

like_ocas_in_the_rain

Generated by Doxygen 1.8.11

Contents

1	Test List	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	_Area Struct Reference	7
4.1.1	Member Data Documentation	7
4.1.1.1	cursor	7
4.1.1.2	height	7
4.1.1.3	width	7
4.1.1.4	x	7
4.1.1.5	y	7
4.2	_Die Struct Reference	8
4.2.1	Member Data Documentation	8
4.2.1.1	id	8
4.2.1.2	result	8
4.3	_F_Command Struct Reference	8
4.3.1	Member Data Documentation	8
4.3.1.1	id	8
4.3.1.2	text	8
4.4	_Game Struct Reference	9

4.4.1	Member Data Documentation	9
4.4.1.1	die	9
4.4.1.2	last_cmd	9
4.4.1.3	links	9
4.4.1.4	objects	9
4.4.1.5	player	9
4.4.1.6	spaces	9
4.4.1.7	sprites	10
4.5	_Graphic_engine Struct Reference	10
4.5.1	Member Data Documentation	10
4.5.1.1	banner	10
4.5.1.2	descript	10
4.5.1.3	feedback	10
4.5.1.4	help	10
4.5.1.5	map	11
4.6	_Inventory Struct Reference	11
4.6.1	Member Data Documentation	11
4.6.1.1	id_max	11
4.6.1.2	ids	11
4.7	_Link Struct Reference	11
4.7.1	Member Data Documentation	12
4.7.1.1	direction	12
4.7.1.2	door	12
4.7.1.3	linkId	12
4.7.1.4	linkspace1	12
4.7.1.5	linkspace2	12
4.8	_Object Struct Reference	12
4.8.1	Member Data Documentation	12
4.8.1.1	description	12
4.8.1.2	description_al	13

4.8.1.3	hidden	13
4.8.1.4	id	13
4.8.1.5	iluminati	13
4.8.1.6	mobile	13
4.8.1.7	moved	13
4.8.1.8	name	13
4.8.1.9	on	13
4.8.1.10	open	13
4.9	_Player Struct Reference	13
4.9.1	Member Data Documentation	14
4.9.1.1	id	14
4.9.1.2	inv	14
4.9.1.3	location_id	14
4.9.1.4	name	14
4.10	_Set Struct Reference	14
4.10.1	Member Data Documentation	14
4.10.1.1	id_list	14
4.10.1.2	id_total	15
4.11	_Space Struct Reference	15
4.11.1	Member Data Documentation	15
4.11.1.1	curentSprite	15
4.11.1.2	description	15
4.11.1.3	gdesc	15
4.11.1.4	id	16
4.11.1.5	light	16
4.11.1.6	linkDown	16
4.11.1.7	linkEast	16
4.11.1.8	linkNorth	16
4.11.1.9	linkSouth	16
4.11.1.10	linkUp	16
4.11.1.11	linkWest	16
4.11.1.12	name	16
4.11.1.13	objects	16
4.11.1.14	spriteld	17
4.12	_Sprite Struct Reference	17
4.12.1	Member Data Documentation	17
4.12.1.1	data	17
4.12.1.2	id	17

5	File Documentation	19
5.1	command.c File Reference	19
5.1.1	Detailed Description	20
5.1.2	Macro Definition Documentation	20
5.1.2.1	CMD_LENHT	20
5.1.2.2	N_CMD	20
5.1.3	Function Documentation	20
5.1.3.1	command_create()	20
5.1.3.2	command_free(F_Command *cmd)	20
5.1.3.3	command_get_id(F_Command *cmd)	20
5.1.3.4	command_getCmd(F_Command *cmd)	20
5.1.3.5	command_set_id(F_Command *cmd, char *string)	20
5.1.3.6	command_setCmd(F_Command *cmd, T_Command command)	20
5.1.3.7	get_user_input(F_Command *command)	20
5.1.4	Variable Documentation	20
5.1.4.1	cmd_to_str	21
5.1.4.2	short_cmd_to_str	21
5.2	command.h File Reference	21
5.2.1	Detailed Description	21
5.2.2	Typedef Documentation	22
5.2.2.1	F_Command	22
5.2.2.2	T_Command	22
5.2.3	Enumeration Type Documentation	22
5.2.3.1	enum_Command	22
5.2.4	Function Documentation	22
5.2.4.1	command_create()	22
5.2.4.2	command_free(F_Command *)	22
5.2.4.3	command_get_id(F_Command *)	22
5.2.4.4	command_getCmd(F_Command *)	22
5.2.4.5	command_set_id(F_Command *, char *)	22

5.2.4.6	<code>command_setCmd(F_Command *, T_Command)</code>	22
5.2.4.7	<code>get_user_input(F_Command *)</code>	22
5.3	<code>command_test.c</code> File Reference	22
5.3.1	Macro Definition Documentation	23
5.3.1.1	<code>MAX_TESTS</code>	23
5.3.2	Function Documentation	23
5.3.2.1	<code>main(int argc, char **argv)</code>	23
5.3.2.2	<code>test1_command_create()</code>	23
5.3.2.3	<code>test1_command_get_cmd()</code>	23
5.3.2.4	<code>test1_command_get_id()</code>	23
5.3.2.5	<code>test1_command_set_cmd()</code>	23
5.3.2.6	<code>test1_command_set_id()</code>	23
5.3.2.7	<code>test2_command_set_cmd()</code>	23
5.3.2.8	<code>test2_command_set_id()</code>	23
5.4	<code>command_test.h</code> File Reference	23
5.4.1	Detailed Description	24
5.4.2	Function Documentation	24
5.4.2.1	<code>test1_command_create()</code>	24
5.4.2.2	<code>test1_command_get_cmd()</code>	24
5.4.2.3	<code>test1_command_get_id()</code>	24
5.4.2.4	<code>test1_command_set_cmd()</code>	24
5.4.2.5	<code>test1_command_set_id()</code>	24
5.4.2.6	<code>test2_command_set_cmd()</code>	24
5.4.2.7	<code>test2_command_set_id()</code>	24
5.5	<code>die.c</code> File Reference	24
5.5.1	Detailed Description	25
5.5.2	Function Documentation	25
5.5.2.1	<code>die_die_die(Die *die)</code>	25
5.5.2.2	<code>die_get_last_roll(Die *die)</code>	25
5.5.2.3	<code>die_ini(Id id)</code>	25

5.5.2.4	<code>die_print(FILE *f, Die *die)</code>	26
5.5.2.5	<code>die_roll(Die *die)</code>	26
5.6	<code>die.h</code> File Reference	26
5.6.1	Detailed Description	26
5.6.2	Typedef Documentation	27
5.6.2.1	<code>Die</code>	27
5.6.3	Function Documentation	27
5.6.3.1	<code>die_die_die(Die *)</code>	27
5.6.3.2	<code>die_get_last_roll(Die *)</code>	27
5.6.3.3	<code>die_ini(Id)</code>	27
5.6.3.4	<code>die_print(FILE *, Die *)</code>	27
5.6.3.5	<code>die_roll(Die *)</code>	27
5.7	<code>die_test.c</code> File Reference	27
5.7.1	Macro Definition Documentation	28
5.7.1.1	<code>MAX_TESTS</code>	28
5.7.2	Function Documentation	28
5.7.2.1	<code>main(int argc, char **argv)</code>	28
5.7.2.2	<code>test1_die_create()</code>	28
5.7.2.3	<code>test1_die_get_last_roll()</code>	28
5.7.2.4	<code>test1_die_roll()</code>	28
5.7.2.5	<code>test2_die_get_last_roll()</code>	28
5.7.2.6	<code>test2_die_roll()</code>	28
5.8	<code>die_test.h</code> File Reference	28
5.8.1	Detailed Description	29
5.8.2	Function Documentation	29
5.8.2.1	<code>test1_die_create()</code>	29
5.8.2.2	<code>test1_die_get_last_roll()</code>	29
5.8.2.3	<code>test1_die_roll()</code>	29
5.8.2.4	<code>test2_die_get_last_roll()</code>	29
5.8.2.5	<code>test2_die_roll()</code>	29

5.9	game.c File Reference	29
5.9.1	Detailed Description	31
5.9.2	Macro Definition Documentation	31
5.9.2.1	DIE_SEED	31
5.9.2.2	N_CALLBACK	31
5.9.2.3	NO_LIGHT_SPRITE	31
5.9.2.4	PLAYER_ID	31
5.9.2.5	STARTING_SPACE	31
5.9.3	Typedef Documentation	31
5.9.3.1	callback_fn	31
5.9.4	Function Documentation	31
5.9.4.1	game_add_object(Game *game, Object *object)	31
5.9.4.2	game_add_space(Game *game, Space *space)	31
5.9.4.3	game_add_sprite(Game *game, Sprite *sprite, int i)	31
5.9.4.4	game_areSpacesAdjacent(Game *g, Id space1, Id space2)	31
5.9.4.5	game_callback_check(Game *game)	31
5.9.4.6	game_callback_drop(Game *game)	31
5.9.4.7	game_callback_exit(Game *game)	31
5.9.4.8	game_callback_move(Game *game)	32
5.9.4.9	game_callback_open(Game *game)	32
5.9.4.10	game_callback_pickup(Game *game)	32
5.9.4.11	game_callback_roll(Game *game)	32
5.9.4.12	game_callback_turnOff(Game *game)	32
5.9.4.13	game_callback_turnOn(Game *game)	32
5.9.4.14	game_callback_unknown(Game *game)	32
5.9.4.15	game_create()	32
5.9.4.16	game_create_from_file(Game *game, char *filename)	32
5.9.4.17	game_destroy(Game *game)	32
5.9.4.18	game_get_last_command(Game *game)	32
5.9.4.19	game_get_last_command_text(Game *game)	32

5.9.4.20	<code>game_get_last_roll(Game *game)</code>	32
5.9.4.21	<code>game_get_link(Game *game, Id id)</code>	32
5.9.4.22	<code>game_get_link_id_at(Game *game, int pos)</code>	32
5.9.4.23	<code>game_get_object(Game *game, char *object_name)</code>	32
5.9.4.24	<code>game_get_object_from_id(Game *game, Id id)</code>	32
5.9.4.25	<code>game_get_object_location(Game *game, Id id)</code>	32
5.9.4.26	<code>game_get_player(Game *game)</code>	32
5.9.4.27	<code>game_get_player_location(Game *game)</code>	32
5.9.4.28	<code>game_get_space(Game *game, Id id)</code>	32
5.9.4.29	<code>game_get_space_id_at(Game *game, int position)</code>	32
5.9.4.30	<code>game_get_sprite(Game *game, Id id)</code>	33
5.9.4.31	<code>game_is_over(Game *game)</code>	33
5.9.4.32	<code>game_print_opened_links(Game *game)</code>	33
5.9.4.33	<code>game_set_link(Game *game, Id link_id, Id space_id0, Id space_id1, int direction, LinkStatus door)</code>	33
5.9.4.34	<code>game_set_object_location(Game *game, Id id, Id obj_id, char *name, char *description)</code>	33
5.9.4.35	<code>game_set_player_location(Game *game, Id id)</code>	33
5.9.4.36	<code>game_update(Game *game, F_Command *cmd)</code>	33
5.9.4.37	<code>update_sprites(Game *game)</code>	33
5.9.5	Variable Documentation	33
5.9.5.1	<code>game_callback_fn_list</code>	33
5.10	game.h File Reference	33
5.10.1	Detailed Description	34
5.10.2	Macro Definition Documentation	35
5.10.2.1	<code>MAX_OBJECTS</code>	35
5.10.3	Typedef Documentation	35
5.10.3.1	<code>Game</code>	35
5.10.4	Function Documentation	35
5.10.4.1	<code>game_add_object(Game *game, Object *object)</code>	35
5.10.4.2	<code>game_add_space(Game *, Space *)</code>	35
5.10.4.3	<code>game_add_sprite(Game *game, Sprite *sprite, int i)</code>	35

5.10.4.4	<code>game_areSpacesAdjacent(Game *, Id, Id)</code>	35
5.10.4.5	<code>game_create()</code>	35
5.10.4.6	<code>game_create_from_file(Game *, char *)</code>	35
5.10.4.7	<code>game_destroy(Game *)</code>	35
5.10.4.8	<code>game_get_last_command(Game *)</code>	35
5.10.4.9	<code>game_get_last_command_text(Game *)</code>	35
5.10.4.10	<code>game_get_last_roll(Game *)</code>	35
5.10.4.11	<code>game_get_link(Game *, Id)</code>	35
5.10.4.12	<code>game_get_link_id_at(Game *, int)</code>	35
5.10.4.13	<code>game_get_object(Game *, char *)</code>	35
5.10.4.14	<code>game_get_object_from_id(Game *game, Id id)</code>	35
5.10.4.15	<code>game_get_object_location(Game *, Id)</code>	35
5.10.4.16	<code>game_get_player(Game *)</code>	35
5.10.4.17	<code>game_get_player_location(Game *)</code>	35
5.10.4.18	<code>game_get_space(Game *, Id)</code>	35
5.10.4.19	<code>game_get_space_id_at(Game *, int)</code>	36
5.10.4.20	<code>game_get_sprite(Game *game, Id id)</code>	36
5.10.4.21	<code>game_is_over(Game *)</code>	36
5.10.4.22	<code>game_load_spaces(Game *, char *)</code>	36
5.10.4.23	<code>game_print_data(Game *)</code>	36
5.10.4.24	<code>game_print_screen(Game *)</code>	36
5.10.4.25	<code>game_set_link(Game *, Id, Id, Id, int, LinkStatus)</code>	36
5.10.4.26	<code>game_set_object_location(Game *, Id, Id, char *, char *)</code>	36
5.10.4.27	<code>game_set_player_location(Game *, Id)</code>	36
5.10.4.28	<code>game_update(Game *, F_Command *)</code>	36
5.10.4.29	<code>update_sprites(Game *game)</code>	36
5.11	<code>game_loop.c</code> File Reference	36
5.11.1	Detailed Description	36
5.11.2	Function Documentation	37
5.11.2.1	<code>main(int argc, char *argv[])</code>	37

5.12	game_reader.c File Reference	37
5.12.1	Detailed Description	37
5.12.2	Function Documentation	37
5.12.2.1	game_load_spaces(Game *game, char *filename)	37
5.13	game_reader.h File Reference	37
5.13.1	Detailed Description	38
5.13.2	Function Documentation	38
5.13.2.1	game_load_links(Game *game, char *filename)	38
5.13.2.2	game_load_spaces(Game *game, char *filename)	38
5.14	game_rules.c File Reference	38
5.14.1	Function Documentation	38
5.14.1.1	update_rules()	38
5.15	game_rules.h File Reference	38
5.16	graphic_engine.c File Reference	39
5.16.1	Detailed Description	39
5.16.2	Macro Definition Documentation	40
5.16.2.1	STD_SPACE	40
5.16.2.2	STD_SPACE1	40
5.16.3	Function Documentation	40
5.16.3.1	create_objects_string(Game *game, Id id)	40
5.16.3.2	graphic_engine_create()	40
5.16.3.3	graphic_engine_destroy(Graphic_engine *ge)	40
5.16.3.4	graphic_engine_paint_game(Graphic_engine *ge, Game *game)	40
5.16.3.5	print_new_line(Area *area, int number)	40
5.17	graphic_engine.h File Reference	40
5.17.1	Detailed Description	41
5.17.2	Typedef Documentation	41
5.17.2.1	Graphic_engine	41
5.17.3	Function Documentation	41
5.17.3.1	create_objects_string(Game *, Id)	41

5.17.3.2	graphic_engine_create()	41
5.17.3.3	graphic_engine_destroy(Graphic_engine *)	41
5.17.3.4	graphic_engine_paint_game(Graphic_engine *, Game *)	41
5.17.3.5	graphic_engine_paint_space(Graphic_engine *, Game *, int)	41
5.17.3.6	print_new_line(Area *, int number)	41
5.18	inventory.c File Reference	41
5.18.1	Detailed Description	42
5.18.2	Function Documentation	42
5.18.2.1	inventory_add_id(Inventory *inv, Id id)	42
5.18.2.2	inventory_create(int size)	42
5.18.2.3	inventory_del_id(Inventory *inv, Id id)	42
5.18.2.4	inventory_destroy(Inventory *inv)	42
5.18.2.5	inventory_get_id_at(Inventory *inv, int num)	42
5.18.2.6	inventory_get_id_max(Inventory *inv)	42
5.18.2.7	inventory_get_ids(Inventory *inv)	42
5.18.2.8	inventory_print(Inventory *inv)	42
5.18.2.9	inventory_set_id_max(Inventory *inv, int id_max)	42
5.18.2.10	inventory_set_ids(Inventory *inv, Set *ids)	42
5.19	inventory.h File Reference	43
5.19.1	Detailed Description	43
5.19.2	Typedef Documentation	44
5.19.2.1	Inventory	44
5.19.3	Function Documentation	44
5.19.3.1	inventory_add_id(Inventory *inv, Id id)	44
5.19.3.2	inventory_create(int)	44
5.19.3.3	inventory_del_id(Inventory *inv, Id id)	44
5.19.3.4	inventory_destroy(Inventory *inv)	44
5.19.3.5	inventory_get_id_at(Inventory *inv, int num)	44
5.19.3.6	inventory_get_ids(Inventory *inv)	44
5.19.3.7	inventory_print(Inventory *)	44

5.19.3.8	inventory_set_ids(Inventory *inv, Set *ids)	44
5.20	inventory_test.c File Reference	44
5.20.1	Macro Definition Documentation	45
5.20.1.1	MAX_TESTS	45
5.20.2	Function Documentation	45
5.20.2.1	main(int argc, char **argv)	45
5.20.2.2	test1_inventory_add_id()	45
5.20.2.3	test1_inventory_create()	45
5.20.2.4	test1_inventory_del_id()	45
5.20.2.5	test1_inventory_get_id_at()	45
5.20.2.6	test1_inventory_get_ids()	45
5.20.2.7	test1_inventory_set_ids()	45
5.20.2.8	test2_inventory_add_id()	45
5.20.2.9	test2_inventory_del_id()	45
5.20.2.10	test2_inventory_get_id_at()	45
5.21	inventory_test.h File Reference	45
5.21.1	Detailed Description	46
5.21.2	Function Documentation	46
5.21.2.1	test1_inventory_add_id()	46
5.21.2.2	test1_inventory_create()	46
5.21.2.3	test1_inventory_del_id()	46
5.21.2.4	test1_inventory_get_id_at()	46
5.21.2.5	test1_inventory_get_ids()	46
5.21.2.6	test1_inventory_get_max()	46
5.21.2.7	test1_inventory_is_empty()	46
5.21.2.8	test1_inventory_is_full()	46
5.21.2.9	test1_inventory_set_ids()	47
5.21.2.10	test1_inventory_set_max()	47
5.21.2.11	test2_inventory_add_id()	47
5.21.2.12	test2_inventory_del_id()	47

5.21.2.13 test2_inventory_get_id_at()	47
5.21.2.14 test2_inventory_is_empty()	47
5.21.2.15 test2_inventory_is_full()	47
5.21.2.16 test2_inventory_set_max()	47
5.22 link.c File Reference	47
5.22.1 Detailed Description	48
5.22.2 Function Documentation	48
5.22.2.1 link_create(Id id)	48
5.22.2.2 link_destroy(Link *l)	48
5.22.2.3 link_getDestination(Link *l, Id originId)	48
5.22.2.4 link_getDirection(Link *l)	48
5.22.2.5 link_getId(Link *l)	48
5.22.2.6 link_getSpace1(Link *l)	48
5.22.2.7 link_getSpace2(Link *l)	48
5.22.2.8 link_getStatus(Link *l)	48
5.22.2.9 link_print(Link *link)	48
5.22.2.10 link_setDirection(Link *l, int direction)	48
5.22.2.11 link_setId(Link *l, Id id)	48
5.22.2.12 link_setSpaces(Link *l, Id space1, Id space2)	48
5.22.2.13 link_setStatus(Link *l, LinkStatus door)	48
5.23 link.h File Reference	48
5.23.1 Detailed Description	49
5.23.2 Macro Definition Documentation	49
5.23.2.1 MAX_LINK	49
5.23.3 Typedef Documentation	49
5.23.3.1 Link	49
5.23.4 Function Documentation	49
5.23.4.1 link_create(Id)	49
5.23.4.2 link_destroy(Link *)	50
5.23.4.3 link_getDestination(Link *, Id)	50

5.23.4.4	<code>link_getDirection(Link *)</code>	50
5.23.4.5	<code>link_getId(Link *)</code>	50
5.23.4.6	<code>link_getSpace1(Link *)</code>	50
5.23.4.7	<code>link_getSpace2(Link *)</code>	50
5.23.4.8	<code>link_getStatus(Link *)</code>	50
5.23.4.9	<code>link_print(Link *)</code>	50
5.23.4.10	<code>link_setDirection(Link *, int)</code>	50
5.23.4.11	<code>link_setId(Link *, Id)</code>	50
5.23.4.12	<code>link_setSpaces(Link *, Id, Id)</code>	50
5.23.4.13	<code>link_setStatus(Link *, LinkStatus)</code>	50
5.24	<code>link_test.c</code> File Reference	50
5.24.1	Macro Definition Documentation	51
5.24.1.1	<code>MAX_TESTS</code>	51
5.24.2	Function Documentation	51
5.24.2.1	<code>main(int argc, char **argv)</code>	51
5.24.2.2	<code>test1_link_create()</code>	51
5.24.2.3	<code>test1_link_get_destination()</code>	51
5.24.2.4	<code>test1_link_get_id()</code>	51
5.24.2.5	<code>test1_link_get_space1()</code>	51
5.24.2.6	<code>test1_link_get_space2()</code>	51
5.24.2.7	<code>test1_link_get_status()</code>	51
5.24.2.8	<code>test1_link_set_id()</code>	51
5.24.2.9	<code>test1_link_set_spaces()</code>	51
5.24.2.10	<code>test1_link_set_status()</code>	52
5.24.2.11	<code>test2_link_create()</code>	52
5.24.2.12	<code>test2_link_set_id()</code>	52
5.24.2.13	<code>test2_link_set_spaces()</code>	52
5.24.2.14	<code>test2_link_set_status()</code>	52
5.25	<code>link_test.h</code> File Reference	52
5.25.1	Detailed Description	52

5.25.2	Function Documentation	53
5.25.2.1	test1_link_create()	53
5.25.2.2	test1_link_get_destination()	53
5.25.2.3	test1_link_get_id()	53
5.25.2.4	test1_link_get_space1()	53
5.25.2.5	test1_link_get_space2()	53
5.25.2.6	test1_link_get_status()	53
5.25.2.7	test1_link_set_id()	53
5.25.2.8	test1_link_set_spaces()	53
5.25.2.9	test1_link_set_status()	53
5.25.2.10	test2_link_create()	53
5.25.2.11	test2_link_set_id()	53
5.25.2.12	test2_link_set_spaces()	53
5.25.2.13	test2_link_set_status()	53
5.26	object.c File Reference	53
5.26.1	Detailed Description	54
5.26.2	Function Documentation	54
5.26.2.1	object_create(char *name, Id id, BOOL mobile, BOOL hidden, Id open, BOOL lights, BOOL on)	54
5.26.2.2	object_description_al_print(Object *obj, FILE *f)	54
5.26.2.3	object_description_print(Object *obj, FILE *f)	54
5.26.2.4	object_destroy(Object *obj)	54
5.26.2.5	object_get_description(Object *obj)	55
5.26.2.6	object_get_description_alternative(Object *obj)	55
5.26.2.7	object_get_hidden(Object *obj)	55
5.26.2.8	object_get_id(Object *obj)	55
5.26.2.9	object_get_iluminati(Object *obj)	55
5.26.2.10	object_get_mobile(Object *obj)	55
5.26.2.11	object_get_moved(Object *obj)	55
5.26.2.12	object_get_name(Object *obj)	55
5.26.2.13	object_get_on(Object *obj)	55

5.26.2.14	<code>object_get_open(Object *obj)</code>	55
5.26.2.15	<code>object_set_description(Object *obj, char *description)</code>	55
5.26.2.16	<code>object_set_description_alternative(Object *obj, char *description_al)</code>	55
5.26.2.17	<code>object_set_hidden(Object *obj, BOOL hidden)</code>	55
5.26.2.18	<code>object_set_id(Object *obj, Id id)</code>	55
5.26.2.19	<code>object_set_iluminati(Object *obj, BOOL iluminati)</code>	55
5.26.2.20	<code>object_set_mobile(Object *obj, BOOL mobile)</code>	55
5.26.2.21	<code>object_set_moved(Object *obj, BOOL moved)</code>	55
5.26.2.22	<code>object_set_name(Object *obj, char *name)</code>	55
5.26.2.23	<code>object_set_on(Object *obj, BOOL on)</code>	55
5.26.2.24	<code>object_set_open(Object *obj, Id open)</code>	55
5.27	<code>object.h</code> File Reference	55
5.27.1	Detailed Description	56
5.27.2	Typedef Documentation	57
5.27.2.1	Object	57
5.27.3	Function Documentation	57
5.27.3.1	<code>object_create(char *name, Id id, BOOL mobile, BOOL hidden, Id open, BOOL lights, BOOL on)</code>	57
5.27.3.2	<code>object_description_al_print(Object *obj, FILE *f)</code>	57
5.27.3.3	<code>object_description_print(Object *obj, FILE *f)</code>	57
5.27.3.4	<code>object_destroy(Object *obj)</code>	57
5.27.3.5	<code>object_get_description(Object *obj)</code>	57
5.27.3.6	<code>object_get_description_alternative(Object *obj)</code>	57
5.27.3.7	<code>object_get_hidden(Object *obj)</code>	57
5.27.3.8	<code>object_get_id(Object *obj)</code>	57
5.27.3.9	<code>object_get_iluminati(Object *obj)</code>	57
5.27.3.10	<code>object_get_mobile(Object *obj)</code>	57
5.27.3.11	<code>object_get_moved(Object *obj)</code>	57
5.27.3.12	<code>object_get_name(Object *obj)</code>	57
5.27.3.13	<code>object_get_on(Object *obj)</code>	57
5.27.3.14	<code>object_get_open(Object *obj)</code>	57

5.27.3.15	object_set_description(Object *obj, char *descript)	57
5.27.3.16	object_set_description_alternative(Object *obj, char *description_al)	57
5.27.3.17	object_set_hidden(Object *obj, BOOL hidden)	57
5.27.3.18	object_set_id(Object *obj, Id id)	57
5.27.3.19	object_set_ilumnati(Object *obj, BOOL iluminati)	57
5.27.3.20	object_set_mobile(Object *obj, BOOL mobile)	57
5.27.3.21	object_set_moved(Object *obj, BOOL moved)	58
5.27.3.22	object_set_name(Object *obj, char *name)	58
5.27.3.23	object_set_on(Object *obj, BOOL on)	58
5.27.3.24	object_set_open(Object *obj, Id open)	58
5.28	object_test.c File Reference	58
5.28.1	Macro Definition Documentation	58
5.28.1.1	MAX_TESTS	58
5.28.2	Function Documentation	58
5.28.2.1	main(int argc, char **argv)	58
5.28.2.2	test1_object_create()	59
5.28.2.3	test1_object_get_description()	59
5.28.2.4	test1_object_get_id()	59
5.28.2.5	test1_object_get_name()	59
5.28.2.6	test1_object_set_description()	59
5.28.2.7	test1_object_set_id()	59
5.28.2.8	test1_object_set_name()	59
5.28.2.9	test2_object_create()	59
5.28.2.10	test2_object_set_description()	59
5.28.2.11	test2_object_set_id()	59
5.28.2.12	test2_object_set_name()	59
5.29	object_test.h File Reference	59
5.29.1	Detailed Description	60
5.29.2	Function Documentation	60
5.29.2.1	test1_object_create()	60

5.29.2.2	test1_object_get_description()	60
5.29.2.3	test1_object_get_id()	60
5.29.2.4	test1_object_get_name()	60
5.29.2.5	test1_object_set_description()	60
5.29.2.6	test1_object_set_id()	60
5.29.2.7	test1_object_set_name()	60
5.29.2.8	test2_object_create()	60
5.29.2.9	test2_object_set_description()	60
5.29.2.10	test2_object_set_id()	60
5.29.2.11	test2_object_set_name()	60
5.30	Objectives_Tracker.txt File Reference	60
5.30.1	Variable Documentation	61
5.30.1.1	POINTS	61
5.31	player.c File Reference	61
5.31.1	Detailed Description	61
5.31.2	Function Documentation	62
5.31.2.1	player_create(char *name, Id location_id, Id object_id, Id id)	62
5.31.2.2	player_destroy(Player *player)	62
5.31.2.3	player_getId(Player *player)	62
5.31.2.4	player_getLocId(Player *player)	62
5.31.2.5	player_getName(Player *player)	62
5.31.2.6	player_getObjId(Player *player, int num)	62
5.31.2.7	player_removeObjId(Player *player, Id id)	62
5.31.2.8	player_setId(Player *player, Id new_id)	62
5.31.2.9	player_setLocId(Player *player, Id new_locId)	62
5.31.2.10	player_setName(Player *player, char *newName)	62
5.31.2.11	player_setObjId(Player *player, Id new_objId)	62
5.32	player.h File Reference	62
5.32.1	Detailed Description	63
5.32.2	Typedef Documentation	63

5.32.2.1	Player	63
5.32.3	Function Documentation	63
5.32.3.1	player_create(char *, Id, Id, Id)	63
5.32.3.2	player_destroy(Player *)	64
5.32.3.3	player_getId(Player *)	64
5.32.3.4	player_getLocId(Player *)	64
5.32.3.5	player_getName(Player *)	64
5.32.3.6	player_getObjId(Player *, int)	64
5.32.3.7	player_removeObjId(Player *, Id)	64
5.32.3.8	player_setId(Player *, Id)	64
5.32.3.9	player_setLocId(Player *, Id)	64
5.32.3.10	player_setName(Player *, char *)	64
5.32.3.11	player_setObjId(Player *, Id)	64
5.33	player_test.c File Reference	64
5.33.1	Macro Definition Documentation	65
5.33.1.1	MAX_TESTS	65
5.33.2	Function Documentation	65
5.33.2.1	main(int argc, char **argv)	65
5.33.2.2	test1_player_create()	65
5.33.2.3	test1_player_get_id()	65
5.33.2.4	test1_player_get_LocId()	65
5.33.2.5	test1_player_get_name()	65
5.33.2.6	test1_player_get_ObjId()	65
5.33.2.7	test1_player_remove_object_id()	65
5.33.2.8	test1_player_set_id()	65
5.33.2.9	test1_player_set_LocId()	65
5.33.2.10	test1_player_set_name()	65
5.33.2.11	test1_player_set_ObjId()	65
5.33.2.12	test2_player_remove_object_id()	65
5.33.2.13	test2_player_set_id()	65

5.33.2.14 test2_player_set_LocId()	65
5.33.2.15 test2_player_set_name()	65
5.33.2.16 test2_player_set_ObjId()	65
5.34 player_test.h File Reference	65
5.34.1 Detailed Description	66
5.34.2 Function Documentation	66
5.34.2.1 test1_player_create()	66
5.34.2.2 test1_player_get_id()	66
5.34.2.3 test1_player_get_LocId()	66
5.34.2.4 test1_player_get_name()	66
5.34.2.5 test1_player_get_ObjId()	66
5.34.2.6 test1_player_remove_object_id()	66
5.34.2.7 test1_player_set_id()	66
5.34.2.8 test1_player_set_LocId()	66
5.34.2.9 test1_player_set_name()	67
5.34.2.10 test1_player_set_ObjId()	67
5.34.2.11 test2_player_remove_object_id()	67
5.34.2.12 test2_player_set_id()	67
5.34.2.13 test2_player_set_LocId()	67
5.34.2.14 test2_player_set_name()	67
5.34.2.15 test2_player_set_ObjId()	67
5.35 screen.c File Reference	67
5.35.1 Detailed Description	68
5.35.2 Macro Definition Documentation	68
5.35.2.1 ACCESS	68
5.35.2.2 BG_CHAR	68
5.35.2.3 COLUMNS	68
5.35.2.4 FG_CHAR	68
5.35.2.5 PROMPT	68
5.35.2.6 ROWS	68

5.35.2.7	TOTAL_DATA	68
5.35.3	Function Documentation	68
5.35.3.1	screen_area_clear(Area *area)	68
5.35.3.2	screen_area_cursor_is_out_of_bounds(Area *area)	68
5.35.3.3	screen_area_destroy(Area *area)	68
5.35.3.4	screen_area_init(int x, int y, int width, int height)	68
5.35.3.5	screen_area_puts(Area *area, char *str)	68
5.35.3.6	screen_area_reset_cursor(Area *area)	68
5.35.3.7	screen_area_scroll_up(Area *area)	69
5.35.3.8	screen_destroy()	69
5.35.3.9	screen_gets(char *str)	69
5.35.3.10	screen_init()	69
5.35.3.11	screen_paint()	69
5.35.3.12	screen_utils_replaces_special_chars(char *str)	69
5.35.4	Variable Documentation	69
5.35.4.1	__data	69
5.36	screen.h File Reference	69
5.36.1	Detailed Description	70
5.36.2	Macro Definition Documentation	70
5.36.2.1	SCREEN_MAX_STR	70
5.36.3	Typedef Documentation	70
5.36.3.1	Area	70
5.36.4	Function Documentation	70
5.36.4.1	screen_area_clear(Area *area)	70
5.36.4.2	screen_area_destroy(Area *area)	70
5.36.4.3	screen_area_init(int x, int y, int width, int height)	70
5.36.4.4	screen_area_puts(Area *area, char *str)	70
5.36.4.5	screen_area_reset_cursor(Area *area)	70
5.36.4.6	screen_destroy()	70
5.36.4.7	screen_gets(char *str)	70

5.36.4.8	<code>screen_init()</code>	70
5.36.4.9	<code>screen_paint()</code>	70
5.37	set.c File Reference	70
5.37.1	Detailed Description	71
5.37.2	Function Documentation	71
5.37.2.1	<code>set_add(Set *set, Id id)</code>	71
5.37.2.2	<code>set_cp_all(Set *set)</code>	71
5.37.2.3	<code>set_create(int inv_size)</code>	71
5.37.2.4	<code>set_del(Set *set, Id id)</code>	71
5.37.2.5	<code>set_destroy(Set *set)</code>	71
5.37.2.6	<code>set_get_id(Set *set, int num)</code>	71
5.37.2.7	<code>set_print_debug(FILE *f, Set *set)</code>	71
5.37.2.8	<code>set_rearrange(Set *set)</code>	71
5.37.2.9	<code>set_rm_all(Set *set)</code>	71
5.38	set.h File Reference	72
5.38.1	Detailed Description	72
5.38.2	Macro Definition Documentation	73
5.38.2.1	<code>MAX_INV_SIZE</code>	73
5.38.3	Typedef Documentation	73
5.38.3.1	<code>Set</code>	73
5.38.4	Function Documentation	73
5.38.4.1	<code>set_add(Set *, Id)</code>	73
5.38.4.2	<code>set_cp_all(Set *)</code>	73
5.38.4.3	<code>set_create(int)</code>	73
5.38.4.4	<code>set_del(Set *, Id)</code>	73
5.38.4.5	<code>set_destroy(Set *)</code>	73
5.38.4.6	<code>set_get_id(Set *, int)</code>	73
5.38.4.7	<code>set_print_debug(FILE *, Set *)</code>	73
5.38.4.8	<code>set_rearrange(Set *)</code>	73
5.38.4.9	<code>set_rm_all(Set *)</code>	73

5.39 set_test.c File Reference	73
5.39.1 Macro Definition Documentation	74
5.39.1.1 MAX_TESTS	74
5.39.2 Function Documentation	74
5.39.2.1 main(int argc, char **argv)	74
5.39.2.2 test1_set_add()	74
5.39.2.3 test1_set_cp_all()	74
5.39.2.4 test1_set_create()	74
5.39.2.5 test1_set_del()	74
5.39.2.6 test1_set_get_id()	74
5.39.2.7 test1_set_rearrange()	74
5.39.2.8 test1_set_rm_all()	74
5.39.2.9 test2_set_add()	74
5.39.2.10 test2_set_cp_all()	75
5.39.2.11 test2_set_del()	75
5.39.2.12 test2_set_get_id()	75
5.39.2.13 test2_set_rearrange()	75
5.39.2.14 test2_set_rm_all()	75
5.40 set_test.h File Reference	75
5.40.1 Detailed Description	75
5.40.2 Function Documentation	76
5.40.2.1 test1_set_add()	76
5.40.2.2 test1_set_cp_all()	76
5.40.2.3 test1_set_create()	76
5.40.2.4 test1_set_del()	76
5.40.2.5 test1_set_get_id()	76
5.40.2.6 test1_set_rearrange()	76
5.40.2.7 test1_set_rm_all()	76
5.40.2.8 test2_set_add()	76
5.40.2.9 test2_set_cp_all()	76

5.40.2.10 test2_set_del()	76
5.40.2.11 test2_set_get_id()	76
5.40.2.12 test2_set_rearrange()	76
5.40.2.13 test2_set_rm_all()	76
5.41 space.c File Reference	76
5.41.1 Function Documentation	77
5.41.1.1 space_add_object(Space *space, Id obj_id)	77
5.41.1.2 space_create(Id id)	77
5.41.1.3 space_destroy(Space *space)	77
5.41.1.4 space_get_description(Space *space)	77
5.41.1.5 space_get_down(Space *space)	77
5.41.1.6 space_get_east(Space *space)	78
5.41.1.7 space_get_gdesc_0(Space *space)	78
5.41.1.8 space_get_gdesc_1(Space *space)	78
5.41.1.9 space_get_gdesc_2(Space *space)	78
5.41.1.10 space_get_id(Space *space)	78
5.41.1.11 space_get_light(Space *space)	78
5.41.1.12 space_get_name(Space *space)	78
5.41.1.13 space_get_north(Space *space)	78
5.41.1.14 space_get_objects_id(Space *space)	78
5.41.1.15 space_get_south(Space *space)	78
5.41.1.16 space_get_up(Space *space)	78
5.41.1.17 space_get_west(Space *space)	78
5.41.1.18 space_getCurentSprite(Space *space)	78
5.41.1.19 space_getSprite(Space *space, int i)	78
5.41.1.20 space_light_print(Space *space)	78
5.41.1.21 space_print(Space *space)	78
5.41.1.22 space_remove_object(Space *space, Id id)	78
5.41.1.23 space_set_description(Space *space, char *description)	78
5.41.1.24 space_set_down(Space *space, Id id)	78

5.41.1.25	<code>space_set_east(Space *space, Id id)</code>	78
5.41.1.26	<code>space_set_gdesc_0(Space *space, char *cadena)</code>	78
5.41.1.27	<code>space_set_gdesc_1(Space *space, char *cadena)</code>	78
5.41.1.28	<code>space_set_gdesc_2(Space *space, char *cadena)</code>	78
5.41.1.29	<code>space_set_light(Space *space, BOOL light)</code>	79
5.41.1.30	<code>space_set_name(Space *space, char *name)</code>	79
5.41.1.31	<code>space_set_north(Space *space, Id id)</code>	79
5.41.1.32	<code>space_set_south(Space *space, Id id)</code>	79
5.41.1.33	<code>space_set_up(Space *space, Id id)</code>	79
5.41.1.34	<code>space_set_west(Space *space, Id id)</code>	79
5.41.1.35	<code>space_setCurrentSprite(Space *space, int i)</code>	79
5.41.1.36	<code>space_setSprite(Space *space, Id spriteId, int i)</code>	79
5.42	<code>space.h</code> File Reference	79
5.42.1	Detailed Description	80
5.42.2	Macro Definition Documentation	81
5.42.2.1	<code>FIRST_SPACE</code>	81
5.42.2.2	<code>MAX_SPACES</code>	81
5.42.3	Typedef Documentation	81
5.42.3.1	<code>Space</code>	81
5.42.4	Function Documentation	81
5.42.4.1	<code>space_add_object(Space *space, Id obj_id)</code>	81
5.42.4.2	<code>space_create(Id id)</code>	81
5.42.4.3	<code>space_destroy(Space *space)</code>	81
5.42.4.4	<code>space_get_description(Space *space)</code>	81
5.42.4.5	<code>space_get_down(Space *space)</code>	81
5.42.4.6	<code>space_get_east(Space *space)</code>	81
5.42.4.7	<code>space_get_gdesc_0(Space *)</code>	81
5.42.4.8	<code>space_get_gdesc_1(Space *)</code>	81
5.42.4.9	<code>space_get_gdesc_2(Space *)</code>	81
5.42.4.10	<code>space_get_id(Space *space)</code>	81

5.42.4.11 <code>space_get_light(Space *space)</code>	81
5.42.4.12 <code>space_get_name(Space *space)</code>	81
5.42.4.13 <code>space_get_north(Space *space)</code>	81
5.42.4.14 <code>space_get_objects_id(Space *space)</code>	81
5.42.4.15 <code>space_get_south(Space *space)</code>	81
5.42.4.16 <code>space_get_up(Space *space)</code>	81
5.42.4.17 <code>space_get_west(Space *space)</code>	81
5.42.4.18 <code>space_getCurentSprite(Space *space)</code>	82
5.42.4.19 <code>space_getSprite(Space *, int)</code>	82
5.42.4.20 <code>space_light_print(Space *space)</code>	82
5.42.4.21 <code>space_print(Space *space)</code>	82
5.42.4.22 <code>space_remove_object(Space *space, Id obj_id)</code>	82
5.42.4.23 <code>space_set_description(Space *space, char *description)</code>	82
5.42.4.24 <code>space_set_down(Space *space, Id id)</code>	82
5.42.4.25 <code>space_set_east(Space *space, Id id)</code>	82
5.42.4.26 <code>space_set_gdesc_0(Space *, char *)</code>	82
5.42.4.27 <code>space_set_gdesc_1(Space *, char *)</code>	82
5.42.4.28 <code>space_set_gdesc_2(Space *, char *)</code>	82
5.42.4.29 <code>space_set_light(Space *space, BOOL light)</code>	82
5.42.4.30 <code>space_set_name(Space *space, char *name)</code>	82
5.42.4.31 <code>space_set_north(Space *space, Id id)</code>	82
5.42.4.32 <code>space_set_south(Space *space, Id id)</code>	82
5.42.4.33 <code>space_set_up(Space *space, Id id)</code>	82
5.42.4.34 <code>space_set_west(Space *space, Id id)</code>	82
5.42.4.35 <code>space_setCurrentSprite(Space *space, int i)</code>	82
5.42.4.36 <code>space_setSprite(Space *, Id, int)</code>	82
5.43 <code>space_test.c</code> File Reference	82
5.43.1 Detailed Description	83
5.43.2 Macro Definition Documentation	83
5.43.2.1 <code>MAX_TESTS</code>	83

5.43.3 Function Documentation	83
5.43.3.1 main(int argc, char **argv)	83
5.43.3.2 test1_space_create()	84
5.43.3.3 test1_space_get_east()	84
5.43.3.4 test1_space_get_id()	84
5.43.3.5 test1_space_get_name()	84
5.43.3.6 test1_space_get_north()	84
5.43.3.7 test1_space_get_object()	84
5.43.3.8 test1_space_get_south()	84
5.43.3.9 test1_space_get_west()	84
5.43.3.10 test1_space_set_east()	84
5.43.3.11 test1_space_set_name()	84
5.43.3.12 test1_space_set_north()	85
5.43.3.13 test1_space_set_object()	85
5.43.3.14 test1_space_set_south()	85
5.43.3.15 test1_space_set_west()	85
5.43.3.16 test2_space_create()	85
5.43.3.17 test2_space_get_east()	85
5.43.3.18 test2_space_get_id()	85
5.43.3.19 test2_space_get_name()	85
5.43.3.20 test2_space_get_north()	85
5.43.3.21 test2_space_get_object()	85
5.43.3.22 test2_space_get_south()	85
5.43.3.23 test2_space_get_west()	85
5.43.3.24 test2_space_set_east()	85
5.43.3.25 test2_space_set_name()	85
5.43.3.26 test2_space_set_north()	86
5.43.3.27 test2_space_set_object()	86
5.43.3.28 test2_space_set_south()	86
5.43.3.29 test2_space_set_west()	86

5.43.3.30 test3_space_get_object()	86
5.43.3.31 test3_space_set_name()	86
5.44 space_test.h File Reference	86
5.44.1 Detailed Description	87
5.44.2 Function Documentation	87
5.44.2.1 test1_space_create()	87
5.44.2.2 test1_space_get_east()	88
5.44.2.3 test1_space_get_id()	88
5.44.2.4 test1_space_get_name()	88
5.44.2.5 test1_space_get_north()	88
5.44.2.6 test1_space_get_object()	88
5.44.2.7 test1_space_get_south()	88
5.44.2.8 test1_space_get_west()	88
5.44.2.9 test1_space_set_east()	88
5.44.2.10 test1_space_set_name()	88
5.44.2.11 test1_space_set_north()	88
5.44.2.12 test1_space_set_object()	88
5.44.2.13 test1_space_set_south()	88
5.44.2.14 test1_space_set_west()	88
5.44.2.15 test2_space_create()	88
5.44.2.16 test2_space_get_east()	89
5.44.2.17 test2_space_get_id()	89
5.44.2.18 test2_space_get_name()	89
5.44.2.19 test2_space_get_north()	89
5.44.2.20 test2_space_get_object()	89
5.44.2.21 test2_space_get_south()	89
5.44.2.22 test2_space_get_west()	89
5.44.2.23 test2_space_set_east()	89
5.44.2.24 test2_space_set_name()	89
5.44.2.25 test2_space_set_north()	89

5.44.2.26 test2_space_set_object()	89
5.44.2.27 test2_space_set_south()	89
5.44.2.28 test2_space_set_west()	89
5.44.2.29 test3_space_get_object()	89
5.44.2.30 test3_space_set_east()	89
5.44.2.31 test3_space_set_name()	89
5.44.2.32 test3_space_set_north()	90
5.44.2.33 test3_space_set_south()	90
5.44.2.34 test3_space_set_west()	90
5.44.2.35 test4_space_set_east()	90
5.44.2.36 test4_space_set_north()	90
5.44.2.37 test4_space_set_south()	90
5.44.2.38 test4_space_set_west()	90
5.45 sprite.c File Reference	90
5.45.1 Detailed Description	90
5.45.2 Function Documentation	91
5.45.2.1 sprite_create(Id id)	91
5.45.2.2 sprite_destroy(Sprite *sprite)	91
5.45.2.3 sprite_getData(Sprite *sprite, int line)	91
5.45.2.4 sprite_getId(Sprite *sprite)	91
5.45.2.5 sprite_print(Sprite *sprite)	91
5.45.2.6 sprite_putLine(Sprite *sprite, char *string, int line)	91
5.46 sprite.h File Reference	91
5.46.1 Detailed Description	92
5.46.2 Typedef Documentation	92
5.46.2.1 Sprite	92
5.46.3 Function Documentation	92
5.46.3.1 sprite_create(Id id)	92
5.46.3.2 sprite_destroy(Sprite *sprite)	92
5.46.3.3 sprite_getData(Sprite *sprite, int line)	92

5.46.3.4	<code>sprite_getId(Sprite *sprite)</code>	92
5.46.3.5	<code>sprite_print(Sprite *sprite)</code>	92
5.46.3.6	<code>sprite_putLine(Sprite *sprite, char *string, int line)</code>	92
5.47	<code>sprite_loader.c</code> File Reference	92
5.47.1	Detailed Description	93
5.47.2	Function Documentation	93
5.47.2.1	<code>sprite_loader_map(Game *game, char *filename)</code>	93
5.48	<code>sprite_loader.h</code> File Reference	93
5.48.1	Detailed Description	93
5.48.2	Function Documentation	94
5.48.2.1	<code>sprite_loader_map(Game *game, char *filename)</code>	94
5.49	<code>test.h</code> File Reference	94
5.49.1	Detailed Description	94
5.49.2	Macro Definition Documentation	95
5.49.2.1	KCYN	95
5.49.2.2	KGRN	95
5.49.2.3	KRED	95
5.49.2.4	KYEL	95
5.49.2.5	PRINT_PASSED_PERCENTAGE	95
5.49.2.6	PRINT_TEST_RESULT	95
5.49.2.7	RESET	95
5.49.3	Variable Documentation	95
5.49.3.1	__pass	95
5.49.3.2	__test_counter	95
5.49.3.3	__test_passed	95
5.50	<code>types.h</code> File Reference	95
5.50.1	Detailed Description	96
5.50.2	Macro Definition Documentation	96
5.50.2.1	MAX_SPRITES	96
5.50.2.2	MAX_STRING	96
5.50.2.3	NO_ID	96
5.50.2.4	STDSIZE	96
5.50.2.5	WORD_SIZE	96
5.50.3	Typedef Documentation	96
5.50.3.1	Id	96
5.50.4	Enumeration Type Documentation	96
5.50.4.1	BOOL	96
5.50.4.2	DIRECTION	97
5.50.4.3	LinkStatus	97
5.50.4.4	STATUS	97

Chapter 1

Test List

Member [test1_space_create\(\)](#)

Prueba la función de creación de un espacio

Member [test1_space_set_name\(\)](#)

Prueba la función para establecer el nombre de un espacio

Member [test2_space_create\(\)](#)

Prueba la función de creación de un espacio

Member [test2_space_set_name\(\)](#)

Prueba la función para establecer el nombre de un espacio

Member [test3_space_set_name\(\)](#)

Prueba la función para establecer el nombre de un espacio

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

_Area	7
_Die	8
_F_Command	8
_Game	9
_Graphic_engine	10
_Inventory	11
_Link	11
_Object	12
_Player	13
_Set	14
_Space	15
_Sprite	17

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

command.c	Commands and user input	19
command.h	Commands and user input	21
command_test.c	22
command_test.h	It declares the tests for the command module	23
die.c	It declares the die module	24
die.h	It declares the die module	26
die_test.c	27
die_test.h	It declares the tests for the die module	28
game.c	It implements the game interface and all the associated callbacks for each command	29
game.h	Main function	33
game_loop.c	Main loop	36
game_reader.c	Reads data for the game from files	37
game_reader.h	Reads data for the game from files	37
game_rules.c	38
game_rules.h	38
graphic_engine.c	Uses screen.* to create the UI	39
graphic_engine.h	Uses screen.* to create the UI	40
inventory.c	Module for player's inventory	41
inventory.h	Module for player's inventory	43
inventory_test.c	44

inventory_test.h	It declares the tests for the inventory module	45
link.c	Creates the links between spaces	47
link.h	Creates the links between spaces	48
link_test.c	50
link_test.h	It declares the tests for the link module	52
object.c	Functions for the creation of objects	53
object.h	Functions for the creation of objects	55
object_test.c	58
object_test.h	It declares the tests for the object module	59
player.c	Functions for the creation of players	61
player.h	Functions for the creation of players	62
player_test.c	64
player_test.h	It declares the tests for the player module	65
screen.c	Functions for the creation of players	67
screen.h	Functions used by graphic_engine.*	69
set.c	Low level stack and queue functions	70
set.h	Low level stack and queue functions	72
set_test.c	73
set_test.h	It declares the tests for the set module	75
space.c	76
space.h	Defines functions for space manipulation	79
space_test.c	It tests space module	82
space_test.h	It declares the tests for the space module	86
sprite.c	It declares the sprite module	90
sprite.h	It declares the sprite module	91
sprite_loader.c	Reads the sprites from a file	92
sprite_loader.h	Reads the sprites from a file	93
test.h	Test low level functions	94
types.h	Global typedefs	95

Chapter 4

Class Documentation

4.1 `_Area` Struct Reference

Public Attributes

- `int x`
- `int y`
- `int width`
- `int height`
- `char * cursor`

4.1.1 Member Data Documentation

4.1.1.1 `char* _Area::cursor`

`cursor`

4.1.1.2 `int _Area::height`

`dimesiones`

4.1.1.3 `int _Area::width`

4.1.1.4 `int _Area::x`

4.1.1.5 `int _Area::y`

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

- `screen.c`

4.2 `_Die` Struct Reference

Public Attributes

- [Id](#) `id`
- short int [result](#)

4.2.1 Member Data Documentation

4.2.1.1 `Id_Die::id`

Id del dado

4.2.1.2 short int `_Die::result`

Valor de la ultima tirada

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

- [die.c](#)

4.3 `_F_Command` Struct Reference

Public Attributes

- [T_Command](#) `text`
- char [id](#) [[CMD LENGHT](#)]

4.3.1 Member Data Documentation

4.3.1.1 char `_F_Command::id`[[CMD LENGHT](#)]

Id

4.3.1.2 [T_Command](#) `_F_Command::text`

Instruction

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

- [command.c](#)

4.4 _Game Struct Reference

Collaboration diagram for _Game:

Public Attributes

- [Player](#) * [player](#)
- [Object](#) * [objects](#) [MAX_OBJECTS]
- [Space](#) * [spaces](#) [MAX_SPACES+1]
- [Link](#) * [links](#) [MAX_LINK]
- [Die](#) * [die](#)
- [F_Command](#) * [last_cmd](#)
- [Sprite](#) * [sprites](#) [MAX_SPRITES]

4.4.1 Member Data Documentation

4.4.1.1 [Die](#)* _Game::die

Dado

4.4.1.2 [F_Command](#)* _Game::last_cmd

ultimo comando

4.4.1.3 [Link](#)* _Game::links[MAX_LINK]

Array de links

4.4.1.4 [Object](#)* _Game::objects[MAX_OBJECTS]

Array de objetos

4.4.1.5 [Player](#)* _Game::player

Jugador

4.4.1.6 [Space](#)* _Game::spaces[MAX_SPACES+1]

Array de espacios

4.4.1.7 `Sprite* _Game::sprites[MAX_SPRITES]`

sprites del mapa

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

- [game.c](#)

4.5 `_Graphic_engine` Struct Reference

Collaboration diagram for `_Graphic_engine`:

Public Attributes

- [Area](#) * `map`
- [Area](#) * `descript`
- [Area](#) * `banner`
- [Area](#) * `help`
- [Area](#) * `feedback`

4.5.1 Member Data Documentation

4.5.1.1 `Area* _Graphic_engine::banner`

banner

4.5.1.2 `Area* _Graphic_engine::descript`

donde van las descripciones

4.5.1.3 `Area* _Graphic_engine::feedback`

Commandos realizados e info

4.5.1.4 `Area* _Graphic_engine::help`

ayuda

4.5.1.5 Area* _Graphic_engine::map

Mapa, dibujo

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

- [graphic_engine.c](#)

4.6 _Inventory Struct Reference

Collaboration diagram for _Inventory:

Public Attributes

- [Set * ids](#)
- [int id_max](#)

4.6.1 Member Data Documentation

4.6.1.1 int _Inventory::id_max

Maximo de lugares

4.6.1.2 Set* _Inventory::ids

Inventario

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

- [inventory.c](#)

4.7 _Link Struct Reference

Public Attributes

- [Id linkId](#)
- [Id linkspace1](#)
- [Id linkspace2](#)
- [int direction](#)
- [LinkStatus door](#)

4.7.1 Member Data Documentation

4.7.1.1 `int _Link::direction`

Norte, sur, este u oeste

4.7.1.2 `LinkStatus _Link::door`

Estatus del link OPENED/CLOSED

4.7.1.3 `Id _Link::linkId`

Id del link

4.7.1.4 `Id _Link::linkspace1`

4.7.1.5 `Id _Link::linkspace2`

id de los espacios conectados

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

- [link.c](#)

4.8 `_Object` Struct Reference

Public Attributes

- char [name](#) [STDSIZE]
- char [description](#) [STDSIZE]
- [Id id](#)
- [BOOL mobile](#)
- [BOOL moved](#)
- [BOOL hidden](#)
- [Id open](#)
- [BOOL iluminati](#)
- [BOOL on](#)
- char [description_al](#) [STDSIZE]

4.8.1 Member Data Documentation

4.8.1.1 `char _Object::description[STDSIZE]`

Descripcion del objeto

4.8.1.2 char _Object::description_al[STDSIZE]

Descripcion alternativa del objeto

4.8.1.3 BOOL _Object::hidden

indica si el objeto esta oculto

4.8.1.4 Id _Object::id

Identificador

4.8.1.5 BOOL _Object::iluminati

4.8.1.6 BOOL _Object::mobile

indica si es movable o no

4.8.1.7 BOOL _Object::moved

indica si se ha movido o no

4.8.1.8 char _Object::name[STDSIZE]

Nombre del objeto

4.8.1.9 BOOL _Object::on

4.8.1.10 Id _Object::open

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

- [object.c](#)

4.9 _Player Struct Reference

Collaboration diagram for _Player:

Public Attributes

- char [name](#) [[STDSIZE](#)]
- [Id](#) [location_id](#)
- [Inventory](#) * [inv](#)
- [Id](#) [id](#)

4.9.1 Member Data Documentation

4.9.1.1 [Id](#) [_Player::id](#)

id del jugador

4.9.1.2 [Inventory](#)* [_Player::inv](#)

inventario del jugador

4.9.1.3 [Id](#) [_Player::location_id](#)

id de donde esta

4.9.1.4 char [_Player::name](#)[[STDSIZE](#)]

Nombre del jugador

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

- [player.c](#)

4.10 [_Set](#) Struct Reference

Public Attributes

- [Id](#) [id_list](#) [[MAX_INV_SIZE](#)]
- int [id_total](#)

4.10.1 Member Data Documentation

4.10.1.1 [Id](#) [_Set::id_list](#)[[MAX_INV_SIZE](#)]

Array

4.10.1.2 int _Set::id_total

Total del array

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

- [set.c](#)

4.11 _Space Struct Reference

Collaboration diagram for _Space:

Public Attributes

- [Id](#) id
- [char](#) name [[WORD_SIZE](#)]
- [char](#) description [[WORD_SIZE](#)]
- [Id](#) linkNorth
- [Id](#) linkSouth
- [Id](#) linkEast
- [Id](#) linkWest
- [Id](#) linkUp
- [Id](#) linkDown
- [Id](#) spriteId [17]
- [int](#) curentSprite
- [char](#) gdesc [3][21]
- [Set](#) * objects
- [BOOL](#) light

4.11.1 Member Data Documentation

4.11.1.1 int _Space::curentSprite

sprite que actual

4.11.1.2 char _Space::description[WORD_SIZE]

descripcion

4.11.1.3 char _Space::gdesc[3][21]

strings para objetos ASCII

4.11.1.4 Id_Space::id

id del espacio

4.11.1.5 BOOL_Space::light

Bool que indica si la casilla esta iluminada

4.11.1.6 Id_Space::linkDown

id del link hacia abajo

4.11.1.7 Id_Space::linkEast

id del link al este

4.11.1.8 Id_Space::linkNorth

id del link al norte

4.11.1.9 Id_Space::linkSouth

id del link al sur

4.11.1.10 Id_Space::linkUp

id del link hacia arriba

4.11.1.11 Id_Space::linkWest

id del link al oeste

4.11.1.12 char_Space::name[WORD_SIZE]

nombre del espacio

4.11.1.13 Set*_Space::objects

Set de objetos

4.11.1.14 Id_Space::spriteld[17]

id del sprite

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

- [space.c](#)

4.12 _Sprite Struct Reference

Public Attributes

- [Id](#) id
- char [data](#) [17][39]

4.12.1 Member Data Documentation

4.12.1.1 char _Sprite::data[17][39]

Datos en ascii

4.12.1.2 Id_Sprite::id

Id del sprite

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

- [sprite.c](#)

Chapter 5

File Documentation

5.1 command.c File Reference

Commands and user input.

```
#include <stdio.h>
#include <strings.h>
#include "../include/command.h"
Include dependency graph for command.c:
```

Classes

- struct [_F_Command](#)

Macros

- #define [CMD LENGHT](#) 30
- #define [N_CMD](#) 11

Functions

- [STATUS get_user_input](#) ([F_Command](#) *command)
- [F_Command](#) * [command_create](#) ()
- [STATUS command_setCmd](#) ([F_Command](#) *cmd, [T_Command](#) command)
- [T_Command](#) [command_getCmd](#) ([F_Command](#) *cmd)
- char * [command_get_id](#) ([F_Command](#) *cmd)
- [STATUS command_set_id](#) ([F_Command](#) *cmd, char *string)
- void [command_free](#) ([F_Command](#) *cmd)

Variables

- char * [cmd_to_str](#) [[N_CMD](#)] = {"No command", "Unknown", "Exit", "Pickup", "Drop", "Roll", "Move", "Check", "Turnon", "Turnoff", "Open"}
- char * [short_cmd_to_str](#) [[N_CMD](#)] = {"", "", "e", "p", "d", "r", "m", "c", "to", "tf", "o"}

5.1.1 Detailed Description

Commands and user input.

Defines functions for space manipulation.

Author

Antonio Solana

Copyright

GNU Public License

Author

Cataln Rotaru

Copyright

GNU Public License

5.1.2 Macro Definition Documentation

5.1.2.1 `#define CMD_LENHT 30`

5.1.2.2 `#define N_CMD 11`

5.1.3 Function Documentation

5.1.3.1 `F_Command* command_create ()`

5.1.3.2 `void command_free (F_Command * cmd)`

5.1.3.3 `char* command_get_id (F_Command * cmd)`

5.1.3.4 `T_Command command_getCmd (F_Command * cmd)`

5.1.3.5 `STATUS command_set_id (F_Command * cmd, char * string)`

5.1.3.6 `STATUS command_setCmd (F_Command * cmd, T_Command command)`

5.1.3.7 `STATUS get_user_input (F_Command * command)`

5.1.4 Variable Documentation

5.1.4.1 `char* cmd_to_str[N_CMD] = {"No command", "Unknown", "Exit", "Pickup", "Drop", "Roll", "Move", "Check", "Turnon", "Turnoff", "Open"}`

5.1.4.2 `char* short_cmd_to_str[N_CMD] = {"", "", "e", "p", "d", "r", "m", "c", "to", "tf", "o"}`

5.2 command.h File Reference

Commands and user input.

```
#include <string.h>
#include <stdlib.h>
#include "../include/types.h"
Include dependency graph for command.h:
```

This graph shows which files directly or indirectly include this file:

Typedefs

- typedef enum [enum_Command](#) [T_Command](#)
- typedef struct [_F_Command](#) [F_Command](#)

Enumerations

- enum [enum_Command](#) {
[NO_CMD](#) = -1, [UNKNOWN](#), [EXIT](#), [PICK_UP](#),
[DROP](#), [ROLL](#), [MOVE](#), [CHECK](#),
[TURNON](#), [TURNOFF](#), [OPEN](#) }

Functions

- [STATUS](#) [get_user_input](#) ([F_Command](#) *)
- [F_Command](#) * [command_create](#) ()
- void [command_free](#) ([F_Command](#) *)
- [STATUS](#) [command_setCmd](#) ([F_Command](#) *, [T_Command](#))
- [T_Command](#) [command_getCmd](#) ([F_Command](#) *)
- [STATUS](#) [command_set_id](#) ([F_Command](#) *, char *)
- char * [command_get_id](#) ([F_Command](#) *)

5.2.1 Detailed Description

Commands and user input.

Author

Antonio Solana

Copyright

GNU Public License

5.2.2 Typedef Documentation

5.2.2.1 typedef struct `_F_Command` `F_Command`

5.2.2.2 typedef enum `enum_Command` `T_Command`

5.2.3 Enumeration Type Documentation

5.2.3.1 enum `enum_Command`

Enumerator

NO_CMD
UNKNOWN
EXIT
PICK_UP
DROP
ROLL
MOVE
CHECK
TURNON
TURNOFF
OPEN

5.2.4 Function Documentation

5.2.4.1 `F_Command*` `command_create` ()

5.2.4.2 void `command_free` (`F_Command *`)

5.2.4.3 `char*` `command_get_id` (`F_Command *`)

5.2.4.4 `T_Command` `command_getCmd` (`F_Command *`)

5.2.4.5 `STATUS` `command_set_id` (`F_Command *`, `char *`)

5.2.4.6 `STATUS` `command_setCmd` (`F_Command *`, `T_Command`)

5.2.4.7 `STATUS` `get_user_input` (`F_Command *`)

5.3 `command_test.c` File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "../include/command.h"
#include "../include/command_test.h"
#include "../include/test.h"
Include dependency graph for command_test.c:
```

Macros

- #define `MAX_TESTS` 8

Functions

- int `main` (int argc, char **argv)
Funcion principal de pruebas para el modulo Space.
- void `test1_command_create` ()
- void `test1_command_set_cmd` ()
- void `test2_command_set_cmd` ()
- void `test1_command_get_cmd` ()
- void `test1_command_set_id` ()
- void `test2_command_set_id` ()
- void `test1_command_get_id` ()

5.3.1 Macro Definition Documentation

5.3.1.1 #define MAX_TESTS 8

5.3.2 Function Documentation

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

Funcion principal de pruebas para el modulo Space.

Dos modos de ejecucion: 1.-Si se ejecuta sin parametros se ejecutan todas las pruebas 2.-Si se ejecuta con un numero entre 1 y el numero de pruebas solo ejecuta la prueba indicada

5.3.2.2 void test1_command_create ()

5.3.2.3 void test1_command_get_cmd ()

5.3.2.4 void test1_command_get_id ()

5.3.2.5 void test1_command_set_cmd ()

5.3.2.6 void test1_command_set_id ()

5.3.2.7 void test2_command_set_cmd ()

5.3.2.8 void test2_command_set_id ()

5.4 command_test.h File Reference

It declares the tests for the command module.

This graph shows which files directly or indirectly include this file:

Functions

- void [test1_command_create](#) ()
- void [test1_command_set_cmd](#) ()
- void [test2_command_set_cmd](#) ()
- void [test1_command_get_cmd](#) ()
- void [test1_command_set_id](#) ()
- void [test2_command_set_id](#) ()
- void [test1_command_get_id](#) ()

5.4.1 Detailed Description

It declares the tests for the command module.

Author

Pablo Sánchez Redondo

Copyright

GNU Public License

5.4.2 Function Documentation

5.4.2.1 void [test1_command_create](#) ()

5.4.2.2 void [test1_command_get_cmd](#) ()

5.4.2.3 void [test1_command_get_id](#) ()

5.4.2.4 void [test1_command_set_cmd](#) ()

5.4.2.5 void [test1_command_set_id](#) ()

5.4.2.6 void [test2_command_set_cmd](#) ()

5.4.2.7 void [test2_command_set_id](#) ()

5.5 die.c File Reference

It declares the die module.

```
#include "../include/die.h"  
Include dependency graph for die.c:
```


Classes

- struct `_Die`

Functions

- `Die * die_ini (Id id)`
esta funcion se encarga de crear el dado reservando memoria para el mismo.
- void `die_die_die (Die *die)`
- `STATUS die_roll (Die *die)`
- `STATUS die_print (FILE *f, Die *die)`
- short int `die_get_last_roll (Die *die)`

5.5.1 Detailed Description

It declares the die module.

Author

Pablo Sánchez Redondo

Copyright

GNU Public License

5.5.2 Function Documentation

5.5.2.1 void `die_die_die (Die * die)`

5.5.2.2 short int `die_get_last_roll (Die * die)`

5.5.2.3 `Die* die_ini (Id)`

esta funcion se encarga de crear el dado reservando memoria para el mismo.

Author

Pablo Sánchez ID, el id del dado.

Returns

`newdie`, el dado creado, o NULL si algo no ha salido como esperaba.

5.5.2.4 **STATUS** `die_print (FILE * f, Die * die)`

5.5.2.5 **STATUS** `die_roll (Die * die)`

5.6 `die.h` File Reference

It declares the die module.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "types.h"
Include dependency graph for die.h:
```

This graph shows which files directly or indirectly include this file:

Typedefs

- typedef struct [_Die](#) [Die](#)

Functions

- [Die *](#) [die_ini](#) ([Id](#))
esta funcion se encarga de crear el dado reservando memoria para el mismo.
- void [die_die_die](#) ([Die *](#))
- **STATUS** [die_roll](#) ([Die *](#))
- short int [die_get_last_roll](#) ([Die *](#))
- **STATUS** [die_print](#) ([FILE *](#), [Die *](#))

5.6.1 Detailed Description

It declares the die module.

Author

Pablo Sánchez Redondo

Copyright

GNU Public License

5.6.2 Typedef Documentation

5.6.2.1 typedef struct _Die Die

la estructura `_Die` consta de dos componentes, uno de ellos, `ID` (donde se guardara el id de los dados) e `int ult_tirada` (donde se guarda el ultimo resultado de la funcion `roll_die()`).

5.6.3 Function Documentation

5.6.3.1 void die_die_die (Die *)

5.6.3.2 short int die_get_last_roll (Die *)

5.6.3.3 Die* die_ini (Id)

esta funcion se encarga de crear el dado reservando memoria para el mismo.

Author

Pablo Sánchez ID, el id del dado.

Returns

`newdie`, el dado creado, o `NULL` si algo no ha salido como esperaba.

5.6.3.4 STATUS die_print (FILE *, Die *)

5.6.3.5 STATUS die_roll (Die *)

5.7 die_test.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "../include/die.h"
#include "../include/die_test.h"
#include "../include/test.h"
Include dependency graph for die_test.c:
```

Macros

- #define `MAX_TESTS` 6

Functions

- int [main](#) (int argc, char **argv)
Funcion principal de pruebas para el modulo Space.
- void [test1_die_create](#) ()
- void [test1_die_roll](#) ()
- void [test2_die_roll](#) ()
- void [test1_die_get_last_roll](#) ()
- void [test2_die_get_last_roll](#) ()

5.7.1 Macro Definition Documentation

5.7.1.1 `#define MAX_TESTS 6`

5.7.2 Function Documentation

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

Funcion principal de pruebas para el modulo Space.

Dos modos de ejecucion: 1.-Si se ejecuta sin parametros se ejecutan todas las pruebas 2.-Si se ejecuta con un numero entre 1 y el numero de pruebas solo ejecuta la prueba indicada

5.7.2.2 `void test1_die_create ()`

5.7.2.3 `void test1_die_get_last_roll ()`

5.7.2.4 `void test1_die_roll ()`

5.7.2.5 `void test2_die_get_last_roll ()`

5.7.2.6 `void test2_die_roll ()`

5.8 `die_test.h` File Reference

It declares the tests for the die module.

This graph shows which files directly or indirectly include this file:

Functions

- void [test1_die_create](#) ()
- void [test1_die_roll](#) ()
- void [test2_die_roll](#) ()
- void [test1_die_get_last_roll](#) ()
- void [test2_die_get_last_roll](#) ()

5.8.1 Detailed Description

It declares the tests for the die module.

Author

Pablo Sánchez Redondo

Copyright

GNU Public License

5.8.2 Function Documentation

5.8.2.1 void test1_die_create ()

5.8.2.2 void test1_die_get_last_roll ()

5.8.2.3 void test1_die_roll ()

5.8.2.4 void test2_die_get_last_roll ()

5.8.2.5 void test2_die_roll ()

5.9 game.c File Reference

It implements the game interface and all the associated callbacks for each command.

```
#include <stdio.h>
#include <stdlib.h>
#include <strings.h>
#include "../include/game.h"
#include "../include/game_reader.h"
#include "../include/sprite_loader.h"
#include "../include/sprite.h"
Include dependency graph for game.c:
```

Classes

- struct [_Game](#)

Macros

- #define [N_CALLBACK](#) 11
- #define [PLAYER_ID](#) 1
- #define [DIE_SEED](#) 666
- #define [STARTING_SPACE](#) 25
- #define [NO_LIGHT_SPRITE](#) 16

Typedefs

- typedef void(* [callback_fn](#)) ([Game](#) *game)

Functions

- void [game_callback_unknown](#) ([Game](#) *game)
- void [game_callback_exit](#) ([Game](#) *game)
- void [game_callback_pickup](#) ([Game](#) *game)
- void [game_callback_drop](#) ([Game](#) *game)
- void [game_callback_roll](#) ([Game](#) *game)
- void [game_callback_move](#) ([Game](#) *game)
- void [game_callback_check](#) ([Game](#) *game)
- void [game_callback_turnOn](#) ([Game](#) *game)
- void [game_callback_turnOff](#) ([Game](#) *game)
- void [game_callback_open](#) ([Game](#) *game)
- [Game](#) * [game_create](#) ()
- [STATUS](#) [game_create_from_file](#) ([Game](#) *game, char *filename)
- [STATUS](#) [game_destroy](#) ([Game](#) *game)
- [Space](#) * [game_get_space](#) ([Game](#) *game, [Id](#) id)
- [Player](#) * [game_get_player](#) ([Game](#) *game)
- [Object](#) * [game_get_object](#) ([Game](#) *game, char *object_name)
- [Object](#) * [game_get_object_from_id](#) ([Game](#) *game, [Id](#) id)
- [Link](#) * [game_get_link](#) ([Game](#) *game, [Id](#) id)
- [Id](#) [game_get_link_id_at](#) ([Game](#) *game, int pos)
- [Id](#) [game_get_player_location](#) ([Game](#) *game)
- [Id](#) [game_get_object_location](#) ([Game](#) *game, [Id](#) id)
- [STATUS](#) [game_update](#) ([Game](#) *game, [F_Command](#) *cmd)
- [F_Command](#) * [game_get_last_command](#) ([Game](#) *game)
- [T_Command](#) [game_get_last_command_text](#) ([Game](#) *game)
- void [game_print_opened_links](#) ([Game](#) *game)
- [STATUS](#) [game_add_space](#) ([Game](#) *game, [Space](#) *space)
- [STATUS](#) [game_add_object](#) ([Game](#) *game, [Object](#) *object)
- [Sprite](#) * [game_get_sprite](#) ([Game](#) *game, [Id](#) id)
- [STATUS](#) [game_add_sprite](#) ([Game](#) *game, [Sprite](#) *sprite, int i)
- [Id](#) [game_get_space_id_at](#) ([Game](#) *game, int position)
- [STATUS](#) [game_set_player_location](#) ([Game](#) *game, [Id](#) id)
- [STATUS](#) [game_set_link](#) ([Game](#) *game, [Id](#) link_id, [Id](#) space_id0, [Id](#) space_id1, int direction, [LinkStatus](#) door)
- [STATUS](#) [game_set_object_location](#) ([Game](#) *game, [Id](#) id, [Id](#) obj_id, char *name, char *description)
- int [game_get_last_roll](#) ([Game](#) *game)
- [BOOL](#) [game_areSpacesAdjacent](#) ([Game](#) *g, [Id](#) space1, [Id](#) space2)
- [STATUS](#) [update_sprites](#) ([Game](#) *game)
- [BOOL](#) [game_is_over](#) ([Game](#) *game)

Variables

- static [callback_fn](#) [game_callback_fn_list](#) [[N_CALLBACK](#)]

5.9.1 Detailed Description

It implements the game interface and all the associated callbacks for each command.

Author

Profesores PPROG

Copyright

GNU Public License

5.9.2 Macro Definition Documentation

5.9.2.1 `#define DIE_SEED 666`

5.9.2.2 `#define N_CALLBACK 11`

5.9.2.3 `#define NO_LIGHT_SPRITE 16`

5.9.2.4 `#define PLAYER_ID 1`

5.9.2.5 `#define STARTING_SPACE 25`

5.9.3 Typedef Documentation

5.9.3.1 `typedef void(* callback_fn) (Game *game)`

Define the function type for the callbacks

5.9.4 Function Documentation

5.9.4.1 `STATUS game_add_object (Game * game, Object * object)`

5.9.4.2 `STATUS game_add_space (Game * game, Space * space)`

5.9.4.3 `STATUS game_add_sprite (Game * game, Sprite * sprite, int i)`

5.9.4.4 `BOOL game_areSpacesAdjacent (Game * g, Id space1, Id space2)`

5.9.4.5 `void game_callback_check (Game * game)`

5.9.4.6 `void game_callback_drop (Game * game)`

5.9.4.7 `void game_callback_exit (Game * game)`

5.9.4.8 void game_callback_move (Game * *game*)

5.9.4.9 void game_callback_open (Game * *game*)

5.9.4.10 void game_callback_pickup (Game * *game*)

5.9.4.11 void game_callback_roll (Game * *game*)

5.9.4.12 void game_callback_turnOff (Game * *game*)

5.9.4.13 void game_callback_turnOn (Game * *game*)

5.9.4.14 void game_callback_unknown (Game * *game*)

List of callbacks for each command in the game

5.9.4.15 Game* game_create ()

5.9.4.16 STATUS game_create_from_file (Game * *game*, char * *filename*)

5.9.4.17 STATUS game_destroy (Game * *game*)

5.9.4.18 F_Command* game_get_last_command (Game * *game*)

5.9.4.19 T_Command game_get_last_command_text (Game * *game*)

5.9.4.20 int game_get_last_roll (Game * *game*)

5.9.4.21 Link* game_get_link (Game * *game*, Id *id*)

5.9.4.22 Id game_get_link_id_at (Game * *game*, int *pos*)

5.9.4.23 Object* game_get_object (Game * *game*, char * *object_name*)

5.9.4.24 Object* game_get_object_from_id (Game * *game*, Id *id*)

5.9.4.25 Id game_get_object_location (Game * *game*, Id *id*)

5.9.4.26 Player* game_get_player (Game * *game*)

5.9.4.27 Id game_get_player_location (Game * *game*)

5.9.4.28 Space* game_get_space (Game * *game*, Id *id*)

5.9.4.29 Id game_get_space_id_at (Game * *game*, int *position*)

5.9.4.30 `Sprite* game_get_sprite (Game * game, Id id)`

5.9.4.31 `BOOL game_is_over (Game * game)`

5.9.4.32 `void game_print_opened_links (Game * game)`

5.9.4.33 `STATUS game_set_link (Game * game, Id link_id, Id space_id0, Id space_id1, int direction, LinkStatus door)`

5.9.4.34 `STATUS game_set_object_location (Game * game, Id id, Id obj_id, char * name, char * description)`

5.9.4.35 `STATUS game_set_player_location (Game * game, Id id)`

5.9.4.36 `STATUS game_update (Game * game, F_Command * cmd)`

5.9.4.37 `STATUS update_sprites (Game * game)`

5.9.5 Variable Documentation

5.9.5.1 `callback_fn game_callback_fn_list[N_CALLBACK] [static]`

Initial value:

```
=
{
    game_callback_unknown,
    game_callback_exit,
    game_callback_pickup,
    game_callback_drop,
    game_callback_roll,
    game_callback_move,
    game_callback_check,
    game_callback_turnOn,
    game_callback_turnOff,
    game_callback_open}
```

5.10 game.h File Reference

Main function.

```
#include "../include/command.h"
#include "../include/space.h"
#include "../include/object.h"
#include "../include/player.h"
#include "../include/die.h"
#include "../include/link.h"
#include "../include/inventory.h"
#include "../include/sprite.h"
Include dependency graph for game.h:
```

This graph shows which files directly or indirectly include this file:

Macros

- `#define MAX_OBJECTS 1024`

Typedefs

- `typedef struct _Game Game`

Functions

- `Game * game_create ()`
- `STATUS game_create_from_file (Game *, char *)`
- `STATUS game_update (Game *, F_Command *)`
- `STATUS game_destroy (Game *)`
- `BOOL game_is_over (Game *)`
- `void game_print_screen (Game *)`
- `void game_print_data (Game *)`
- `Space * game_get_space (Game *, Id)`
- `Player * game_get_player (Game *)`
- `Object * game_get_object (Game *, char *)`
- `Object * game_get_object_from_id (Game *game, Id id)`
- `Link * game_get_link (Game *, Id)`
- `Id game_get_link_id_at (Game *, int)`
- `Id game_get_player_location (Game *)`
- `Id game_get_object_location (Game *, Id)`
- `Id game_get_space_id_at (Game *, int)`
- `int game_get_last_roll (Game *)`
- `STATUS game_add_space (Game *, Space *)`
- `STATUS game_add_object (Game *game, Object *object)`
- `STATUS game_set_player_location (Game *, Id)`
- `STATUS game_set_object_location (Game *, Id, Id, char *, char *)`
- `STATUS game_set_link (Game *, Id, Id, Id, int, LinkStatus)`
- `STATUS game_load_spaces (Game *, char *)`
- `F_Command * game_get_last_command (Game *)`
- `T_Command game_get_last_command_text (Game *)`
- `BOOL game_areSpacesAdjacent (Game *, Id, Id)`
- `Sprite * game_get_sprite (Game *game, Id id)`
- `STATUS game_add_sprite (Game *game, Sprite *sprite, int i)`
- `STATUS update_sprites (Game *game)`

5.10.1 Detailed Description

Main function.

Author

Bernardo Zambrano

Copyright

GNU Public License

5.10.2 Macro Definition Documentation

5.10.2.1 `#define MAX_OBJECTS 1024`

5.10.3 Typedef Documentation

5.10.3.1 `typedef struct _Game Game`

5.10.4 Function Documentation

5.10.4.1 `STATUS game_add_object (Game * game, Object * object)`

5.10.4.2 `STATUS game_add_space (Game *, Space *)`

5.10.4.3 `STATUS game_add_sprite (Game * game, Sprite * sprite, int i)`

5.10.4.4 `BOOL game_areSpacesAdjacent (Game *, Id, Id)`

5.10.4.5 `Game* game_create ()`

5.10.4.6 `STATUS game_create_from_file (Game *, char *)`

5.10.4.7 `STATUS game_destroy (Game *)`

5.10.4.8 `F_Command* game_get_last_command (Game *)`

5.10.4.9 `T_Command game_get_last_command_text (Game *)`

5.10.4.10 `int game_get_last_roll (Game *)`

5.10.4.11 `Link* game_get_link (Game *, Id)`

5.10.4.12 `Id game_get_link_id_at (Game *, int)`

5.10.4.13 `Object* game_get_object (Game *, char *)`

5.10.4.14 `Object* game_get_object_from_id (Game * game, Id id)`

5.10.4.15 `Id game_get_object_location (Game *, Id)`

5.10.4.16 `Player* game_get_player (Game *)`

5.10.4.17 `Id game_get_player_location (Game *)`

5.10.4.18 `Space* game_get_space (Game *, Id)`

- 5.10.4.19 `Id game_get_space_id_at (Game *, int)`
- 5.10.4.20 `Sprite* game_get_sprite (Game * game, Id id)`
- 5.10.4.21 `BOOL game_is_over (Game *)`
- 5.10.4.22 `STATUS game_load_spaces (Game *, char *)`
- 5.10.4.23 `void game_print_data (Game *)`
- 5.10.4.24 `void game_print_screen (Game *)`
- 5.10.4.25 `STATUS game_set_link (Game *, Id, Id, Id, int, LinkStatus)`
- 5.10.4.26 `STATUS game_set_object_location (Game *, Id, Id, char *, char *)`
- 5.10.4.27 `STATUS game_set_player_location (Game *, Id)`
- 5.10.4.28 `STATUS game_update (Game *, F_Command *)`
- 5.10.4.29 `STATUS update_sprites (Game * game)`

5.11 game_loop.c File Reference

Main loop.

```
#include <stdio.h>
#include <stdlib.h>
#include "../include/graphic_engine.h"
Include dependency graph for game_loop.c:
```

Functions

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

5.11.1 Detailed Description

Main loop.

Author

Antonio Solana

Copyright

GNU Public License

5.11.2 Function Documentation

5.11.2.1 `int main (int argc, char * argv[])`

5.12 game_reader.c File Reference

Reads data for the game from files.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "../include/game_reader.h"
Include dependency graph for game_reader.c:
```

Functions

- [STATUS game_load_spaces](#) ([Game](#) *game, char *filename)

5.12.1 Detailed Description

Reads data for the game from files.

Author

Catalin Rotaru

Copyright

GNU Public License

5.12.2 Function Documentation

5.12.2.1 `STATUS game_load_spaces (Game * game, char * filename)`

5.13 game_reader.h File Reference

Reads data for the game from files.

```
#include "../include/types.h"
#include "../include/game.h"
Include dependency graph for game_reader.h:
```

This graph shows which files directly or indirectly include this file:

Functions

- [STATUS game_load_spaces](#) ([Game](#) *game, char *filename)
- [STATUS game_load_links](#) ([Game](#) *game, char *filename)

5.13.1 Detailed Description

Reads data for the game from files.

Author

Catalin Rotaru

Copyright

GNU Public License

5.13.2 Function Documentation

5.13.2.1 **STATUS** game_load_links (**Game** * *game*, char * *filename*)

5.13.2.2 **STATUS** game_load_spaces (**Game** * *game*, char * *filename*)

5.14 game_rules.c File Reference

Functions

- [STATUS update_rules](#) ()

5.14.1 Function Documentation

5.14.1.1 **STATUS** update_rules ()

5.15 game_rules.h File Reference

```
#include "../include/game.h"
```

Include dependency graph for game_rules.h:

5.16 `graphic_engine.c` File Reference

Uses `screen.*` to create the UI.

```
#include <stdlib.h>
#include <stdio.h>
#include "../include/screen.h"
#include "../include/graphic_engine.h"
#include "../include/set.h"
Include dependency graph for graphic_engine.c:
```

Classes

- [struct `_Graphic_engine`](#)

Macros

- `#define` [STD_SPACE](#) " "
- `#define` [STD_SPACE1](#) " "

Functions

- [Graphic_engine * `graphic_engine_create`](#) ()
- `void` [`graphic_engine_destroy`](#) ([Graphic_engine *`ge`](#))
- `void` [`graphic_engine_paint_game`](#) ([Graphic_engine *`ge`](#), [Game *`game`](#))
- `char *` [`create_objects_string`](#) ([Game *`game`](#), `Id` [id](#))
- `void` [`print_new_line`](#) ([Area *`area`](#), `int` [number](#))

5.16.1 Detailed Description

Uses `screen.*` to create the UI.

Author

Antonio Solana

Copyright

GNU Public License

5.16.2 Macro Definition Documentation

5.16.2.1 `#define STD_SPACE " "`

5.16.2.2 `#define STD_SPACE1 " "`

5.16.3 Function Documentation

5.16.3.1 `char* create_objects_string (Game * game, Id id)`

5.16.3.2 `Graphic_engine* graphic_engine_create ()`

5.16.3.3 `void graphic_engine_destroy (Graphic_engine * ge)`

5.16.3.4 `void graphic_engine_paint_game (Graphic_engine * ge, Game * game)`

5.16.3.5 `void print_new_line (Area * area, int number)`

5.17 `graphic_engine.h` File Reference

Uses `screen.*` to create the UI.

```
#include "../include/game.h"
#include "../include/screen.h"
Include dependency graph for graphic_engine.h:
```

This graph shows which files directly or indirectly include this file:

Typedefs

- `typedef struct _Graphic_engine Graphic_engine`

Functions

- `Graphic_engine * graphic_engine_create ()`
- `void graphic_engine_destroy (Graphic_engine *)`
- `void graphic_engine_paint_game (Graphic_engine *, Game *)`
- `void graphic_engine_paint_space (Graphic_engine *, Game *, int)`
- `void print_new_line (Area *, int number)`
- `char * create_objects_string (Game *, Id)`

5.17.1 Detailed Description

Uses screen.* to create the UI.

Author

Antonio Solana

Copyright

GNU Public License

5.17.2 Typedef Documentation

5.17.2.1 `typedef struct _Graphic_engine Graphic_engine`

5.17.3 Function Documentation

5.17.3.1 `char* create_objects_string (Game *, Id)`

5.17.3.2 `Graphic_engine* graphic_engine_create ()`

5.17.3.3 `void graphic_engine_destroy (Graphic_engine *)`

5.17.3.4 `void graphic_engine_paint_game (Graphic_engine *, Game *)`

5.17.3.5 `void graphic_engine_paint_space (Graphic_engine *, Game *, int)`

5.17.3.6 `void print_new_line (Area *, int number)`

5.18 inventory.c File Reference

Module for player's inventory.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "../include/inventory.h"
Include dependency graph for inventory.c:
```

Classes

- [struct _Inventory](#)

Functions

- `Inventory * inventory_create` (int size)
- `STATUS inventory_destroy` (`Inventory *inv`)
- `STATUS inventory_set_ids` (`Inventory *inv`, `Set *ids`)
- `Set * inventory_get_ids` (`Inventory *inv`)
- `Id inventory_get_id_at` (`Inventory *inv`, int num)
- `STATUS inventory_set_id_max` (`Inventory *inv`, int id_max)
- int `inventory_get_id_max` (`Inventory *inv`)
- `STATUS inventory_add_id` (`Inventory *inv`, `Id id`)
- `STATUS inventory_del_id` (`Inventory *inv`, `Id id`)
- void `inventory_print` (`Inventory *inv`)

5.18.1 Detailed Description

Module for player's inventory.

Author

Guillermo Ríos

Copyright

GNU Public License

5.18.2 Function Documentation

5.18.2.1 `STATUS inventory_add_id (Inventory * inv, Id id)`

5.18.2.2 `Inventory* inventory_create (int size)`

5.18.2.3 `STATUS inventory_del_id (Inventory * inv, Id id)`

5.18.2.4 `STATUS inventory_destroy (Inventory * inv)`

5.18.2.5 `Id inventory_get_id_at (Inventory * inv, int num)`

5.18.2.6 `int inventory_get_id_max (Inventory * inv)`

5.18.2.7 `Set* inventory_get_ids (Inventory * inv)`

5.18.2.8 `void inventory_print (Inventory * inv)`

5.18.2.9 `STATUS inventory_set_id_max (Inventory * inv, int id_max)`

5.18.2.10 `STATUS inventory_set_ids (Inventory * inv, Set * ids)`

5.19 inventory.h File Reference

Module for player's inventory.

```
#include "../include/types.h"
#include "../include/set.h"
Include dependency graph for inventory.h:
```

This graph shows which files directly or indirectly include this file:

Typedefs

- typedef struct [_Inventory](#) Inventory

Functions

- [Inventory](#) * [inventory_create](#) (int)
- [STATUS](#) [inventory_destroy](#) ([Inventory](#) *inv)
- [STATUS](#) [inventory_set_ids](#) ([Inventory](#) *inv, [Set](#) *ids)
- [Set](#) * [inventory_get_ids](#) ([Inventory](#) *inv)
- [Id](#) [inventory_get_id_at](#) ([Inventory](#) *inv, int num)
- [STATUS](#) [inventory_add_id](#) ([Inventory](#) *inv, [Id](#) id)
- [STATUS](#) [inventory_del_id](#) ([Inventory](#) *inv, [Id](#) id)
- void [inventory_print](#) ([Inventory](#) *)

5.19.1 Detailed Description

Module for player's inventory.

Author

Guillermo Ríos

Copyright

GNU Public License

5.19.2 Typedef Documentation

5.19.2.1 typedef struct _Inventory Inventory

5.19.3 Function Documentation

5.19.3.1 STATUS inventory_add_id (Inventory * inv, Id id)

5.19.3.2 Inventory* inventory_create (int)

5.19.3.3 STATUS inventory_del_id (Inventory * inv, Id id)

5.19.3.4 STATUS inventory_destroy (Inventory * inv)

5.19.3.5 Id inventory_get_id_at (Inventory * inv, int num)

5.19.3.6 Set* inventory_get_ids (Inventory * inv)

5.19.3.7 void inventory_print (Inventory *)

5.19.3.8 STATUS inventory_set_ids (Inventory * inv, Set * ids)

5.20 inventory_test.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "../include/inventory.h"
#include "../include/inventory_test.h"
#include "../include/set.h"
#include "../include/test.h"
```

Include dependency graph for inventory_test.c:

Macros

- #define [MAX_TESTS](#) 9

Functions

- int [main](#) (int argc, char **argv)
Funcion principal de pruebas para el modulo Space.
- void [test1_inventory_create](#) ()
- void [test1_inventory_set_ids](#) ()
- void [test1_inventory_get_ids](#) ()
- void [test1_inventory_get_id_at](#) ()
- void [test2_inventory_get_id_at](#) ()
- void [test1_inventory_add_id](#) ()
- void [test2_inventory_add_id](#) ()
- void [test1_inventory_del_id](#) ()
- void [test2_inventory_del_id](#) ()

5.20.1 Macro Definition Documentation

5.20.1.1 `#define MAX_TESTS 9`

5.20.2 Function Documentation

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

Funcion principal de pruebas para el modulo Space.

Dos modos de ejecucion: 1.-Si se ejecuta sin parametros se ejecutan todas las pruebas 2.-Si se ejecuta con un numero entre 1 y el numero de pruebas solo ejecuta la prueba indicada

5.20.2.2 `void test1_inventory_add_id ()`

5.20.2.3 `void test1_inventory_create ()`

5.20.2.4 `void test1_inventory_del_id ()`

5.20.2.5 `void test1_inventory_get_id_at ()`

5.20.2.6 `void test1_inventory_get_ids ()`

5.20.2.7 `void test1_inventory_set_ids ()`

5.20.2.8 `void test2_inventory_add_id ()`

5.20.2.9 `void test2_inventory_del_id ()`

5.20.2.10 `void test2_inventory_get_id_at ()`

5.21 inventory_test.h File Reference

It declares the tests for the inventory module.

This graph shows which files directly or indirectly include this file:

Functions

- void [test1_inventory_create](#) ()
- void [test1_inventory_is_full](#) ()
- void [test2_inventory_is_full](#) ()
- void [test1_inventory_is_empty](#) ()
- void [test2_inventory_is_empty](#) ()
- void [test1_inventory_set_ids](#) ()
- void [test1_inventory_get_ids](#) ()
- void [test1_inventory_get_id_at](#) ()
- void [test2_inventory_get_id_at](#) ()
- void [test1_inventory_set_max](#) ()
- void [test2_inventory_set_max](#) ()
- void [test1_inventory_add_id](#) ()
- void [test2_inventory_add_id](#) ()
- void [test1_inventory_del_id](#) ()
- void [test2_inventory_del_id](#) ()
- void [test1_inventory_get_max](#) ()

5.21.1 Detailed Description

It declares the tests for the inventory module.

Author

Pablo Sánchez Redondo

Copyright

GNU Public License

5.21.2 Function Documentation

5.21.2.1 void [test1_inventory_add_id](#) ()

5.21.2.2 void [test1_inventory_create](#) ()

5.21.2.3 void [test1_inventory_del_id](#) ()

5.21.2.4 void [test1_inventory_get_id_at](#) ()

5.21.2.5 void [test1_inventory_get_ids](#) ()

5.21.2.6 void [test1_inventory_get_max](#) ()

5.21.2.7 void [test1_inventory_is_empty](#) ()

5.21.2.8 void [test1_inventory_is_full](#) ()

5.21.2.9 void test1_inventory_set_ids ()

5.21.2.10 void test1_inventory_set_max ()

5.21.2.11 void test2_inventory_add_id ()

5.21.2.12 void test2_inventory_del_id ()

5.21.2.13 void test2_inventory_get_id_at ()

5.21.2.14 void test2_inventory_is_empty ()

5.21.2.15 void test2_inventory_is_full ()

5.21.2.16 void test2_inventory_set_max ()

5.22 link.c File Reference

Creates the links between spaces.

```
#include <stdio.h>
#include <stdlib.h>
#include "../include/link.h"
Include dependency graph for link.c:
```

Classes

- struct [_Link](#)

Functions

- [Link *](#) [link_create](#) ([Id](#) id)
- void [link_destroy](#) ([Link *](#)l)
- [STATUS](#) [link_setDirection](#) ([Link *](#)l, int direction)
- [STATUS](#) [link_setId](#) ([Link *](#)l, [Id](#) id)
- [STATUS](#) [link_setStatus](#) ([Link *](#)l, [LinkStatus](#) door)
- [STATUS](#) [link_setSpaces](#) ([Link *](#)l, [Id](#) space1, [Id](#) space2)
- [Id](#) [link_getDirection](#) ([Link *](#)l)
- [Id](#) [link_getId](#) ([Link *](#)l)
- [Id](#) [link_getSpace1](#) ([Link *](#)l)
- [Id](#) [link_getSpace2](#) ([Link *](#)l)
- [LinkStatus](#) [link_getStatus](#) ([Link *](#)l)
- [STATUS](#) [link_print](#) ([Link *](#)link)
- [Id](#) [link_getDestination](#) ([Link *](#)l, [Id](#) originId)

5.22.1 Detailed Description

Creates the links between spaces.

Author

Pablo Sánchez Redondo

Copyright

GNU Public License

5.22.2 Function Documentation

5.22.2.1 **Link*** link_create (**Id** *id*)

5.22.2.2 **void** link_destroy (**Link** * *l*)

5.22.2.3 **Id** link_getDestination (**Link** * *l*, **Id** *originId*)

5.22.2.4 **Id** link_getDirection (**Link** * *l*)

5.22.2.5 **Id** link_getId (**Link** * *l*)

5.22.2.6 **Id** link_getSpace1 (**Link** * *l*)

5.22.2.7 **Id** link_getSpace2 (**Link** * *l*)

5.22.2.8 **LinkStatus** link_getStatus (**Link** * *l*)

5.22.2.9 **STATUS** link_print (**Link** * *link*)

5.22.2.10 **STATUS** link_setDirection (**Link** * *l*, **int** *direction*)

5.22.2.11 **STATUS** link_setId (**Link** * *l*, **Id** *id*)

5.22.2.12 **STATUS** link_setSpaces (**Link** * *l*, **Id** *space1*, **Id** *space2*)

5.22.2.13 **STATUS** link_setStatus (**Link** * *l*, **LinkStatus** *door*)

5.23 link.h File Reference

Creates the links between spaces.

```
#include "../include/types.h"
Include dependency graph for link.h:
```

This graph shows which files directly or indirectly include this file:

Macros

- `#define MAX_LINK 1024`

Typedefs

- `typedef struct _Link Link`

Functions

- `Link * link_create (Id)`
- `void link_destroy (Link *)`
- `STATUS link_setId (Link *, Id)`
- `STATUS link_setStatus (Link *, LinkStatus)`
- `STATUS link_setSpaces (Link *, Id, Id)`
- `Id link_getId (Link *)`
- `Id link_getSpace1 (Link *)`
- `Id link_getSpace2 (Link *)`
- `LinkStatus link_getStatus (Link *)`
- `Id link_getDestination (Link *, Id)`
- `Id link_getDirection (Link *)`
- `STATUS link_setDirection (Link *, int)`
- `STATUS link_print (Link *)`

5.23.1 Detailed Description

Creates the links between spaces.

Author

Pablo Sánchez Redondo

Copyright

GNU Public License

5.23.2 Macro Definition Documentation

5.23.2.1 `#define MAX_LINK 1024`

5.23.3 Typedef Documentation

5.23.3.1 `typedef struct _Link Link`

5.23.4 Function Documentation

5.23.4.1 `Link* link_create (Id)`

5.23.4.2 void link_destroy (Link *)

5.23.4.3 Id link_getDestination (Link *, Id)

5.23.4.4 Id link_getDirection (Link *)

5.23.4.5 Id link_getId (Link *)

5.23.4.6 Id link_getSpace1 (Link *)

5.23.4.7 Id link_getSpace2 (Link *)

5.23.4.8 LinkStatus link_getStatus (Link *)

5.23.4.9 STATUS link_print (Link *)

5.23.4.10 STATUS link_setDirection (Link *, int)

5.23.4.11 STATUS link_setId (Link *, Id)

5.23.4.12 STATUS link_setSpaces (Link *, Id, Id)

5.23.4.13 STATUS link_setStatus (Link *, LinkStatus)

5.24 link_test.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "../include/link.h"
#include "../include/link_test.h"
#include "../include/test.h"
Include dependency graph for link_test.c:
```

Macros

- #define MAX_TESTS 14

Functions

- int [main](#) (int argc, char **argv)
Funcion principal de pruebas para el modulo Space.
- void [test1_link_create](#) ()
- void [test2_link_create](#) ()
- void [test1_link_set_id](#) ()
- void [test2_link_set_id](#) ()
- void [test1_link_set_status](#) ()
- void [test2_link_set_status](#) ()
- void [test1_link_set_spaces](#) ()
- void [test2_link_set_spaces](#) ()
- void [test1_link_get_id](#) ()
- void [test1_link_get_space1](#) ()
- void [test1_link_get_space2](#) ()
- void [test1_link_get_status](#) ()
- void [test1_link_get_destination](#) ()

5.24.1 Macro Definition Documentation

5.24.1.1 #define MAX_TESTS 14

5.24.2 Function Documentation

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

Funcion principal de pruebas para el modulo Space.

Dos modos de ejecucion: 1.-Si se ejecuta sin parametros se ejecutan todas las pruebas 2.-Si se ejecuta con un numero entre 1 y el numero de pruebas solo ejecuta la prueba indicada

5.24.2.2 void test1_link_create ()

5.24.2.3 void test1_link_get_destination ()

5.24.2.4 void test1_link_get_id ()

5.24.2.5 void test1_link_get_space1 ()

5.24.2.6 void test1_link_get_space2 ()

5.24.2.7 void test1_link_get_status ()

5.24.2.8 void test1_link_set_id ()

5.24.2.9 void test1_link_set_spaces ()

5.24.2.10 void test1_link_set_status ()

5.24.2.11 void test2_link_create ()

5.24.2.12 void test2_link_set_id ()

5.24.2.13 void test2_link_set_spaces ()

5.24.2.14 void test2_link_set_status ()

5.25 link_test.h File Reference

It declares the tests for the link module.

This graph shows which files directly or indirectly include this file:

Functions

- void [test1_link_create](#) ()
- void [test2_link_create](#) ()
- void [test1_link_set_id](#) ()
- void [test2_link_set_id](#) ()
- void [test1_link_set_status](#) ()
- void [test2_link_set_status](#) ()
- void [test1_link_set_spaces](#) ()
- void [test2_link_set_spaces](#) ()
- void [test1_link_get_id](#) ()
- void [test1_link_get_space1](#) ()
- void [test1_link_get_space2](#) ()
- void [test1_link_get_status](#) ()
- void [test1_link_get_destination](#) ()

5.25.1 Detailed Description

It declares the tests for the link module.

Author

Pablo Sánchez Redondo

Copyright

GNU Public License

5.25.2 Function Documentation

5.25.2.1 void test1_link_create ()

5.25.2.2 void test1_link_get_destination ()

5.25.2.3 void test1_link_get_id ()

5.25.2.4 void test1_link_get_space1 ()

5.25.2.5 void test1_link_get_space2 ()

5.25.2.6 void test1_link_get_status ()

5.25.2.7 void test1_link_set_id ()

5.25.2.8 void test1_link_set_spaces ()

5.25.2.9 void test1_link_set_status ()

5.25.2.10 void test2_link_create ()

5.25.2.11 void test2_link_set_id ()

5.25.2.12 void test2_link_set_spaces ()

5.25.2.13 void test2_link_set_status ()

5.26 object.c File Reference

Functions for the creation of objects.

```
#include <string.h>
#include "../include/object.h"
Include dependency graph for object.c:
```

Classes

- [struct _Object](#)

Functions

- `Object * object_create` (char *name, `Id` id, `BOOL` mobile, `BOOL` hidden, `Id` open, `BOOL` lights, `BOOL` on)
- `void object_destroy` (`Object` *obj)
- `STATUS object_set_name` (`Object` *obj, char *name)
- `STATUS object_set_description` (`Object` *obj, char *description)
- `STATUS object_set_id` (`Object` *obj, `Id` id)
- `char * object_get_name` (`Object` *obj)
- `char * object_get_description` (`Object` *obj)
- `STATUS object_description_print` (`Object` *obj, `FILE` *f)
- `Id object_get_id` (`Object` *obj)
- `BOOL object_get_mobile` (`Object` *obj)
- `STATUS object_set_mobile` (`Object` *obj, `BOOL` mobile)
- `BOOL object_get_moved` (`Object` *obj)
- `STATUS object_set_moved` (`Object` *obj, `BOOL` moved)
- `BOOL object_get_hidden` (`Object` *obj)
- `STATUS object_set_hidden` (`Object` *obj, `BOOL` hidden)
- `BOOL object_get_iluminati` (`Object` *obj)
- `STATUS object_set_ilumnati` (`Object` *obj, `BOOL` iluminati)
- `Id object_get_open` (`Object` *obj)
- `STATUS object_set_open` (`Object` *obj, `Id` open)
- `BOOL object_get_on` (`Object` *obj)
- `STATUS object_set_on` (`Object` *obj, `BOOL` on)
- `char * object_get_description_alternative` (`Object` *obj)
- `STATUS object_set_description_alternative` (`Object` *obj, char *description_al)
- `STATUS object_description_al_print` (`Object` *obj, `FILE` *f)

5.26.1 Detailed Description

Functions for the creation of objects.

Author

Pablo Snchez y Guillermo Ros

Copyright

GNU Public License

5.26.2 Function Documentation

5.26.2.1 `Object* object_create` (char * name, `Id` id, `BOOL` mobile, `BOOL` hidden, `Id` open, `BOOL` lights, `BOOL` on)

5.26.2.2 `STATUS object_description_al_print` (`Object` * obj, `FILE` * f)

5.26.2.3 `STATUS object_description_print` (`Object` * obj, `FILE` * f)

5.26.2.4 `void object_destroy` (`Object` * obj)

- 5.26.2.5 `char* object_get_description (Object * obj)`
- 5.26.2.6 `char* object_get_description_alternative (Object * obj)`
- 5.26.2.7 `BOOL object_get_hidden (Object * obj)`
- 5.26.2.8 `Id object_get_id (Object * obj)`
- 5.26.2.9 `BOOL object_get_iluminati (Object * obj)`
- 5.26.2.10 `BOOL object_get_mobile (Object * obj)`
- 5.26.2.11 `BOOL object_get_moved (Object * obj)`
- 5.26.2.12 `char* object_get_name (Object * obj)`
- 5.26.2.13 `BOOL object_get_on (Object * obj)`
- 5.26.2.14 `Id object_get_open (Object * obj)`
- 5.26.2.15 `STATUS object_set_description (Object * obj, char * description)`
- 5.26.2.16 `STATUS object_set_description_alternative (Object * obj, char * description_al)`
- 5.26.2.17 `STATUS object_set_hidden (Object * obj, BOOL hidden)`
- 5.26.2.18 `STATUS object_set_id (Object * obj, Id id)`
- 5.26.2.19 `STATUS object_set_ilumnati (Object * obj, BOOL iluminati)`
- 5.26.2.20 `STATUS object_set_mobile (Object * obj, BOOL mobile)`
- 5.26.2.21 `STATUS object_set_moved (Object * obj, BOOL moved)`
- 5.26.2.22 `STATUS object_set_name (Object * obj, char * name)`
- 5.26.2.23 `STATUS object_set_on (Object * obj, BOOL on)`
- 5.26.2.24 `STATUS object_set_open (Object * obj, Id open)`

5.27 object.h File Reference

Functions for the creation of objects.

```
#include <stdio.h>
#include <stdlib.h>
#include "types.h"
Include dependency graph for object.h:
```

This graph shows which files directly or indirectly include this file:

Typedefs

- typedef struct [_Object](#) Object

Functions

- [Object *](#) [object_create](#) (char *name, [Id](#) id, [BOOL](#) mobile, [BOOL](#) hidden, [Id](#) open, [BOOL](#) lights, [BOOL](#) on)
- void [object_destroy](#) ([Object](#) *obj)
- [STATUS](#) [object_set_name](#) ([Object](#) *obj, char *name)
- [STATUS](#) [object_set_description](#) ([Object](#) *obj, char *descript)
- [STATUS](#) [object_set_id](#) ([Object](#) *obj, [Id](#) id)
- char * [object_get_name](#) ([Object](#) *obj)
- char * [object_get_description](#) ([Object](#) *obj)
- [STATUS](#) [object_description_print](#) ([Object](#) *obj, FILE *f)
- [Id](#) [object_get_id](#) ([Object](#) *obj)
- [BOOL](#) [object_get_mobile](#) ([Object](#) *obj)
- [STATUS](#) [object_set_mobile](#) ([Object](#) *obj, [BOOL](#) mobile)
- [BOOL](#) [object_get_moved](#) ([Object](#) *obj)
- [STATUS](#) [object_set_moved](#) ([Object](#) *obj, [BOOL](#) moved)
- [BOOL](#) [object_get_hidden](#) ([Object](#) *obj)
- [STATUS](#) [object_set_hidden](#) ([Object](#) *obj, [BOOL](#) hidden)
- [BOOL](#) [object_get_iluminati](#) ([Object](#) *obj)
- [STATUS](#) [object_set_ilumnati](#) ([Object](#) *obj, [BOOL](#) iluminati)
- [Id](#) [object_get_open](#) ([Object](#) *obj)
- [STATUS](#) [object_set_open](#) ([Object](#) *obj, [Id](#) open)
- [BOOL](#) [object_get_on](#) ([Object](#) *obj)
- [STATUS](#) [object_set_on](#) ([Object](#) *obj, [BOOL](#) on)
- char * [object_get_description_alternative](#) ([Object](#) *obj)
- [STATUS](#) [object_set_description_alternative](#) ([Object](#) *obj, char *description_al)
- [STATUS](#) [object_description_al_print](#) ([Object](#) *obj, FILE *f)

5.27.1 Detailed Description

Functions for the creation of objects.

Author

Guillermo Ros

Copyright

GNU Public License

5.27.2 Typedef Documentation

5.27.2.1 typedef struct _Object Object

5.27.3 Function Documentation

5.27.3.1 **Object*** object_create (char * *name*, Id *id*, **BOOL** *mobile*, **BOOL** *hidden*, Id *open*, **BOOL** *lights*, **BOOL** *on*)

5.27.3.2 **STATUS** object_description_al_print (**Object** * *obj*, FILE * *f*)

5.27.3.3 **STATUS** object_description_print (**Object** * *obj*, FILE * *f*)

5.27.3.4 void object_destroy (**Object** * *obj*)

5.27.3.5 char* object_get_description (**Object** * *obj*)

5.27.3.6 char* object_get_description_alternative (**Object** * *obj*)

5.27.3.7 **BOOL** object_get_hidden (**Object** * *obj*)

5.27.3.8 Id object_get_id (**Object** * *obj*)

5.27.3.9 **BOOL** object_get_iluminati (**Object** * *obj*)

5.27.3.10 **BOOL** object_get_mobile (**Object** * *obj*)

5.27.3.11 **BOOL** object_get_moved (**Object** * *obj*)

5.27.3.12 char* object_get_name (**Object** * *obj*)

5.27.3.13 **BOOL** object_get_on (**Object** * *obj*)

5.27.3.14 Id object_get_open (**Object** * *obj*)

5.27.3.15 **STATUS** object_set_description (**Object** * *obj*, char * *descript*)

5.27.3.16 **STATUS** object_set_description_alternative (**Object** * *obj*, char * *description_al*)

5.27.3.17 **STATUS** object_set_hidden (**Object** * *obj*, **BOOL** *hidden*)

5.27.3.18 **STATUS** object_set_id (**Object** * *obj*, Id *id*)

5.27.3.19 **STATUS** object_set_ilumnati (**Object** * *obj*, **BOOL** *iluminati*)

5.27.3.20 **STATUS** object_set_mobile (**Object** * *obj*, **BOOL** *mobile*)

5.27.3.21 **STATUS** `object_set_moved (Object * obj, BOOL moved)`

5.27.3.22 **STATUS** `object_set_name (Object * obj, char * name)`

5.27.3.23 **STATUS** `object_set_on (Object * obj, BOOL on)`

5.27.3.24 **STATUS** `object_set_open (Object * obj, Id open)`

5.28 object_test.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "../include/object.h"
#include "../include/object_test.h"
#include "../include/test.h"
Include dependency graph for object_test.c:
```

Macros

- `#define MAX_TESTS 28`

Functions

- `int main (int argc, char **argv)`
Funcion principal de pruebas para el modulo Space.
- `void test1_object_create ()`
- `void test2_object_create ()`
- `void test1_object_set_name ()`
- `void test2_object_set_name ()`
- `void test1_object_set_id ()`
- `void test2_object_set_id ()`
- `void test1_object_set_description ()`
- `void test2_object_set_description ()`
- `void test1_object_get_name ()`
- `void test1_object_get_description ()`
- `void test1_object_get_id ()`

5.28.1 Macro Definition Documentation

5.28.1.1 `#define MAX_TESTS 28`

5.28.2 Function Documentation

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

Funcion principal de pruebas para el modulo Space.

Dos modos de ejecucion: 1.-Si se ejecuta sin parametros se ejecutan todas las pruebas 2.-Si se ejecuta con un numero entre 1 y el numero de pruebas solo ejecuta la prueba indicada

5.28.2.2 void test1_object_create ()

5.28.2.3 void test1_object_get_description ()

5.28.2.4 void test1_object_get_id ()

5.28.2.5 void test1_object_get_name ()

5.28.2.6 void test1_object_set_description ()

5.28.2.7 void test1_object_set_id ()

5.28.2.8 void test1_object_set_name ()

5.28.2.9 void test2_object_create ()

5.28.2.10 void test2_object_set_description ()

5.28.2.11 void test2_object_set_id ()

5.28.2.12 void test2_object_set_name ()

5.29 object_test.h File Reference

It declares the tests for the object module.

This graph shows which files directly or indirectly include this file:

Functions

- void [test1_object_create](#) ()
- void [test2_object_create](#) ()
- void [test1_object_set_name](#) ()
- void [test2_object_set_name](#) ()
- void [test1_object_set_id](#) ()
- void [test2_object_set_id](#) ()
- void [test1_object_set_description](#) ()
- void [test2_object_set_description](#) ()
- void [test1_object_get_name](#) ()
- void [test1_object_get_description](#) ()
- void [test1_object_get_id](#) ()

5.29.1 Detailed Description

It declares the tests for the object module.

Author

Pablo Sánchez Redondo

Copyright

GNU Public License

5.29.2 Function Documentation

5.29.2.1 void test1_object_create ()

5.29.2.2 void test1_object_get_description ()

5.29.2.3 void test1_object_get_id ()

5.29.2.4 void test1_object_get_name ()

5.29.2.5 void test1_object_set_description ()

5.29.2.6 void test1_object_set_id ()

5.29.2.7 void test1_object_set_name ()

5.29.2.8 void test2_object_create ()

5.29.2.9 void test2_object_set_description ()

5.29.2.10 void test2_object_set_id ()

5.29.2.11 void test2_object_set_name ()

5.30 Objectives_Tracker.txt File Reference

Variables

- [POINTS](#)

5.30.1 Variable Documentation

5.30.1.1 POINTS

5.31 player.c File Reference

Functions for the creation of players.

```
#include <string.h>
#include "../include/player.h"
#include "../include/object.h"
#include "../include/set.h"
#include "../include/inventory.h"
Include dependency graph for player.c:
```

Classes

- [struct _Player](#)

Functions

- [Player * player_create](#) (char *name, [Id](#) location_id, [Id](#) object_id, [Id](#) id)
- void [player_destroy](#) ([Player](#) *player)
- [STATUS](#) [player_setName](#) ([Player](#) *player, char *newName)
- [STATUS](#) [player_setLocId](#) ([Player](#) *player, [Id](#) new_locId)
- [STATUS](#) [player_setObjId](#) ([Player](#) *player, [Id](#) new_objId)
- [STATUS](#) [player_setId](#) ([Player](#) *player, [Id](#) new_id)
- char * [player_getName](#) ([Player](#) *player)
- [Id](#) [player_getLocId](#) ([Player](#) *player)
- [Id](#) [player_getObjId](#) ([Player](#) *player, int num)
- [Id](#) [player_getId](#) ([Player](#) *player)
- [STATUS](#) [player_removeObjId](#) ([Player](#) *player, [Id](#) id)

5.31.1 Detailed Description

Functions for the creation of players.

Author

Antonio Solana

Copyright

GNU Public License

5.31.2 Function Documentation

5.31.2.1 **Player*** `player_create (char * name, Id location_id, Id object_id, Id id)`

Returns null if no name is given to the player Returns pointer to the newly created player if ok

5.31.2.2 **void** `player_destroy (Player * player)`

5.31.2.3 **Id** `player_getId (Player * player)`

5.31.2.4 **Id** `player_getLocId (Player * player)`

5.31.2.5 **char*** `player_getName (Player * player)`

5.31.2.6 **Id** `player_getObjId (Player * player, int num)`

5.31.2.7 **STATUS** `player_removeObjId (Player * player, Id id)`

5.31.2.8 **STATUS** `player_setId (Player * player, Id new_id)`

5.31.2.9 **STATUS** `player_setLocId (Player * player, Id new_locId)`

5.31.2.10 **STATUS** `player_setName (Player * player, char * newName)`

5.31.2.11 **STATUS** `player_setObjId (Player * player, Id new_objId)`

5.32 player.h File Reference

Functions for the creation of players.

```
#include <stdio.h>
#include <stdlib.h>
#include "types.h"
Include dependency graph for player.h:
```

This graph shows which files directly or indirectly include this file:

Typedefs

- typedef struct [_Player](#) [Player](#)

Functions

- `Player * player_create (char *, Id, Id, Id)`
- `void player_destroy (Player *)`
- `STATUS player_setName (Player *, char *)`
- `STATUS player_setLocId (Player *, Id)`
- `STATUS player_setObjId (Player *, Id)`
- `STATUS player_setId (Player *, Id)`
- `char * player_getName (Player *)`
- `Id player_getLocId (Player *)`
- `Id player_getObjId (Player *, int)`
- `Id player_getId (Player *)`
- `STATUS player_removeObjId (Player *, Id)`

5.32.1 Detailed Description

Functions for the creation of players.

Author

Guillermo Ríos

Copyright

GNU Public License

5.32.2 Typedef Documentation

5.32.2.1 typedef struct _Player Player

5.32.3 Function Documentation

5.32.3.1 `Player* player_create (char *, Id , Id , Id)`

Returns null if no name is given to the player Returns pointer to the newly created player if ok

5.32.3.2 void player_destroy (Player *)

5.32.3.3 Id player_getId (Player *)

5.32.3.4 Id player_getLocId (Player *)

5.32.3.5 char* player_getName (Player *)

5.32.3.6 Id player_getObjId (Player *, int)

5.32.3.7 STATUS player_removeObjId (Player *, Id)

5.32.3.8 STATUS player_setId (Player *, Id)

5.32.3.9 STATUS player_setLocId (Player *, Id)

5.32.3.10 STATUS player_setName (Player *, char *)

5.32.3.11 STATUS player_setObjId (Player *, Id)

5.33 player_test.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "../include/player.h"
#include "../include/player_test.h"
#include "../include/test.h"
Include dependency graph for player_test.c:
```

Macros

- #define MAX_TESTS 16

Functions

- int main (int argc, char **argv)
Funcion principal de pruebas para el modulo Space.
- void test1_player_create ()
- void test1_player_set_name ()
- void test2_player_set_name ()
- void test1_player_set_LocId ()
- void test2_player_set_LocId ()
- void test1_player_set_ObjId ()
- void test2_player_set_ObjId ()
- void test1_player_set_id ()
- void test2_player_set_id ()
- void test1_player_get_name ()
- void test1_player_get_LocId ()
- void test1_player_get_ObjId ()
- void test1_player_get_id ()
- void test1_player_remove_object_id ()
- void test2_player_remove_object_id ()

5.33.1 Macro Definition Documentation

5.33.1.1 `#define MAX_TESTS 16`

5.33.2 Function Documentation

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

Funcion principal de pruebas para el modulo Space.

Dos modos de ejecucion: 1.-Si se ejecuta sin parametros se ejecutan todas las pruebas 2.-Si se ejecuta con un numero entre 1 y el numero de pruebas solo ejecuta la prueba indicada

5.33.2.2 `void test1_player_create ()`

5.33.2.3 `void test1_player_get_id ()`

5.33.2.4 `void test1_player_get_LocId ()`

5.33.2.5 `void test1_player_get_name ()`

5.33.2.6 `void test1_player_get_ObjId ()`

5.33.2.7 `void test1_player_remove_object_id ()`

5.33.2.8 `void test1_player_set_id ()`

5.33.2.9 `void test1_player_set_LocId ()`

5.33.2.10 `void test1_player_set_name ()`

5.33.2.11 `void test1_player_set_ObjId ()`

5.33.2.12 `void test2_player_remove_object_id ()`

5.33.2.13 `void test2_player_set_id ()`

5.33.2.14 `void test2_player_set_LocId ()`

5.33.2.15 `void test2_player_set_name ()`

5.33.2.16 `void test2_player_set_ObjId ()`

5.34 player_test.h File Reference

It declares the tests for the player module.

This graph shows which files directly or indirectly include this file:

Functions

- void [test1_player_create](#) ()
- void [test1_player_set_name](#) ()
- void [test2_player_set_name](#) ()
- void [test1_player_set_LocId](#) ()
- void [test2_player_set_LocId](#) ()
- void [test1_player_set_ObjId](#) ()
- void [test2_player_set_ObjId](#) ()
- void [test1_player_set_id](#) ()
- void [test2_player_set_id](#) ()
- void [test1_player_get_name](#) ()
- void [test1_player_get_LocId](#) ()
- void [test1_player_get_ObjId](#) ()
- void [test1_player_get_id](#) ()
- void [test1_player_remove_object_id](#) ()
- void [test2_player_remove_object_id](#) ()

5.34.1 Detailed Description

It declares the tests for the player module.

Author

Pablo Sánchez Redondo

Copyright

GNU Public License

5.34.2 Function Documentation

5.34.2.1 void [test1_player_create](#) ()

5.34.2.2 void [test1_player_get_id](#) ()

5.34.2.3 void [test1_player_get_LocId](#) ()

5.34.2.4 void [test1_player_get_name](#) ()

5.34.2.5 void [test1_player_get_ObjId](#) ()

5.34.2.6 void [test1_player_remove_object_id](#) ()

5.34.2.7 void [test1_player_set_id](#) ()

5.34.2.8 void [test1_player_set_LocId](#) ()

5.34.2.9 void test1_player_set_name ()

5.34.2.10 void test1_player_set_ObjId ()

5.34.2.11 void test2_player_remove_object_id ()

5.34.2.12 void test2_player_set_id ()

5.34.2.13 void test2_player_set_LocId ()

5.34.2.14 void test2_player_set_name ()

5.34.2.15 void test2_player_set_ObjId ()

5.35 screen.c File Reference

Functions for the creation of players.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "../include/screen.h"
Include dependency graph for screen.c:
```

Classes

- struct [_Area](#)

Macros

- #define [ROWS](#) 29
- #define [COLUMNS](#) 75
- #define [TOTAL_DATA](#) ([ROWS](#) * [COLUMNS](#)) + 1
- #define [BG_CHAR](#) '~'
- #define [FG_CHAR](#) ''
- #define [PROMPT](#) " prompt:> "
- #define [ACCESS](#)(d, x, y) (d + ((y)*[COLUMNS](#)) + (x))

Functions

- int [screen_area_cursor_is_out_of_bounds](#) ([Area](#) *area)
- void [screen_area_scroll_up](#) ([Area](#) *area)
- void [screen_utils_replaces_special_chars](#) (char *str)
- void [screen_init](#) ()
- void [screen_destroy](#) ()
- void [screen_paint](#) ()
- void [screen_gets](#) (char *str)
- [Area](#) * [screen_area_init](#) (int x, int y, int width, int height)
- void [screen_area_destroy](#) ([Area](#) *area)
- void [screen_area_clear](#) ([Area](#) *area)
- void [screen_area_reset_cursor](#) ([Area](#) *area)
- void [screen_area_puts](#) ([Area](#) *area, char *str)

Variables

- char * [__data](#)

5.35.1 Detailed Description

Functions for the creation of players.

Author

Profesores Pprog

Copyright

GNU Public License

5.35.2 Macro Definition Documentation

5.35.2.1 `#define ACCESS(d, x, y) (d + ((y)*COLUMNS) + (x))`

5.35.2.2 `#define BG_CHAR '~'`

5.35.2.3 `#define COLUMNS 75`

5.35.2.4 `#define FG_CHAR ' '`

5.35.2.5 `#define PROMPT " prompt:> "`

5.35.2.6 `#define ROWS 29`

5.35.2.7 `#define TOTAL_DATA (ROWS * COLUMNS) + 1`

5.35.3 Function Documentation

5.35.3.1 `void screen_area_clear (Area * area)`

5.35.3.2 `int screen_area_cursor_is_out_of_bounds (Area * area)`

5.35.3.3 `void screen_area_destroy (Area * area)`

5.35.3.4 `Area* screen_area_init (int x, int y, int width, int height)`

5.35.3.5 `void screen_area_puts (Area * area, char * str)`

5.35.3.6 `void screen_area_reset_cursor (Area * area)`

5.35.3.7 void `screen_area_scroll_up` (`Area * area`)

5.35.3.8 void `screen_destroy` ()

5.35.3.9 void `screen_gets` (`char * str`)

5.35.3.10 void `screen_init` ()

5.35.3.11 void `screen_paint` ()

5.35.3.12 void `screen_utils_replaces_special_chars` (`char * str`)

5.35.4 Variable Documentation

5.35.4.1 `char* __data`

5.36 screen.h File Reference

Functions used by `graphic_engine.*`.

This graph shows which files directly or indirectly include this file:

Macros

- `#define SCREEN_MAX_STR` 80

Typedefs

- `typedef struct _Area Area`

Functions

- void `screen_init` ()
- void `screen_destroy` ()
- void `screen_paint` ()
- void `screen_gets` (`char *str`)
- `Area * screen_area_init` (`int x`, `int y`, `int width`, `int height`)
- void `screen_area_destroy` (`Area *area`)
- void `screen_area_clear` (`Area *area`)
- void `screen_area_reset_cursor` (`Area *area`)
- void `screen_area_puts` (`Area *area`, `char *str`)

5.36.1 Detailed Description

Functions used by `graphic_engine.*`.

Author

Profesores Pprog

Copyright

GNU Public License

5.36.2 Macro Definition Documentation

5.36.2.1 `#define SCREEN_MAX_STR 80`

5.36.3 Typedef Documentation

5.36.3.1 `typedef struct _Area Area`

5.36.4 Function Documentation

5.36.4.1 `void screen_area_clear (Area * area)`

5.36.4.2 `void screen_area_destroy (Area * area)`

5.36.4.3 `Area* screen_area_init (int x, int y, int width, int height)`

5.36.4.4 `void screen_area_puts (Area * area, char * str)`

5.36.4.5 `void screen_area_reset_cursor (Area * area)`

5.36.4.6 `void screen_destroy ()`

5.36.4.7 `void screen_gets (char * str)`

5.36.4.8 `void screen_init ()`

5.36.4.9 `void screen_paint ()`

5.37 set.c File Reference

Low level stack and queue functions.

```
#include "../include/set.h"
```

Include dependency graph for `set.c`:

Classes

- [struct _Set](#)

Functions

- [Set * set_create](#) (int *inv_size*)
- [void set_destroy](#) ([Set](#) **set*)
- [STATUS set_add](#) ([Set](#) **set*, [Id](#) *id*)
- [STATUS set_del](#) ([Set](#) **set*, [Id](#) *id*)
- [Id set_get_id](#) ([Set](#) **set*, int *num*)
- [STATUS set_rm_all](#) ([Set](#) **set*)
- [Set * set_cp_all](#) ([Set](#) **set*)
- [STATUS set_rearrange](#) ([Set](#) **set*)
- [STATUS set_print_debug](#) (FILE **f*, [Set](#) **set*)

5.37.1 Detailed Description

Low level stack and queue functions.

Author

Bernardo Zambrano

Copyright

GNU Public License

5.37.2 Function Documentation

5.37.2.1 [STATUS set_add](#) ([Set](#) * *set*, [Id](#) *id*)

5.37.2.2 [Set*](#) [set_cp_all](#) ([Set](#) * *set*)

5.37.2.3 [Set*](#) [set_create](#) (int *inv_size*)

5.37.2.4 [STATUS set_del](#) ([Set](#) * *set*, [Id](#) *id*)

5.37.2.5 [void set_destroy](#) ([Set](#) * *set*)

5.37.2.6 [Id set_get_id](#) ([Set](#) * *set*, int *num*)

5.37.2.7 [STATUS set_print_debug](#) (FILE * *f*, [Set](#) * *set*)

5.37.2.8 [STATUS set_rearrange](#) ([Set](#) * *set*)

5.37.2.9 [STATUS set_rm_all](#) ([Set](#) * *set*)

5.38 set.h File Reference

Low level stack and queue functions.

```
#include <stdio.h>
#include <stdlib.h>
#include "types.h"
Include dependency graph for set.h:
```

This graph shows which files directly or indirectly include this file:

Macros

- `#define MAX_INV_SIZE 1024`

Typedefs

- `typedef struct _Set Set`

Functions

- `Set * set_create (int)`
- `void set_destroy (Set *)`
- `STATUS set_add (Set *, Id)`
- `STATUS set_del (Set *, Id)`
- `Id set_get_id (Set *, int)`
- `STATUS set_rm_all (Set *)`
- `STATUS set_rearrange (Set *)`
- `Set * set_cp_all (Set *)`
- `STATUS set_print_debug (FILE *, Set *)`

5.38.1 Detailed Description

Low level stack and queue functions.

Author

Bernardo Zambrano

Copyright

GNU Public License

5.38.2 Macro Definition Documentation

5.38.2.1 `#define MAX_INV_SIZE 1024`

5.38.3 Typedef Documentation

5.38.3.1 `typedef struct _Set Set`

5.38.4 Function Documentation

5.38.4.1 `STATUS set_add (Set *, Id)`

5.38.4.2 `Set* set_cp_all (Set *)`

5.38.4.3 `Set* set_create (int)`

5.38.4.4 `STATUS set_del (Set *, Id)`

5.38.4.5 `void set_destroy (Set *)`

5.38.4.6 `Id set_get_id (Set *, int)`

5.38.4.7 `STATUS set_print_debug (FILE *, Set *)`

5.38.4.8 `STATUS set_rearrange (Set *)`

5.38.4.9 `STATUS set_rm_all (Set *)`

5.39 set_test.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "../include/set.h"
#include "../include/set_test.h"
#include "../include/test.h"
Include dependency graph for set_test.c:
```

Macros

- `#define MAX_TESTS 28`

Functions

- int `main` (int argc, char **argv)
Funcion principal de pruebas para el modulo Space.
- void `test1_set_create` ()
- void `test1_set_add` ()
- void `test2_set_add` ()
- void `test1_set_del` ()
- void `test2_set_del` ()
- void `test1_set_get_id` ()
- void `test2_set_get_id` ()
- void `test1_set_rm_all` ()
- void `test2_set_rm_all` ()
- void `test1_set_rearrange` ()
- void `test2_set_rearrange` ()
- void `test1_set_cp_all` ()
- void `test2_set_cp_all` ()

5.39.1 Macro Definition Documentation

5.39.1.1 `#define MAX_TESTS 28`

5.39.2 Function Documentation

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

Funcion principal de pruebas para el modulo Space.

Dos modos de ejecucion: 1.-Si se ejecuta sin parametros se ejecutan todas las pruebas 2.-Si se ejecuta con un numero entre 1 y el numero de pruebas solo ejecuta la prueba indicada

5.39.2.2 `void test1_set_add ()`

5.39.2.3 `void test1_set_cp_all ()`

5.39.2.4 `void test1_set_create ()`

5.39.2.5 `void test1_set_del ()`

5.39.2.6 `void test1_set_get_id ()`

5.39.2.7 `void test1_set_rearrange ()`

5.39.2.8 `void test1_set_rm_all ()`

5.39.2.9 `void test2_set_add ()`

5.39.2.10 void test2_set_cp_all ()

5.39.2.11 void test2_set_del ()

5.39.2.12 void test2_set_get_id ()

5.39.2.13 void test2_set_rearrange ()

5.39.2.14 void test2_set_rm_all ()

5.40 set_test.h File Reference

It declares the tests for the set module.

This graph shows which files directly or indirectly include this file:

Functions

- void [test1_set_create](#) ()
- void [test1_set_add](#) ()
- void [test2_set_add](#) ()
- void [test1_set_del](#) ()
- void [test2_set_del](#) ()
- void [test1_set_get_id](#) ()
- void [test2_set_get_id](#) ()
- void [test1_set_rm_all](#) ()
- void [test2_set_rm_all](#) ()
- void [test1_set_rearrange](#) ()
- void [test2_set_rearrange](#) ()
- void [test1_set_cp_all](#) ()
- void [test2_set_cp_all](#) ()

5.40.1 Detailed Description

It declares the tests for the set module.

Author

Pablo Sánchez Redondo

Copyright

GNU Public License

5.40.2 Function Documentation

5.40.2.1 void test1_set_add ()

5.40.2.2 void test1_set_cp_all ()

5.40.2.3 void test1_set_create ()

5.40.2.4 void test1_set_del ()

5.40.2.5 void test1_set_get_id ()

5.40.2.6 void test1_set_rearrange ()

5.40.2.7 void test1_set_rm_all ()

5.40.2.8 void test2_set_add ()

5.40.2.9 void test2_set_cp_all ()

5.40.2.10 void test2_set_del ()

5.40.2.11 void test2_set_get_id ()

5.40.2.12 void test2_set_rearrange ()

5.40.2.13 void test2_set_rm_all ()

5.41 space.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "../include/types.h"
#include "../include/space.h"
#include "../include/set.h"
```

Include dependency graph for space.c:

Classes

- [struct _Space](#)

Functions

- `Space * space_create (Id id)`
- `STATUS space_destroy (Space *space)`
- `STATUS space_set_name (Space *space, char *name)`
- `STATUS space_setSprite (Space *space, Id spriteId, int i)`
- `STATUS space_setCurrentSprite (Space *space, int i)`
- `STATUS space_set_description (Space *space, char *description)`
- `STATUS space_set_north (Space *space, Id id)`
- `STATUS space_set_south (Space *space, Id id)`
- `STATUS space_set_east (Space *space, Id id)`
- `STATUS space_set_west (Space *space, Id id)`
- `STATUS space_set_up (Space *space, Id id)`
- `STATUS space_set_down (Space *space, Id id)`
- `STATUS space_set_light (Space *space, BOOL light)`
- `BOOL space_get_light (Space *space)`
- `STATUS space_add_object (Space *space, Id obj_id)`
- `STATUS space_remove_object (Space *space, Id id)`
- `Id space_getSprite (Space *space, int i)`
- `int space_getCurentSprite (Space *space)`
- `const char * space_get_name (Space *space)`
- `const char * space_get_description (Space *space)`
- `Id space_get_id (Space *space)`
- `Id space_get_north (Space *space)`
- `Id space_get_south (Space *space)`
- `Id space_get_east (Space *space)`
- `Id space_get_west (Space *space)`
- `Id space_get_up (Space *space)`
- `Id space_get_down (Space *space)`
- `Set * space_get_objects_id (Space *space)`
- `STATUS space_light_print (Space *space)`
- `STATUS space_print (Space *space)`
- `STATUS space_set_gdesc_0 (Space *space, char *cadena)`
- `STATUS space_set_gdesc_1 (Space *space, char *cadena)`
- `STATUS space_set_gdesc_2 (Space *space, char *cadena)`
- `char * space_get_gdesc_0 (Space *space)`
- `char * space_get_gdesc_1 (Space *space)`
- `char * space_get_gdesc_2 (Space *space)`

5.41.1 Function Documentation

5.41.1.1 `STATUS space_add_object (Space * space, Id obj_id)`

5.41.1.2 `Space* space_create (Id id)`

5.41.1.3 `STATUS space_destroy (Space * space)`

5.41.1.4 `const char* space_get_description (Space * space)`

5.41.1.5 `Id space_get_down (Space * space)`

- 5.41.1.6 **Id** space_get_east (**Space** * *space*)
- 5.41.1.7 **char*** space_get_gdesc_0 (**Space** * *space*)
- 5.41.1.8 **char*** space_get_gdesc_1 (**Space** * *space*)
- 5.41.1.9 **char*** space_get_gdesc_2 (**Space** * *space*)
- 5.41.1.10 **Id** space_get_id (**Space** * *space*)
- 5.41.1.11 **BOOL** space_get_light (**Space** * *space*)
- 5.41.1.12 **const char*** space_get_name (**Space** * *space*)
- 5.41.1.13 **Id** space_get_north (**Space** * *space*)
- 5.41.1.14 **Set*** space_get_objects_id (**Space** * *space*)
- 5.41.1.15 **Id** space_get_south (**Space** * *space*)
- 5.41.1.16 **Id** space_get_up (**Space** * *space*)
- 5.41.1.17 **Id** space_get_west (**Space** * *space*)
- 5.41.1.18 **int** space_getCurentSprite (**Space** * *space*)
- 5.41.1.19 **Id** space_getSprite (**Space** * *space*, **int** *i*)
- 5.41.1.20 **STATUS** space_light_print (**Space** * *space*)
- 5.41.1.21 **STATUS** space_print (**Space** * *space*)
- 5.41.1.22 **STATUS** space_remove_object (**Space** * *space*, **Id** *id*)
- 5.41.1.23 **STATUS** space_set_description (**Space** * *space*, **char** * *description*)
- 5.41.1.24 **STATUS** space_set_down (**Space** * *space*, **Id** *id*)
- 5.41.1.25 **STATUS** space_set_east (**Space** * *space*, **Id** *id*)
- 5.41.1.26 **STATUS** space_set_gdesc_0 (**Space** * *space*, **char** * *cadena*)
- 5.41.1.27 **STATUS** space_set_gdesc_1 (**Space** * *space*, **char** * *cadena*)
- 5.41.1.28 **STATUS** space_set_gdesc_2 (**Space** * *space*, **char** * *cadena*)

5.41.1.29 STATUS `space_set_light (Space * space, BOOL light)`

5.41.1.30 STATUS `space_set_name (Space * space, char * name)`

5.41.1.31 STATUS `space_set_north (Space * space, Id id)`

5.41.1.32 STATUS `space_set_south (Space * space, Id id)`

5.41.1.33 STATUS `space_set_up (Space * space, Id id)`

5.41.1.34 STATUS `space_set_west (Space * space, Id id)`

5.41.1.35 STATUS `space_setCurrentSprite (Space * space, int i)`

5.41.1.36 STATUS `space_setSprite (Space * space, Id sprited, int i)`

5.42 space.h File Reference

Defines functions for space manipulation.

```
#include "../include/types.h"
#include "../include/object.h"
#include "../include/set.h"
Include dependency graph for space.h:
```

This graph shows which files directly or indirectly include this file:

Macros

- `#define MAX_SPACES 100`
- `#define FIRST_SPACE 1`

Typedefs

- `typedef struct _Space Space`

Functions

- `Space * space_create (Id id)`
- `STATUS space_destroy (Space *space)`
- `Id space_get_id (Space *space)`
- `STATUS space_set_name (Space *space, char *name)`
- `const char * space_get_name (Space *space)`
- `STATUS space_set_description (Space *space, char *description)`
- `const char * space_get_description (Space *space)`
- `STATUS space_set_north (Space *space, Id id)`
- `Id space_get_north (Space *space)`
- `STATUS space_set_south (Space *space, Id id)`
- `Id space_get_south (Space *space)`
- `STATUS space_set_east (Space *space, Id id)`
- `Id space_get_east (Space *space)`
- `STATUS space_set_west (Space *space, Id id)`
- `Id space_get_west (Space *space)`
- `STATUS space_set_up (Space *space, Id id)`
- `Id space_get_up (Space *space)`
- `STATUS space_set_down (Space *space, Id id)`
- `Id space_get_down (Space *space)`
- `STATUS space_set_light (Space *space, BOOL light)`
- `BOOL space_get_light (Space *space)`
- `STATUS space_add_object (Space *space, Id obj_id)`
- `STATUS space_remove_object (Space *space, Id obj_id)`
- `Set * space_get_objects_id (Space *space)`
- `STATUS space_set_gdesc_0 (Space *, char *)`
- `STATUS space_set_gdesc_1 (Space *, char *)`
- `STATUS space_set_gdesc_2 (Space *, char *)`
- `char * space_get_gdesc_0 (Space *)`
- `char * space_get_gdesc_1 (Space *)`
- `char * space_get_gdesc_2 (Space *)`
- `STATUS space_light_print (Space *space)`
- `STATUS space_print (Space *space)`
- `Id space_getSprite (Space *, int)`
- `STATUS space_setSprite (Space *, Id, int)`
- `STATUS space_setCurrentSprite (Space *space, int i)`
- `int space_getCurentSprite (Space *space)`

5.42.1 Detailed Description

Defines functions for space manipulation.

Author

Catalín Rotaru

Copyright

GNU Public License

5.42.2 Macro Definition Documentation

5.42.2.1 `#define FIRST_SPACE 1`

5.42.2.2 `#define MAX_SPACES 100`

5.42.3 Typedef Documentation

5.42.3.1 `typedef struct _Space Space`

5.42.4 Function Documentation

5.42.4.1 `STATUS space_add_object (Space * space, Id obj_id)`

5.42.4.2 `Space* space_create (Id id)`

5.42.4.3 `STATUS space_destroy (Space * space)`

5.42.4.4 `const char* space_get_description (Space * space)`

5.42.4.5 `Id space_get_down (Space * space)`

5.42.4.6 `Id space_get_east (Space * space)`

5.42.4.7 `char* space_get_gdesc_0 (Space *)`

5.42.4.8 `char* space_get_gdesc_1 (Space *)`

5.42.4.9 `char* space_get_gdesc_2 (Space *)`

5.42.4.10 `Id space_get_id (Space * space)`

5.42.4.11 `BOOL space_get_light (Space * space)`

5.42.4.12 `const char* space_get_name (Space * space)`

5.42.4.13 `Id space_get_north (Space * space)`

5.42.4.14 `Set* space_get_objects_id (Space * space)`

5.42.4.15 `Id space_get_south (Space * space)`

5.42.4.16 `Id space_get_up (Space * space)`

5.42.4.17 `Id space_get_west (Space * space)`

- 5.42.4.18 `int space_getCurentSprite (Space * space)`
- 5.42.4.19 `Id space_getSprite (Space * , int)`
- 5.42.4.20 `STATUS space_light_print (Space * space)`
- 5.42.4.21 `STATUS space_print (Space * space)`
- 5.42.4.22 `STATUS space_remove_object (Space * space, Id obj_id)`
- 5.42.4.23 `STATUS space_set_description (Space * space, char * description)`
- 5.42.4.24 `STATUS space_set_down (Space * space, Id id)`
- 5.42.4.25 `STATUS space_set_east (Space * space, Id id)`
- 5.42.4.26 `STATUS space_set_gdesc_0 (Space * , char *)`
- 5.42.4.27 `STATUS space_set_gdesc_1 (Space * , char *)`
- 5.42.4.28 `STATUS space_set_gdesc_2 (Space * , char *)`
- 5.42.4.29 `STATUS space_set_light (Space * space, BOOL light)`
- 5.42.4.30 `STATUS space_set_name (Space * space, char * name)`
- 5.42.4.31 `STATUS space_set_north (Space * space, Id id)`
- 5.42.4.32 `STATUS space_set_south (Space * space, Id id)`
- 5.42.4.33 `STATUS space_set_up (Space * space, Id id)`
- 5.42.4.34 `STATUS space_set_west (Space * space, Id id)`
- 5.42.4.35 `STATUS space_setCurrentSprite (Space * space, int i)`
- 5.42.4.36 `STATUS space_setSprite (Space * , Id , int)`

5.43 space_test.c File Reference

It tests space module.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "../include/space.h"
#include "../include/space_test.h"
#include "../include/test.h"
Include dependency graph for space_test.c:
```

Macros

- `#define MAX_TESTS 28`

Functions

- `int main (int argc, char **argv)`
Funcion principal de pruebas para el modulo Space.
- `void test1_space_create ()`
- `void test2_space_create ()`
- `void test1_space_set_name ()`
- `void test2_space_set_name ()`
- `void test3_space_set_name ()`
- `void test1_space_set_north ()`
- `void test2_space_set_north ()`
- `void test1_space_set_south ()`
- `void test2_space_set_south ()`
- `void test1_space_set_east ()`
- `void test2_space_set_east ()`
- `void test1_space_set_west ()`
- `void test2_space_set_west ()`
- `void test1_space_set_object ()`
- `void test2_space_set_object ()`
- `void test1_space_get_name ()`
- `void test2_space_get_name ()`
- `void test1_space_get_object ()`
- `void test2_space_get_object ()`
- `void test3_space_get_object ()`
- `void test1_space_get_north ()`
- `void test2_space_get_north ()`
- `void test1_space_get_south ()`
- `void test2_space_get_south ()`
- `void test1_space_get_east ()`
- `void test2_space_get_east ()`
- `void test1_space_get_west ()`
- `void test2_space_get_west ()`
- `void test1_space_get_id ()`
- `void test2_space_get_id ()`

5.43.1 Detailed Description

It tests space module.

5.43.2 Macro Definition Documentation

5.43.2.1 `#define MAX_TESTS 28`

5.43.3 Function Documentation

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

Funcion principal de pruebas para el modulo Space.

Dos modos de ejecucion: 1.-Si se ejecuta sin parametros se ejecutan todas las pruebas 2.-Si se ejecuta con un numero entre 1 y el numero de pruebas solo ejecuta la prueba indicada

5.43.3.2 void test1_space_create ()

Test Prueba la función de creación de un espacio

Precondition

Un identificador como parámetro

Postcondition

Un puntero no nulo al espacio creado

5.43.3.3 void test1_space_get_east ()

5.43.3.4 void test1_space_get_id ()

5.43.3.5 void test1_space_get_name ()

5.43.3.6 void test1_space_get_north ()

5.43.3.7 void test1_space_get_object ()

5.43.3.8 void test1_space_get_south ()

5.43.3.9 void test1_space_get_west ()

5.43.3.10 void test1_space_set_east ()

5.43.3.11 void test1_space_set_name ()

Test Prueba la función para establecer el nombre de un espacio

Precondition

Nombre que establecer al espacio

Postcondition

La salida debe ser OK

5.43.3.12 void test1_space_set_north ()

5.43.3.13 void test1_space_set_object ()

5.43.3.14 void test1_space_set_south ()

5.43.3.15 void test1_space_set_west ()

5.43.3.16 void test2_space_create ()

Test Prueba la función de creación de un espacio

Precondition

Un identificador como parámetro

Postcondition

El identificador del espacio es el introducido

5.43.3.17 void test2_space_get_east ()

5.43.3.18 void test2_space_get_id ()

5.43.3.19 void test2_space_get_name ()

5.43.3.20 void test2_space_get_north ()

5.43.3.21 void test2_space_get_object ()

5.43.3.22 void test2_space_get_south ()

5.43.3.23 void test2_space_get_west ()

5.43.3.24 void test2_space_set_east ()

5.43.3.25 void test2_space_set_name ()

Test Prueba la función para establecer el nombre de un espacio

Precondition

El espacio al que establecer el nombre es un puntero a NULL

Postcondition

La salida debe ser ERROR

5.43.3.26 void test2_space_set_north ()

5.43.3.27 void test2_space_set_object ()

5.43.3.28 void test2_space_set_south ()

5.43.3.29 void test2_space_set_west ()

5.43.3.30 void test3_space_get_object ()

5.43.3.31 void test3_space_set_name ()

Test Prueba la función para establecer el nombre de un espacio

Precondition

El espacio es un puntero no NULL, pero el nombre a establecer es NULL

Postcondition

La salida debe ser ERROR

5.44 space_test.h File Reference

It declares the tests for the space module.

This graph shows which files directly or indirectly include this file:

Functions

- void [test1_space_create](#) ()
- void [test2_space_create](#) ()
- void [test1_space_set_name](#) ()
- void [test2_space_set_name](#) ()
- void [test3_space_set_name](#) ()
- void [test1_space_set_north](#) ()
- void [test2_space_set_north](#) ()
- void [test3_space_set_north](#) ()
- void [test4_space_set_north](#) ()
- void [test1_space_set_south](#) ()
- void [test2_space_set_south](#) ()
- void [test3_space_set_south](#) ()
- void [test4_space_set_south](#) ()
- void [test1_space_set_east](#) ()
- void [test2_space_set_east](#) ()
- void [test3_space_set_east](#) ()

- void [test4_space_set_east](#) ()
- void [test1_space_set_west](#) ()
- void [test2_space_set_west](#) ()
- void [test3_space_set_west](#) ()
- void [test4_space_set_west](#) ()
- void [test1_space_get_id](#) ()
- void [test2_space_get_id](#) ()
- void [test1_space_set_object](#) ()
- void [test2_space_set_object](#) ()
- void [test1_space_get_name](#) ()
- void [test2_space_get_name](#) ()
- void [test1_space_get_north](#) ()
- void [test2_space_get_north](#) ()
- void [test1_space_get_south](#) ()
- void [test2_space_get_south](#) ()
- void [test1_space_get_east](#) ()
- void [test2_space_get_east](#) ()
- void [test1_space_get_west](#) ()
- void [test2_space_get_west](#) ()
- void [test1_space_get_object](#) ()
- void [test2_space_get_object](#) ()
- void [test3_space_get_object](#) ()

5.44.1 Detailed Description

It declares the tests for the space module.

Author

Profesores Pprog

Copyright

GNU Public License

5.44.2 Function Documentation

5.44.2.1 void [test1_space_create](#) ()

Test Prueba la función de creación de un espacio

Precondition

Un identificador como parámetro

Postcondition

Un puntero no nulo al espacio creado

5.44.2.2 void test1_space_get_east ()

5.44.2.3 void test1_space_get_id ()

5.44.2.4 void test1_space_get_name ()

5.44.2.5 void test1_space_get_north ()

5.44.2.6 void test1_space_get_object ()

5.44.2.7 void test1_space_get_south ()

5.44.2.8 void test1_space_get_west ()

5.44.2.9 void test1_space_set_east ()

5.44.2.10 void test1_space_set_name ()

Test Prueba la función para establecer el nombre de un espacio

Precondition

Nombre que establecer al espacio

Postcondition

La salida debe ser OK

5.44.2.11 void test1_space_set_north ()

5.44.2.12 void test1_space_set_object ()

5.44.2.13 void test1_space_set_south ()

5.44.2.14 void test1_space_set_west ()

5.44.2.15 void test2_space_create ()

Test Prueba la función de creación de un espacio

Precondition

Un identificador como parámetro

Postcondition

El identificador del espacio es el introducido

5.44.2.16 void test2_space_get_east ()

5.44.2.17 void test2_space_get_id ()

5.44.2.18 void test2_space_get_name ()

5.44.2.19 void test2_space_get_north ()

5.44.2.20 void test2_space_get_object ()

5.44.2.21 void test2_space_get_south ()

5.44.2.22 void test2_space_get_west ()

5.44.2.23 void test2_space_set_east ()

5.44.2.24 void test2_space_set_name ()

Test Prueba la función para establecer el nombre de un espacio

Precondition

El espacio al que establecer el nombre es un puntero a NULL

Postcondition

La salida debe ser ERROR

5.44.2.25 void test2_space_set_north ()

5.44.2.26 void test2_space_set_object ()

5.44.2.27 void test2_space_set_south ()

5.44.2.28 void test2_space_set_west ()

5.44.2.29 void test3_space_get_object ()

5.44.2.30 void test3_space_set_east ()

5.44.2.31 void test3_space_set_name ()

Test Prueba la función para establecer el nombre de un espacio

Precondition

El espacio es un puntero no NULL, pero el nombre a establecer es NULL

Postcondition

La salida debe ser ERROR

5.44.2.32 void test3_space_set_north ()

5.44.2.33 void test3_space_set_south ()

5.44.2.34 void test3_space_set_west ()

5.44.2.35 void test4_space_set_east ()

5.44.2.36 void test4_space_set_north ()

5.44.2.37 void test4_space_set_south ()

5.44.2.38 void test4_space_set_west ()

5.45 sprite.c File Reference

It declares the sprite module.

```
#include "../include/sprite.h"
```

Include dependency graph for sprite.c:

Classes

- struct [_Sprite](#)

Functions

- [Sprite](#) * [sprite_create](#) (Id id)
- void [sprite_destroy](#) ([Sprite](#) *sprite)
- Id [sprite_getId](#) ([Sprite](#) *sprite)
- char * [sprite_getData](#) ([Sprite](#) *sprite, int line)
- [STATUS](#) [sprite_putLine](#) ([Sprite](#) *sprite, char *string, int line)
- void [sprite_print](#) ([Sprite](#) *sprite)

5.45.1 Detailed Description

It declares the sprite module.

Author

Antonio Solana

Copyright

GNU Public License

5.45.2 Function Documentation

5.45.2.1 `Sprite* sprite_create (Id id)`

5.45.2.2 `void sprite_destroy (Sprite * sprite)`

5.45.2.3 `char* sprite_getData (Sprite * sprite, int line)`

5.45.2.4 `Id sprite_getId (Sprite * sprite)`

5.45.2.5 `void sprite_print (Sprite * sprite)`

5.45.2.6 `STATUS sprite_putLine (Sprite * sprite, char * string, int line)`

5.46 sprite.h File Reference

It declares the sprite module.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include "types.h"
Include dependency graph for sprite.h:
```

This graph shows which files directly or indirectly include this file:

Typedefs

- `typedef struct _Sprite Sprite`

Functions

- `Sprite * sprite_create (Id id)`
- `void sprite_destroy (Sprite *sprite)`
- `Id sprite_getId (Sprite *sprite)`
- `char * sprite_getData (Sprite *sprite, int line)`
- `STATUS sprite_putLine (Sprite *sprite, char *string, int line)`
- `void sprite_print (Sprite *sprite)`

5.46.1 Detailed Description

It declares the sprite module.

Author

Antonio Solana

Copyright

GNU Public License

5.46.2 Typedef Documentation

5.46.2.1 typedef struct _Sprite Sprite

5.46.3 Function Documentation

5.46.3.1 Sprite* sprite_create (Id id)

5.46.3.2 void sprite_destroy (Sprite * sprite)

5.46.3.3 char* sprite_getData (Sprite * sprite, int line)

5.46.3.4 Id sprite_getId (Sprite * sprite)

5.46.3.5 void sprite_print (Sprite * sprite)

5.46.3.6 STATUS sprite_putLine (Sprite * sprite, char * string, int line)

5.47 sprite_loader.c File Reference

Reads the sprites from a file.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "../include/sprite_loader.h"
Include dependency graph for sprite_loader.c:
```

Functions

- [STATUS sprite_loader_map](#) (Game *game, char *filename)

5.47.1 Detailed Description

Reads the sprites from a file.

Author

Antonio Solana

Copyright

GNU Public License

5.47.2 Function Documentation

5.47.2.1 **STATUS** `sprite_loader_map` (`Game * game`, `char * filename`)

5.48 `sprite_loader.h` File Reference

Reads the sprites from a file.

```
#include "../include/types.h"
#include "../include/game.h"
Include dependency graph for sprite_loader.h:
```

This graph shows which files directly or indirectly include this file:

Functions

- [STATUS `sprite_loader_map`](#) (`Game *game`, `char *filename`)

5.48.1 Detailed Description

Reads the sprites from a file.

Author

Antonio Solana

Copyright

GNU Public License

5.48.2 Function Documentation

5.48.2.1 STATUS sprite_loader_map (Game * game, char * filename)

5.49 test.h File Reference

Test low level functions.

This graph shows which files directly or indirectly include this file:

Macros

- #define `KRED` `"\x1B[31m"`
- #define `KGRN` `"\x1B[32m"`
- #define `KYEL` `"\x1B[33m"`
- #define `KCYN` `"\x1B[36m"`
- #define `RESET` `"\033[0m"`
- #define `PRINT_TEST_RESULT`(x)
- #define `PRINT_PASSED_PERCENTAGE` `printf("Tests passed %d%%\n", ((__test_passed * 100) / __test_counter))`

Variables

- static int `__test_counter` = 0
- static int `__test_passed` = 0
- static int `__pass` = 0

5.49.1 Detailed Description

Test low level functions.

Author

Profesores Pprog

Copyright

GNU Public License

5.49.2 Macro Definition Documentation

5.49.2.1 `#define KCYN "\x1B[36m"`

5.49.2.2 `#define KGRN "\x1B[32m"`

5.49.2.3 `#define KRED "\x1B[31m"`

5.49.2.4 `#define KYEL "\x1B[33m"`

5.49.2.5 `#define PRINT_PASSED_PERCENTAGE printf("Tests passed %d%%\n", ((__test_passed * 100) / __test_counter))`

5.49.2.6 `#define PRINT_TEST_RESULT(x)`

Value:

```
do{\
    __test_counter++; \
    __pass = (x); \
    __test_passed = (__pass)? __test_passed + 1 : \
    __test_passed; \
    printf(KYEL "%s" RESET " line " "%d " KCYN "%s" RESET ": %s\n", \
        __FILE__, __LINE__, __FUNCTION__, \
        ((!__pass) ? KRED "NOT PASS" RESET : KGRN "PASS" RESET)); \
} while (0)
```

5.49.2.7 `#define RESET "\033[0m"`

5.49.3 Variable Documentation

5.49.3.1 `int __pass = 0 [static]`

5.49.3.2 `int __test_counter = 0 [static]`

5.49.3.3 `int __test_passed = 0 [static]`

5.50 types.h File Reference

Global typedefs.

This graph shows which files directly or indirectly include this file:

Macros

- `#define WORD_SIZE 1000`
- `#define NO_ID -1`
- `#define STDSIZE 1024`
- `#define MAX_SPRITES 1000`
- `#define MAX_STRING 20`

Typedefs

- typedef long [ld](#)

Enumerations

- enum [BOOL](#) { [FALSE](#), [TRUE](#) }
- enum [STATUS](#) { [ERROR](#), [OK](#) }
- enum [DIRECTION](#) { [NORTH](#), [EAST](#), [SOUTH](#), [WEST](#) }
- enum [LinkStatus](#) { [OPENED](#), [CLOSED](#), [NO_LINK](#) }

5.50.1 Detailed Description

Global typedefs.

Author

NONAME

Copyright

GNU Public License

5.50.2 Macro Definition Documentation

5.50.2.1 `#define MAX_SPRITES 1000`

5.50.2.2 `#define MAX_STRING 20`

5.50.2.3 `#define NO_ID -1`

5.50.2.4 `#define STDSIZE 1024`

5.50.2.5 `#define WORD_SIZE 1000`

5.50.3 Typedef Documentation

5.50.3.1 typedef long `ld`

5.50.4 Enumeration Type Documentation

5.50.4.1 enum `BOOL`

Enumerator

`FALSE`

`TRUE`

5.50.4.2 enum DIRECTION

Enumerator

NORTH***EAST******SOUTH******WEST***

5.50.4.3 enum LinkStatus

Enumerator

OPENED***CLOSED******NO_LINK***

5.50.4.4 enum STATUS

Enumerator

ERROR***OK***

