )
```

Funzione che stampa lo sfondo della schermata.

Parametri:

- 1) bmp_bg - Bitmap relativo allo sfondo;
- 2) offset_bg - Scostamento in pixel dello sfondo dal lato sinistro della schermata (per rotazione sfondo);
- 3) playing - Flag sul quale si basa la velocità di rotazione (durante il gioco risulta essere più lenta).

4.4.1.5 `draw_gameover_animation()`

```
void draw_gameover_animation (
    const int & score_goal,
    const int & dast_goal,
    ALLEGRO_BITMAP * bg_bmp,
    float & bg_offset,
    ALLEGRO_FONT ** arr_fonts )
```

Funzione che gestisce l'animazione di fine partita.

Parametri:

- 1) score_goal - Score effettiva del giocatore;
- 2) dast_goal - Asteroidi effettivamente distrutti;
- 4) bg_bmp - Bitmap di sfondo;
- 5) bg_offset - Scostamento laterale dello sfondo;
- 6) arr_fonts - Array dei font necessari.

4.4.1.6 `draw_head()`

```
void draw_head (
    char * head_str,
    ALLEGRO_FONT * font )
```

Funzione che stampa il titolo della schermata.

Parametri:

- 1) head_str - Stringa relativa al titolo;
- 2) font - Font relativo al titolo.

4.4.1.7 draw_instructions()

```
void draw_instructions (
    ALLEGRO_FONT ** arr_fonts,
    ALLEGRO_BITMAP ** arr_bitmaps )
```

Funzione che disegna le istruzioni durante la pausa.

Parametri:

- 1) arr_fonts - Array contenente i font utilizzati;
- 2) arr_bitmaps - Array contenente i bitmap utilizzati.

4.4.1.8 draw_intro_init()

```
void draw_intro_init (
    ALLEGRO_BITMAP ** arr_bitmaps,
    float & bg_offset )
```

Funzione che gestisce l'inizializzazione dell'animazione di introduzione.

Parametri:

- 1) arr_bitmaps - Array di bitmap;
- 2) bg_offset - Offset iniziale dello sfondo.

4.4.1.9 draw_intro_menu_fadein()

```
void draw_intro_menu_fadein (
    ALLEGRO_BITMAP ** arr_bitmaps,
    ALLEGRO_FONT ** arr_fonts,
    float & bg_offset )
```

Funzione che gestisce il fadein del menu nell'animazione di introduzione.

Parametri:

- 1) arr_bitmaps - Array di bitmap;
- 2) arr_fonts - Array di font;
- 3) bg_offset - Offset iniziale dello sfondo.

4.4.1.10 draw_intro_title_slide()

```
void draw_intro_title_slide (
    ALLEGRO_BITMAP ** arr_bitmaps,
    float & bg_offset )
```

Funzione che gestisce lo slide del titolo nell'animazione di introduzione.

Parametri:

- 1) arr_bitmaps - Array di bitmap;
- 2) bg_offset - Offset iniziale dello sfondo.

4.4.1.11 `draw_menu_copyright()`

```
void draw_menu_copyright (
    ALLEGRO_FONT * font )
```

Funzione che stampa la stringa di copyright (menu).

Parametri:

- 1) font - Font relativo alla stringa di copyright.

4.4.1.12 `draw_menu_option()`

```
void draw_menu_option (
    const int & index,
    ALLEGRO_BITMAP * bmp_option,
    ALLEGRO_FONT * font,
    char * str_scelta,
    const bool settings )
```

Funzione che stampa l'opzione indicata (menu).

Parametri:

- 1) index - Indice dell'opzione;
- 2) bmp_option - Bitmap relativo all'opzione indicata;
- 3) font - Font relativo all'opzione indicata;
- 4) str_scelta - Stringa relativa all'opzione indicata;
- 5) settings - Se vero, stampa le opzioni leggermente più in alto rispetto al menu principale.

4.4.1.13 `draw_menu_title()`

```
void draw_menu_title (
    ALLEGRO_BITMAP * bmp_title )
```

Funzione che stampa il bitmap relativo al titolo presente nel menu principale.

Parametri:

- 1) bmp_title - Bitmap relativo al titolo.

4.4.1.14 `draw_playagain_button()`

```
void draw_playagain_button (
    ALLEGRO_FONT * font )
```

Funzione che disegna il bottone "play again" nella schermata di Game Over.

Parametri:

- 1) arr_fonts - Array dei font necessari;

4.4.1.15 draw_playwa_asteroids_bullets()

```
void draw_playwa_asteroids_bullets (
    const match_vars_t & match_vars,
    const bullet_t * arr_bullet,
    ALLEGRO_BITMAP ** arr_bitmaps,
    ALLEGRO_FONT * font )
```

Funzione che stampa lo gli asteroidi ed i proiettili presenti in gioco.

Parametri:

- 1) match_vars - Struct contenente le variabili di gioco;
- 2) arr_bullet - Array contenente i proiettili in gioco;
- 3) arr_bitmaps - Array contenente i bitmap di gioco;
- 4) font - Font relativo agli asteroidi.

4.4.1.16 draw_playwa_bonus_music()

```
void draw_playwa_bonus_music (
    const match_vars_t & match_vars,
    ALLEGRO_BITMAP ** arr_bitmaps,
    ALLEGRO_FONT * font_score,
    ALLEGRO_FONT * font_bonus,
    ALLEGRO_FONT * font_bonus_hint,
    settings_t settings )
```

Funzione che stampa i bonus della partita (e le icone relative al suono).

Parametri:

- 1) match_vars - Struct contenente le variabili di gioco;
- 2) arr_bitmaps - Array contenente i bitmap di gioco;
- 3) font_score - Font relativo al punteggio (utilizzato per stampare l'intestazione dei bonus);
- 4) font_bonus - Font relativo alla numerosità dei bonus;
- 5) font_bonus_hint - Font relativo al suggerimento per l'attivazione del bonus.

4.4.1.17 draw_playwa_score_hint()

```
void draw_playwa_score_hint (
    const match_vars_t & match_vars,
    ALLEGRO_FONT * font_score,
    ALLEGRO_FONT * font_hints,
    const bool visible_hints )
```

Funzione che stampa il punteggio e i suggerimenti (schermata di gioco).

Parametri:

- 1) match_vars - Struct contenente le variabili di gioco;
- 2) font_score - Font relativo al punteggio;
- 3) font_hints - Font relativo ai suggerimenti;
- 4) visible_hints - Se true, mostra i suggerimenti di gioco.

4.4.1.18 `draw_playwa_shooter()`

```
void draw_playwa_shooter (
    const match_vars_t & match_vars,
    const bullet_t * arr_bullet,
    const shooter_t & shooter,
    ALLEGRO_BITMAP ** arr_bitmaps )
```

Funzione che stampa lo shooter, la sua area (bianca normalmente, rossa in caso di pericolo o azzurra nel caso in cui il bonus "shield" venga attivato).

Parametri:

- 1) `match_vars` - Struct contenente le variabili di gioco;
- 2) `arr_bullet` - Array contenente i proiettili in gioco;
- 3) `shooter` - Struct contenente le variabili relative allo shooter;
- 4) `arr_bitmaps` - Array contenente i bitmap di gioco.

4.4.1.19 `draw_results()`

```
void draw_results (
    ALLEGRO_FONT * font,
    result_t res )
```

Funzione che disegna i risultati estratti dagli archivi corrispondenti alla password inserita dell'utente.

Parametri:

- 1) `arr_fonts` - Array dei font necessari;
- 2) `res` - Risultati dei match estratti;

4.4.1.20 `draw_stat()`

```
void draw_stat (
    const int & idx_stat,
    int stat_value,
    ALLEGRO_FONT * font )
```

Funzione che disegna un singolo risultato.

Parametri:

- 1) `idx_stat` - Indice del risultato;
- 2) `stat_value` - Valore del risultato;
- 3) `font` - Font utilizzato.

4.4.1.21 `draw_stats_blocks()`

```
void draw_stats_blocks (
    ALLEGRO_FONT * font )
```

Funzione che disegna i blocchi di sfondo relativi ai risultati di fine partita.

Parametri:

- 1) `font` - Font utilizzato.

4.4.1.22 draw_username()

```
void draw_username (
    char * username,
    const int & cur_pos,
    ALLEGRO_FONT * font,
    const bool bar_visible )
```

Funzione che disegna lo username.

- 1) username - Stringa dello username;
- 2) cur_pos - Posizione corrente all'interno della stringa;
- 3) font - Font utilizzato;
- 4) bar_visible - Se vero, il cursore è visibile.

4.4.1.23 draw_username_block()

```
void draw_username_block (
    const bool enter_active,
    ALLEGRO_FONT * font )
```

Funzione che disegna il blocco di inserimento dello username.

Parametri:

- 1) enter_active - Se true, è possibile cliccare il tasto di invio dello username;
- 2) font - Font utilizzato.

4.4.1.24 generate_instr_bitmaps()

```
ALLEGRO_BITMAP** generate_instr_bitmaps ( )
```

Funzione che genera l'array contenente i puntatori ai bitmap relativi alla schermata di pausa.

Ritorna un puntatore all'array generato.

4.4.1.25 generate_instr_fonts()

```
ALLEGRO_FONT** generate_instr_fonts ( )
```

Funzione che genera l'array contenente i puntatori ai font relativi alla schermata di pausa.

Ritorna un puntatore all'array generato.

4.4.1.26 generate_menu_bitmaps()

```
ALLEGRO_BITMAP** generate_menu_bitmaps ( )
```

Funzione che genera l'array contenente i puntatori ai bitmap relativi alla schermata del menu principale.

Ritorna un puntatore all'array generato.

4.4.1.27 generate_menu_fonts()

```
ALLEGRO_FONT** generate_menu_fonts ( )
```

Funzione che genera l'array contenente i puntatori ai font relativi ai menu.

Ritorna un puntatore all'array generato.

4.4.1.28 generate_playwa_bitmaps()

```
ALLEGRO_BITMAP** generate_playwa_bitmaps ( )
```

Funzione che genera l'array contenente i puntatori ai bitmap relativi alla schermata di gioco.

Ritorna un puntatore all'array generato.

4.4.1.29 generate_playwa_fonts()

```
ALLEGRO_FONT** generate_playwa_fonts ( )
```

Funzione che genera l'array contenente i puntatori ai font relativi alla schermata di gioco.

Ritorna un puntatore all'array generato.

4.4.1.30 generate_settings_fonts()

```
ALLEGRO_FONT** generate_settings_fonts ( )
```

Funzione che genera l'array contenente i puntatori ai font relativi alla schermata delle impostazioni.

Ritorna un puntatore all'array generato.

4.4.1.31 generate_user_fonts()

```
ALLEGRO_FONT** generate_user_fonts ( )
```

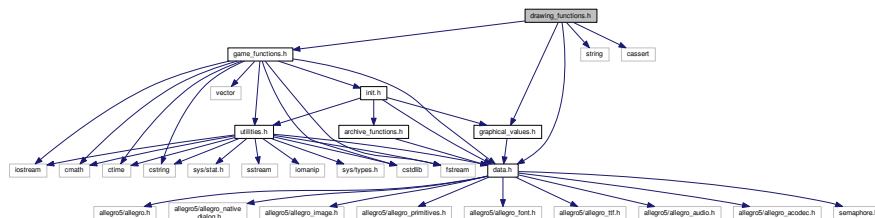
Funzione che genera l'array contenente i puntatori ai font relativi alla schermata di gameover.

Ritorna un puntatore all'array generato.

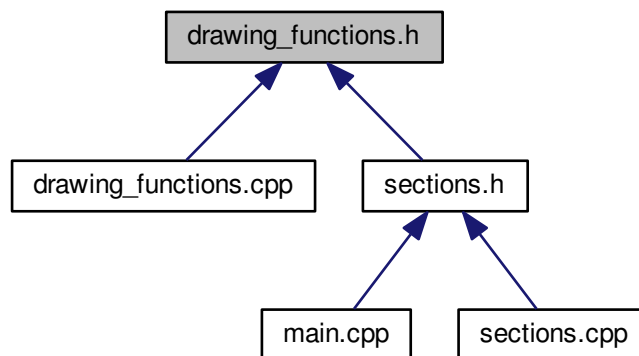
4.5 Riferimenti per il file drawing_functions.h

```
#include "data.h"
#include "graphical_values.h"
#include "game_functions.h"
#include <string>
#include <cassert>
```

Grafo delle dipendenze di inclusione per drawing_functions.h:



Questo grafo mostra quali altri file includono direttamente o indirettamente questo file:



Funzioni

- ALLEGRO_BITMAP ** [generate_playwa_bitmaps](#) ()
Funzione che genera l'array contenente i puntatori ai bitmap relativi alla schermata di gioco.
- ALLEGRO_FONT ** [generate_playwa_fonts](#) ()
Funzione che genera l'array contenente i puntatori ai font relativi alla schermata di gioco.
- ALLEGRO_BITMAP ** [generate_menu_bitmaps](#) ()
Funzione che genera l'array contenente i puntatori ai bitmap relativi alla schermata del menu principale.
- ALLEGRO_BITMAP ** [generate_instr_bitmaps](#) ()
Funzione che genera l'array contenente i puntatori ai bitmap relativi alla schermata di pausa.
- ALLEGRO_FONT ** [generate_menu_fonts](#) ()
Funzione che genera l'array contenente i puntatori ai font relativi ai menu.

- `ALLEGRO_FONT ** generate_instr_fonts ()`
Funzione che genera l'array contenente i puntatori ai font relativi alla schermata di pausa.
- `ALLEGRO_FONT ** generate_user_fonts ()`
Funzione che genera l'array contenente i puntatori ai font relativi alla schermata di gameover.
- `ALLEGRO_FONT ** generate_settings_fonts ()`
Funzione che genera l'array contenente i puntatori ai font relativi alla schermata delle impostazioni.
- `void destroy_arr_bitmaps (ALLEGRO_BITMAP **arr, const int num)`
Funzione che dealloca un array di bitmap, oltre che tutti i bitmap in esso contenuti.
- `void destroy_arr_fonts (ALLEGRO_FONT **arr, const int num)`
Funzione che dealloca un array di font, oltre che tutti i font in esso contenuti.
- `void draw_background (ALLEGRO_BITMAP *bmp_bg, float &offset_bg, const bool playing)`
Funzione che stampa lo sfondo della schermata.
- `void draw_head (char *head_str, ALLEGRO_FONT *font)`
Funzione che stampa il titolo della schermata.
- `void draw_back (ALLEGRO_FONT *font, const bool pause)`
Funzione che stampa la stringa di ritorno al gioco, o al menu principale (in base alla variabile "pause").
- `void draw_intro_init (ALLEGRO_BITMAP **arr_bitmaps, float &bg_offset)`
Funzione che gestisce l'inizializzazione dell'animazione di introduzione.
- `void draw_intro_title_slide (ALLEGRO_BITMAP **arr_bitmaps, float &bg_offset)`
Funzione che gestisce lo slide del titolo nell'animazione di introduzione.
- `void draw_intro_menu_fadein (ALLEGRO_BITMAP **arr_bitmaps, ALLEGRO_FONT **arr_fonts, float &bg_offset)`
Funzione che gestisce il fadein del menu nell'animazione di introduzione.
- `void draw_menu_title (ALLEGRO_BITMAP *bmp_title)`
Funzione che stampa il bitmap relativo al titolo presente nel menu principale.
- `void draw_menu_option (const int &index, ALLEGRO_BITMAP *bmp_option, ALLEGRO_FONT *font, char *str_scelta, const bool settings)`
Funzione che stampa l'opzione indicata (menu).
- `void draw_menu_copyright (ALLEGRO_FONT *font)`
Funzione che stampa la stringa di copyright (menu).
- `void draw_playwa_score_hint (const match_vars_t &match_vars, ALLEGRO_FONT *font_score, ALLEGRO_FONT *font_hints, const bool visible_hints)`
Funzione che stampa il punteggio e i suggerimenti (schermata di gioco).
- `void draw_playwa_shooter (const match_vars_t &match_vars, const bullet_t *arr_bullet, const shooter_t &shooter, ALLEGRO_BITMAP **arr_bitmaps)`
Funzione che stampa lo shooter, la sua area (bianca normalmente, rossa in caso di pericolo o azzurra nel caso in cui il bonus "shield" venga attivato).
- `void draw_playwa_asteroids_bullets (const match_vars_t &match_vars, const bullet_t *arr_bullet, ALLEGRO_BITMAP **arr_bitmaps, ALLEGRO_FONT *font)`
Funzione che stampa gli asteroidi ed i proiettili presenti in gioco.
- `void draw_playwa_bonus_music (const match_vars_t &match_vars, ALLEGRO_BITMAP **arr_bitmaps, ALLEGRO_FONT *font_score, ALLEGRO_FONT *font_bonus, ALLEGRO_FONT *font_bonus_hint, settings_t settings)`
Funzione che stampa i bonus della partita (e le icone relative al suono).
- `void draw_gameover_animation (const int &score_goal, const int &dast_goal, ALLEGRO_BITMAP *bg_bmp, float &bg_offset, ALLEGRO_FONT **arr_fonts)`
Funzione che gestisce l'animazione di fine partita.
- `void draw_results (ALLEGRO_FONT *font, result_t res)`
Funzione che disegna i risultati estratti dagli archivi corrispondenti alla password inserita dell'utente.
- `void draw_playagain_button (ALLEGRO_FONT *font)`
Funzione che disegna il bottone "play again" nella schermata di Game Over.
- `void draw_stats_blocks (ALLEGRO_FONT *font)`

Funzione che disegna i blocchi di sfondo relativi ai risultati di fine partita.

- void [draw_stat](#) (const int &idx_stat, int stat_value, ALLEGRO_FONT *font)

Funzione che disegna un singolo risultato.

- void [draw_username_block](#) (const bool enter_active, ALLEGRO_FONT *font)

Funzione che disegna il blocco di inserimento dello username.

- void [draw_username](#) (char *username, const int &cur_pos, ALLEGRO_FONT *font, const bool bar_visible)

Funzione che disegna lo username.

- void [draw_instructions](#) (ALLEGRO_FONT **arr_fonts, ALLEGRO_BITMAP **arr_bitmaps)

Funzione che disegna le istruzioni durante la pausa.

Variabili

- [display_info_t](#) [display_info](#)
- [general_gv_t](#) [GENERAL_GV](#)

Struct contenente le proporzioni grafiche generali.

- [menu_gv_t](#) [MENU_GV](#)

Struct contenente le proporzioni grafiche dei menu.

- [playwa_gv_t](#) [PLAYWA_GV](#)

Struct contenente le proporzioni grafiche della schermata di gioco.

- [instr_gv_t](#) [INSTR_GV](#)

Struct contenente le proporzioni grafiche della schermata di pausa.

- [usr_gv_t](#) [USR_GV](#)

Struct contenente le proporzioni grafiche della schermata di fine partita.

4.5.1 Documentazione delle funzioni

4.5.1.1 [destroy_arr_bitmaps\(\)](#)

```
void destroy_arr_bitmaps (
    ALLEGRO_BITMAP ** arr,
    const int num )
```

Funzione che dealloca un array di bitmap, oltre che tutti i bitmap in esso contenuti.

Parametri:

- 1) arr - Array di bitmap;
- 2) num - Dimensione dell'array di bitmap.

4.5.1.2 [destroy_arr_fonts\(\)](#)

```
void destroy_arr_fonts (
    ALLEGRO_FONT ** arr,
    const int num )
```

Funzione che dealloca un array di font, oltre che tutti i font in esso contenuti.

Parametri:

- 1) arr - Array di font;
- 2) num - Dimensione dell'array di font.

4.5.1.3 `draw_back()`

```
void draw_back (
    ALLEGRO_FONT * font,
    const bool pause )
```

Funzione che stampa la stringa di ritorno al gioco, o al menu principale (in base alla variabile "pause").

Parametri:

- 1) font - Font relativo alla stringa di ritorno;
- 2) pause - In base al valore assunto da questa variabile viene stampata la stringa adatta.

4.5.1.4 `draw_background()`

```
void draw_background (
    ALLEGRO_BITMAP * bmp_bg,
    float & offset_bg,
    const bool playing )
```

Funzione che stampa lo sfondo della schermata.

Parametri:

- 1) bmp_bg - Bitmap relativo allo sfondo;
- 2) offset_bg - Scostamento in pixel dello sfondo dal lato sinistro della schermata (per rotazione sfondo);
- 3) playing - Flag sul quale si basa la velocità di rotazione (durante il gioco risulta essere più lenta).

4.5.1.5 `draw_gameover_animation()`

```
void draw_gameover_animation (
    const int & score_goal,
    const int & dast_goal,
    ALLEGRO_BITMAP * bg_bmp,
    float & bg_offset,
    ALLEGRO_FONT ** arr_fonts )
```

Funzione che gestisce l'animazione di fine partita.

Parametri:

- 1) score_goal - Score effettiva del giocatore;
- 2) dast_goal - Asteroidi effettivamente distrutti;
- 4) bg_bmp - Bitmap di sfondo;
- 5) bg_offset - Scostamento laterale dello sfondo;
- 6) arr_fonts - Array dei font necessari.

4.5.1.6 `draw_head()`

```
void draw_head (
    char * head_str,
    ALLEGRO_FONT * font )
```

Funzione che stampa il titolo della schermata.

Parametri:

- 1) head_str - Stringa relativa al titolo;
- 2) font - Font relativo al titolo.

4.5.1.7 draw_instructions()

```
void draw_instructions (
    ALLEGRO_FONT ** arr_fonts,
    ALLEGRO_BITMAP ** arr_bitmaps )
```

Funzione che disegna le istruzioni durante la pausa.

Parametri:

- 1) arr_fonts - Array contenente i font utilizzati;
- 2) arr_bitmaps - Array contenente i bitmap utilizzati.

4.5.1.8 draw_intro_init()

```
void draw_intro_init (
    ALLEGRO_BITMAP ** arr_bitmaps,
    float & bg_offset )
```

Funzione che gestisce l'inizializzazione dell'animazione di introduzione.

Parametri:

- 1) arr_bitmaps - Array di bitmap;
- 2) bg_offset - Offset iniziale dello sfondo.

4.5.1.9 draw_intro_menu_fadein()

```
void draw_intro_menu_fadein (
    ALLEGRO_BITMAP ** arr_bitmaps,
    ALLEGRO_FONT ** arr_fonts,
    float & bg_offset )
```

Funzione che gestisce il fadein del menu nell'animazione di introduzione.

Parametri:

- 1) arr_bitmaps - Array di bitmap;
- 2) arr_fonts - Array di font;
- 3) bg_offset - Offset iniziale dello sfondo.

4.5.1.10 draw_intro_title_slide()

```
void draw_intro_title_slide (
    ALLEGRO_BITMAP ** arr_bitmaps,
    float & bg_offset )
```

Funzione che gestisce lo slide del titolo nell'animazione di introduzione.

Parametri:

- 1) arr_bitmaps - Array di bitmap;
- 2) bg_offset - Offset iniziale dello sfondo.

4.5.1.11 `draw_menu_copyright()`

```
void draw_menu_copyright (
    ALLEGRO_FONT * font )
```

Funzione che stampa la stringa di copyright (menu).

Parametri:

- 1) font - Font relativo alla stringa di copyright.

4.5.1.12 `draw_menu_option()`

```
void draw_menu_option (
    const int & index,
    ALLEGRO_BITMAP * bmp_option,
    ALLEGRO_FONT * font,
    char * str_scelta,
    const bool settings )
```

Funzione che stampa l'opzione indicata (menu).

Parametri:

- 1) index - Indice dell'opzione;
- 2) bmp_option - Bitmap relativo all'opzione indicata;
- 3) font - Font relativo all'opzione indicata;
- 4) str_scelta - Stringa relativa all'opzione indicata;
- 5) settings - Se vero, stampa le opzioni leggermente più in alto rispetto al menu principale.

4.5.1.13 `draw_menu_title()`

```
void draw_menu_title (
    ALLEGRO_BITMAP * bmp_title )
```

Funzione che stampa il bitmap relativo al titolo presente nel menu principale.

Parametri:

- 1) bmp_title - Bitmap relativo al titolo.

4.5.1.14 `draw_playagain_button()`

```
void draw_playagain_button (
    ALLEGRO_FONT * font )
```

Funzione che disegna il bottone "play again" nella schermata di Game Over.

Parametri:

- 1) arr_fonts - Array dei font necessari;

4.5.1.15 draw_playwa_asteroids_bullets()

```
void draw_playwa_asteroids_bullets (
    const match_vars_t & match_vars,
    const bullet_t * arr_bullet,
    ALLEGRO_BITMAP ** arr_bitmaps,
    ALLEGRO_FONT * font )
```

Funzione che stampa lo gli asteroidi ed i proiettili presenti in gioco.

Parametri:

- 1) match_vars - Struct contenente le variabili di gioco;
- 2) arr_bullet - Array contenente i proiettili in gioco;
- 3) arr_bitmaps - Array contenente i bitmap di gioco;
- 4) font - Font relativo agli asteroidi.

4.5.1.16 draw_playwa_bonus_music()

```
void draw_playwa_bonus_music (
    const match_vars_t & match_vars,
    ALLEGRO_BITMAP ** arr_bitmaps,
    ALLEGRO_FONT * font_score,
    ALLEGRO_FONT * font_bonus,
    ALLEGRO_FONT * font_bonus_hint,
    settings_t settings )
```

Funzione che stampa i bonus della partita (e le icone relative al suono).

Parametri:

- 1) match_vars - Struct contenente le variabili di gioco;
- 2) arr_bitmaps - Array contenente i bitmap di gioco;
- 3) font_score - Font relativo al punteggio (utilizzato per stampare l'intestazione dei bonus);
- 4) font_bonus - Font relativo alla numerosità dei bonus;
- 5) font_bonus_hint - Font relativo al suggerimento per l'attivazione del bonus.

4.5.1.17 draw_playwa_score_hint()

```
void draw_playwa_score_hint (
    const match_vars_t & match_vars,
    ALLEGRO_FONT * font_score,
    ALLEGRO_FONT * font_hints,
    const bool visible_hints )
```

Funzione che stampa il punteggio e i suggerimenti (schermata di gioco).

Parametri:

- 1) match_vars - Struct contenente le variabili di gioco;
- 2) font_score - Font relativo al punteggio;
- 3) font_hints - Font relativo ai suggerimenti;
- 4) visible_hints - Se true, mostra i suggerimenti di gioco.

4.5.1.18 `draw_playwa_shooter()`

```
void draw_playwa_shooter (
    const match_vars_t & match_vars,
    const bullet_t * arr_bullet,
    const shooter_t & shooter,
    ALLEGRO_BITMAP ** arr_bitmaps )
```

Funzione che stampa lo shooter, la sua area (bianca normalmente, rossa in caso di pericolo o azzurra nel caso in cui il bonus "shield" venga attivato).

Parametri:

- 1) `match_vars` - Struct contenente le variabili di gioco;
- 2) `arr_bullet` - Array contenente i proiettili in gioco;
- 3) `shooter` - Struct contenente le variabili relative allo shooter;
- 4) `arr_bitmaps` - Array contenente i bitmap di gioco.

4.5.1.19 `draw_results()`

```
void draw_results (
    ALLEGRO_FONT * font,
    result_t res )
```

Funzione che disegna i risultati estratti dagli archivi corrispondenti alla password inserita dell'utente.

Parametri:

- 1) `arr_fonts` - Array dei font necessari;
- 2) `res` - Risultati dei match estratti;

4.5.1.20 `draw_stat()`

```
void draw_stat (
    const int & idx_stat,
    int stat_value,
    ALLEGRO_FONT * font )
```

Funzione che disegna un singolo risultato.

Parametri:

- 1) `idx_stat` - Indice del risultato;
- 2) `stat_value` - Valore del risultato;
- 3) `font` - Font utilizzato.

4.5.1.21 `draw_stats_blocks()`

```
void draw_stats_blocks (
    ALLEGRO_FONT * font )
```

Funzione che disegna i blocchi di sfondo relativi ai risultati di fine partita.

Parametri:

- 1) `font` - Font utilizzato.

4.5.1.22 draw_username()

```
void draw_username (
    char * username,
    const int & cur_pos,
    ALLEGRO_FONT * font,
    const bool bar_visible )
```

Funzione che disegna lo username.

- 1) username - Stringa dello username;
- 2) cur_pos - Posizione corrente all'interno della stringa;
- 3) font - Font utilizzato;
- 4) bar_visible - Se vero, il cursore è visibile.

4.5.1.23 draw_username_block()

```
void draw_username_block (
    const bool enter_active,
    ALLEGRO_FONT * font )
```

Funzione che disegna il blocco di inserimento dello username.

Parametri:

- 1) enter_active - Se true, è possibile cliccare il tasto di invio dello username;
- 2) font - Font utilizzato.

4.5.1.24 generate_instr_bitmaps()

```
ALLEGRO_BITMAP** generate_instr_bitmaps ( )
```

Funzione che genera l'array contenente i puntatori ai bitmap relativi alla schermata di pausa.

Ritorna un puntatore all'array generato.

4.5.1.25 generate_instr_fonts()

```
ALLEGRO_FONT** generate_instr_fonts ( )
```

Funzione che genera l'array contenente i puntatori ai font relativi alla schermata di pausa.

Ritorna un puntatore all'array generato.

4.5.1.26 generate_menu_bitmaps()

```
ALLEGRO_BITMAP** generate_menu_bitmaps ( )
```

Funzione che genera l'array contenente i puntatori ai bitmap relativi alla schermata del menu principale.

Ritorna un puntatore all'array generato.

4.5.1.27 `generate_menu_fonts()`

```
ALLEGRO_FONT** generate_menu_fonts ( )
```

Funzione che genera l'array contenente i puntatori ai font relativi ai menu.

Ritorna un puntatore all'array generato.

4.5.1.28 `generate_playwa_bitmaps()`

```
ALLEGRO_BITMAP** generate_playwa_bitmaps ( )
```

Funzione che genera l'array contenente i puntatori ai bitmap relativi alla schermata di gioco.

Ritorna un puntatore all'array generato.

4.5.1.29 `generate_playwa_fonts()`

```
ALLEGRO_FONT** generate_playwa_fonts ( )
```

Funzione che genera l'array contenente i puntatori ai font relativi alla schermata di gioco.

Ritorna un puntatore all'array generato.

4.5.1.30 `generate_settings_fonts()`

```
ALLEGRO_FONT** generate_settings_fonts ( )
```

Funzione che genera l'array contenente i puntatori ai font relativi alla schermata delle impostazioni.

Ritorna un puntatore all'array generato.

4.5.1.31 `generate_user_fonts()`

```
ALLEGRO_FONT** generate_user_fonts ( )
```

Funzione che genera l'array contenente i puntatori ai font relativi alla schermata di gameover.

Ritorna un puntatore all'array generato.

4.5.2 Documentazione delle variabili

4.5.2.1 display_info

`display_info_t` display_info

4.5.2.2 GENERAL_GV

`general_gv_t` GENERAL_GV

Struct contenente le proporzioni grafiche generali.

4.5.2.3 INSTR_GV

`instr_gv_t` INSTR_GV

Struct contenente le proporzioni grafiche della schermata di pausa.

4.5.2.4 MENU_GV

`menu_gv_t` MENU_GV

Struct contenente le proporzioni grafiche dei menu.

4.5.2.5 PLAYWA_GV

`playwa_gv_t` PLAYWA_GV

Struct contenente le proporzioni grafiche della schermata di gioco.

4.5.2.6 USR_GV

`usr_gv_t` USR_GV

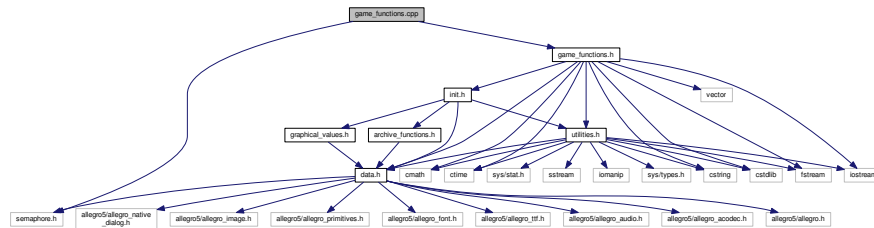
Struct contenente le proporzioni grafiche della schermata di fine partita.

4.6 Riferimenti per il file game_functions.cpp

```
#include <semaphore.h>
```

```
#include "game_functions.h"
```

Grafo delle dipendenze di inclusione per game_functions.cpp:



Funzioni

- void * **spawn_asteroid_thread** (void *mv)
Funzione che implementa il consumer thread.
- void **sound_play** (ALLEGRO_SAMPLE *sound, const **enable_t** &en_sound)
Funzione che fa partire il rispettivo suono del gioco, se i suoni sono attivati.
- void **music_wa_play** ()
Funzione che fa partire la musica del gioco.
- **asteroid_t generate_asteroid** (const **asteroid_special_t** s, const int &next_p)
Funzione che genera un singolo asteroide (di tipo compreso tra 3 e lev) in un punto casuale del bordo della zona di gioco.
- void **move_bullets** (**bullet_t** *bullet, **match_vars_t** &match_vars, **enable_t** &en_sound)
Funzione che muove i bullets, e quando c'è una collisione viene definitivamente eliminato l'asteroide dalla lista, e aggiornati i bonus corrispondenti agli asteroidi distrutti.
- void **move_asteroids** (**asteroid_list** *ast_queues, const int &extract_index)
Funzione che muove l'asteroide in relazione al suo angolo con lo shooter, in modo che segua la retta che congiunge shooter - asteroide.
- void **move_shooter** (**shooter_t** &shooter, const **asteroid_list** *ast_queues, const int current_asteroid)
Funzione che ruota lo shooter in base all'asteroide puntato.
- bool **check_gameover** (**asteroid_list** *ast_queues, const **shooter_t** &shooter, const **bullet_t** *arr_bullet)
Funzione che, in base alla posizione degli asteroidi in gioco e dello shooter, controlla se ci siano collisioni, ovvero se lo shooter e un asteroide in gioco si siano toccati, e il gioco in questo caso finisce.
- float **check_distance** (const **asteroid_list** *ast_queues, const **bullet_t** *arr_bullet, const **shooter_t** &shooter)
Funzione che restituisce la distanza tra il meteorite più vicino all'asteroide e l'asteroide stesso.
- void **check_shield** (**match_vars_t** &match_vars, **bullet_t** *arr_bullet, **shooter_t** &shooter, **enable_t** &en_sound)
Funzione che controlla lo stato dello scudo.
- void **handler_timer** (ALLEGRO_EVENT ev, **match_vars_t** &match_vars, **settings_t** &settings)
Funzione di routine che gestisce l'evento della scadenza di un timer.
- void **handler_key_pressed** (ALLEGRO_EVENT ev, **match_vars_t** &match_vars, **bullet_t** *arr_bullet, **shooter_t** &shooter, **settings_t** &settings)
Funzione di routine che gestisce l'evento della scadenza di un timer.
- void **change_music** (const char *path, const **enable_t** &music)
Funzione che mette cambia la musica tra le varie schermate.

4.6.1 Documentazione delle funzioni

4.6.1.1 change_music()

```
void change_music (
    const char * path,
    const enable_t & music )
```

Funzione che mette cambia la musica tra le varie schermate.

Parametri:

1) path - Percorso del nuovo audio.

4.6.1.2 check_distance()

```
float check_distance (
    const asteroid_list * ast_queues,
    const bullet_t * arr_bullet,
    const shooter_t & shooter )
```

Funzione che restituisce la distanza tra il meteorite più vicino all'asteroide e l'asteroide stesso.

Parametri:

1) ast_queues - Vettore di liste di asteroidi;

2) arr_bullet - Vettore dei proiettili

3) shooter - Informazioni relative allo shooter.

4.6.1.3 check_gameover()

```
bool check_gameover (
    asteroid_list * ast_queues,
    const shooter_t & shooter,
    const bullet_t * arr_bullet )
```

Funzione che, in base alla posizione degli asteroidi in gioco e dello shooter, controlla se ci siano collisioni, ovvero se lo shooter e un asteroide in gioco si siano toccati, e il gioco in questo caso finisce.

Parametri:

1) ast_queues - Vettore di liste di asteroidi;

2) shooter - Informazioni relative allo shooter;

3) arr_bullet - Vettore dei proiettili.

4.6.1.4 `check_shield()`

```
void check_shield (
    match_vars_t & match_vars,
    bullet_t * arr_bullet,
    shooter_t & shooter,
    enable_t & en_sound )
```

Funzione che controlla lo stato dello scudo.

Se c'è gameover ma lo scudo è attivo grazie al bonus 'shield', l'asteroide che colpisce lo scudo viene distrutto.

Parametri:

- 2) `match_vars` - Contenitore delle variabili di gioco;
- 2) `arr_bullet` - Vettore dei proiettili;
- 3) `shooter` - Informazioni relative allo shooter.

4.6.1.5 `generate_asteroid()`

```
asteroid_t generate_asteroid (
    const asteroid_special_t s,
    const int & next_p )
```

Funzione che genera un singolo asterioide (di tipo compreso tra 3 e lev) in un punto casuale del bordo della zona di gioco.

Parametri:

- 1) `lev` - Livello attuale;
- 2) `v` - lingua della parole dell'asteroide
- 3) `s` - Caratteristiche speciali dell'asteroide
- 4) `next_p` - In caso di conflitto, viene utilizzata come posizione in cui inserire l'asteroide

4.6.1.6 `handler_key_pressed()`

```
void handler_key_pressed (
    ALLEGRO_EVENT ev,
    match_vars_t & match_vars,
    bullet_t * arr_bullet,
    shooter_t & shooter,
    settings_t & settings )
```

Funzione di routine che gestisce l'evento della scadenza di un timer.

In base al timer scaduto (spawn, shield, rallenty) genera l'evento associato.

Parametri:

- 1) `ev` - Evento generato;
- 2) `match_vars` - Contenitore delle variabili di gioco;
- 3) `arr_bullet` - Vettore dei proiettili;
- 4) `shooter` - Informazioni relative allo shooter;
- 5) `settings` - Impostazioni di gioco.

4.6.1.7 handler_timer()

```
void handler_timer (
    ALLEGRO_EVENT ev,
    match_vars_t & match_vars,
    settings_t & settings )
```

Funzione di routine che gestisce l'evento della scadenza di un timer.

In base al timer scaduto (spawn, shield, rallenty) genera l'evento associato.

Parametri:

- 1) ev - Evento generato;
- 2) match_vars - Contenitore dei tre timer;
- 3) settings - Impostazioni di gioco.

4.6.1.8 move_asteroids()

```
void move_asteroids (
    asteroid_list * ast_queues,
    const int & extract_index )
```

Funzione che muove l'asteroide in relazione al suo angolo con lo shooter, in modo che segua la retta che congiunge shooter - asteroide.

Parametri:

- 1) ast_queues - Vettore di liste di asteroidi;
- 2) extract_index - Indice dell'asteroide in campo ma da non far comparire.

4.6.1.9 move_bullets()

```
void move_bullets (
    bullet_t * bullet,
    match_vars_t & match_vars,
    enable_t & en_sound )
```

Funzione che muove i bullets, e quando c'è una collisione viene definitivamente eliminato l'asteroide dalla lista, e aggiornati i bonus corrispondenti agli asteroidi distrutti.

Parametri:

- 1) bullet - Vettore dei bullet;
- 2) match_vars - Contenitore delle variabili di gioco;
- 3) items - Vettore dei bonus.

4.6.1.10 move_shooter()

```
void move_shooter (
    shooter_t & shooter,
    const asteroid_list * ast_queues,
    const int current_asteroid )
```

Funzione che ruota lo shooter in base all'asteroide puntato.

Parametri:

- 1) shooter - Informazioni relative allo shooter;
- 2) ast_queues - Vettore di liste di asteroidi;
- 3) current_asteroid - Carattere corrispondente all'asteroide puntato.

4.6.1.11 music_wa_play()

```
void music_wa_play ( )
```

Funzione che fa partire la musica del gioco.

4.6.1.12 sound_play()

```
void sound_play (
    ALLEGRO_SAMPLE * sound,
    const enable_t & en_sound )
```

Funzione che fa partire il rispettivo suono del gioco, se i suoni sono attivati.

Parametri:

- 1) sound - Suono da far partire;
- 2) en_sound - Impostazioni che indica se i suoni sono ON/OFF.

4.6.1.13 spawn_asteroid_thread()

```
void* spawn_asteroid_thread (
    void * mv )
```

Funzione che implementa il consumer thread.

Ad intervalli regolari viene estratta una parola ed inserita all'interno del gioco.

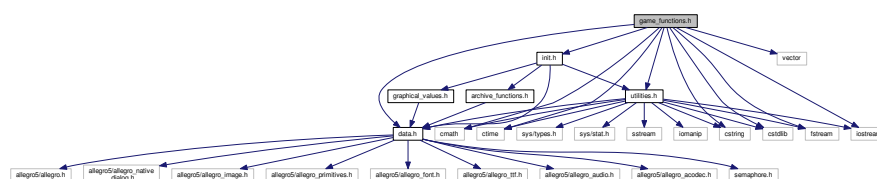
Parametri:

- 1) mv - Puntatore alle match vars, le variabili di gioco;

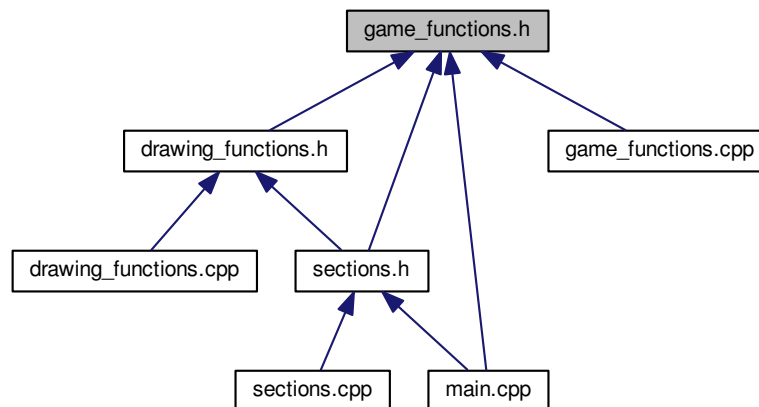
4.7 Riferimenti per il file game_functions.h

```
#include "data.h"
#include "utilities.h"
#include "init.h"
#include <iostream>
#include <vector>
#include <fstream>
#include <cmath>
#include <ctime>
#include <cstring>
#include <cstdlib>
```

Grafo delle dipendenze di inclusione per game_functions.h:



Questo grafo mostra quali altri file includono direttamente o indirettamente questo file:



Funzioni

- void `sound_play` (ALLEGRO_SAMPLE *sound, const `enable_t` &en_sound)
Funzione che fa partire il rispettivo suono del gioco, se i suoni sono attivati.
- void `music_wa_play` ()
Funzione che fa partire la musica del gioco.
- `asteroid_t` `generate_asteroid` (const `asteroid_special_t` s, const int &next_p)
Funzione che genera un singolo asteroide (di tipo compreso tra 3 e lev) in un punto casuale del bordo della zona di gioco.
- void `move_bullets` (`bullet_t` *bullet, `match_vars_t` &match_vars, `enable_t` &en_sound)
Funzione che muove i bullets, e quando c'è una collisione viene definitivamente eliminato l'asteroide dalla lista, e aggiornati i bonus corrispondenti agli asteroidi distrutti.
- void `move_asteroids` (`asteroid_list` *ast_queues, const int &extract_index)
Funzione che muove l'asteroide in relazione al suo angolo con lo shooter, in modo che segua la retta che congiunge shooter - asteroide.
- void `move_shooter` (`shooter_t` &shooter, const `asteroid_list` *ast_queues, const int current_asteroid)
Funzione che ruota lo shooter in base all'asteroide puntato.
- bool `check_gameover` (`asteroid_list` *ast_queues, const `shooter_t` &shooter, const `bullet_t` *arr_bullet)
Funzione che, in base alla posizione degli asteroidi in gioco e dello shooter, controlla se ci siano collisioni, ovvero se lo shooter e un asteroide in gioco si siano toccati, e il gioco in questo caso finisce.
- float `check_distance` (const `asteroid_list` *ast_queues, const `bullet_t` *arr_bullet, const `shooter_t` &shooter)
Funzione che restituisce la distanza tra il meteorite più vicino all'asteroide e l'asteroide stesso.
- void `check_shield` (`match_vars_t` &match_vars, `bullet_t` *arr_bullet, `shooter_t` &shooter, `enable_t` &en_sound)
Funzione che controlla lo stato dello scudo.
- void `handler_timer` (ALLEGRO_EVENT ev, `match_vars_t` &match_vars, `settings_t` &settings)
Funzione di routine che gestisce l'evento della scadenza di un timer.
- void `handler_key_pressed` (ALLEGRO_EVENT ev, `match_vars_t` &match_vars, `bullet_t` *arr_bullet, `shooter_t` &shooter, `settings_t` &settings)
Funzione di routine che gestisce l'evento della scadenza di un timer.
- void `change_music` (const char *path, const `enable_t` &music)
Funzione che mette cambia la musica tra le varie schermate.
- void * `spawn_asteroid_thread` (void *mv)
Funzione che implementa il consumer thread.

Variabili

- `display_info_t` `display_info`
- `ALLEGRO_SAMPLE` * `music_wa`
- `playwa_gv_t` `PLAYWA_GV`

Struct contenente le proporzioni grafiche della schermata di gioco.

- `thread_manager_t` `thread_manager`
- `words_buffer_t` `words_buffer`

4.7.1 Documentazione delle funzioni

4.7.1.1 `change_music()`

```
void change_music (
    const char * path,
    const enable_t & music )
```

Funzione che mette cambia la musica tra le varie schermate.

Parametri:

- 1) path - Percorso del nuovo audio.

4.7.1.2 `check_distance()`

```
float check_distance (
    const asteroid_list * ast_queues,
    const bullet_t * arr_bullet,
    const shooter_t & shooter )
```

Funzione che restituisce la distanza tra il meteorite più vicino all'asteroide e l'asteroide stesso.

Parametri:

- 1) `ast_queues` - Vettore di liste di asteroidi;
- 2) `arr_bullet` - Vettore dei proiettili
- 3) `shooter` - Informazioni relative allo shooter.

4.7.1.3 `check_gameover()`

```
bool check_gameover (
    asteroid_list * ast_queues,
    const shooter_t & shooter,
    const bullet_t * arr_bullet )
```

Funzione che, in base alla posizione degli asteroidi in gioco e dello shooter, controlla se ci siano collisioni, ovvero se lo shooter e un asteroide in gioco si siano toccati, e il gioco in questo caso finisce.

Parametri:

- 1) `ast_queues` - Vettore di liste di asteroidi;
- 2) `shooter` - Informazioni relative allo shooter;
- 3) `arr_bullet` - Vettore dei proiettili.

4.7.1.4 check_shield()

```
void check_shield (
    match_vars_t & match_vars,
    bullet_t * arr_bullet,
    shooter_t & shooter,
    enable_t & en_sound )
```

Funzione che controlla lo stato dello scudo.

Se c'è gameover ma lo scudo è attivo grazie al bonus 'shield', l'asteroide che colpisce lo scudo viene distrutto.

Parametri:

- 2) match_vars - Contenitore delle variabili di gioco;
- 2) arr_bullet - Vettore dei proiettili;
- 3) shooter - Informazioni relative allo shooter.

4.7.1.5 generate_asteroid()

```
asteroid_t generate_asteroid (
    const asteroid_special_t s,
    const int & next_p )
```

Funzione che genera un singolo asteroide (di tipo compreso tra 3 e lev) in un punto casuale del bordo della zona di gioco.

Parametri:

- 1) lev - Livello attuale;
- 2) v - lingua della parole dell'asteroide
- 3) s - Caratteristiche speciali dell'asteroide
- 4) next_p - In caso di conflitto, viene utilizzata come posizione in cui inserire l'asteroide

4.7.1.6 handler_key_pressed()

```
void handler_key_pressed (
    ALLEGRO_EVENT ev,
    match_vars_t & match_vars,
    bullet_t * arr_bullet,
    shooter_t & shooter,
    settings_t & settings )
```

Funzione di routine che gestisce l'evento della scadenza di un timer.

In base al timer scaduto (spawn, shield, rallenty) genera l'evento associato.

Parametri:

- 1) ev - Evento generato;
- 2) match_vars - Contenitore delle variabili di gioco;
- 3) arr_bullet - Vettore dei proiettili;
- 4) shooter - Informazioni relative allo shooter;
- 5) settings - Impostazioni di gioco.

4.7.1.7 `handler_timer()`

```
void handler_timer (
    ALLEGRO_EVENT ev,
    match_vars_t & match_vars,
    settings_t & settings )
```

Funzione di routine che gestisce l'evento della scadenza di un timer.

In base al timer scaduto (spawn, shield, rallenty) genera l'evento associato.

Parametri:

- 1) ev - Evento generato;
- 2) match_vars - Contenitore dei tre timer;
- 3) settings - Impostazioni di gioco.

4.7.1.8 `move_asteroids()`

```
void move_asteroids (
    asteroid_list * ast_queues,
    const int & extract_index )
```

Funzione che muove l'asteroide in relazione al suo angolo con lo shooter, in modo che segua la retta che congiunge shooter - asteroide.

Parametri:

- 1) ast_queues - Vettore di liste di asteroidi;
- 2) extract_index - Indice dell'asteroide in campo ma da non far comparire.

4.7.1.9 `move_bullets()`

```
void move_bullets (
    bullet_t * bullet,
    match_vars_t & match_vars,
    enable_t & en_sound )
```

Funzione che muove i bullets, e quando c'è una collisione viene definitivamente eliminato l'asteroide dalla lista, e aggiornati i bonus corrispondenti agli asteroidi distrutti.

Parametri:

- 1) bullet - Vettore dei bullet;
- 2) match_vars - Contenitore delle variabili di gioco;
- 3) items - Vettore dei bonus.

4.7.1.10 `move_shooter()`

```
void move_shooter (
    shooter_t & shooter,
    const asteroid_list * ast_queues,
    const int current_asteroid )
```

Funzione che ruota lo shooter in base all'asteroide puntato.

Parametri:

- 1) shooter - Informazioni relative allo shooter;
- 2) ast_queues - Vettore di liste di asteroidi;
- 3) current_asteroid - Carattere corrispondente all'asteroide puntato.

4.7.1.11 music_wa_play()

```
void music_wa_play ( )
```

Funzione che fa partire la musica del gioco.

4.7.1.12 sound_play()

```
void sound_play (
    ALLEGRO_SAMPLE * sound,
    const enable_t & en_sound )
```

Funzione che fa partire il rispettivo suono del gioco, se i suoni sono attivati.

Parametri:

- 1) sound - Suono da far partire;
- 2) en_sound - Impostazioni che indica se i suoni sono ON/OFF.

4.7.1.13 spawn_asteroid_thread()

```
void* spawn_asteroid_thread (
    void * mv )
```

Funzione che implementa il consumer thread.

Ad intervalli regolari viene estratta una parola ed inserita all'interno del gioco.

Parametri:

- 1) mv - Puntatore alle match vars, le variabili di gioco;

4.7.2 Documentazione delle variabili

4.7.2.1 display_info

```
display_info_t display_info
```

4.7.2.2 music_wa

```
ALLEGRO_SAMPLE* music_wa
```

4.7.2.3 PLAYWA_GV

```
playwa_gv_t PLAYWA_GV
```

Struct contenente le proporzioni grafiche della schermata di gioco.

4.7.2.4 thread_manager

```
thread_manager_t thread_manager
```

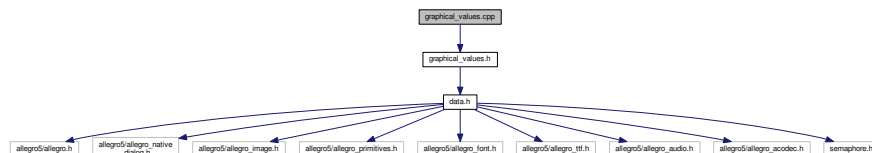
4.7.2.5 words_buffer

```
words_buffer_t words_buffer
```

4.8 Riferimenti per il file graphical_values.cpp

```
#include "graphical_values.h"
```

Grafo delle dipendenze di inclusione per graphical_values.cpp:



Funzioni

- void `update_graphical_values()`
Aggiorna le proporzioni grafiche dei singoli oggetti in proporzione all'attuale dimensione del display.

Variabili

- `general_gv_t GENERAL_GV`
Struct contenente le proporzioni grafiche generali.
- `menu_gv_t MENU_GV`
Struct contenente le proporzioni grafiche dei menu.
- `playwa_gv_t PLAYWA_GV`
Struct contenente le proporzioni grafiche della schermata di gioco.
- `instr_gv_t INSTR_GV`
Struct contenente le proporzioni grafiche della schermata di pausa.
- `usr_gv_t USR_GV`
Struct contenente le proporzioni grafiche della schermata di fine partita.

4.8.1 Documentazione delle funzioni

4.8.1.1 `update_graphical_values()`

```
void update_graphical_values ( )
```

Aggiorna le proporzioni grafiche dei singoli oggetti in proporzione all'attuale dimensione del display.

4.8.2 Documentazione delle variabili

4.8.2.1 `GENERAL_GV`

```
general_gv_t GENERAL_GV
```

Struct contenente le proporzioni grafiche generali.

4.8.2.2 `INSTR_GV`

```
instr_gv_t INSTR_GV
```

Struct contenente le proporzioni grafiche della schermata di pausa.

4.8.2.3 `MENU_GV`

```
menu_gv_t MENU_GV
```

Struct contenente le proporzioni grafiche dei menu.

4.8.2.4 `PLAYWA_GV`

```
playwa_gv_t PLAYWA_GV
```

Struct contenente le proporzioni grafiche della schermata di gioco.

4.8.2.5 USR_GV

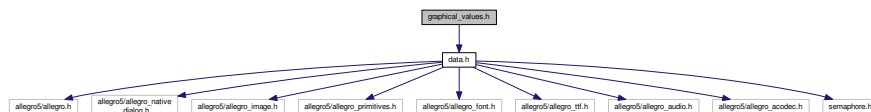
`usr_gv_t` USR_GV

Struct contenente le proporzioni grafiche della schermata di fine partita.

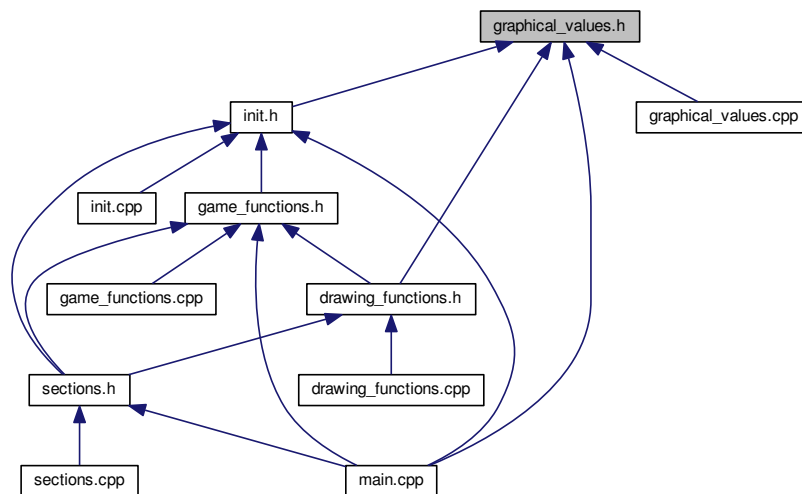
4.9 Riferimenti per il file graphical_values.h

```
#include "data.h"
```

Grafo delle dipendenze di inclusione per graphical_values.h:



Questo grafo mostra quali altri file includono direttamente o indirettamente questo file:



Funzioni

- void `update_graphical_values` ()

Aggiorna le proporzioni grafiche dei singoli oggetti in proporzione all'attuale dimensione del display.

Variabili

- `display_info_t` `display_info`

4.9.1 Documentazione delle funzioni

4.9.1.1 update_graphical_values()

```
void update_graphical_values ( )
```

Aggiorna le proporzioni grafiche dei singoli oggetti in proporzione all'attuale dimensione del display.

4.9.2 Documentazione delle variabili

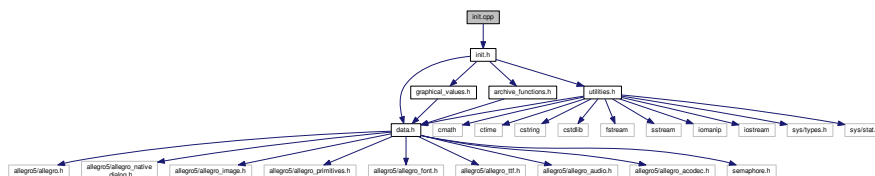
4.9.2.1 display_info

```
display_info_t display_info
```

4.10 Riferimenti per il file init.cpp

```
#include "init.h"
```

Grafo delle dipendenze di inclusione per init.cpp:



Funzioni

- void `sort_file_list` (`ls_t *file_list`)
Funzione che ordina un array di elementi che rappresentano gli archivi in ordine crescente in base alla loro dimensione.
- bool `init_game_requirements` ()
Funzione che inizializza le funzionalità relative all'audio, ai font (tff), alla tastiera.
- `display_info_t` `init_display_res` ()
Funzione che inizializza le dimensioni del display di gioco in base alle proporzioni standard (1280x800), adattandole al monitor del proprio computer.
- void `init_display` (ALLEGRO_DISPLAY *&display)
Funzione che crea il display ed inizializza il titolo e l'icona.
- ALLEGRO_EVENT_QUEUE * `init_queue_event` (ALLEGRO_DISPLAY *&display)
Funzione che crea il puntatore alla coda degli eventi e aggancia alla coda gli eventi relativi alla tastiera e display.
- int `init_target_bonus` (int &instant_ast_dest, const int &asteroids_destroyed)

Funzione che indica dopo quanti asteroidi distrutti ottengo il bonus.

- ALLEGRO_SAMPLE * [init_sound_music](#) (const char *path)

Funzione che inizializza un suono del gioco.

- void [init_match_vars](#) ([match_vars_t](#) &match_vars, ALLEGRO_EVENT_QUEUE *ev_queue)

Funzione che inizializza le variabili di gioco.

- void [init_settings](#) ([settings_t](#) &settings)

Funzione che inizializza le impostazioni di gioco.

- void [init_shooter](#) ([shooter_t](#) &shooter)

Funzione che inizializza le informazioni dello shooter a inizio gioco.

- void [init_bullet](#) ([bullet_t](#) &bullet)

Funzione che inizializza le informazioni di un proiettile.

- void [init_arr_bullet](#) ([bullet_t](#) *arr_bullet)

Funzione che inizializza un array di puntatori a asteroidi.

- void [init_words_buffer](#) ([words_buffer_t](#) &words_buffer)

Funzione che inizializza l'array utilizzato come buffer per le parole tra producer e consumer.

- void [init_thread_manager](#) ([thread_manager_t](#) &thread_manager, char *email)

Funzione che inizializza la struttura dati del thread manager.

4.10.1 Documentazione delle funzioni

4.10.1.1 [init_arr_bullet\(\)](#)

```
void init_arr_bullet (
    bullet\_t * arr_bullet )
```

Funzione che inizializza un array di puntatori a asteroidi.

Parametri:

- 1) arr_bullet - Array dei bullet.

4.10.1.2 [init_bullet\(\)](#)

```
void init_bullet (
    bullet\_t & bullet )
```

Funzione che inizializza le informazioni di un proiettile.

Parametri:

- 1) shooter - Struct relativo al singolo proiettile.

4.10.1.3 init_display()

```
void init_display (
    ALLEGRO_DISPLAY *& display )
```

Funzione che crea il display ed inizializza il titolo e l'icona.

Parametri:

1) display - Puntatore al display del gioco;

4.10.1.4 init_display_res()

```
display_info_t init_display_res ( )
```

Funzione che inizializza le dimensioni del display di gioco in base alle proporzioni standard (1280x800), adattandole al monitor del proprio computer.

Viene lasciata libera una certa porzione di schermo (1/5 in altezza).

Ritorna le dimensioni effettive del display di gioco.

4.10.1.5 init_game_requirements()

```
bool init_game_requirements ( )
```

Funzione che inizializza le funzionalità relative all'audio, ai font (ttf), alla tastiera.

Ritorna un booleano che indica il successo delle installazioni.

4.10.1.6 init_match_vars()

```
void init_match_vars (
    match_vars_t & match_vars,
    ALLEGRO_EVENT_QUEUE * ev_queue )
```

Funzione che inizializza le variabili di gioco.

Parametri:

1) match_vars - Struct contenente le variabili di gioco; 2) ev_queue - Coda degli eventi (per inizializzazione dei timer).

4.10.1.7 init_queue_event()

```
ALLEGRO_EVENT_QUEUE* init_queue_event (
    ALLEGRO_DISPLAY *& display )
```

Funzione che crea il puntatore alla coda degli eventi e aggancia alla coda gli eventi relativi alla tastiera e display.

Parametri:

1) display - Puntatore al display del gioco;

Ritorna il puntatore alla coda del display, o nullptr se si è verificato un errore nella creazione.

4.10.1.8 init_settings()

```
void init_settings (
    settings_t & settings )
```

Funzione che inizializza le impostazioni di gioco.

Parametri:

1) settings - Struct contenente le impostazioni di gioco.

4.10.1.9 init_shooter()

```
void init_shooter (
    shooter_t & shooter )
```

Funzione che inizializza le informazioni dello shooter a inizio gioco.

Parametri:

1) shooter - Struct relativo alla navicella.

4.10.1.10 init_sound_music()

```
ALLEGRO_SAMPLE* init_sound_music (
    const char * path )
```

Funzione che inizializza un suono del gioco.

Le impostazioni di inizializzazione sono standard, e il suono viene ripetuto una sola volta.

Parametri:

1) path - Percorso del file audio;

Ritorna il puntatore al suono inizializzato.

4.10.1.11 init_target_bonus()

```
int init_target_bonus (
    int & instant_ast_dest,
    const int & asteroids_destroyed )
```

Funzione che indica dopo quanti asteroidi distrutti ottengo il bonus.

Inizializza instant_ast_dest, per memorizzare quanti asteroidi sono stati distrutti fino a quel momento, per controllare ad ogni asteroide distrutto se si ha diritto al bonus.

Parametri:

1) instant_ast_dest - Appoggio per sapere quanti asteroidi ho distrutto in quel momento, per capire se ho raggiunto il bonus

2) asteroids_destroyed - Numero di asteroidi distrutti in quel momento

4.10.1.12 init_thread_manager()

```
void init_thread_manager (
    thread_manager_t & thread_manager,
    char * email )
```

Funzione che inizializza la struttura dati del thread manager.

Questa struttura contiene i mutex, le condition variable e i thread utilizzati durante il gioco, al fine di sincronizzare i vari worker.

Parametri:

1) thread_manager - Variabile contenente le strutture per la sincronizzazione; 2) email - Email da cercare con cui viene inizializzato il search manager.

4.10.1.13 init_words_buffer()

```
void init_words_buffer (
    words_buffer_t & words_buffer )
```

Funzione che inizializza l'array utilizzato come buffer per le parole tra producer e consumer.

Parametri:

1) words_buffer - Buffer delle parole.

4.10.1.14 sort_file_list()

```
void sort_file_list (
    ls_t * file_list )
```

Funzione che ordina un array di elementi che rappresentano gli archivi in ordine crescente in base alla loro dimensione.

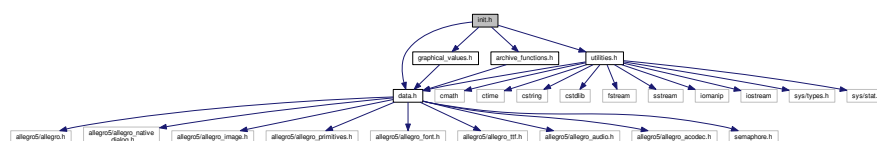
Parametri:

1) file_list - Lista degli archivi;

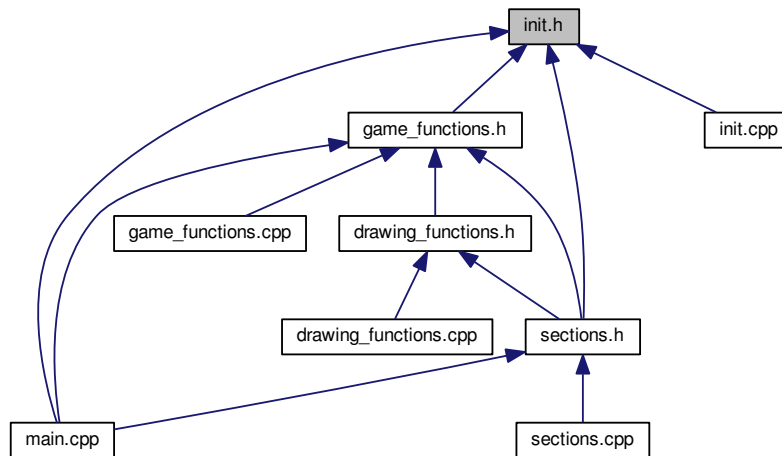
4.11 Riferimenti per il file init.h

```
#include "data.h"
#include "graphical_values.h"
#include "utilities.h"
#include "archive_functions.h"
```

Grafo delle dipendenze di inclusione per init.h:



Questo grafo mostra quali altri file includono direttamente o indirettamente questo file:



Funzioni

- `bool init_game_requirements ()`
Funzione che inizializza le funzionalità relative all'audio, ai font (ttf), alla tastiera.
- `display_info_t init_display_res ()`
Funzione che inizializza le dimensioni del display di gioco in base alle proporzioni standard (1280x800), adattandole al monitor del proprio computer.
- `ALLEGRO_EVENT_QUEUE * init_queue_event (ALLEGRO_DISPLAY *&display)`
Funzione che crea il puntatore alla coda degli eventi e aggancia alla coda gli eventi relativi alla tastiera e display.
- `void init_display (ALLEGRO_DISPLAY *&display)`
Funzione che crea il display ed inizializza il titolo e l'icona.
- `int init_target_bonus (int &instant_ast_dest, const int &asteroids_destroyed)`
Funzione che indica dopo quanti asteroidi distrutti ottengo il bonus.
- `ALLEGRO_SAMPLE * init_sound_music (const char *path)`
Funzione che inizializza un suono del gioco.
- `void init_match_vars (match_vars_t &match_vars, ALLEGRO_EVENT_QUEUE *ev_queue)`
Funzione che inizializza le variabili di gioco.
- `void init_settings (settings_t &settings)`
Funzione che inizializza le impostazioni di gioco.
- `void init_shooter (shooter_t &shooter)`
Funzione che inizializza le informazioni dello shooter a inizio gioco.
- `void init_bullet (bullet_t &bullet)`
Funzione che inizializza le informazioni di un proiettile.
- `void init_arr_bullet (bullet_t *arr_bullet)`
Funzione che inizializza un array di puntatori a asteroidi.
- `void init_words_buffer (words_buffer_t &words_buffer)`
Funzione che inizializza l'array utilizzato come buffer per le parole tra producer e consumer.
- `void init_thread_manager (thread_manager_t &thread_manager, char *email)`
Funzione che inizializza la struttura dati del thread manager.

Variabili

- `display_info_t display_info`
- `playwa_gv_t PLAYWA_GV`
Struct contenente le proporzioni grafiche della schermata di gioco.
- `ALLEGRO_SAMPLE * music_wa`

4.11.1 Documentazione delle funzioni

4.11.1.1 `init_arr_bullet()`

```
void init_arr_bullet (
    bullet_t * arr_bullet )
```

Funzione che inizializza un array di puntatori a asteroidi.

Parametri:

- 1) `arr_bullet` - Array dei bullet.

4.11.1.2 `init_bullet()`

```
void init_bullet (
    bullet_t & bullet )
```

Funzione che inizializza le informazioni di un proiettile.

Parametri:

- 1) `shooter` - Struct relativo al singolo proiettile.

4.11.1.3 `init_display()`

```
void init_display (
    ALLEGRO_DISPLAY *& display )
```

Funzione che crea il display ed inizializza il titolo e l'icona.

Parametri:

- 1) `display` - Puntatore al display del gioco;

4.11.1.4 init_display_res()

```
display_info_t init_display_res ( )
```

Funzione che inizializza le dimensioni del display di gioco in base alle proporzioni standard (1280x800), adattandole al monitor del proprio computer.

Viene lasciata libera una certa porzione di schermo (1/5 in altezza).
Ritorna le dimensioni effettive del display di gioco.

4.11.1.5 init_game_requirements()

```
bool init_game_requirements ( )
```

Funzione che inizializza le funzionalità relative all'audio, ai font (ttf), alla tastiera.

Ritorna un booleano che indica il successo delle installazioni.

4.11.1.6 init_match_vars()

```
void init_match_vars (
    match_vars_t & match_vars,
    ALLEGRO_EVENT_QUEUE * ev_queue )
```

Funzione che inizializza le variabili di gioco.

Parametri:

1) match_vars - Struct contenente le variabili di gioco; 2) ev_queue - Coda degli eventi (per inizializzazione dei timer).

4.11.1.7 init_queue_event()

```
ALLEGRO_EVENT_QUEUE* init_queue_event (
    ALLEGRO_DISPLAY *& display )
```

Funzione che crea il puntatore alla coda degli eventi e aggancia alla coda gli eventi relativi alla tastiera e display.

Parametri:

1) display - Puntatore al display del gioco;
Ritorna il puntatore alla coda del display, o nullptr se si è verificato un errore nella creazione.

4.11.1.8 init_settings()

```
void init_settings (
    settings_t & settings )
```

Funzione che inizializza le impostazioni di gioco.

Parametri:

1) settings - Struct contenente le impostazioni di gioco.

4.11.1.9 init_shooter()

```
void init_shooter (
    shooter_t & shooter )
```

Funzione che inizializza le informazioni dello shooter a inizio gioco.

Parametri:

1) shooter - Struct relativo alla navicella.

4.11.1.10 init_sound_music()

```
ALLEGRO_SAMPLE* init_sound_music (
    const char * path )
```

Funzione che inizializza un suono del gioco.

Le impostazioni di inizializzazione sono standard, e il suono viene ripetuto una sola volta.

Parametri:

1) path - Percorso del file audio;

Ritorna il puntatore al suono inizializzato.

4.11.1.11 init_target_bonus()

```
int init_target_bonus (
    int & instant_ast_dest,
    const int & asteroids_destroyed )
```

Funzione che indica dopo quanti asteroidi distrutti ottengo il bonus.

Inizializza instant_ast_dest, per memorizzare quanti asteroidi sono stati distrutti fino a quel momento, per controllare ad ogni asteroide distrutto se si ha diritto al bonus.

Parametri:

1) instant_ast_dest - Appoggio per sapere quanti asteroidi ho distrutto in quel momento, per capire se ho raggiunto il bonus

2) asteroids_destroyed - Numero di asteroidi distrutti in quel momento

4.11.1.12 init_thread_manager()

```
void init_thread_manager (
    thread_manager_t & thread_manager,
    char * email )
```

Funzione che inizializza la struttura dati del thread manager.

Questa struttura contiene i mutex, le condition variable e i thread utilizzati durante il gioco, al fine di sincronizzare i vari worker.

Parametri:

1) thread_manager - Variabile contenente le strutture per la sincronizzazione; 2) email - Email da cercare con cui viene inizializzato il search manager.

4.11.1.13 init_words_buffer()

```
void init_words_buffer (
    words_buffer_t & words_buffer )
```

Funzione che inizializza l'array utilizzato come buffer per le parole tra producer e consumer.

Parametri:

1) words_buffer - Buffer delle parole.

4.11.2 Documentazione delle variabili

4.11.2.1 display_info

```
display_info_t display_info
```

4.11.2.2 music_wa

ALLEGRO_SAMPLE* music_wa

4.11.2.3 PLAYWA_GV

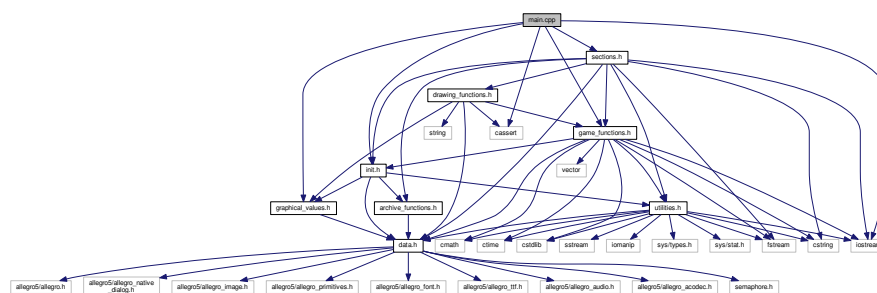
```
playwa_gv_t PLAYWA_GV
```

Struct contenente le proporzioni grafiche della schermata di gioco.

4.12 Riferimenti per il file main.cpp

```
#include "init.h"
#include "graphical_values.h"
#include "game_functions.h"
#include "sections.h"
#include <iostream>
#include <cassert>
```

Grafo delle dipendenze di inclusione per main.cpp:



Funzioni

- int `main` (int argc, char **argv)

Variabili

- `display_info_t` `display_info`
- ALLEGRO_SAMPLE * `music_wa` = nullptr
- `thread_manager_t` `thread_manager`
- `words_buffer_t` `words_buffer`

4.12.1 Documentazione delle funzioni

4.12.1.1 `main()`

```
int main (
    int argc,
    char ** argv )
```

Inizializzazione dimensioni display, e aggiornamento proporzioni grafiche Inizializzazione della coda degli eventi e delle impostazioni Inizializzazione e avvio della musica

4.12.2 Documentazione delle variabili

4.12.2.1 `display_info`

```
display_info_t display_info
```

4.12.2.2 `music_wa`

```
ALLEGRO_SAMPLE* music_wa = nullptr
```

4.12.2.3 `thread_manager`

```
thread_manager_t thread_manager
```


4.12.2.4 words_buffer

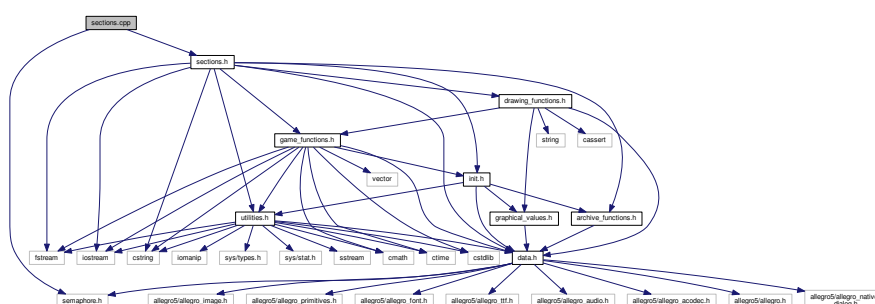
```
words_buffer_t words_buffer
```

4.13 Riferimenti per il file sections.cpp

```
#include <semaphore.h>
```

```
#include "sections.h"
```

Grafo delle dipendenze di inclusione per sections.cpp:



Funzioni

- float `intro` (ALLEGRO_DISPLAY *display)
Funzione che lancia l'animazione di introduzione.
- int `main_menu` (ALLEGRO_DISPLAY *display, ALLEGRO_EVENT_QUEUE *ev_queue, const float &prev↵
_bg_offset)
Funzione che gestisce la schermata del menu principale.
- bool `settings_menu` (ALLEGRO_DISPLAY *display, ALLEGRO_EVENT_QUEUE *ev_queue, `settings_t`
&settings)
Funzione che gestisce la schermata del menu delle impostazioni.
- bool `play_wa` (ALLEGRO_DISPLAY *display, ALLEGRO_EVENT_QUEUE *ev_queue, `settings_t` &settings)
Funzione che gestisce la schermata di gioco.

4.13.1 Documentazione delle funzioni

4.13.1.1 intro()

```
float intro (
    ALLEGRO_DISPLAY * display )
```

Funzione che lancia l'animazione di introduzione.

Parametri:

1) display - Display corrente.

Ritorna la posizione dello sfondo a fine funzione (espressa in pixel).

4.13.1.2 main_menu()

```
int main_menu (
    ALLEGRO_DISPLAY * display,
    ALLEGRO_EVENT_QUEUE * ev_queue,
    const float & prev_bg_offset )
```

Funzione che gestisce la schermata del menu principale.

Parametri:

- 1) display - Display corrente;
 - 2) ev_queue - Coda degli eventi;
 - 3) prev_bg_offset - Posizione dello sfondo al momento del lancio della funzione.
- Ritorna la scelta effettuata dall'utente.

4.13.1.3 play_wa()

```
bool play_wa (
    ALLEGRO_DISPLAY * display,
    ALLEGRO_EVENT_QUEUE * ev_queue,
    settings_t & settings )
```

Funzione che gestisce la schermata di gioco.

Parametri:

- 1) display - Display corrente;
 - 2) ev_queue - Coda degli eventi;
 - 3) settings - Struct contenente le impostazioni di gioco.
- Ritorna true se la partita termina normalmente, false se l'utente decide di uscire dal gioco.

4.13.1.4 settings_menu()

```
bool settings_menu (
    ALLEGRO_DISPLAY * display,
    ALLEGRO_EVENT_QUEUE * ev_queue,
    settings_t & settings )
```

Funzione che gestisce la schermata del menu delle impostazioni.

Parametri:

- 1) display - Display corrente;
 - 2) ev_queue - Coda degli eventi;
 - 3) settings - Struct contenente le impostazioni di gioco.
- Ritorna true se l'utente decide di tornare al menu principale, false se decide di uscire dal gioco.

Variabili

- `display_info_t display_info`
- `thread_manager_t thread_manager`
- `words_buffer_t words_buffer`
- `menu_gv_t MENU_GV`
Struct contenente le proporzioni grafiche dei menu.
- `usr_gv_t USR_GV`
Struct contenente le proporzioni grafiche della schermata di fine partita.
- `ALLEGRO_SAMPLE * music_wa`

4.14.1 Documentazione delle funzioni

4.14.1.1 `intro()`

```
float intro (
    ALLEGRO_DISPLAY * display )
```

Funzione che lancia l'animazione di introduzione.

Parametri:

1) `display` - Display corrente.

Ritorna la posizione dello sfondo a fine funzione (espressa in pixel).

4.14.1.2 `main_menu()`

```
int main_menu (
    ALLEGRO_DISPLAY * display,
    ALLEGRO_EVENT_QUEUE * ev_queue,
    const float & prev_bg_offset )
```

Funzione che gestisce la schermata del menu principale.

Parametri:

1) `display` - Display corrente;

2) `ev_queue` - Coda degli eventi;

3) `prev_bg_offset` - Posizione dello sfondo al momento del lancio della funzione.

Ritorna la scelta effettuata dall'utente.

4.14.1.3 `play_wa()`

```
bool play_wa (
    ALLEGRO_DISPLAY * display,
    ALLEGRO_EVENT_QUEUE * ev_queue,
    settings_t & settings )
```

Funzione che gestisce la schermata di gioco.

Parametri:

1) `display` - Display corrente;

2) `ev_queue` - Coda degli eventi;

3) `settings` - Struct contenente le impostazioni di gioco.

Ritorna true se la partita termina normalmente, false se l'utente decide di uscire dal gioco.

4.14.1.4 settings_menu()

```
bool settings_menu (
    ALLEGRO_DISPLAY * display,
    ALLEGRO_EVENT_QUEUE * ev_queue,
    settings_t & settings )
```

Funzione che gestisce la schermata del menu delle impostazioni.

Parametri:

- 1) display - Display corrente;
 - 2) ev_queue - Coda degli eventi;
 - 3) settings - Struct contenente le impostazioni di gioco.
- Ritorna true se l'utente decide di tornare al menu principale, false se decide di uscire dal gioco.

4.14.2 Documentazione delle variabili

4.14.2.1 display_info

```
display_info_t display_info
```

4.14.2.2 MENU_GV

```
menu_gv_t MENU_GV
```

Struct contenente le proporzioni grafiche dei menu.

4.14.2.3 music_wa

```
ALLEGRO_SAMPLE* music_wa
```

4.14.2.4 thread_manager

```
thread_manager_t thread_manager
```

4.14.2.5 USR_GV

`usr_gv_t` USR_GV

Struct contenente le proporzioni grafiche della schermata di fine partita.

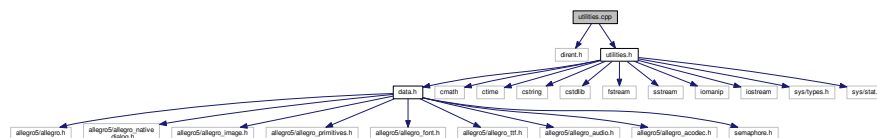
4.14.2.6 words_buffer

`words_buffer_t` words_buffer

4.15 Riferimenti per il file utilities.cpp

```
#include <dirent.h>
#include "utilities.h"
```

Grafo delle dipendenze di inclusione per utilities.cpp:



Funzioni

- bool `exit` (ALLEGRO_DISPLAY *display, const bool playing)
Funzione ausiliaria che genera un alert con la richiesta di uscire dal gioco.
- char * `int_to_str_score` (const int &match_score)
Funzione che converte il punteggio (intero) in una stringa di otto caratteri.
- int `which_quadrant` (const float x, const float y)
Funzione che, dato un certo punto del display, restituisce il quadrante (1-4) al quale appartiene.
- void `extract_elem` (asteroid_list *ast_queues, const int &index_to_extract)
Funzione che estrai l'elemento dalla coda.
- int `insert_elem_head` (asteroid_list *ast_queues, const asteroid_t &asteroid)
Funzione che inserisce un elemento nella lista, con un controllo se l'elemento è inserito in testa o in coda.
- void `change_enable` (enable_t &enabled)
Funzione adibita al set/reset di una determinata impostazione di gioco.
- void `delete_ast_queues` (asteroid_list *ast_queues)
Funzione che dealloca gli asteroidi in gioco a fine partita e le relativi immagini.
- void `delete_timer` (match_vars_t &match_vars)
Funzione che dealloca i timer utilizzati come appoggio durante la partita.
- void `delete_display_evqueue_music` (ALLEGRO_DISPLAY *display, ALLEGRO_EVENT_QUEUE *ev_queue, ALLEGRO_SAMPLE *sound)
Funzione che dealloca il display, la coda degli eventi e la musica alla chiusura del gioco.
- void `delete_sounds` (match_vars_t &match_vars)
Funzione che dealloca i suoni durante la singola partita.
- bool `ls_tar` (ls_t *ls)
Funzione che restituisce la lista di archivi presenti nella cartella.
- bool `archives_exist` ()
Funzione true nel caso la cartella indicata durante la compilazione del programma contenga archivi validi.

4.15.1 Documentazione delle funzioni

4.15.1.1 archives_exist()

```
bool archives_exist ( )
```

Funzione true nel caso la cartella indicata durante la compilazione del programma contenga archivi validi.

Ritorna false altrimenti.

4.15.1.2 change_enable()

```
void change_enable (
    enable_t & enable )
```

Funzione adibita al set/reset di una determinata impostazione di gioco.

Parametri:

1) enable - Impostazione corrente.

4.15.1.3 delete_ast_queues()

```
void delete_ast_queues (
    asteroid_list * ast_queues )
```

Funzione che dealloca gli asteroidi in gioco a fine partita e le relativi immagini.

Parametri:

1) ast_queues - Vettore di liste di asteroidi.

4.15.1.4 delete_display_evqueue_music()

```
void delete_display_evqueue_music (
    ALLEGRO_DISPLAY * display,
    ALLEGRO_EVENT_QUEUE * ev_queue,
    ALLEGRO_SAMPLE * sound )
```

Funzione che dealloca il display, la coda degli eventi e la musica alla chiusura del gioco.

Parametri:

1) display - Puntatore al display del gioco;
2) ev_queue - Puntatore alla coda di eventi;
3) sound - Puntatore alla musica del gioco.

4.15.1.5 delete_sounds()

```
void delete_sounds (
    match_vars_t & match_vars )
```

Funzione che dealloca i suoni durante la singola partita.

Parametri:

1) match_vars - Contenitore dei puntatori agli audio.

4.15.1.6 delete_timer()

```
void delete_timer (
    match_vars_t & match_vars )
```

Funzione che dealloca i timer utilizzati come appoggio durante la partita.

Parametri:

1) match_vars - Contenitore dei puntatori ai tre timer.

4.15.1.7 exit()

```
bool exit (
    ALLEGRO_DISPLAY * display,
    const bool playing )
```

Funzione ausiliaria che genera un alert con la richiesta di uscire dal gioco.

Parametri:

1) display - Puntatore al display del gioco;

2) playing - Flag per indicare se si sta giocando, in modo da modificare il messaggio;

Ritorna un booleano, e indica se si desidera uscire (true) o continuare a giocare (false).

4.15.1.8 extract_elem()

```
void extract_elem (
    asteroid_list * ast_queues,
    const int & index_to_extract )
```

Funzione che estrai l'elemento dalla coda.

Non sono necessari controlli prima dell'estrazione, perchè sono fatti prima della chiamata a questa funzione.

Parametri:

1) ast_queues - Vettore di liste di asteroidi;

2) character - Iniziale dell'elemento da eliminare, cioè indice dell'array.

4.15.1.9 `insert_elem_head()`

```
int insert_elem_head (
    asteroid_list * ast_queues,
    const asteroid_t & asteroid )
```

Funzione che inserisce un elemento nella lista, con un controllo se l'elemento è inserito in testa o in coda.

Ricavo l'indice della coda direttamente da `asteroid.word[0]`.

Parametri:

- 1) `ast_queues` - Vettore di liste di asteroidi;
- 2) `asteroid` - Asteroide da inserire. Ritorna un intero, -1 se l'asteroide è stato inserito in testa altrimenti un indice che rappresenta il numero della coda.

4.15.1.10 `int_to_str_score()`

```
char* int_to_str_score (
    const int & match_score )
```

Funzione che converte il punteggio (intero) in una stringa di otto caratteri.

Parametri:

- 1) `match_score` - Il punteggio intero.
- Ritorna il puntatore contenente la stringa formattata.

4.15.1.11 `ls_tar()`

```
bool ls_tar (
    ls_t * ls )
```

Funzione che restituisce la lista di archivi presenti nella cartella.

Parametri:

- 1) `ls` - struttura che rappresenta i file contenuti nella cartella. Ritorno false in caso di errore, true altrimenti.

4.15.1.12 `which_quadrant()`

```
int which_quadrant (
    const float x,
    const float y )
```

Funzione che, dato un certo punto del display, restituisce il quadrante (1-4) al quale appartiene.

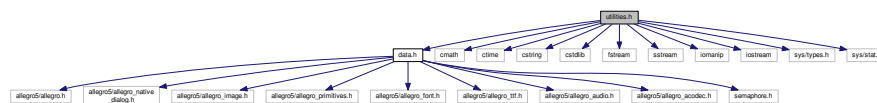
Parametri:

- 1) `x` - Ascissa punto;
- 2) `y` - Ordinata punto.

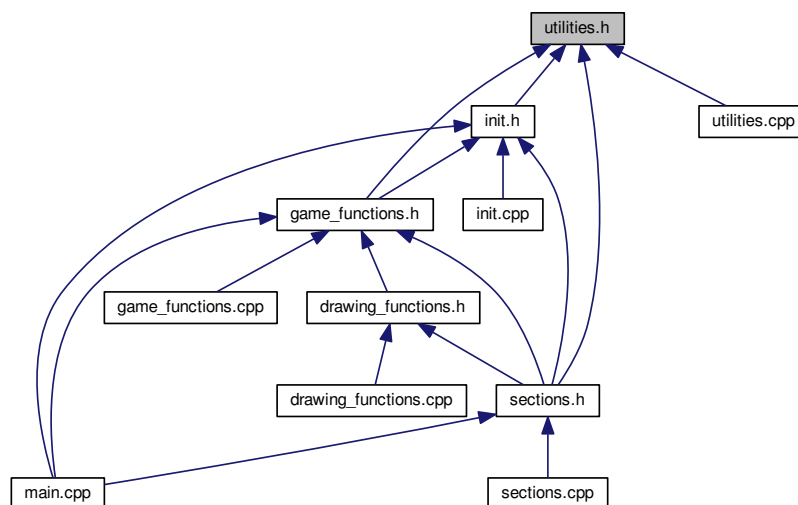
4.16 Riferimenti per il file utilities.h

```
#include "data.h"
#include <cmath>
#include <ctime>
#include <cstring>
#include <cstdlib>
#include <fstream>
#include <sstream>
#include <iomanip>
#include <iostream>
#include <sys/types.h>
#include <sys/stat.h>
```

Grafo delle dipendenze di inclusione per utilities.h:



Questo grafo mostra quali altri file includono direttamente o indirettamente questo file:



Funzioni

- bool **exit** (ALLEGRO_DISPLAY *display, const bool playing)
Funzione ausiliaria che genera un alert con la richiesta di uscire dal gioco.
- char * **int_to_str_score** (const int &match_score)
Funzione che converte il punteggio (intero) in una stringa di otto caratteri.
- int **which_quadrant** (const float x, const float y)
Funzione che, dato un certo punto del display, restituisce il quadrante (1-4) al quale appartiene.
- void **extract_elem** (asteroid_list *ast_queues, const int &index_to_extract)

Funzione che estrai l'elemento dalla coda.

- int `insert_elem_head` (`asteroid_list` *ast_queues, const `asteroid_t` &asteroid)

Funzione che inserisce un elemento nella lista, con un controllo se l'elemento è inserito in testa o in coda.

- void `change_enable` (`enable_t` &enable)

Funzione adibita al set/reset di una determinata impostazione di gioco.

- void `delete_ast_queues` (`asteroid_list` *ast_queues)

Funzione che dealloca gli asteroidi in gioco a fine partita e le relativi immagini.

- void `delete_timer` (`match_vars_t` &match_vars)

Funzione che dealloca i timer utilizzati come appoggio durante la partita.

- void `delete_display_evqueue_music` (`ALLEGRO_DISPLAY` *display, `ALLEGRO_EVENT_QUEUE` *ev_queue, `ALLEGRO_SAMPLE` *sound)

Funzione che dealloca il display, la coda degli eventi e la musica alla chiusura del gioco.

- void `delete_sounds` (`match_vars_t` &match_vars)

Funzione che dealloca i suoni durante la singola partita.

- bool `ls_tar` (`ls_t` *ls)

Funzione che restituisce la lista di archivi presenti nella cartella.

- bool `archives_exist` ()

Funzione true nel caso la cartella indicata durante la compilazione del programma contenga archivi validi.

Variabili

- `display_info_t` `display_info`

4.16.1 Documentazione delle funzioni

4.16.1.1 `archives_exist()`

```
bool archives_exist ( )
```

Funzione true nel caso la cartella indicata durante la compilazione del programma contenga archivi validi.

Ritorna false altrimenti.

4.16.1.2 `change_enable()`

```
void change_enable (
    enable_t & enable )
```

Funzione adibita al set/reset di una determinata impostazione di gioco.

Parametri:

- 1) enable - Impostazione corrente.

4.16.1.3 delete_ast_queues()

```
void delete_ast_queues (
    asteroid_list * ast_queues )
```

Funzione che dealloca gli asteroidi in gioco a fine partita e le relativi immagini.

Parametri:

1) ast_queues - Vettore di liste di asteroidi.

4.16.1.4 delete_display_evqueue_music()

```
void delete_display_evqueue_music (
    ALLEGRO_DISPLAY * display,
    ALLEGRO_EVENT_QUEUE * ev_queue,
    ALLEGRO_SAMPLE * sound )
```

Funzione che dealloca il display, la coda degli eventi e la musica alla chiusura del gioco.

Parametri:

- 1) display - Puntatore al display del gioco;
- 2) ev_queue - Puntatore alla coda di eventi;
- 3) sound - Puntatore alla musica del gioco.

4.16.1.5 delete_sounds()

```
void delete_sounds (
    match_vars_t & match_vars )
```

Funzione che dealloca i suoni durante la singola partita.

Parametri:

1) match_vars - Contenitore dei puntatori agli audio.

4.16.1.6 delete_timer()

```
void delete_timer (
    match_vars_t & match_vars )
```

Funzione che dealloca i timer utilizzati come appoggio durante la partita.

Parametri:

1) match_vars - Contenitore dei puntatori ai tre timer.

4.16.1.7 `exit()`

```
bool exit (
    ALLEGRO_DISPLAY * display,
    const bool playing )
```

Funzione ausiliaria che genera un alert con la richiesta di uscire dal gioco.

Parametri:

- 1) `display` - Puntatore al display del gioco;
 - 2) `playing` - Flag per indicare se si sta giocando, in modo da modificare il messaggio;
- Ritorna un booleano, e indica se si desidera uscire (true) o continuare a giocare (false).

4.16.1.8 `extract_elem()`

```
void extract_elem (
    asteroid_list * ast_queues,
    const int & index_to_extract )
```

Funzione che estrai l'elemento dalla coda.

Non sono necessari controlli prima dell'estrazione, perchè sono fatti prima della chiamata a questa funzione.

Parametri:

- 1) `ast_queues` - Vettore di liste di asteroidi;
- 2) `character` - Iniziale dell'elemento da eliminare, cioè indice dell'array.

4.16.1.9 `insert_elem_head()`

```
int insert_elem_head (
    asteroid_list * ast_queues,
    const asteroid_t & asteroid )
```

Funzione che inserisce un elemento nella lista, con un controllo se l'elemento è inserito in testa o in coda.

Ricavo l'indice della coda direttamente da `asteroid.word[0]`.

Parametri:

- 1) `ast_queues` - Vettore di liste di asteroidi;
- 2) `asteroid` - Asteroide da inserire. Ritorna un intero, -1 se l'asteroide è stato inserito in testa altrimenti un indice che rappresenta il numero della coda.

4.16.1.10 `int_to_str_score()`

```
char* int_to_str_score (
    const int & match_score )
```

Funzione che converte il punteggio (intero) in una stringa di otto caratteri.

Parametri:

- 1) `match_score` - Il punteggio intero.
- Ritorna il puntatore contenente la stringa formattata.

4.16.1.11 ls_tar()

```
bool ls_tar (
    ls_t * ls )
```

Funzione che restituisce la lista di archivi presenti nella cartella.

Parametri:

1) ls - struttura che rappresenta i file contenuti nella cartella. Ritorno false in caso di errore, true altrimenti.

4.16.1.12 which_quadrant()

```
int which_quadrant (
    const float x,
    const float y )
```

Funzione che, dato un certo punto del display, restituisce il quadrante (1-4) al quale appartiene.

Parametri:

- 1) x - Ascissa punto;
- 2) y - Ordinata punto.

4.16.2 Documentazione delle variabili

4.16.2.1 display_info

```
display_info_t display_info
```

Indice analitico

ASTEROID_FONTSIZE
 playwa_gv_t, [23](#)
ASTEROID_SHADOW_GAP
 playwa_gv_t, [23](#)
ASTEROID_TEXT_SHADOW_GAP
 playwa_gv_t, [23](#)
archive_functions.cpp, [37](#)
 start_searching, [37](#)
archive_functions.h, [38](#)
 start_searching, [39](#)
 thread_manager, [39](#)
 words_buffer, [39](#)
archives_exist
 utilities.cpp, [101](#)
 utilities.h, [105](#)
ast_queues
 match_vars_t, [17](#)
asteroid
 elem_asteroid_t, [9](#)
asteroid_head
 match_vars_t, [17](#)
asteroid_list
 data.h, [47](#)
asteroid_special_t
 data.h, [48](#)
asteroid_t, [5](#)
 bmp, [5](#)
 radius, [6](#)
 slope, [6](#)
 special, [6](#)
 word, [6](#)
 x, [6](#)
 y, [6](#)
asteroid_to_bonus
 match_vars_t, [18](#)
asteroid_to_generate
 match_vars_t, [18](#)
asteroids_destroyed
 match_vars_t, [18](#)

BACK_FONTSIZE
 general_gv_t, [11](#)
BACK_LEFT_OFFSET
 general_gv_t, [11](#)
BACK_TEXT_SHADOW_GAP
 general_gv_t, [11](#)
BACK_TOP_OFFSET
 general_gv_t, [11](#)
BG_REFRESH_TIME
 data.h, [43](#)

BMP_PATH_BASE
 data.h, [48](#)
BONUS_CASE_WIDTH
 instr_gv_t, [13](#)
BONUS_HINT_FONTSIZE
 playwa_gv_t, [23](#)
BONUS_LEFT_OFFSET
 playwa_gv_t, [24](#)
BONUS_NUM_FONTSIZE
 playwa_gv_t, [24](#)
BONUS_RADIUS
 instr_gv_t, [13](#)
BONUS_TOP_OFFSET
 playwa_gv_t, [24](#)
BULLET_RADIUS
 playwa_gv_t, [24](#)
BULLET_SCALE
 playwa_gv_t, [24](#)
bmp
 asteroid_t, [5](#)
bullet_t, [7](#)
 radius, [7](#)
 slope, [7](#)
 x, [7](#)
 y, [7](#)

COPYRIGHT_FONTSIZE
 menu_gv_t, [21](#)
COPYRIGHT_TOP_OFFSET
 menu_gv_t, [21](#)
change_enable
 utilities.cpp, [101](#)
 utilities.h, [105](#)
change_music
 game_functions.cpp, [72](#)
 game_functions.h, [77](#)
check_distance
 game_functions.cpp, [72](#)
 game_functions.h, [77](#)
check_gameover
 game_functions.cpp, [72](#)
 game_functions.h, [77](#)
check_shield
 game_functions.cpp, [72](#)
 game_functions.h, [77](#)
consumer_sem
 thread_manager_t, [32](#)
count_result
 result_t, [26](#)
current_asteroid

- match_vars_t, 18
- DANGER_RADIUS
 - playwa_gv_t, 24
- data.h, 40
 - asteroid_list, 47
 - asteroid_special_t, 48
 - BG_REFRESH_TIME, 43
 - BMP_PATH_BASE, 48
 - enable_t, 48
 - FONT_PATH, 48
 - GAMERES_H, 43
 - GAMERES_W, 43
 - MAX_ASTEROID_HEAD, 43
 - MAX_DIGIT_SCORE, 44
 - MAX_LENGTH_WORD, 44
 - MAX_MATCHING_PASS, 44
 - MAX_USERNAME_LENGTH, 44
 - MENU_CPY_COLOR, 44
 - MENU_FONT_COLOR, 44
 - MIN_TIME_SPAWN, 45
 - MONITOR_OFFSET_RATE, 45
 - MUSIC_GAMEOVER_PATH_WA, 49
 - MUSIC_MENU_PATH_WA, 49
 - MUSIC_PLAYWA_PATH_WA, 49
 - N_QUEUES, 49
 - NUM_INSTR_BITMAPS, 45
 - NUM_INSTR_FONTS, 45
 - NUM_ITEMS, 45
 - NUM_MENU_BITMAPS, 45
 - NUM_MENU_FONTS, 46
 - NUM_OPTIONS_SETTINGS, 46
 - NUM_OPTIONS, 46
 - NUM_PLAYWA_BITMAPS, 46
 - NUM_PLAYWA_FONTS, 46
 - NUM_SETTINGS_FONTS, 46
 - NUM_USER_FONTS, 47
 - SCORE_INCREMENT, 47
 - SHADOW_COLOR, 47
 - SOUND_PATH_ERROR, 49
 - SOUND_PATH_EXPLOSION, 49
 - SOUND_PATH_FIRE, 50
 - SOUND_PATH_GAMEOVER, 50
 - TIMER_RALLENTY_BONUS, 47
 - TIMER_SHIELD_BONUS, 47
- delete_ast_queues
 - utilities.cpp, 101
 - utilities.h, 105
- delete_display_evqueue_music
 - utilities.cpp, 101
 - utilities.h, 106
- delete_sounds
 - utilities.cpp, 101
 - utilities.h, 106
- delete_timer
 - utilities.cpp, 102
 - utilities.h, 106
- destroy_arr_bitmaps
 - drawing_functions.cpp, 52
- drawing_functions.h, 62
- destroy_arr_fonts
 - drawing_functions.cpp, 52
 - drawing_functions.h, 62
- display_info
 - drawing_functions.h, 69
 - game_functions.h, 80
 - graphical_values.h, 84
 - init.h, 93
 - main.cpp, 94
 - sections.h, 99
 - utilities.h, 108
- display_info_t, 8
 - height, 8
 - width, 8
- draw_back
 - drawing_functions.cpp, 52
 - drawing_functions.h, 62
- draw_background
 - drawing_functions.cpp, 53
 - drawing_functions.h, 63
- draw_gameover_animation
 - drawing_functions.cpp, 53
 - drawing_functions.h, 63
- draw_head
 - drawing_functions.cpp, 53
 - drawing_functions.h, 63
- draw_instructions
 - drawing_functions.cpp, 53
 - drawing_functions.h, 63
- draw_intro_init
 - drawing_functions.cpp, 54
 - drawing_functions.h, 64
- draw_intro_menu_fadein
 - drawing_functions.cpp, 54
 - drawing_functions.h, 64
- draw_intro_title_slide
 - drawing_functions.cpp, 54
 - drawing_functions.h, 64
- draw_menu_copyright
 - drawing_functions.cpp, 54
 - drawing_functions.h, 64
- draw_menu_option
 - drawing_functions.cpp, 55
 - drawing_functions.h, 65
- draw_menu_title
 - drawing_functions.cpp, 55
 - drawing_functions.h, 65
- draw_playagain_button
 - drawing_functions.cpp, 55
 - drawing_functions.h, 65
- draw_playwa_asteroids_bullets
 - drawing_functions.cpp, 55
 - drawing_functions.h, 65
- draw_playwa_bonus_music
 - drawing_functions.cpp, 56
 - drawing_functions.h, 66
- draw_playwa_score_hint

- drawing_functions.cpp, 56
- drawing_functions.h, 66
- draw_playwa_shooter
 - drawing_functions.cpp, 56
 - drawing_functions.h, 66
- draw_results
 - drawing_functions.cpp, 57
 - drawing_functions.h, 67
- draw_stat
 - drawing_functions.cpp, 57
 - drawing_functions.h, 67
- draw_stats_blocks
 - drawing_functions.cpp, 57
 - drawing_functions.h, 67
- draw_username
 - drawing_functions.cpp, 57
 - drawing_functions.h, 67
- draw_username_block
 - drawing_functions.cpp, 58
 - drawing_functions.h, 68
- drawing_functions.cpp, 50
 - destroy_arr_bitmaps, 52
 - destroy_arr_fonts, 52
 - draw_back, 52
 - draw_background, 53
 - draw_gameover_animation, 53
 - draw_head, 53
 - draw_instructions, 53
 - draw_intro_init, 54
 - draw_intro_menu_fadein, 54
 - draw_intro_title_slide, 54
 - draw_menu_copyright, 54
 - draw_menu_option, 55
 - draw_menu_title, 55
 - draw_playagain_button, 55
 - draw_playwa_asteroids_bullets, 55
 - draw_playwa_bonus_music, 56
 - draw_playwa_score_hint, 56
 - draw_playwa_shooter, 56
 - draw_results, 57
 - draw_stat, 57
 - draw_stats_blocks, 57
 - draw_username, 57
 - draw_username_block, 58
 - generate_instr_bitmaps, 58
 - generate_instr_fonts, 58
 - generate_menu_bitmaps, 58
 - generate_menu_fonts, 58
 - generate_playwa_bitmaps, 59
 - generate_playwa_fonts, 59
 - generate_settings_fonts, 59
 - generate_user_fonts, 59
- drawing_functions.h, 60
 - destroy_arr_bitmaps, 62
 - destroy_arr_fonts, 62
 - display_info, 69
 - draw_back, 62
 - draw_background, 63
 - draw_gameover_animation, 63
 - draw_head, 63
 - draw_instructions, 63
 - draw_intro_init, 64
 - draw_intro_menu_fadein, 64
 - draw_intro_title_slide, 64
 - draw_menu_copyright, 64
 - draw_menu_option, 65
 - draw_menu_title, 65
 - draw_playagain_button, 65
 - draw_playwa_asteroids_bullets, 65
 - draw_playwa_bonus_music, 66
 - draw_playwa_score_hint, 66
 - draw_playwa_shooter, 66
 - draw_results, 67
 - draw_stat, 67
 - draw_stats_blocks, 67
 - draw_username, 67
 - draw_username_block, 68
 - GENERAL_GV, 70
 - generate_instr_bitmaps, 68
 - generate_instr_fonts, 68
 - generate_menu_bitmaps, 68
 - generate_menu_fonts, 68
 - generate_playwa_bitmaps, 69
 - generate_playwa_fonts, 69
 - generate_settings_fonts, 69
 - generate_user_fonts, 69
 - INSTR_GV, 70
 - MENU_GV, 70
 - PLAYWA_GV, 70
 - USR_GV, 70
- ENTER_FONTSIZE
 - usr_gv_t, 34
- ENTER_HEIGHT
 - usr_gv_t, 34
- ENTER_LEFT_OFFSET
 - usr_gv_t, 34
- ENTER_TOP_OFFSET
 - usr_gv_t, 34
- ENTER_WIDTH
 - usr_gv_t, 34
- elem_asteroid_t, 9
 - asteroid, 9
 - next, 9
- email
 - look_into_t, 14
 - search_manager_t, 27
- enable_t
 - data.h, 48
- exit
 - utilities.cpp, 102
 - utilities.h, 106
- extract_elem
 - utilities.cpp, 102
 - utilities.h, 107
- FONT_PATH

- data.h, 48
- file_list
 - search_manager_t, 28
- file_t, 10
 - filename, 10
 - size, 10
- filename
 - file_t, 10
 - look_into_t, 14
- files
 - ls_t, 15
- fps
 - match_vars_t, 18
- GAMERES_H
 - data.h, 43
- GAMERES_W
 - data.h, 43
- GENERAL_GV
 - drawing_functions.h, 70
 - graphical_values.cpp, 82
- game_functions.cpp, 71
 - change_music, 72
 - check_distance, 72
 - check_gameover, 72
 - check_shield, 72
 - generate_asteroid, 73
 - handler_key_pressed, 73
 - handler_timer, 73
 - move_asteroids, 74
 - move_bullets, 74
 - move_shooter, 74
 - music_wa_play, 74
 - sound_play, 75
 - spawn_asteroid_thread, 75
- game_functions.h, 75
 - change_music, 77
 - check_distance, 77
 - check_gameover, 77
 - check_shield, 77
 - display_info, 80
 - generate_asteroid, 78
 - handler_key_pressed, 78
 - handler_timer, 78
 - move_asteroids, 79
 - move_bullets, 79
 - move_shooter, 79
 - music_wa, 80
 - music_wa_play, 79
 - PLAYWA_GV, 80
 - sound_play, 80
 - spawn_asteroid_thread, 80
 - thread_manager, 81
 - words_buffer, 81
- gameover
 - match_vars_t, 18
- gameover_cond
 - thread_manager_t, 32
- gameover_mutex
 - thread_manager_t, 32
- general_gv_t, 11
 - BACK_FONTSIZE, 11
 - BACK_LEFT_OFFSET, 11
 - BACK_TEXT_SHADOW_GAP, 11
 - BACK_TOP_OFFSET, 11
 - HEAD_FONTSIZE, 12
 - HEAD_TEXT_SHADOW_GAP, 12
 - HEAD_TOP_OFFSET, 12
 - RESULT_TOP_OFFSET, 12
- generate_asteroid
 - game_functions.cpp, 73
 - game_functions.h, 78
- generate_instr_bitmaps
 - drawing_functions.cpp, 58
 - drawing_functions.h, 68
- generate_instr_fonts
 - drawing_functions.cpp, 58
 - drawing_functions.h, 68
- generate_menu_bitmaps
 - drawing_functions.cpp, 58
 - drawing_functions.h, 68
- generate_menu_fonts
 - drawing_functions.cpp, 58
 - drawing_functions.h, 68
- generate_playwa_bitmaps
 - drawing_functions.cpp, 59
 - drawing_functions.h, 69
- generate_playwa_fonts
 - drawing_functions.cpp, 59
 - drawing_functions.h, 69
- generate_settings_fonts
 - drawing_functions.cpp, 59
 - drawing_functions.h, 69
- generate_user_fonts
 - drawing_functions.cpp, 59
 - drawing_functions.h, 69
- graphical_values.cpp, 81
 - GENERAL_GV, 82
 - INSTR_GV, 82
 - MENU_GV, 82
 - PLAYWA_GV, 82
 - USR_GV, 82
 - update_graphical_values, 82
- graphical_values.h, 83
 - display_info, 84
 - update_graphical_values, 84
- HEAD_FONTSIZE
 - general_gv_t, 12
- HEAD_TEXT_SHADOW_GAP
 - general_gv_t, 12
- HEAD_TOP_OFFSET
 - general_gv_t, 12
- HEIGHT_BONUS
 - playwa_gv_t, 24
- HINT_FONTSIZE
 - playwa_gv_t, 24
- HINT_LEFT_OFFSET

- playwa_gv_t, 25
- HINT_TOP_OFFSET
 - playwa_gv_t, 25
- handler_key_pressed
 - game_functions.cpp, 73
 - game_functions.h, 78
- handler_timer
 - game_functions.cpp, 73
 - game_functions.h, 78
- height
 - display_info_t, 8
- INSTR_GV
 - drawing_functions.h, 70
 - graphical_values.cpp, 82
- INTERNAL_OFFSET
 - menu_gv_t, 21
- init.cpp, 84
 - init_arr_bullet, 85
 - init_bullet, 85
 - init_display, 85
 - init_display_res, 86
 - init_game_requirements, 86
 - init_match_vars, 86
 - init_queue_event, 86
 - init_settings, 86
 - init_shooter, 87
 - init_sound_music, 87
 - init_target_bonus, 87
 - init_thread_manager, 87
 - init_words_buffer, 88
 - sort_file_list, 88
- init.h, 88
 - display_info, 93
 - init_arr_bullet, 90
 - init_bullet, 90
 - init_display, 90
 - init_display_res, 90
 - init_game_requirements, 91
 - init_match_vars, 91
 - init_queue_event, 91
 - init_settings, 91
 - init_shooter, 91
 - init_sound_music, 92
 - init_target_bonus, 92
 - init_thread_manager, 92
 - init_words_buffer, 92
 - music_wa, 93
 - PLAYWA_GV, 93
- init_arr_bullet
 - init.cpp, 85
 - init.h, 90
- init_bullet
 - init.cpp, 85
 - init.h, 90
- init_display
 - init.cpp, 85
 - init.h, 90
- init_display_res
- init.cpp, 86
 - init.h, 90
- init_game_requirements
 - init.cpp, 86
 - init.h, 91
- init_match_vars
 - init.cpp, 86
 - init.h, 91
- init_queue_event
 - init.cpp, 86
 - init.h, 91
- init_settings
 - init.cpp, 86
 - init.h, 91
- init_shooter
 - init.cpp, 87
 - init.h, 91
- init_sound_music
 - init.cpp, 87
 - init.h, 92
- init_target_bonus
 - init.cpp, 87
 - init.h, 92
- init_thread_manager
 - init.cpp, 87
 - init.h, 92
- init_words_buffer
 - init.cpp, 88
 - init.h, 92
- insert_asteroid_index
 - match_vars_t, 18
- insert_elem_head
 - utilities.cpp, 102
 - utilities.h, 107
- instant_ast_dest
 - match_vars_t, 18
- instr_gv_t, 12
 - BONUS_CASE_WIDTH, 13
 - BONUS_RADIUS, 13
 - SUBTITLE_FONTSIZE, 13
 - TEXT_FONTSIZE, 13
- int_to_str_score
 - utilities.cpp, 103
 - utilities.h, 107
- intro
 - sections.cpp, 95
 - sections.h, 98
- is_searcher
 - look_into_t, 14
- items
 - match_vars_t, 19
- key_error
 - match_vars_t, 19
- load
 - words_buffer_t, 35
- look_into_t, 13
 - email, 14

- filename, [14](#)
- is_searcher, [14](#)
- ls_t, [15](#)
 - files, [15](#)
 - nfile, [15](#)
- ls_tar
 - utilities.cpp, [103](#)
 - utilities.h, [107](#)
- MAX_ASTEROID_HEAD
 - data.h, [43](#)
- MAX_DIGIT_SCORE
 - data.h, [44](#)
- MAX_LENGTH_WORD
 - data.h, [44](#)
- MAX_MATCHING_PASS
 - data.h, [44](#)
- MAX_USERNAME_LENGTH
 - data.h, [44](#)
- MENU_CPY_COLOR
 - data.h, [44](#)
- MENU_FONT_COLOR
 - data.h, [44](#)
- MENU_GV
 - drawing_functions.h, [70](#)
 - graphical_values.cpp, [82](#)
 - sections.h, [99](#)
- MIN_TIME_SPAWN
 - data.h, [45](#)
- MONITOR_OFFSET_RATE
 - data.h, [45](#)
- MUSIC_GAMEOVER_PATH_WA
 - data.h, [49](#)
- MUSIC_MENU_PATH_WA
 - data.h, [49](#)
- MUSIC_PLAYWA_PATH_WA
 - data.h, [49](#)
- main
 - main.cpp, [94](#)
- main.cpp, [93](#)
 - display_info, [94](#)
 - main, [94](#)
 - music_wa, [94](#)
 - thread_manager, [94](#)
 - words_buffer, [94](#)
- main_menu
 - sections.cpp, [95](#)
 - sections.h, [98](#)
- match_vars_t, [16](#)
 - ast_queues, [17](#)
 - asteroid_head, [17](#)
 - asteroid_to_bonus, [18](#)
 - asteroid_to_generate, [18](#)
 - asteroids_destroyed, [18](#)
 - current_asteroid, [18](#)
 - fps, [18](#)
 - gameover, [18](#)
 - insert_asteroid_index, [18](#)
 - instant_ast_dest, [18](#)
 - items, [19](#)
 - key_error, [19](#)
 - music_playwa, [19](#)
 - playing, [19](#)
 - score, [19](#)
 - sound_error, [19](#)
 - sound_exp, [19](#)
 - sound_fire, [19](#)
 - sound_gameover, [20](#)
 - spawn_interval, [20](#)
 - spawn_timer, [20](#)
 - timer_rallenty_bonus, [20](#)
 - timer_shield_bonus, [20](#)
- menu_gv_t, [20](#)
 - COPYRIGHT_FONTSIZE, [21](#)
 - COPYRIGHT_TOP_OFFSET, [21](#)
 - INTERNAL_OFFSET, [21](#)
 - OPTION_FONTSIZE, [21](#)
 - OPTION_TEXT_OFFSET, [22](#)
 - OPTION_H, [21](#)
 - OPTION_W, [22](#)
 - TITLE_H, [22](#)
 - TITLE_W, [22](#)
 - UPPER_OFFSET, [22](#)
- move_asteroids
 - game_functions.cpp, [74](#)
 - game_functions.h, [79](#)
- move_bullets
 - game_functions.cpp, [74](#)
 - game_functions.h, [79](#)
- move_shooter
 - game_functions.cpp, [74](#)
 - game_functions.h, [79](#)
- music
 - settings_t, [29](#)
- music_playwa
 - match_vars_t, [19](#)
- music_wa
 - game_functions.h, [80](#)
 - init.h, [93](#)
 - main.cpp, [94](#)
 - sections.h, [99](#)
- music_wa_play
 - game_functions.cpp, [74](#)
 - game_functions.h, [79](#)
- N_QUEUES
 - data.h, [49](#)
- NUM_INSTR_BITMAPS
 - data.h, [45](#)
- NUM_INSTR_FONTS
 - data.h, [45](#)
- NUM_ITEMS
 - data.h, [45](#)
- NUM_MENU_BITMAPS
 - data.h, [45](#)
- NUM_MENU_FONTS
 - data.h, [46](#)
- NUM_OPTIONS_SETTINGS

- data.h, [46](#)
- NUM_OPTIONS
 - data.h, [46](#)
- NUM_PLAYWA_BITMAPS
 - data.h, [46](#)
- NUM_PLAYWA_FONTS
 - data.h, [46](#)
- NUM_SETTINGS_FONTS
 - data.h, [46](#)
- NUM_USER_FONTS
 - data.h, [47](#)
- next
 - elem_asteroid_t, [9](#)
- nfile
 - ls_t, [15](#)
- OPTION_FONTSIZE
 - menu_gv_t, [21](#)
- OPTION_TEXT_OFFSET
 - menu_gv_t, [22](#)
- OPTION_H
 - menu_gv_t, [21](#)
- OPTION_W
 - menu_gv_t, [22](#)
- PLAYWA_GV
 - drawing_functions.h, [70](#)
 - game_functions.h, [80](#)
 - graphical_values.cpp, [82](#)
 - init.h, [93](#)
- pass_result
 - result_t, [26](#)
- play_wa
 - sections.cpp, [96](#)
 - sections.h, [98](#)
- playing
 - match_vars_t, [19](#)
- playwa_gv_t, [22](#)
 - ASTEROID_FONTSIZE, [23](#)
 - ASTEROID_SHADOW_GAP, [23](#)
 - ASTEROID_TEXT_SHADOW_GAP, [23](#)
 - BONUS_HINT_FONTSIZE, [23](#)
 - BONUS_LEFT_OFFSET, [24](#)
 - BONUS_NUM_FONTSIZE, [24](#)
 - BONUS_TOP_OFFSET, [24](#)
 - BULLET_RADIUS, [24](#)
 - BULLET_SCALE, [24](#)
 - DANGER_RADIUS, [24](#)
 - HEIGHT_BONUS, [24](#)
 - HINT_FONTSIZE, [24](#)
 - HINT_LEFT_OFFSET, [25](#)
 - HINT_TOP_OFFSET, [25](#)
 - SCORE_FONTSIZE, [25](#)
 - SCORE_LEFT_OFFSET, [25](#)
 - SCORE_TOP_OFFSET, [25](#)
 - SHOOTER_AREA_RAD, [25](#)
 - SHOOTER_SCALE, [25](#)
 - TEXT_INTERNAL_OFFSET, [25](#)
- RESULT_TOP_OFFSET
 - general_gv_t, [12](#)
- radius
 - asteroid_t, [6](#)
 - bullet_t, [7](#)
 - shooter_t, [30](#)
- res
 - search_manager_t, [28](#)
- result_t, [26](#)
 - count_result, [26](#)
 - pass_result, [26](#)
- SCORE_FONTSIZE
 - playwa_gv_t, [25](#)
- SCORE_INCREMENT
 - data.h, [47](#)
- SCORE_LEFT_OFFSET
 - playwa_gv_t, [25](#)
- SCORE_TOP_OFFSET
 - playwa_gv_t, [25](#)
- SHADOW_COLOR
 - data.h, [47](#)
- SHOOTER_AREA_RAD
 - playwa_gv_t, [25](#)
- SHOOTER_SCALE
 - playwa_gv_t, [25](#)
- SOUND_PATH_ERROR
 - data.h, [49](#)
- SOUND_PATH_EXPLOSION
 - data.h, [49](#)
- SOUND_PATH_FIRE
 - data.h, [50](#)
- SOUND_PATH_GAMEOVER
 - data.h, [50](#)
- STATS_FONTSIZE
 - usr_gv_t, [34](#)
- SUBTITLE_FONTSIZE
 - instr_gv_t, [13](#)
- score
 - match_vars_t, [19](#)
- search_manager
 - thread_manager_t, [32](#)
- search_manager_t, [27](#)
 - email, [27](#)
 - file_list, [28](#)
 - res, [28](#)
 - threads, [28](#)
- search_result_mutex
 - thread_manager_t, [32](#)
- searcher_sem
 - thread_manager_t, [32](#)
- searcher_thread
 - thread_manager_t, [32](#)
- sections.cpp, [95](#)
 - intro, [95](#)
 - main_menu, [95](#)
 - play_wa, [96](#)
 - settings_menu, [96](#)
- sections.h, [97](#)

- display_info, 99
- intro, 98
- MENU_GV, 99
- main_menu, 98
- music_wa, 99
- play_wa, 98
- settings_menu, 98
- thread_manager, 99
- USR_GV, 99
- words_buffer, 100
- settings_menu
 - sections.cpp, 96
 - sections.h, 98
- settings_t, 28
 - music, 29
 - sound, 29
 - visible_hints, 29
- shooter_t, 29
 - radius, 30
 - slope, 30
 - x, 30
 - y, 30
- size
 - file_t, 10
- slope
 - asteroid_t, 6
 - bullet_t, 7
 - shooter_t, 30
- sort_file_list
 - init.cpp, 88
- sound
 - settings_t, 29
- sound_error
 - match_vars_t, 19
- sound_exp
 - match_vars_t, 19
- sound_fire
 - match_vars_t, 19
- sound_gameover
 - match_vars_t, 20
- sound_play
 - game_functions.cpp, 75
 - game_functions.h, 80
- spawn_asteroid_thread
 - game_functions.cpp, 75
 - game_functions.h, 80
- spawn_cond
 - thread_manager_t, 32
- spawn_interval
 - match_vars_t, 20
- spawn_mutex
 - thread_manager_t, 33
- spawn_timer
 - match_vars_t, 20
- spawner_thread
 - thread_manager_t, 33
- special
 - asteroid_t, 6
- start_searching
 - archive_functions.cpp, 37
 - archive_functions.h, 39
- TEXT_FONTSIZE
 - instr_gv_t, 13
- TEXT_INTERNAL_OFFSET
 - playwa_gv_t, 25
- TIMER_RALLENTY_BONUS
 - data.h, 47
- TIMER_SHIELD_BONUS
 - data.h, 47
- TITLE_H
 - menu_gv_t, 22
- TITLE_W
 - menu_gv_t, 22
- thread_manager
 - archive_functions.h, 39
 - game_functions.h, 81
 - main.cpp, 94
 - sections.h, 99
- thread_manager_t, 30
 - consumer_sem, 32
 - gameover_cond, 32
 - gameover_mutex, 32
 - search_manager, 32
 - search_result_mutex, 32
 - searcher_sem, 32
 - searcher_thread, 32
 - spawn_cond, 32
 - spawn_mutex, 33
 - spawner_thread, 33
 - words_buffer_mutex, 33
- threads
 - search_manager_t, 28
- timer_rallenty_bonus
 - match_vars_t, 20
- timer_shield_bonus
 - match_vars_t, 20
- UPPER_OFFSET
 - menu_gv_t, 22
- USERNAME_BLOCK_TOP_OFFSET
 - usr_gv_t, 34
- USERNAME_FONTSIZE
 - usr_gv_t, 34
- USR_GV
 - drawing_functions.h, 70
 - graphical_values.cpp, 82
 - sections.h, 99
- update_graphical_values
 - graphical_values.cpp, 82
 - graphical_values.h, 84
- usr_gv_t, 33
 - ENTER_FONTSIZE, 34
 - ENTER_HEIGHT, 34
 - ENTER_LEFT_OFFSET, 34
 - ENTER_TOP_OFFSET, 34
 - ENTER_WIDTH, 34

- STATS_FONTSIZE, [34](#)
 - USERNAME_BLOCK_TOP_OFFSET, [34](#)
 - USERNAME_FONTSIZE, [34](#)
- utilities.cpp, [100](#)
 - archives_exist, [101](#)
 - change_enable, [101](#)
 - delete_ast_queues, [101](#)
 - delete_display_evqueue_music, [101](#)
 - delete_sounds, [101](#)
 - delete_timer, [102](#)
 - exit, [102](#)
 - extract_elem, [102](#)
 - insert_elem_head, [102](#)
 - int_to_str_score, [103](#)
 - ls_tar, [103](#)
 - which_quadrant, [103](#)
- utilities.h, [104](#)
 - archives_exist, [105](#)
 - change_enable, [105](#)
 - delete_ast_queues, [105](#)
 - delete_display_evqueue_music, [106](#)
 - delete_sounds, [106](#)
 - delete_timer, [106](#)
 - display_info, [108](#)
 - exit, [106](#)
 - extract_elem, [107](#)
 - insert_elem_head, [107](#)
 - int_to_str_score, [107](#)
 - ls_tar, [107](#)
 - which_quadrant, [108](#)
- visible_hints
 - settings_t, [29](#)
- which_quadrant
 - utilities.cpp, [103](#)
 - utilities.h, [108](#)
- width
 - display_info_t, [8](#)
- word
 - asteroid_t, [6](#)
- words
 - words_buffer_t, [35](#)
- words_buffer
 - archive_functions.h, [39](#)
 - game_functions.h, [81](#)
 - main.cpp, [94](#)
 - sections.h, [100](#)
- words_buffer_mutex
 - thread_manager_t, [33](#)
- words_buffer_t, [35](#)
 - load, [35](#)
 - words, [35](#)
- x
 - asteroid_t, [6](#)
 - bullet_t, [7](#)
 - shooter_t, [30](#)
- y
 - asteroid_t, [6](#)
 - bullet_t, [7](#)
 - shooter_t, [30](#)