Indiana Jones 2D - En busca del arca perdida

Generado por Doxygen 1.8.6

Lunes, 1 de Mayo de 2017 20:29:04

Índice general

1 Indiana Jones 2D - Juego Conversacional **1**

1.1 Desarolladores . . . 1

2 Lista de pruebas **3**

3 Indice de archivos **57**

3.1 Lista de archivos . . . 57

4 Documentación de archivos **61**

4.1 Referencia del Archivo include/command.h . . . 61

4.1.1 Descripción detallada . . . 62

4.1.2 Documentación de los 'typedefs' . . . 62

4.1.2.1 Command . . . 62

4.1.3 Documentación de las enumeraciones . . . 62

4.1.3.1 enum_Command . . . 62

4.1.4 Documentación de las funciones . . . 63

4.1.4.1 Command_clear . . . 63

4.1.4.2 Command_destroy . . . 63

4.1.4.3 Command_get_cmd . . . 63

4.1.4.4 Command_get_cmd_arg . . . 64

4.1.4.5 Command_ini . . . 64

4.1.4.6 Command_set_cmd . . . 64

4.1.4.7 Command_set_cmd_arg . . . 65

4.1.4.8 get_user_input . . . 65

4.2 Referencia del Archivo include/dialogue.h . . . 65

4.2.1 Descripción detallada . . . 67

4.2.2 Documentación de los 'defines' . . . 67

4.2.2.1 GLOBAL_NO_ARGS . . . 67

4.2.3 Documentación de los 'typedefs' . . . 67

4.2.3.1 Dialogue . . . 67

4.2.4 Documentación de las enumeraciones . . . 68

4.2.4.1 DIALOGUE_SAVE_STATUS . . . 68

4.2.5 Documentación de las funciones . . . 68
4.2.5.1 dialogue_attack . . . 68
4.2.5.2 dialogue_destroy . . . 68
4.2.5.3 dialogue_dir . . . 68
4.2.5.4 dialogue_game_rule . . . 68
4.2.5.5 dialogue_get_text . . . 69
4.2.5.6 dialogue_go . . . 69
4.2.5.7 dialogue_help . . . 69
4.2.5.8 dialogue_ini . . . 69
4.2.5.9 dialogue_inspect . . . 70
4.2.5.10 dialogue_leave . . . 70
4.2.5.11 dialogue_load . . . 70
4.2.5.12 dialogue_open . . . 70
4.2.5.13 dialogue_save . . . 71
4.2.5.14 dialogue_take . . . 71
4.2.5.15 dialogue_turn_off . . . 71
4.2.5.16 dialogue_turn_on . . . 71
4.2.5.17 dialogue_unknown . . . 72
4.3 Referencia del Archivo include/die.h . . . 72
4.3.1 Descripción detallada . . . 73
4.3.2 Documentación de los 'defines' . . . 73
4.3.2.1 DIE_ID . . . 73
4.3.3 Documentación de los 'typedefs' . . . 73
4.3.3.1 Die . . . 73
4.3.4 Documentación de las funciones . . . 73
4.3.4.1 die_create . . . 73
4.3.4.2 die_destroy . . . 73
4.3.4.3 die_get_faces . . . 74
4.3.4.4 die_get_number . . . 74
4.3.4.5 die_print . . . 74
4.3.4.6 die_roll . . . 75
4.3.4.7 die_set_faces . . . 75
4.4 Referencia del Archivo include/game.h . . . 75
4.4.1 Descripción detallada . . . 77
4.4.2 Documentación de los 'defines' . . . 77
4.4.2.1 MAX_OBJECTS . . . 77
4.4.2.2 MAX_SPACES . . . 77
4.4.3 Documentación de los 'typedefs' . . . 77
4.4.3.1 Game . . . 77
4.4.4 Documentación de las funciones . . . 77

4.4.4.1 game_add_link . . . 77
4.4.4.2 game_add_object . . . 78
4.4.4.3 game_add_player . . . 78
4.4.4.4 game_add_space . . . 78
4.4.4.5 game_create . . . 79
4.4.4.6 game_destroy . . . 79
4.4.4.7 game_get_dialogue . . . 79
4.4.4.8 game_get_die . . . 80
4.4.4.9 game_get_last_inspected_object . . . 80
4.4.4.10 game_get_last_inspected_space . . . 80
4.4.4.11 game_get_link . . . 80
4.4.4.12 game_get_link_at . . . 81
4.4.4.13 game_get_obj_list_as_str . . . 81
4.4.4.14 game_get_object_at . . . 81
4.4.4.15 game_get_object_location . . . 82
4.4.4.16 game_get_player . . . 82
4.4.4.17 game_get_player_location . . . 82
4.4.4.18 game_get_space . . . 83
4.4.4.19 game_get_space_at . . . 83
4.4.4.20 game_is_over . . . 83
4.4.4.21 game_print_data . . . 84
4.4.4.22 game_update . . . 84
4.5 Referencia del Archivo include/game_management.h . . . 84
4.5.1 Descripción detallada . . . 85
4.5.2 Documentación de las enumeraciones . . . 85
4.5.2.1 SAVE_STATUS . . . 85
4.5.3 Documentación de las funciones . . . 85
4.5.3.1 game_management_load . . . 85
4.5.3.2 game_management_save . . . 86
4.5.3.3 game_management_start_from_file . . . 86
4.6 Referencia del Archivo include/game_rules.h . . . 86
4.6.1 Descripción detallada . . . 87
4.6.2 Documentación de las funciones . . . 87
4.6.2.1 game_rules_run_random_rule . . . 87
4.7 Referencia del Archivo include/graphic_engine.h . . . 87
4.7.1 Descripción detallada . . . 88
4.7.2 Documentación de las funciones . . . 89
4.7.2.1 graphic_engine_create . . . 89
4.7.2.2 graphic_engine_destroy . . . 89
4.7.2.3 graphic_engine_game_over . . . 89

4.7.2.4 graphic_engine_paint_directions . . . 89
4.7.2.5 graphic_engine_paint_game . . . 90
4.7.2.6 graphic_engine_paint_help . . . 90
4.7.2.7 graphic_engine_play_intro . . . 90
4.8 Referencia del Archivo include/inventory.h . . . 90
4.8.1 Descripción detallada . . . 91
4.8.2 Documentación de las funciones . . . 91
4.8.2.1 inventory_add_object . . . 91
4.8.2.2 inventory_create . . . 92
4.8.2.3 inventory_destroy . . . 92
4.8.2.4 inventory_get_max . . . 92
4.8.2.5 inventory_get_set . . . 92
4.8.2.6 inventory_print . . . 92
4.8.2.7 inventory_remove_object . . . 93
4.8.2.8 inventory_set_max . . . 93
4.9 Referencia del Archivo include/link.h . . . 93
4.9.1 Descripción detallada . . . 94
4.9.2 Documentación de los 'defines' . . . 94
4.9.2.1 MAX_LINK . . . 94
4.9.3 Documentación de las enumeraciones . . . 94
4.9.3.1 State . . . 94
4.9.4 Documentación de las funciones . . . 95
4.9.4.1 link_create . . . 95
4.9.4.2 link_destroy . . . 95
4.9.4.3 link_get_dest_from . . . 95
4.9.4.4 link_get_id . . . 95
4.9.4.5 link_get_name . . . 96
4.9.4.6 link_get_space1 . . . 96
4.9.4.7 link_get_space2 . . . 96
4.9.4.8 link_get_state . . . 97
4.9.4.9 link_print . . . 97
4.9.4.10 link_set_id . . . 97
4.9.4.11 link_set_name . . . 97
4.9.4.12 link_set_space1 . . . 98
4.9.4.13 link_set_space2 . . . 98
4.9.4.14 link_set_state . . . 98
4.10 Referencia del Archivo include/object.h . . . 99
4.10.1 Descripción detallada . . . 100
4.10.2 Documentación de las funciones . . . 100
4.10.2.1 object_create . . . 100

4.10.2.2 object_destroy . . . 101
4.10.2.3 object_Get_Description . . . 101
4.10.2.4 object_Get_Description2 . . . 101
4.10.2.5 object_Get_Graphics . . . 101
4.10.2.6 object_Get_Hidden . . . 101
4.10.2.7 object_Get_Id . . . 102
4.10.2.8 object_Get_Illuminates . . . 102
4.10.2.9 object_Get_Light . . . 102
4.10.2.10 object_Get_Mobile . . . 102
4.10.2.11 object_Get_Moved . . . 103
4.10.2.12 object_Get_Name . . . 104
4.10.2.13 object_Get_Open . . . 104
4.10.2.14 object_print . . . 104
4.10.2.15 object_Set_Description . . . 104
4.10.2.16 object_Set_Description2 . . . 105
4.10.2.17 object_Set_Graphics . . . 105
4.10.2.18 object_Set_Hidden . . . 105
4.10.2.19 object_Set_Id . . . 105
4.10.2.20 object_Set_Illuminates . . . 106
4.10.2.21 object_Set_Light . . . 106
4.10.2.22 object_Set_Mobile . . . 106
4.10.2.23 object_Set_Moved . . . 106
4.10.2.24 object_Set_Name . . . 107
4.10.2.25 object_Set_Open . . . 107
4.11 Referencia del Archivo include/player.h . . . 107
4.11.1 Descripción detallada . . . 108
4.11.2 Documentación de las funciones . . . 109
4.11.2.1 player_Add_Object . . . 109
4.11.2.2 player_create . . . 109
4.11.2.3 player_destroy . . . 109
4.11.2.4 player_Get_Id . . . 109
4.11.2.5 player_Get_Location . . . 109
4.11.2.6 player_Get_Max_Objects . . . 110
4.11.2.7 player_Get_Name . . . 110
4.11.2.8 player_Has_Object . . . 110
4.11.2.9 player_Print . . . 110
4.11.2.10 player_Remove_Object . . . 111
4.11.2.11 player_Set_Id . . . 111
4.11.2.12 player_Set_Location . . . 111
4.11.2.13 player_Set_Max_Objects . . . 111

4.11.2.14 player_Set_Name . . . 112
4.12 Referencia del Archivo include/screen.h . . . 112
4.12.1 Descripción detallada . . . 113
4.12.2 Documentación de los 'defines' . . . 113
4.12.2.1 SCREEN_MAX_STR . . . 113
4.12.3 Documentación de los 'typedefs' . . . 113
4.12.3.1 Area . . . 113
4.12.4 Documentación de las funciones . . . 113
4.12.4.1 screen_area_clear . . . 113
4.12.4.2 screen_area_destroy . . . 113
4.12.4.3 screen_area_init . . . 113
4.12.4.4 screen_area_puts . . . 114
4.12.4.5 screen_area_reset_cursor . . . 114
4.12.4.6 screen_gets . . . 114
4.12.4.7 screen_init . . . 114
4.13 Referencia del Archivo include/set.h . . . 114
4.13.1 Descripción detallada . . . 115
4.13.2 Documentación de los 'defines' . . . 115
4.13.2.1 MAX_SET . . . 115
4.13.3 Documentación de las funciones . . . 115
4.13.3.1 set_addId . . . 115
4.13.3.2 set_create . . . 116
4.13.3.3 set_delId . . . 116
4.13.3.4 set_destroy . . . 116
4.13.3.5 set_getNumberOfIds . . . 117
4.13.3.6 set_Id_is_in . . . 117
4.13.3.7 set_print . . . 117
4.14 Referencia del Archivo include/space.h . . . 117
4.14.1 Descripción detallada . . . 119
4.14.2 Documentación de los 'defines' . . . 119
4.14.2.1 G_COLUMNS . . . 119
4.14.2.2 G_ROWS . . . 120
4.14.3 Documentación de los 'typedefs' . . . 120
4.14.3.1 Space . . . 120
4.14.4 Documentación de las funciones . . . 120
4.14.4.1 space_add_object . . . 120
4.14.4.2 space_contains_object . . . 120
4.14.4.3 space_create . . . 120
4.14.4.4 space_destroy . . . 120
4.14.4.5 space_get_description . . . 121

4.14.4.6 space_get_down . . . 121
4.14.4.7 space_get_east . . . 121
4.14.4.8 space_get_graphics . . . 121
4.14.4.9 space_get_id . . . 122
4.14.4.10 space_get_iluminated . . . 122
4.14.4.11 space_get_long_description . . . 122
4.14.4.12 space_get_name . . . 122
4.14.4.13 space_get_north . . . 122
4.14.4.14 space_get_objects . . . 123
4.14.4.15 space_get_south . . . 123
4.14.4.16 space_get_up . . . 123
4.14.4.17 space_get_west . . . 123
4.14.4.18 space_graphics_areEmpty . . . 124
4.14.4.19 space_print . . . 125
4.14.4.20 space_print_graphics . . . 125
4.14.4.21 space_remove_object . . . 125
4.14.4.22 space_set_description . . . 125
4.14.4.23 space_set_down . . . 126
4.14.4.24 space_set_east . . . 126
4.14.4.25 space_set_graphics . . . 126
4.14.4.26 space_set_iluminated . . . 126
4.14.4.27 space_set_long_description . . . 127
4.14.4.28 space_set_name . . . 127
4.14.4.29 space_set_north . . . 127
4.14.4.30 space_set_south . . . 127
4.14.4.31 space_set_up . . . 128
4.14.4.32 space_set_west . . . 128
4.15 Referencia del Archivo include/tests/command_test.h . . . 128
4.15.1 Descripción detallada . . . 129
4.15.2 Documentación de las funciones . . . 129
4.15.2.1 test1_command_clear . . . 129
4.15.2.2 test1_command_destroy . . . 129
4.15.2.3 test1_command_get_cmd . . . 130
4.15.2.4 test1_command_get_cmd_arg . . . 130
4.15.2.5 test1_command_ini . . . 130
4.15.2.6 test1_command_set_cmd . . . 130
4.15.2.7 test1_command_set_cmd_arg . . . 130
4.15.2.8 test2_command_clear . . . 131
4.15.2.9 test2_command_destroy . . . 131
4.15.2.10 test2_command_get_cmd . . . 131

4.15.2.11 test2_command_get_cmd_arg . . . 131
4.15.2.12 test2_command_set_cmd . . . 131
4.15.2.13 test2_command_set_cmd_arg . . . 132
4.15.2.14 test3_command_get_cmd . . . 132
4.15.2.15 test3_command_get_cmd_arg . . . 132
4.15.2.16 test3_command_set_cmd . . . 132
4.15.2.17 test3_command_set_cmd_arg . . . 132
4.15.2.18 test4_command_set_cmd_arg . . . 133
4.16 Referencia del Archivo include/tests/dialogue_test.h . . . 133
4.16.1 Descripción detallada . . . 134
4.16.2 Documentación de las funciones . . . 134
4.16.2.1 test1_dialogue_attack . . . 134
4.16.2.2 test1_dialogue_dir . . . 134
4.16.2.3 test1_dialogue_game_rule . . . 134
4.16.2.4 test1_dialogue_get_text . . . 135
4.16.2.5 test1_dialogue_go . . . 135
4.16.2.6 test1_dialogue_help . . . 135
4.16.2.7 test1_dialogue_ini . . . 135
4.16.2.8 test1_dialogue_inspect . . . 135
4.16.2.9 test1_dialogue_leave . . . 136
4.16.2.10 test1_dialogue_load . . . 136
4.16.2.11 test1_dialogue_open . . . 136
4.16.2.12 test1_dialogue_save . . . 136
4.16.2.13 test1_dialogue_take . . . 136
4.16.2.14 test1_dialogue_turn_off . . . 137
4.16.2.15 test1_dialogue_turn_on . . . 137
4.16.2.16 test1_dialogue_unknown . . . 137
4.16.2.17 test2_dialogue_attack . . . 137
4.16.2.18 test2_dialogue_dir . . . 137
4.16.2.19 test2_dialogue_game_rule . . . 138
4.16.2.20 test2_dialogue_get_text . . . 138
4.16.2.21 test2_dialogue_go . . . 138
4.16.2.22 test2_dialogue_help . . . 138
4.16.2.23 test2_dialogue_inspect . . . 138
4.16.2.24 test2_dialogue_leave . . . 139
4.16.2.25 test2_dialogue_load . . . 139
4.16.2.26 test2_dialogue_open . . . 139
4.16.2.27 test2_dialogue_save . . . 139
4.16.2.28 test2_dialogue_take . . . 139
4.16.2.29 test2_dialogue_turn_off . . . 140

4.16.2.30 test2_dialogue_turn_on . . . 140
4.16.2.31 test2_dialogue_unknown . . . 140
4.17 Referencia del Archivo include/tests/game_management_test.h . . . 140
4.17.1 Descripción detallada . . . 141
4.17.2 Documentación de las funciones . . . 141
4.17.2.1 test1_game_management_load . . . 141
4.17.2.2 test1_game_management_save . . . 141
4.17.2.3 test1_game_management_start_from_file . . . 141
4.17.2.4 test2_game_management_load . . . 142
4.17.2.5 test2_game_management_save . . . 142
4.17.2.6 test2_game_management_start_from_file . . . 142
4.18 Referencia del Archivo include/tests/game_rules_test.h . . . 142
4.18.1 Descripción detallada . . . 142
4.18.2 Documentación de las funciones . . . 143
4.18.2.1 test1_game_rules_run_random_rule . . . 143
4.18.2.2 test2_game_rules_run_random_rule . . . 143
4.19 Referencia del Archivo include/tests/game_test.h . . . 143
4.19.1 Descripción detallada . . . 144
4.19.2 Documentación de las funciones . . . 145
4.19.2.1 test1_game_add_link . . . 145
4.19.2.2 test1_game_add_object . . . 145
4.19.2.3 test1_game_add_player . . . 145
4.19.2.4 test1_game_add_space . . . 145
4.19.2.5 test1_game_create . . . 145
4.19.2.6 test1_game_destroy . . . 146
4.19.2.7 test1_game_get_dialogue . . . 146
4.19.2.8 test1_game_get_die . . . 146
4.19.2.9 test1_game_get_last_inspected_object . . . 146
4.19.2.10 test1_game_get_last_inspected_space . . . 146
4.19.2.11 test1_game_get_link . . . 147
4.19.2.12 test1_game_get_link_at . . . 147
4.19.2.13 test1_game_get_obj_list_as_str . . . 147
4.19.2.14 test1_game_get_object_at . . . 147
4.19.2.15 test1_game_get_object_location . . . 147
4.19.2.16 test1_game_get_player . . . 148
4.19.2.17 test1_game_get_player_location . . . 148
4.19.2.18 test1_game_get_space . . . 148
4.19.2.19 test1_game_get_space_at . . . 148
4.19.2.20 test1_game_is_over . . . 148
4.19.2.21 test1_game_update . . . 149

4.19.2.22 test2_game_add_link . . . 149
4.19.2.23 test2_game_add_object . . . 149
4.19.2.24 test2_game_add_player . . . 149
4.19.2.25 test2_game_add_space . . . 149
4.19.2.26 test2_game_destroy . . . 150
4.19.2.27 test2_game_get_dialogue . . . 150
4.19.2.28 test2_game_get_die . . . 150
4.19.2.29 test2_game_get_last_inspected_object . . . 150
4.19.2.30 test2_game_get_last_inspected_space . . . 150
4.19.2.31 test2_game_get_link . . . 151
4.19.2.32 test2_game_get_link_at . . . 151
4.19.2.33 test2_game_get_obj_list_as_str . . . 151
4.19.2.34 test2_game_get_object_at . . . 151
4.19.2.35 test2_game_get_object_location . . . 151
4.19.2.36 test2_game_get_player . . . 152
4.19.2.37 test2_game_get_player_location . . . 152
4.19.2.38 test2_game_get_space . . . 152
4.19.2.39 test2_game_get_space_at . . . 152
4.19.2.40 test2_game_is_over . . . 152
4.19.2.41 test2_game_update . . . 153
4.19.2.42 test3_game_add_link . . . 153
4.19.2.43 test3_game_add_object . . . 153
4.19.2.44 test3_game_add_player . . . 153
4.19.2.45 test3_game_add_space . . . 153
4.19.2.46 test3_game_get_last_inspected_object . . . 154
4.19.2.47 test3_game_get_link . . . 154
4.19.2.48 test3_game_get_link_at . . . 154
4.19.2.49 test3_game_get_object_at . . . 154
4.19.2.50 test3_game_get_object_location . . . 154
4.19.2.51 test3_game_get_space . . . 155
4.19.2.52 test3_game_get_space_at . . . 155
4.19.2.53 test3_game_update . . . 155
4.20 Referencia del Archivo include/tests/graphic_engine_test.h . . . 155
4.20.1 Descripción detallada . . . 155
4.20.2 Documentación de las funciones . . . 156
4.20.2.1 test1_graphic_engine_create . . . 156
4.21 Referencia del Archivo include/tests/inventory_test.h . . . 156
4.21.1 Descripción detallada . . . 156
4.21.2 Documentación de las funciones . . . 157
4.21.2.1 test1_inventory_add_object . . . 157

4.21.2.2 test1_inventory_create . . . 157
4.21.2.3 test1_inventory_get_max . . . 157
4.21.2.4 test1_inventory_get_set . . . 157
4.21.2.5 test1_inventory_remove_object . . . 158
4.21.2.6 test1_inventory_set_max . . . 158
4.21.2.7 test2_inventory_add_object . . . 158
4.21.2.8 test2_inventory_get_max . . . 158
4.21.2.9 test2_inventory_get_set . . . 158
4.21.2.10 test2_inventory_remove_object . . . 159
4.21.2.11 test2_inventory_set_max . . . 159
4.21.2.12 test3_inventory_add_object . . . 159
4.21.2.13 test3_inventory_remove_object . . . 159
4.21.2.14 test3_inventory_set_max . . . 159
4.22 Referencia del Archivo include/tests/link_test.h . . . 160
4.22.1 Descripción detallada . . . 160
4.22.2 Documentación de las funciones . . . 161
4.22.2.1 test1_link_create . . . 161
4.22.2.2 test1_link_get_dest_from . . . 161
4.22.2.3 test1_link_get_id . . . 161
4.22.2.4 test1_link_get_name . . . 161
4.22.2.5 test1_link_get_space1 . . . 162
4.22.2.6 test1_link_get_space2 . . . 162
4.22.2.7 test1_link_get_state . . . 162
4.22.2.8 test1_link_set_id . . . 162
4.22.2.9 test1_link_set_name . . . 162
4.22.2.10 test1_link_set_space1 . . . 163
4.22.2.11 test1_link_set_space2 . . . 163
4.22.2.12 test1_link_set_state . . . 163
4.22.2.13 test2_link_get_dest_from . . . 163
4.22.2.14 test2_link_get_id . . . 163
4.22.2.15 test2_link_get_name . . . 164
4.22.2.16 test2_link_get_space1 . . . 164
4.22.2.17 test2_link_get_space2 . . . 164
4.22.2.18 test2_link_get_state . . . 164
4.22.2.19 test2_link_set_id . . . 164
4.22.2.20 test2_link_set_name . . . 165
4.22.2.21 test2_link_set_space1 . . . 165
4.22.2.22 test2_link_set_space2 . . . 165
4.22.2.23 test2_link_set_state . . . 165
4.22.2.24 test3_link_get_dest_from . . . 165

4.22.2.25 test3_link_set_id . . . 166
4.22.2.26 test3_link_set_name . . . 166
4.22.2.27 test3_link_set_space1 . . . 166
4.22.2.28 test3_link_set_space2 . . . 166
4.22.2.29 test3_link_set_state . . . 166
4.22.2.30 test4_link_get_dest_from . . . 167
4.23 Referencia del Archivo include/tests/object_test.h . . . 167
4.23.1 Descripción detallada . . . 168
4.23.2 Documentación de las funciones . . . 168
4.23.2.1 test1_object_create . . . 168
4.23.2.2 test1_object_Get_Description . . . 168
4.23.2.3 test1_object_Get_Description2 . . . 168
4.23.2.4 test1_object_Get_Graphics . . . 169
4.23.2.5 test1_object_Get_Hidden . . . 169
4.23.2.6 test1_object_Get_Id . . . 169
4.23.2.7 test1_object_Get_Illuminates . . . 169
4.23.2.8 test1_object_Get_Light . . . 169
4.23.2.9 test1_object_Get_Mobile . . . 170
4.23.2.10 test1_object_Get_Moved . . . 170
4.23.2.11 test1_object_Get_Name . . . 170
4.23.2.12 test1_object_Get_Open . . . 170
4.23.2.13 test1_object_set_description . . . 170
4.23.2.14 test1_object_set_description2 . . . 171
4.23.2.15 test1_object_set_graphics . . . 171
4.23.2.16 test1_object_set_Hidden . . . 171
4.23.2.17 test1_object_set_Id . . . 171
4.23.2.18 test1_object_set_Illuminates . . . 171
4.23.2.19 test1_object_set_Light . . . 172
4.23.2.20 test1_object_set_Mobile . . . 172
4.23.2.21 test1_object_set_Moved . . . 172
4.23.2.22 test1_object_set_name . . . 172
4.23.2.23 test1_object_set_Open . . . 172
4.23.2.24 test2_object_Get_Description . . . 173
4.23.2.25 test2_object_Get_Description2 . . . 173
4.23.2.26 test2_object_Get_Graphics . . . 173
4.23.2.27 test2_object_Get_Hidden . . . 173
4.23.2.28 test2_object_Get_Id . . . 173
4.23.2.29 test2_object_Get_Illuminates . . . 174
4.23.2.30 test2_object_Get_Light . . . 174
4.23.2.31 test2_object_Get_Mobile . . . 174

4.23.2.32 test2_object_Get_Moved . . . 174
4.23.2.33 test2_object_Get_Name . . . 174
4.23.2.34 test2_object_Get_Open . . . 175
4.23.2.35 test2_object_set_description . . . 175
4.23.2.36 test2_object_set_description2 . . . 175
4.23.2.37 test2_object_set_graphics . . . 175
4.23.2.38 test2_object_set_Hidden . . . 175
4.23.2.39 test2_object_set_Id . . . 176
4.23.2.40 test2_object_set_Illuminates . . . 176
4.23.2.41 test2_object_set_Light . . . 176
4.23.2.42 test2_object_set_Mobile . . . 176
4.23.2.43 test2_object_set_Moved . . . 176
4.23.2.44 test2_object_set_name . . . 177
4.23.2.45 test2_object_set_Open . . . 177
4.24 Referencia del Archivo include/tests/player_test.h . . . 177
4.24.1 Descripción detallada . . . 178
4.24.2 Documentación de las funciones . . . 178
4.24.2.1 test1_player_add_object . . . 178
4.24.2.2 test1_player_create . . . 178
4.24.2.3 test1_player_get_id . . . 179
4.24.2.4 test1_player_get_location . . . 179
4.24.2.5 test1_player_get_name . . . 179
4.24.2.6 test1_player_has_object . . . 179
4.24.2.7 test1_player_remove_object . . . 179
4.24.2.8 test1_player_set_id . . . 180
4.24.2.9 test1_player_set_location . . . 180
4.24.2.10 test1_player_set_max_objects . . . 180
4.24.2.11 test1_player_set_name . . . 180
4.24.2.12 test2_player_add_object . . . 180
4.24.2.13 test2_player_get_id . . . 181
4.24.2.14 test2_player_get_location . . . 181
4.24.2.15 test2_player_get_name . . . 181
4.24.2.16 test2_player_has_object . . . 181
4.24.2.17 test2_player_remove_object . . . 181
4.24.2.18 test2_player_set_id . . . 182
4.24.2.19 test2_player_set_location . . . 182
4.24.2.20 test2_player_set_max_objects . . . 182
4.24.2.21 test2_player_set_name . . . 182
4.24.2.22 test3_player_add_object . . . 182
4.24.2.23 test3_player_get_id . . . 183

4.24.2.24 test3_player_get_location . . . 183
4.24.2.25 test3_player_get_name . . . 183
4.24.2.26 test3_player_has_object . . . 183
4.24.2.27 test3_player_remove_object . . . 183
4.24.2.28 test3_player_set_id . . . 184
4.24.2.29 test3_player_set_location . . . 184
4.24.2.30 test3_player_set_max_objects . . . 184
4.24.2.31 test3_player_set_name . . . 184
4.24.2.32 test4_player_remove_object . . . 184
4.24.2.33 test4_player_set_max_objects . . . 185
4.25 Referencia del Archivo include/tests/screen_test.h . . . 185
4.25.1 Descripción detallada . . . 185
4.25.2 Documentación de las funciones . . . 185
4.25.2.1 test1_screen_area_init . . . 185
4.25.2.2 test2_screen_area_init . . . 186
4.25.2.3 test3_screen_area_init . . . 186
4.25.2.4 test4_screen_area_init . . . 186
4.25.2.5 test5_screen_area_init . . . 186
4.26 Referencia del Archivo include/tests/space_test.h . . . 186
4.26.1 Descripción detallada . . . 188
4.26.2 Documentación de las funciones . . . 189
4.26.2.1 test1_space_add_object . . . 189
4.26.2.2 test1_space_contains_object . . . 189
4.26.2.3 test1_space_create . . . 189
4.26.2.4 test1_space_get_description . . . 189
4.26.2.5 test1_space_get_down . . . 190
4.26.2.6 test1_space_get_east . . . 190
4.26.2.7 test1_space_get_graphics . . . 190
4.26.2.8 test1_space_get_id . . . 190
4.26.2.9 test1_space_get_iluminated . . . 190
4.26.2.10 test1_space_get_long_description . . . 191
4.26.2.11 test1_space_get_name . . . 191
4.26.2.12 test1_space_get_north . . . 191
4.26.2.13 test1_space_get_objects . . . 191
4.26.2.14 test1_space_get_south . . . 191
4.26.2.15 test1_space_get_up . . . 192
4.26.2.16 test1_space_get_west . . . 192
4.26.2.17 test1_space_graphics_areEmpty . . . 192
4.26.2.18 test1_space_remove_object . . . 192
4.26.2.19 test1_space_set_description . . . 192

4.26.2.20 test1_space_set_down . . . 193
4.26.2.21 test1_space_set_east . . . 193
4.26.2.22 test1_space_set_graphics . . . 193
4.26.2.23 test1_space_set_iluminated . . . 193
4.26.2.24 test1_space_set_long_description . . . 193
4.26.2.25 test1_space_set_name . . . 194
4.26.2.26 test1_space_set_north . . . 194
4.26.2.27 test1_space_set_south . . . 194
4.26.2.28 test1_space_set_up . . . 194
4.26.2.29 test1_space_set_west . . . 194
4.26.2.30 test2_space_add_object . . . 195
4.26.2.31 test2_space_contains_object . . . 195
4.26.2.32 test2_space_create . . . 195
4.26.2.33 test2_space_get_description . . . 195
4.26.2.34 test2_space_get_down . . . 195
4.26.2.35 test2_space_get_east . . . 196
4.26.2.36 test2_space_get_graphics . . . 196
4.26.2.37 test2_space_get_id . . . 196
4.26.2.38 test2_space_get_iluminated . . . 196
4.26.2.39 test2_space_get_long_description . . . 196
4.26.2.40 test2_space_get_name . . . 197
4.26.2.41 test2_space_get_north . . . 197
4.26.2.42 test2_space_get_objects . . . 197
4.26.2.43 test2_space_get_south . . . 197
4.26.2.44 test2_space_get_up . . . 197
4.26.2.45 test2_space_get_west . . . 198
4.26.2.46 test2_space_graphics_areEmpty . . . 198
4.26.2.47 test2_space_remove_object . . . 198
4.26.2.48 test2_space_set_description . . . 198
4.26.2.49 test2_space_set_down . . . 198
4.26.2.50 test2_space_set_east . . . 199
4.26.2.51 test2_space_set_graphics . . . 199
4.26.2.52 test2_space_set_iluminated . . . 199
4.26.2.53 test2_space_set_long_description . . . 199
4.26.2.54 test2_space_set_name . . . 199
4.26.2.55 test2_space_set_north . . . 200
4.26.2.56 test2_space_set_south . . . 200
4.26.2.57 test2_space_set_up . . . 200
4.26.2.58 test2_space_set_west . . . 200
4.26.2.59 test3_space_add_object . . . 200

4.26.2.60 test3_space_contains_object . . . 201
4.26.2.61 test3_space_create . . . 201
4.26.2.62 test3_space_get_description . . . 201
4.26.2.63 test3_space_get_down . . . 201
4.26.2.64 test3_space_get_east . . . 201
4.26.2.65 test3_space_get_long_description . . . 202
4.26.2.66 test3_space_get_name . . . 202
4.26.2.67 test3_space_get_north . . . 202
4.26.2.68 test3_space_get_south . . . 202
4.26.2.69 test3_space_get_up . . . 202
4.26.2.70 test3_space_get_west . . . 203
4.26.2.71 test3_space_graphics_areEmpty . . . 203
4.26.2.72 test3_space_remove_object . . . 203
4.26.2.73 test3_space_set_description . . . 203
4.26.2.74 test3_space_set_down . . . 203
4.26.2.75 test3_space_set_east . . . 204
4.26.2.76 test3_space_set_long_description . . . 204
4.26.2.77 test3_space_set_name . . . 204
4.26.2.78 test3_space_set_north . . . 204
4.26.2.79 test3_space_set_south . . . 204
4.26.2.80 test3_space_set_up . . . 205
4.26.2.81 test3_space_set_west . . . 205
4.26.2.82 test4_space_remove_object . . . 205
4.27 Referencia del Archivo include/tests/test.h . . . 205
4.27.1 Descripción detallada . . . 205
4.28 Referencia del Archivo include/types.h . . . 206
4.28.1 Descripción detallada . . . 206
4.29 Referencia del Archivo src/command.c . . . 207
4.29.1 Descripción detallada . . . 207
4.29.2 Documentación de las funciones . . . 208
4.29.2.1 Command_clear . . . 208
4.29.2.2 Command_destroy . . . 208
4.29.2.3 Command_get_cmd . . . 208
4.29.2.4 Command_get_cmd_arg . . . 208
4.29.2.5 Command_ini . . . 209
4.29.2.6 Command_set_cmd . . . 209
4.29.2.7 Command_set_cmd_arg . . . 209
4.29.2.8 get_user_input . . . 210
4.30 Referencia del Archivo src/command_test.c . . . 210
4.30.1 Descripción detallada . . . 211

4.30.2 Documentación de las funciones . . . 211
4.30.2.1 test1_command_clear . . . 211
4.30.2.2 test1_command_destroy . . . 211
4.30.2.3 test1_command_get_cmd . . . 211
4.30.2.4 test1_command_get_cmd_arg . . . 212
4.30.2.5 test1_command_ini . . . 212
4.30.2.6 test1_command_set_cmd . . . 212
4.30.2.7 test1_command_set_cmd_arg . . . 212
4.30.2.8 test2_command_clear . . . 212
4.30.2.9 test2_command_destroy . . . 213
4.30.2.10 test2_command_get_cmd . . . 213
4.30.2.11 test2_command_get_cmd_arg . . . 213
4.30.2.12 test2_command_set_cmd . . . 213
4.30.2.13 test2_command_set_cmd_arg . . . 213
4.30.2.14 test3_command_get_cmd . . . 214
4.30.2.15 test3_command_get_cmd_arg . . . 214
4.30.2.16 test3_command_set_cmd . . . 214
4.30.2.17 test3_command_set_cmd_arg . . . 214
4.30.2.18 test4_command_set_cmd_arg . . . 214
4.31 Referencia del Archivo src/dialogue.c . . . 215
4.31.1 Descripción detallada . . . 216
4.31.2 Documentación de las funciones . . . 216
4.31.2.1 dialogue_attack . . . 216
4.31.2.2 dialogue_destroy . . . 216
4.31.2.3 dialogue_dir . . . 216
4.31.2.4 dialogue_game_rule . . . 217
4.31.2.5 dialogue_get_text . . . 217
4.31.2.6 dialogue_go . . . 217
4.31.2.7 dialogue_help . . . 217
4.31.2.8 dialogue_ini . . . 218
4.31.2.9 dialogue_inspect . . . 218
4.31.2.10 dialogue_leave . . . 218
4.31.2.11 dialogue_load . . . 218
4.31.2.12 dialogue_open . . . 219
4.31.2.13 dialogue_save . . . 219
4.31.2.14 dialogue_take . . . 219
4.31.2.15 dialogue_turn_off . . . 219
4.31.2.16 dialogue_turn_on . . . 220
4.31.2.17 dialogue_unknown . . . 220
4.32 Referencia del Archivo src/dialogue_test.c . . . 220

4.32.1 Descripción detallada . . . 221
4.32.2 Documentación de las funciones . . . 221
4.32.2.1 test1_dialogue_attack . . . 221
4.32.2.2 test1_dialogue_dir . . . 222
4.32.2.3 test1_dialogue_game_rule . . . 222
4.32.2.4 test1_dialogue_get_text . . . 222
4.32.2.5 test1_dialogue_go . . . 222
4.32.2.6 test1_dialogue_help . . . 223
4.32.2.7 test1_dialogue_ini . . . 223
4.32.2.8 test1_dialogue_inspect . . . 223
4.32.2.9 test1_dialogue_leave . . . 223
4.32.2.10 test1_dialogue_load . . . 223
4.32.2.11 test1_dialogue_open . . . 224
4.32.2.12 test1_dialogue_save . . . 224
4.32.2.13 test1_dialogue_take . . . 224
4.32.2.14 test1_dialogue_turn_off . . . 224
4.32.2.15 test1_dialogue_turn_on . . . 224
4.32.2.16 test1_dialogue_unknown . . . 225
4.32.2.17 test2_dialogue_attack . . . 225
4.32.2.18 test2_dialogue_dir . . . 225
4.32.2.19 test2_dialogue_game_rule . . . 225
4.32.2.20 test2_dialogue_get_text . . . 225
4.32.2.21 test2_dialogue_go . . . 226
4.32.2.22 test2_dialogue_help . . . 226
4.32.2.23 test2_dialogue_inspect . . . 226
4.32.2.24 test2_dialogue_leave . . . 226
4.32.2.25 test2_dialogue_load . . . 226
4.32.2.26 test2_dialogue_open . . . 227
4.32.2.27 test2_dialogue_save . . . 227
4.32.2.28 test2_dialogue_take . . . 227
4.32.2.29 test2_dialogue_turn_off . . . 227
4.32.2.30 test2_dialogue_turn_on . . . 227
4.32.2.31 test2_dialogue_unknown . . . 228
4.33 Referencia del Archivo src/die.c . . . 228
4.33.1 Descripción detallada . . . 228
4.33.2 Documentación de las funciones . . . 229
4.33.2.1 die_create . . . 229
4.33.2.2 die_destroy . . . 229
4.33.2.3 die_get_faces . . . 229
4.33.2.4 die_get_number . . . 229

4.33.2.5 die_print . . . 230
4.33.2.6 die_roll . . . 230
4.33.2.7 die_set_faces . . . 230
4.34 Referencia del Archivo src/die_test.c . . . 231
4.34.1 Descripción detallada . . . 231
4.35 Referencia del Archivo src/game.c . . . 231
4.35.1 Descripción detallada . . . 232
4.35.2 Documentación de las funciones . . . 233
4.35.2.1 game_add_link . . . 233
4.35.2.2 game_add_object . . . 233
4.35.2.3 game_add_player . . . 233
4.35.2.4 game_add_space . . . 234
4.35.2.5 game_create . . . 234
4.35.2.6 game_destroy . . . 234
4.35.2.7 game_get_dialogue . . . 235
4.35.2.8 game_get_die . . . 235
4.35.2.9 game_get_last_inspected_object . . . 235
4.35.2.10 game_get_last_inspected_space . . . 236
4.35.2.11 game_get_link . . . 236
4.35.2.12 game_get_link_at . . . 236
4.35.2.13 game_get_obj_list_as_str . . . 236
4.35.2.14 game_get_object_at . . . 237
4.35.2.15 game_get_object_location . . . 237
4.35.2.16 game_get_player . . . 237
4.35.2.17 game_get_player_location . . . 238
4.35.2.18 game_get_space . . . 238
4.35.2.19 game_get_space_at . . . 238
4.35.2.20 game_is_over . . . 239
4.35.2.21 game_print_data . . . 239
4.35.2.22 game_update . . . 239
4.36 Referencia del Archivo src/game_loop.c . . . 240
4.36.1 Descripción detallada . . . 240
4.37 Referencia del Archivo src/game_management.c . . . 240
4.37.1 Descripción detallada . . . 241
4.37.2 Documentación de las funciones . . . 241
4.37.2.1 game_management_load . . . 241
4.37.2.2 game_management_save . . . 241
4.37.2.3 game_management_start_from_file . . . 242
4.38 Referencia del Archivo src/game_management_test.c . . . 242
4.38.1 Descripción detallada . . . 243

4.38.2 Documentación de las funciones . . . 243
4.38.2.1 test1_game_management_load . . . 243
4.38.2.2 test1_game_management_save . . . 243
4.38.2.3 test1_game_management_start_from_file . . . 243
4.38.2.4 test2_game_management_load . . . 243
4.38.2.5 test2_game_management_save . . . 244
4.38.2.6 test2_game_management_start_from_file . . . 244
4.39 Referencia del Archivo src/game_rules.c . . . 244
4.39.1 Descripción detallada . . . 244
4.40 Referencia del Archivo src/game_rules_test.c . . . 244
4.40.1 Descripción detallada . . . 245
4.40.2 Documentación de las funciones . . . 245
4.40.2.1 test1_game_rules_run_random_rule . . . 245
4.40.2.2 test2_game_rules_run_random_rule . . . 245
4.41 Referencia del Archivo src/game_test.c . . . 246
4.41.1 Descripción detallada . . . 247
4.41.2 Documentación de las funciones . . . 247
4.41.2.1 test1_game_add_link . . . 247
4.41.2.2 test1_game_add_object . . . 247
4.41.2.3 test1_game_add_player . . . 248
4.41.2.4 test1_game_add_space . . . 248
4.41.2.5 test1_game_create . . . 248
4.41.2.6 test1_game_destroy . . . 248
4.41.2.7 test1_game_get_dialogue . . . 248
4.41.2.8 test1_game_get_die . . . 249
4.41.2.9 test1_game_get_last_inspected_object . . . 249
4.41.2.10 test1_game_get_last_inspected_space . . . 249
4.41.2.11 test1_game_get_link . . . 249
4.41.2.12 test1_game_get_link_at . . . 249
4.41.2.13 test1_game_get_obj_list_as_str . . . 250
4.41.2.14 test1_game_get_object_at . . . 250
4.41.2.15 test1_game_get_object_location . . . 250
4.41.2.16 test1_game_get_player . . . 250
4.41.2.17 test1_game_get_player_location . . . 250
4.41.2.18 test1_game_get_space . . . 251
4.41.2.19 test1_game_get_space_at . . . 251
4.41.2.20 test1_game_is_over . . . 251
4.41.2.21 test1_game_update . . . 251
4.41.2.22 test2_game_add_link . . . 251
4.41.2.23 test2_game_add_object . . . 252

4.41.2.24 test2_game_add_player . . . 252
4.41.2.25 test2_game_add_space . . . 252
4.41.2.26 test2_game_destroy . . . 252
4.41.2.27 test2_game_get_dialogue . . . 252
4.41.2.28 test2_game_get_die . . . 253
4.41.2.29 test2_game_get_last_inspected_object . . . 253
4.41.2.30 test2_game_get_last_inspected_space . . . 253
4.41.2.31 test2_game_get_link . . . 253
4.41.2.32 test2_game_get_link_at . . . 253
4.41.2.33 test2_game_get_obj_list_as_str . . . 254
4.41.2.34 test2_game_get_object_at . . . 254
4.41.2.35 test2_game_get_object_location . . . 254
4.41.2.36 test2_game_get_player . . . 254
4.41.2.37 test2_game_get_player_location . . . 254
4.41.2.38 test2_game_get_space . . . 255
4.41.2.39 test2_game_get_space_at . . . 255
4.41.2.40 test2_game_is_over . . . 255
4.41.2.41 test2_game_update . . . 255
4.41.2.42 test3_game_add_link . . . 255
4.41.2.43 test3_game_add_object . . . 256
4.41.2.44 test3_game_add_player . . . 256
4.41.2.45 test3_game_add_space . . . 256
4.41.2.46 test3_game_get_last_inspected_object . . . 256
4.41.2.47 test3_game_get_link . . . 256
4.41.2.48 test3_game_get_link_at . . . 257
4.41.2.49 test3_game_get_object_at . . . 257
4.41.2.50 test3_game_get_object_location . . . 257
4.41.2.51 test3_game_get_space . . . 257
4.41.2.52 test3_game_get_space_at . . . 257
4.41.2.53 test3_game_update . . . 258
4.42 Referencia del Archivo src/graphic_engine_test.c . . . 258
4.42.1 Descripción detallada . . . 258
4.42.2 Documentación de las funciones . . . 258
4.42.2.1 test1_graphic_engine_create . . . 258
4.43 Referencia del Archivo src/inventory.c . . . 259
4.43.1 Descripción detallada . . . 259
4.43.2 Documentación de las funciones . . . 259
4.43.2.1 inventory_add_object . . . 259
4.43.2.2 inventory_create . . . 260
4.43.2.3 inventory_destroy . . . 260

4.43.2.4 inventory_get_max . . . 260
4.43.2.5 inventory_get_set . . . 260
4.43.2.6 inventory_print . . . 261
4.43.2.7 inventory_remove_object . . . 261
4.43.2.8 inventory_set_max . . . 261
4.44 Referencia del Archivo src/inventory_test.c . . . 261
4.44.1 Descripción detallada . . . 262
4.44.2 Documentación de las funciones . . . 262
4.44.2.1 test1_inventory_add_object . . . 262
4.44.2.2 test1_inventory_create . . . 262
4.44.2.3 test1_inventory_get_max . . . 263
4.44.2.4 test1_inventory_get_set . . . 263
4.44.2.5 test1_inventory_remove_object . . . 263
4.44.2.6 test1_inventory_set_max . . . 263
4.44.2.7 test2_inventory_add_object . . . 263
4.44.2.8 test2_inventory_get_max . . . 264
4.44.2.9 test2_inventory_get_set . . . 264
4.44.2.10 test2_inventory_remove_object . . . 264
4.44.2.11 test2_inventory_set_max . . . 264
4.44.2.12 test3_inventory_add_object . . . 264
4.44.2.13 test3_inventory_remove_object . . . 265
4.44.2.14 test3_inventory_set_max . . . 265
4.45 Referencia del Archivo src/link.c . . . 265
4.45.1 Descripción detallada . . . 266
4.45.2 Documentación de las funciones . . . 266
4.45.2.1 link_create . . . 266
4.45.2.2 link_destroy . . . 266
4.45.2.3 link_get_dest_from . . . 267
4.45.2.4 link_get_id . . . 267
4.45.2.5 link_get_name . . . 267
4.45.2.6 link_get_space1 . . . 267
4.45.2.7 link_get_space2 . . . 268
4.45.2.8 link_get_state . . . 268
4.45.2.9 link_print . . . 268
4.45.2.10 link_set_id . . . 269
4.45.2.11 link_set_name . . . 269
4.45.2.12 link_set_space1 . . . 269
4.45.2.13 link_set_space2 . . . 269
4.45.2.14 link_set_state . . . 270
4.46 Referencia del Archivo src/link_test.c . . . 270

4.46.1 Descripción detallada . . . 271
4.46.2 Documentación de las funciones . . . 271
4.46.2.1 test1_link_create . . . 271
4.46.2.2 test1_link_get_dest_from . . . 271
4.46.2.3 test1_link_get_id . . . 271
4.46.2.4 test1_link_get_name . . . 272
4.46.2.5 test1_link_get_space1 . . . 272
4.46.2.6 test1_link_get_space2 . . . 272
4.46.2.7 test1_link_get_state . . . 272
4.46.2.8 test1_link_set_id . . . 272
4.46.2.9 test1_link_set_name . . . 273
4.46.2.10 test1_link_set_space1 . . . 273
4.46.2.11 test1_link_set_space2 . . . 273
4.46.2.12 test1_link_set_state . . . 273
4.46.2.13 test2_link_get_dest_from . . . 273
4.46.2.14 test2_link_get_id . . . 274
4.46.2.15 test2_link_get_name . . . 274
4.46.2.16 test2_link_get_space1 . . . 274
4.46.2.17 test2_link_get_space2 . . . 274
4.46.2.18 test2_link_get_state . . . 274
4.46.2.19 test2_link_set_id . . . 275
4.46.2.20 test2_link_set_name . . . 275
4.46.2.21 test2_link_set_space1 . . . 275
4.46.2.22 test2_link_set_space2 . . . 275
4.46.2.23 test2_link_set_state . . . 275
4.46.2.24 test3_link_get_dest_from . . . 276
4.46.2.25 test3_link_set_id . . . 276
4.46.2.26 test3_link_set_name . . . 276
4.46.2.27 test3_link_set_space1 . . . 276
4.46.2.28 test3_link_set_space2 . . . 276
4.46.2.29 test3_link_set_state . . . 277
4.46.2.30 test4_link_get_dest_from . . . 277
4.47 Referencia del Archivo src/object.c . . . 277
4.47.1 Descripción detallada . . . 278
4.47.2 Documentación de las funciones . . . 279
4.47.2.1 object_create . . . 279
4.47.2.2 object_destroy . . . 279
4.47.2.3 object_Get_Description . . . 279
4.47.2.4 object_Get_Description2 . . . 279
4.47.2.5 object_Get_Graphics . . . 279

4.47.2.6 object_Get_Hidden . . . 280
4.47.2.7 object_Get_Id . . . 280
4.47.2.8 object_Get_Illuminates . . . 280
4.47.2.9 object_Get_Light . . . 280
4.47.2.10 object_Get_Mobile . . . 281
4.47.2.11 object_Get_Moved . . . 282
4.47.2.12 object_Get_Name . . . 282
4.47.2.13 object_Get_Open . . . 282
4.47.2.14 object_print . . . 282
4.47.2.15 object_Set_Description . . . 283
4.47.2.16 object_Set_Description2 . . . 284
4.47.2.17 object_Set_Graphics . . . 284
4.47.2.18 object_Set_Hidden . . . 284
4.47.2.19 object_Set_Id . . . 284
4.47.2.20 object_Set_Illuminates . . . 285
4.47.2.21 object_Set_Light . . . 285
4.47.2.22 object_Set_Mobile . . . 285
4.47.2.23 object_Set_Moved . . . 285
4.47.2.24 object_Set_Name . . . 286
4.47.2.25 object_Set_Open . . . 286
4.48 Referencia del Archivo src/object_test.c . . . 286
4.48.1 Descripción detallada . . . 287
4.48.2 Documentación de las funciones . . . 288
4.48.2.1 test1_object_create . . . 288
4.48.2.2 test1_object_Get_Description . . . 288
4.48.2.3 test1_object_Get_Description2 . . . 288
4.48.2.4 test1_object_Get_Graphics . . . 288
4.48.2.5 test1_object_Get_Hidden . . . 288
4.48.2.6 test1_object_Get_Id . . . 289
4.48.2.7 test1_object_Get_Illuminates . . . 289
4.48.2.8 test1_object_Get_Light . . . 289
4.48.2.9 test1_object_Get_Mobile . . . 289
4.48.2.10 test1_object_Get_Moved . . . 289
4.48.2.11 test1_object_Get_Name . . . 290
4.48.2.12 test1_object_Get_Open . . . 290
4.48.2.13 test1_object_set_description . . . 290
4.48.2.14 test1_object_set_description2 . . . 290
4.48.2.15 test1_object_set_graphics . . . 290
4.48.2.16 test1_object_set_Hidden . . . 291
4.48.2.17 test1_object_set_Id . . . 291

4.48.2.18 test1_object_set_Illuminates . . . 291
4.48.2.19 test1_object_set_Light . . . 291
4.48.2.20 test1_object_set_Mobile . . . 291
4.48.2.21 test1_object_set_Moved . . . 292
4.48.2.22 test1_object_set_name . . . 292
4.48.2.23 test1_object_set_Open . . . 292
4.48.2.24 test2_object_Get_Description . . . 292
4.48.2.25 test2_object_Get_Description2 . . . 292
4.48.2.26 test2_object_Get_Graphics . . . 293
4.48.2.27 test2_object_Get_Hidden . . . 293
4.48.2.28 test2_object_Get_Id . . . 293
4.48.2.29 test2_object_Get_Illuminates . . . 293
4.48.2.30 test2_object_Get_Light . . . 293
4.48.2.31 test2_object_Get_Mobile . . . 294
4.48.2.32 test2_object_Get_Moved . . . 294
4.48.2.33 test2_object_Get_Name . . . 294
4.48.2.34 test2_object_Get_Open . . . 294
4.48.2.35 test2_object_set_description . . . 294
4.48.2.36 test2_object_set_description2 . . . 295
4.48.2.37 test2_object_set_graphics . . . 295
4.48.2.38 test2_object_set_Hidden . . . 295
4.48.2.39 test2_object_set_Id . . . 295
4.48.2.40 test2_object_set_Illuminates . . . 295
4.48.2.41 test2_object_set_Light . . . 296
4.48.2.42 test2_object_set_Mobile . . . 296
4.48.2.43 test2_object_set_Moved . . . 296
4.48.2.44 test2_object_set_name . . . 296
4.48.2.45 test2_object_set_Open . . . 296
4.49 Referencia del Archivo src/player.c . . . 297
4.49.1 Descripción detallada . . . 297
4.49.2 Documentación de las funciones . . . 298
4.49.2.1 player_Add_Object . . . 298
4.49.2.2 player_create . . . 298
4.49.2.3 player_destroy . . . 298
4.49.2.4 player_Get_Id . . . 298
4.49.2.5 player_Get_Location . . . 299
4.49.2.6 player_Get_Max_Objects . . . 299
4.49.2.7 player_Get_Name . . . 299
4.49.2.8 player_Has_Object . . . 299
4.49.2.9 player_Print . . . 300

4.49.2.10 player_Remove_Object . . . 300
4.49.2.11 player_Set_Id . . . 300
4.49.2.12 player_Set_Location . . . 300
4.49.2.13 player_Set_Max_Objects . . . 301
4.49.2.14 player_Set_Name . . . 301
4.50 Referencia del Archivo src/player_test.c . . . 301
4.50.1 Descripción detallada . . . 302
4.50.2 Documentación de las funciones . . . 303
4.50.2.1 test1_player_add_object . . . 303
4.50.2.2 test1_player_create . . . 303
4.50.2.3 test1_player_get_id . . . 303
4.50.2.4 test1_player_get_location . . . 303
4.50.2.5 test1_player_get_name . . . 303
4.50.2.6 test1_player_has_object . . . 304
4.50.2.7 test1_player_remove_object . . . 304
4.50.2.8 test1_player_set_id . . . 304
4.50.2.9 test1_player_set_location . . . 304
4.50.2.10 test1_player_set_max_objects . . . 304
4.50.2.11 test1_player_set_name . . . 305
4.50.2.12 test2_player_add_object . . . 305
4.50.2.13 test2_player_get_id . . . 305
4.50.2.14 test2_player_get_location . . . 305
4.50.2.15 test2_player_get_name . . . 305
4.50.2.16 test2_player_has_object . . . 306
4.50.2.17 test2_player_remove_object . . . 306
4.50.2.18 test2_player_set_id . . . 306
4.50.2.19 test2_player_set_location . . . 306
4.50.2.20 test2_player_set_max_objects . . . 306
4.50.2.21 test2_player_set_name . . . 307
4.50.2.22 test3_player_add_object . . . 307
4.50.2.23 test3_player_get_id . . . 307
4.50.2.24 test3_player_get_location . . . 307
4.50.2.25 test3_player_get_name . . . 307
4.50.2.26 test3_player_has_object . . . 308
4.50.2.27 test3_player_remove_object . . . 308
4.50.2.28 test3_player_set_id . . . 308
4.50.2.29 test3_player_set_location . . . 308
4.50.2.30 test3_player_set_max_objects . . . 308
4.50.2.31 test3_player_set_name . . . 309
4.50.2.32 test4_player_remove_object . . . 309

4.50.2.33 test4_player_set_max_objects . . . 309
4.51 Referencia del Archivo src/screen.c . . . 309
4.51.1 Descripción detallada . . . 310
4.51.2 Documentación de las funciones . . . 310
4.51.2.1 screen_area_clear . . . 310
4.51.2.2 screen_area_destroy . . . 310
4.51.2.3 screen_area_init . . . 310
4.51.2.4 screen_area_puts . . . 311
4.51.2.5 screen_area_reset_cursor . . . 311
4.51.2.6 screen_gets . . . 311
4.51.2.7 screen_init . . . 311
4.52 Referencia del Archivo src/screen_test.c . . . 311
4.52.1 Descripción detallada . . . 312
4.52.2 Documentación de las funciones . . . 312
4.52.2.1 test1_screen_area_init . . . 312
4.52.2.2 test2_screen_area_init . . . 312
4.52.2.3 test3_screen_area_init . . . 312
4.52.2.4 test4_screen_area_init . . . 313
4.52.2.5 test5_screen_area_init . . . 313
4.53 Referencia del Archivo src/set.c . . . 313
4.53.1 Descripción detallada . . . 314
4.53.2 Documentación de las funciones . . . 314
4.53.2.1 set_addId . . . 314
4.53.2.2 set_create . . . 314
4.53.2.3 set_delId . . . 314
4.53.2.4 set_destroy . . . 315
4.53.2.5 set_getNumberOfIds . . . 315
4.53.2.6 set_Id_is_in . . . 315
4.53.2.7 set_print . . . 316
4.54 Referencia del Archivo src/set_test.c . . . 316
4.54.1 Descripción detallada . . . 316
4.55 Referencia del Archivo src/space.c . . . 317
4.55.1 Descripción detallada . . . 318
4.55.2 Documentación de las funciones . . . 318
4.55.2.1 space_add_object . . . 318
4.55.2.2 space_contains_object . . . 319
4.55.2.3 space_create . . . 319
4.55.2.4 space_destroy . . . 319
4.55.2.5 space_get_description . . . 319
4.55.2.6 space_get_down . . . 320

4.55.2.7 space_get_east . . . 320
4.55.2.8 space_get_graphics . . . 320
4.55.2.9 space_get_id . . . 320
4.55.2.10 space_get_iluminated . . . 320
4.55.2.11 space_get_long_description . . . 321
4.55.2.12 space_get_name . . . 321
4.55.2.13 space_get_north . . . 321
4.55.2.14 space_get_objects . . . 321
4.55.2.15 space_get_south . . . 322
4.55.2.16 space_get_up . . . 323
4.55.2.17 space_get_west . . . 323
4.55.2.18 space_graphics_areEmpty . . . 323
4.55.2.19 space_print . . . 323
4.55.2.20 space_print_graphics . . . 324
4.55.2.21 space_remove_object . . . 325
4.55.2.22 space_set_description . . . 325
4.55.2.23 space_set_down . . . 325
4.55.2.24 space_set_east . . . 325
4.55.2.25 space_set_graphics . . . 326
4.55.2.26 space_set_iluminated . . . 326
4.55.2.27 space_set_long_description . . . 326
4.55.2.28 space_set_name . . . 326
4.55.2.29 space_set_north . . . 327
4.55.2.30 space_set_south . . . 327
4.55.2.31 space_set_up . . . 327
4.55.2.32 space_set_west . . . 327
4.56 Referencia del Archivo src/space_test.c . . . 328
4.56.1 Descripción detallada . . . 330
4.56.2 Documentación de las funciones . . . 330
4.56.2.1 test1_space_add_object . . . 330
4.56.2.2 test1_space_contains_object . . . 330
4.56.2.3 test1_space_create . . . 330
4.56.2.4 test1_space_get_description . . . 330
4.56.2.5 test1_space_get_down . . . 331
4.56.2.6 test1_space_get_east . . . 331
4.56.2.7 test1_space_get_graphics . . . 331
4.56.2.8 test1_space_get_id . . . 331
4.56.2.9 test1_space_get_iluminated . . . 331
4.56.2.10 test1_space_get_long_description . . . 332
4.56.2.11 test1_space_get_name . . . 332

4.56.2.12 test1_space_get_north . . . 332
4.56.2.13 test1_space_get_objects . . . 332
4.56.2.14 test1_space_get_south . . . 332
4.56.2.15 test1_space_get_up . . . 333
4.56.2.16 test1_space_get_west . . . 333
4.56.2.17 test1_space_graphics_areEmpty . . . 333
4.56.2.18 test1_space_remove_object . . . 333
4.56.2.19 test1_space_set_description . . . 333
4.56.2.20 test1_space_set_down . . . 334
4.56.2.21 test1_space_set_east . . . 334
4.56.2.22 test1_space_set_graphics . . . 334
4.56.2.23 test1_space_set_iluminated . . . 334
4.56.2.24 test1_space_set_long_description . . . 334
4.56.2.25 test1_space_set_name . . . 335
4.56.2.26 test1_space_set_north . . . 335
4.56.2.27 test1_space_set_south . . . 335
4.56.2.28 test1_space_set_up . . . 335
4.56.2.29 test1_space_set_west . . . 335
4.56.2.30 test2_space_add_object . . . 336
4.56.2.31 test2_space_contains_object . . . 336
4.56.2.32 test2_space_create . . . 336
4.56.2.33 test2_space_get_description . . . 336
4.56.2.34 test2_space_get_down . . . 336
4.56.2.35 test2_space_get_east . . . 337
4.56.2.36 test2_space_get_graphics . . . 337
4.56.2.37 test2_space_get_id . . . 337
4.56.2.38 test2_space_get_iluminated . . . 337
4.56.2.39 test2_space_get_long_description . . . 337
4.56.2.40 test2_space_get_name . . . 338
4.56.2.41 test2_space_get_north . . . 338
4.56.2.42 test2_space_get_objects . . . 338
4.56.2.43 test2_space_get_south . . . 338
4.56.2.44 test2_space_get_up . . . 338
4.56.2.45 test2_space_get_west . . . 339
4.56.2.46 test2_space_graphics_areEmpty . . . 339
4.56.2.47 test2_space_remove_object . . . 339
4.56.2.48 test2_space_set_description . . . 339
4.56.2.49 test2_space_set_down . . . 339
4.56.2.50 test2_space_set_east . . . 340
4.56.2.51 test2_space_set_graphics . . . 340

4.56.2.52 test2_space_set_iluminated . . . 340
4.56.2.53 test2_space_set_long_description . . . 340
4.56.2.54 test2_space_set_name . . . 340
4.56.2.55 test2_space_set_north . . . 341
4.56.2.56 test2_space_set_south . . . 341
4.56.2.57 test2_space_set_up . . . 341
4.56.2.58 test2_space_set_west . . . 341
4.56.2.59 test3_space_add_object . . . 341
4.56.2.60 test3_space_contains_object . . . 342
4.56.2.61 test3_space_create . . . 342
4.56.2.62 test3_space_get_description . . . 342
4.56.2.63 test3_space_get_down . . . 342
4.56.2.64 test3_space_get_east . . . 342
4.56.2.65 test3_space_get_long_description . . . 343
4.56.2.66 test3_space_get_name . . . 343
4.56.2.67 test3_space_get_north . . . 343
4.56.2.68 test3_space_get_south . . . 343
4.56.2.69 test3_space_get_up . . . 343
4.56.2.70 test3_space_get_west . . . 344
4.56.2.71 test3_space_graphics_areEmpty . . . 344
4.56.2.72 test3_space_remove_object . . . 344
4.56.2.73 test3_space_set_description . . . 344
4.56.2.74 test3_space_set_down . . . 344
4.56.2.75 test3_space_set_east . . . 345
4.56.2.76 test3_space_set_long_description . . . 345
4.56.2.77 test3_space_set_name . . . 345
4.56.2.78 test3_space_set_north . . . 345
4.56.2.79 test3_space_set_south . . . 345
4.56.2.80 test3_space_set_up . . . 346
4.56.2.81 test3_space_set_west . . . 346
4.56.2.82 test4_space_remove_object . . . 346

Índice **347**

Capítulo 1

Indiana Jones 2D - Juego Conversacional

En la cuarta iteración (I4) se continúa con el desarrollo del proyecto y de las habilidades y los conceptos necesarios para

ello, así como con la introducción y uso de herramientas apropiadas para dicha actividad. En esta iteración se abandonará

el Juego de la Oca como referencia para el desarrollo, puesto que deja de ser un modelo adecuado para ilustrar las

funcionalidades adicionales necesarias para completar un sistema que soporte Aventuras Conversacionales. Se creará un nuevo juego

que demostrara las habilidades adquiridas por el equipo.

1.1. Desarolladores

- Javier Bernardo
- Laura Bernal
- Mihai Blidaru
- Sandra Benítez

Capítulo 2

Lista de pruebas

Global test1_command_clear ()

Prueba la función que limpia los campos de un commando

Precondición

El gestor de comandos ha sido inicializado previamente y se ha guardado un comando dentro

Postcondición

La salida esperada es NO_CMD y una cadena vacia

Global test1_command_destroy ()

Pruba la función que libera memoria usada por un gestor de comandos

Precondición

El gestor de comandos ha sido previamente inicializado

Postcondición

La salida esperada es OK

Global test1_command_get_cmd ()

Prueba la función que devuelve el tipo de comando guardado en un gestor de comandos

Precondición

El gestor de comandos ha sido inicializado previamente y se le ha asignado el comando GO

Postcondición

La salida esperada es GO, el comando previamente asignado

Global test1_command_get_cmd_arg ()

Prueba la función que devuelve un argumento de un comando guardado en un gestor de comandos

Precondición

El gestor de comandos ha sido inicializado previamente y se le ha asignado un argumento "test"

Postcondición

La salida esperada es "test", el argumento añadido previamente

Global test1_command_ini ()

Prueba la función de creación de un gestor de comandos

Precondición

Condiciones normales para la prueba

Postcondición

Un puntero no nulo al al gestor de comandos creado

Global test1_command_set_cmd ()

Prueba la función que guarda el tipo de comando en un gestor de comandos

Precondición

El gestor de comandos ha sido inicializado previamente

Postcondición

La salida esperada es OK

Global test1_command_set_cmd_arg ()

Prueba la función que guarda un argumento de un comando en un gestor de comandos

Precondición

El gestor de comandos ha sido inicializado previamente

Postcondición

La salida esperada es OK

Global test1_dialogue_attack ()

Prueba la funcion que construye el dialogo para el comando ATTACK

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

Global test1_dialogue_dir ()

Prueba la funcion que construye el dialogo para el comando DIR

Precondición

El dialogo ha sido inicializado

Postcondición

La salida esperada es OK

Global test1_dialogue_game_rule ()

Prueba la funcion que construye el dialogo para las reglas del juego

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

Global test1_dialogue_get_text ()

Prueba la funcion que devuelve el texto del dialogo

Precondición

El dialogo se ha inicializado antes

Postcondición

La salida es un texto no nulo

Global test1_dialogue_go ()

Prueba la funcion que construye el dialogo para el comando GO

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

Global test1_dialogue_help ()

Prueba la funcion que construye el dialogo para el comando HELP

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

Global test1_dialogue_ini ()

Prueba la función que crea un dialogo

Precondición

Se reserva memoria para el dialogo

Postcondición

La salida que se espera es el dialogo inicializado

Global test1_dialogue_inspect ()

Prueba la funcion que construye el dialogo para el comando INSPECT

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

Global test1_dialogue_leave ()

Prueba la funcion que construye el dialogo para el comando LEAVE

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

Global test1_dialogue_load ()

Prueba la funcion que construye el dialogo para el comando LOAD

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

Global test1_dialogue_open ()

Prueba la funcion que construye el dialogo para el comando OPEN

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

Global test1_dialogue_save ()

Prueba la funcion que construye el dialogo para el comando SAVE

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

Global test1_dialogue_take ()

Prueba la funcion que construye el dialogo para el comando TAKE

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

Global test1_dialogue_turn_off ()

Prueba la funcion que construye el dialogo para el comando TURN_OFF

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

Global test1_dialogue_turn_on ()

Prueba la funcion que construye el dialogo para el comando TURN_ON

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

Global test1_dialogue_unknown ()

Prueba la funcion que construye el dialogo para comandos desconocidos

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

Global test1_game_add_link ()

Prueba la función que añade un link al juego

Precondición

El juego y el link han sido inicializados correctamente

Postcondición

La salida esperada es OK

Global test1_game_add_object ()

Prueba la función que añade un objeto al juego

Precondición

El juego y el objeto han sido inicializados correctamente

Postcondición

La salida esperada es OK

Global test1_game_add_player ()

Prueba la función que añade un jugador al juego

Precondición

El juego y el jugador han sido inicializados correctamente

Postcondición

La salida esperada es OK

Global test1_game_add_space ()

Prueba la función que añade un espacio al juego

Precondición

El juego y el espacio han sido inicializados correctamente

Postcondición

La salida esperada es OK

Global test1_game_create ()

Prueba la funcion que reserva memoria para la estructra del jueg

Precondición

Se reserva memoria normalmente

Postcondición

La salida tiene que ser el juego inicializado

Global test1_game_destroy ()

Prueba la funcon que destruye un juego

Precondición

El juego se ha creado previamente

Postcondición

La salida esperada es OK

Global test1_game_get_dialogue ()

Prueba la función que devuelve el dialogo del juego

Precondición

El juego ha sido creado correctamente

Postcondición

La salida debe ser un puntero distinto de NULL

Global test1_game_get_die ()

Prueba la función que devuelve el dado del juego

Precondición

El juego ha sido creado correctamente

Postcondición

La salida debe ser un puntero distinto de NULL

Global test1_game_get_last_inspected_object ()

Prueba la función que devuelve el ultimo objeto inspeccionado

Precondición

El nombre del objeto añadido es "obj"

Postcondición

La salida esperada es el objeto con nombre "obj"

Global test1_game_get_last_inspected_space ()

Prueba la función que devuelve el ultimo espacio inspeccionado

Precondición

El juego ha sido creado, se ha añadido un espacio y se ha colocado al jugador dentro

Postcondición

La salida esperada es la casilla en la que esta el jugador

Global test1_game_get_link ()

Prueba la función que devuelve el link con in id dado

Precondición

El juego y el link han sido creados correctamente

Postcondición

La salida debe ser el link creado anteriormente

Global test1_game_get_link_at ()

Prueba la función que devuelve el link en una posición dada

Precondición

El link ha sido añadido correctamente

Postcondición

La salida debe ser el link añadido anteriormente

Global test1_game_get_obj_list_as_str ()

Prueba la funcion que devuelve la lista de objetos de una casilla como cadena

Precondición

El juego, una casilla, y un objeto han sido añadidos correctamente

Postcondición

La salida esperada es la lista de objetos de la casilla

Global test1_game_get_object_at ()

Prueba la función que devuelve el objeto en una posición dada

Precondición

El objeto ha sido añadido correctamente

Postcondición

La salida debe ser el objeto añadido anteriormente

Global test1_game_get_object_location ()

Prueba la funcion que devuelve la localizacion de un objeto

Precondición

El objeto se ha añadido en la casilla con id 2

Postcondición

La salida es 2, la localización del objeto

Global test1_game_get_player ()

Prueba la función que devuelve el jugador del juego

Precondición

El juego y el jugador han sido creado correctamente

Postcondición

La salida debe ser un puntero distinto de NULL

Global test1_game_get_player_location ()

Prueba la funcion que devuelve la localizacion del jugador

Precondición

El jugador se ha colocado en la casilla con id 2

Postcondición

La salida es 2, la localización del jugador

Global test1_game_get_space ()

Prueba la función que devuelve el espacio con in id dado

Precondición

El juego y el espacio han sido creados correctamente

Postcondición

La salida debe ser el espacio creado anteriormente

Global test1_game_get_space_at ()

Prueba la función que devuelve el espacio en una posición dada

Precondición

El espacio ha sido añadido correctamente

Postcondición

La salida debe ser el espacio añadido anteriormente

Global test1_game_is_over ()

Prueba la función devuelve si el juego ha acabado o no

Precondición

El juego ha sido inicializado

Postcondición

La salida esperada es FALSE

Global test1_game_management_load ()

Prueba la función que carga los datos del juego desde un archivo

Precondición

El nombre del fichero desde donde cargar los datos

Postcondición

La salida esperada es OK

Global test1_game_management_save ()

Prueba que la función guarde los datos de la partida con el mismo formato que los ficheros de carga

Precondición

El juego se ha creado correctamente y el nombre del archivo es valido

Postcondición

La salida esperada es OK

Global test1_game_management_start_from_file ()

Prueba la función que carga los datos del juego desde un archivo

Precondición

Se lee de forma correcta los archivos

Postcondición

La salida esperada es OK

Global test1_game_rules_run_random_rule ()

Prueba la funcion que ejecuta una regla aleatoria

Precondición

El juego se ha inicializado previamente

Postcondición

La salida esperada es OK

Global test1_game_update ()

Prueba la función que actualiza el juego

Precondición

El juego se ha creado y el comando ejecutado es valido

Postcondición

La salida esperada es OK

Global test1_graphic_engine_create ()

Prueba la función que crea el motor grafico

Precondición

El motor grafico se reserva

Postcondición

La salida esperada es el motor grafico inicializado

Global test1_inventory_add_object ()

Prueba la función que añade un objeto a un inventario

Precondición

El inventario ha sido inicializado previamente

Postcondición

La salida esperada es ERROR

Global test1_inventory_create ()

Prueba la función que crea un inventario

Postcondición

La salida debe ser un inventario inicializado (diferente de NULL)

Global test1_inventory_get_max ()

Prueba la función que devuelve el número máximo de objetos de un inventario

Precondición

Se establece previamente el número máximo de objetos a 20

Postcondición

La salida esperada es 20

Global test1_inventory_get_set ()

Prueba la función que devuelve el set de objetos de un inventario

Precondición

El inventario ha sido inicializado previamente

Postcondición

La salida esperada es diferente de NULL

Global test1_inventory_remove_object ()

Prueba la función que elimina un objeto de un inventario

Precondición

Se añade previamente un objeto con el id 10

Postcondición

La salida esperada es OK

Global test1_inventory_set_max ()

Prueba la función que establece el número máximo de objetos de un inventario

Precondición

El inventario ha sido creado previamente

Postcondición

La salida esperada es OK

Global test1_link_create ()

Prueba la función de creación de un link

Precondición

Un identificador como parámetro

Postcondición

Un puntero no nulo al espacio creado

Global test1_link_get_dest_from ()

Prueba la función que devuelve la casilla destino desde otra casilla

Precondición

El enlace se inicializa a (1, 2) y se establece que el id de origen es 1

Postcondición

La salida tiene que ser 2

Global test1_link_get_id ()

Prueba la función para obtener el id de un link

Precondición

Se establece previamente el id del link a 25

Postcondición

La salida debe ser el mismo id: 25

Global test1_link_get_name ()

Prueba la función para obtener el nombre de un link

Precondición

Al link se le pone un nombre previamente

Postcondición

La salida debe ser el mismo nombre establecido

Global test1_link_get_space1 ()

Prueba la función para obtener el space1 de un link

Precondición

Se ha establecido el space1 del link a 26

Postcondición

La salida debe ser el mismo id: 26

Global test1_link_get_space2 ()

Prueba la función para obtener el space2 de un link

Precondición

Se ha establecido el space2 del link a 27

Postcondición

La salida debe ser 27

Global test1_link_get_state ()

Prueba la función para obtener el estado de un link

Precondición

Se establece el estado del enlace como CLOSED

Postcondición

La salida debe ser CLOSED

Global test1_link_set_id ()

Prueba la función para establecer el id de un link

Precondición

El enlace ha sido inicializado previamente

Postcondición

La salida debe ser OK

Global test1_link_set_name ()

Prueba la función para establecer el nombre de un link

Precondición

El link ha sido creado previamente y se le pone un nombre

Postcondición

La salida debe ser OK

Global test1_link_set_space1 ()

Prueba la función para establecer el space1 de un link

Precondición

El enlace ha sido inicializado previamente

Postcondición

La salida debe ser OK

Global test1_link_set_space2 ()

Prueba la función para establecer el space2 de un link

Precondición

El enlace ha sido creado previamente

Postcondición

La salida debe ser OK

Global test1_link_set_state ()

Prueba la función para establecer el estado de un link

Precondición

El link se crea previamente

Postcondición

La salida debe ser OK

Global test1_object_create ()

Prueba si se crea correctamente un objeto

Postcondición

Un puntero no nulo al objeto creado

Global test1_object_Get_Description ()

Prueba leer la descripcion de un objeto

Precondición

Al objeto se le ha asignado previamente la descripcion

Postcondición

La salida esperada es la descripcion asignado antes

Global test1_object_Get_Description2 ()

Prueba leer la descripcion de un objeto

Precondición

Al objeto se le ha asignado previamente la descripcion

Postcondición

La salida esperada es la descripcion asignado antes

Global test1_object_Get_Graphics ()

Prueba leer los graficos de un objeto

Precondición

Al nombre se le ha asignado previamente los graficos "Bryan"

Postcondición

La salida esperada son los graficos asignados antes "Bryan"

Global test1_object_Get_Hidden ()

Prueba leer si se ha escondido un objeto

Precondición

Al objeto se le ha asignado previamente la invibisibilidad

Postcondición

La salida esperada es la movilidad asignada antes

Global test1_object_Get_Id ()

Prueba leer el id de un objeto

Precondición

Al objeto se le ha asignado previamente el id 12

Postcondición

La salida esperada es el id asignado antes (12)

Global test1_object_Get_Illuminates ()

Prueba leer si se ha iluminado un objeto

Precondición

Al objeto se le ha asignado previamente la posibilidad de iluminar

Postcondición

La salida esperada es la movilidad asignada antes

Global test1_object_Get_Light ()

Prueba leer si se puede iluminar un objeto

Precondición

Al objeto se le ha asignado previamente la posibilidad de iluminar

Postcondición

La salida esperada es la movilidad asignada antes

Global test1_object_Get_Mobile ()

Prueba leer la movilidad de un objeto

Precondición

Al objeto se le ha asignado previamente la movilidad

Postcondición

La salida esperada es la movilidad asignada antes

Global test1_object_Get_Moved ()

Prueba leer si se ha movido un objeto

Precondición

Al objeto se le ha asignado previamente la movilidad

Postcondición

La salida esperada es la movilidad asignada antes

Global test1_object_Get_Name ()

Prueba leer el nombre de un objeto

Precondición

Al nombre se le ha asignado previamente el nombre Bryan

Postcondición

La salida esperada es el id asignado antes "Bryan"

Global test1_object_Get_Open ()

Prueba leer si se ha abierto el link a un objeto

Precondición

Al objeto se le ha asignado previamente la disponibilidad

Postcondición

La salida esperada es la movilidad asignada antes

Global test1_object_set_description ()

Prueba si se le asigna correctamente una descripcion a un objeto

Precondición

La descripcion del objeto

Postcondición

La salida tiene que ser el puntero al objeto

Global test1_object_set_description2 ()

Prueba si se le asigna correctamente una descripcion2 a un objeto

Precondición

La descripcion2 del objeto

Postcondición

La salida tiene que ser el puntero al objeto

Global test1_object_set_graphics ()

Prueba si se le asigna correctamente los graficos a un objeto

Precondición

El nombre del objeto

Postcondición

La salida tiene que ser el puntero al objeto

Global test1_object_set_Hidden ()

Prueba si se le asigna correctamente una situacion a un objeto

Precondición

La situacion del objeto

Postcondición

La salida tiene que ser el puntero al objeto

Global test1_object_set_Id ()

Prueba si se le asigna correctamente un Id a un objeto

Precondición

El Id del objeto

Postcondición

La salida tiene que ser el puntero al objeto

Global test1_object_set_Illuminates ()

Prueba si se le asigna correctamente una iluminacion a un objeto

Precondición

La iluminacion del objeto

Postcondición

La salida tiene que ser el puntero al objeto

Global test1_object_set_Light ()

Prueba si se le asigna correctamente una iluminacion a un objeto

Precondición

La iluminacion del objeto

Postcondición

La salida tiene que ser el puntero al objeto

Global test1_object_set_Mobile ()

Prueba si se le asigna correctamente una movilidad a un objeto

Precondición

La movilidad del objeto

Postcondición

La salida tiene que ser el puntero al objeto

Global test1_object_set_Moved ()

Prueba si se le asigna correctamente una movilidad a un objeto

Precondición

La movilidad del objeto

Postcondición

La salida tiene que ser el puntero al objeto

Global test1_object_set_name ()

Prueba si se le asigna correctamente un nombre a un objeto

Precondición

El nombre del objeto

Postcondición

La salida tiene que ser el puntero al objeto

Global test1_object_set_Open ()

Prueba si se le asigna correctamente una situacion a un objeto

Precondición

La situacion del objeto

Postcondición

La salida tiene que ser el puntero al objeto

Global test1_player_add_object ()

Prueba añadir un objeto a un jugador

Precondición

El jugador está sin inicializar

Postcondición

La salida que se espera es ERROR

Global test1_player_create ()

Prueba si se crea correctamente un jugador

Postcondición

Un puntero no nulo al jugador creado

Global test1_player_get_id ()

Prueba leer el id de un jugador

Precondición

Al jugador se le ha asignado previamente el id 12

Postcondición

La salida esperada es el id asignado antes (12)

Global test1_player_get_location ()

Leer la localización de un jugador

Precondición

Al jugador se le ha asignado previamente la localización 12

Postcondición

La salida esperada es la localización previamente asignada (12)

Global test1_player_get_name ()

Intenta leer el nombre de un jugador

Precondición

Al jugador se le ha asignado previamente el nombre "Bob"

Postcondición

La salida esperada es el nombre asignado antes : "Bob"

Global test1_player_has_object ()

Prubea si el jugador tiene un objeto en condiciones normales

Precondición

Se le añade un objeto al jugador

Postcondición

La salida debe ser TRUE

Global test1_player_remove_object ()

Prueba quitar un objeto al jugador en condiciones normales

Precondición

Al jugador se le añade un objeto

Postcondición

La salida que se espera es OK

Global test1_player_set_id ()

Prueba asignar un id a un jugador en condiciones normales

Precondición

Al jugador se le asigna un id cualquiera mayor que cero

Postcondición

La salida tiene que ser el puntero al jugador

Global test1_player_set_location ()

Prueba asignarle una localización a un jugador en condiciones

Precondición

El jugador ha sido previamente inicializado y la localización que se le asigna es valida

Postcondición

La salida es el puntero al jugador

Global test1_player_set_max_objects ()

Prueba poner el número máximo de objetos de un jugador

Precondición

Las condiciones son normales: jugador inicializado, número de objeto dentro de limites.

Postcondición

La salida debe ser el puntero al jugador

Global test1_player_set_name ()

Prueba si se le asigna correctamente un nombre a un jugador

Precondición

El nombre del jugador

Postcondición

La salida tiene que ser el puntero al jugador

Global test1_screen_area_init ()

Prueba la función que inicializa un area

Precondición

Todos los parametros son correctos

Postcondición

La salida esperada es el area inicializado

Global test1_space_add_object ()

Prueba la función que añade un objeto a una casilla

Precondición

El espacio es un puntero a NULL.

Postcondición

La salida debe ser ERROR.

Global test1_space_contains_object ()

Prueba la función que comprueba si una casilla contiene un objeto

Precondición

El espacio se ha inicializado y se ha añadido un objeto

Postcondición

La salida esperada es TRUE

Global test1_space_create ()

Prueba la función de creación de un espacio

Precondición

Un identificador como parámetro

Postcondición

Un puntero no nulo al espacio creado

Global test1_space_get_description ()

Prueba la función que devuelve la descripción de una casilla

Precondición

Al espacio se le ha establecido una descripción previamente

Postcondición

La salda debe ser la descripción previamente establecida

Global test1_space_get_down ()

Prueba la función que devuelve el enlace inferior de una casilla

Precondición

Se ha añadido previamente un enlace on id 7

Postcondición

La salida debe ser el id del enlace 7

Global test1_space_get_east ()

Prueba la función que devuelve el enlace este de una casilla

Precondición

Se ha añadido previamente un enlace on id 7

Postcondición

La salida debe ser el id del enlace 7

Global test1_space_get_graphics ()

Prueba la función que devuelve los graficos de una casilla

Precondición

A la casilla se le han establecido graficos previamente

Postcondición

La salida tiene que ser igual a los graficos establecidos previamente

Global test1_space_get_id ()

Prueba la función que devuelve el id de una casilla

Precondición

Al espacio se le ha establecido un id (12)

Postcondición

La salida esperada es el id establecido 12

Global test1_space_get_iluminated ()

Prueba la funcion que devuelve el estado de la iluminacion en una casilla

Precondición

Se ha creado e iluminado la casilla

Postcondición

La salida esperada es TRUE

Global test1_space_get_long_description ()

Prueba la función que devuelve la descripción larga de una casilla

Precondición

Al espacio se le ha establecido una descripción previamente

Postcondición

La salda debe ser la descripción previamente establecida

Global test1_space_get_name ()

Prueba la función que devuelve el nombre de una casilla

Precondición

Al espacio se le ha establecido un nombre previamente

Postcondición

La salda debe ser el nombre previamente establecido

Global test1_space_get_north ()

Prueba la función que devuelve el enlace norte de una casilla

Precondición

Se ha añadido previamente un enlace on id 7

Postcondición

La salida debe ser el id del enlace 7

Global test1_space_get_objects ()

Prueba la función que devuelve el set de objetos de una casilla

Precondición

El espacio ha sido previamente inicializado.

Postcondición

La salida debe ser diferente de NULL.

Global test1_space_get_south ()

Prueba la función que devuelve el enlace sur de una casilla

Precondición

Se ha añadido previamente un enlace on id 7

Postcondición

La salida debe ser el id del enlace 7

Global test1_space_get_up ()

Prueba la función que devuelve el enlace superior de una casilla

Precondición

Se ha añadido previamente un enlace on id 7

Postcondición

La salida debe ser el id del enlace 7

Global test1_space_get_west ()

Prueba la función que devuelve el enlace oeste de una casilla

Precondición

Se ha añadido previamente un enlace on id 7

Postcondición

La salida debe ser el id del enlace 7

Global test1_space_graphics_areEmpty ()

Prueba la función que comprueba si los graficos de una casilla están vacios.

Precondición

El espacio se ha creado y se le han establecido unos graficos

Postcondición

La salida esperada es FALSE

Global test1_space_remove_object ()

Prueba la función que quita un objeto de una casilla

Precondición

Se añade un objeto previamente

Postcondición

La salida debe ser OK.

Global test1_space_set_description ()

Prueba la función para establecer la descripción de un espacio

Precondición

Nombre que establecer al espacio

Postcondición

La salida debe ser OK

Global test1_space_set_down ()

Prueba la función que establece el enlace inferior de un espacio

Precondición

A la función se le pasa el identificador 4

Postcondición

La salida esperada es OK

Global test1_space_set_east ()

Prueba la función que establece el enlace este de un espacio

Precondición

A la función se le pasa el identificador 4

Postcondición

La salida esperada es OK

Global test1_space_set_graphics ()

Prueba la función que establece los graficos de una casilla

Precondición

Unos graficos aleatorios

Postcondición

La salida esperada es un puntero al espacio

Global test1_space_set_iluminated ()

Prueba la funcion que pone el estado de la iluminacion en una casilla

Precondición

Se ha creado una casiilla con id 5

Postcondición

La salida esperada es OK

Global test1_space_set_long_description ()

Prueba la función para establecer la descripción larga de un espacio

Precondición

Nombre que establecer al espacio

Postcondición

La salida debe ser OK

Global test1_space_set_name ()

Prueba la función para establecer el nombre de un espacio

Precondición

Nombre que establecer al espacio

Postcondición

La salida debe ser OK

Global test1_space_set_north ()

Prueba la función que establece el enlace norte de un espacio

Precondición

Id del enlace

Postcondición

La salida esperada es OK

Global test1_space_set_south ()

Prueba la función que establece el enlace sur de un espacio

Precondición

A la función se le pasa el identificador 4

Postcondición

La salida esperada es OK

Global test1_space_set_up ()

Prueba la función que establece el enlace superior de un espacio

Precondición

A la función se le pasa el identificador 4

Postcondición

La salida esperada es OK

Global test1_space_set_west ()

Prueba la función que establece el enlace oeste de un espacio

Precondición

A la función se le pasa el identificador 4

Postcondición

La salida esperada es OK

Global test2_command_clear ()

Prueba la función que limpia los campos de un commando

Precondición

El gestor de comandos no ha sido inicializado previamente

Postcondición

La salida esperada es ERROR

Global test2_command_destroy ()

Pruba la función que libera memoria usada por un gestor de comandos

Precondición

El gestor de comandos no ha sido inicializado

Postcondición

La salida esperada es ERROR

Global test2_command_get_cmd ()

Prueba la función que devuelve el tipo de comando guardado en un gestor de comandos

Precondición

El gestor de comandos no ha sido inicializado previamente

Postcondición

La salida esperada es UNKNOWN, comando desconocido

Global test2_command_get_cmd_arg ()

Prueba la función que devuelve un argumento de un comando guardado en un gestor de comandos

Precondición

El gestor de comandos no ha sido inicializado previamente

Postcondición

La salida esperada es NULL

Global test2_command_set_cmd ()

Prueba la función que guarda el tipo de comando en un gestor de comandos

Precondición

El gestor de comandos ha sido inicializado previamente pero el comando no es valido

Postcondición

La salida esperada es ERROR

Global test2_command_set_cmd_arg ()

Prueba la función que guarda un argumento de un comando en un gestor de comandos

Precondición

La posición en la que se quiere guardar es invalida

Postcondición

La salida esperada es ERROR

Global test2_dialogue_attack ()

Prueba la funcion que construye el dialogo para el comando ATTACK

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

Global test2_dialogue_dir ()

Prueba la funcion que construye el dialogo para el comando DIR

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

Global test2_dialogue_game_rule ()

Prueba la funcion que construye el dialogo para las reglas del juego

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

Global test2_dialogue_get_text ()

Prueba la funcion que devuelve el texto del dialogo

Precondición

El dialogo no se ha inicializado antes

Postcondición

La salida esperada es NULL

Global test2_dialogue_go ()

Prueba la funcion que construye el dialogo para el comando GO

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

Global test2_dialogue_help ()

Prueba la funcion que construye el dialogo para el comando HELP

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

Global test2_dialogue_inspect ()

Prueba la funcion que construye el dialogo para el comando INSPECT

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

Global test2_dialogue_leave ()

Prueba la funcion que construye el dialogo para el comando LEAVE

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

Global test2_dialogue_load ()

Prueba la funcion que construye el dialogo para el comando LOAD

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

Global test2_dialogue_open ()

Prueba la funcion que construye el dialogo para el comando OPEN

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

Global test2_dialogue_save ()

Prueba la funcion que construye el dialogo para el comando SAVE

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

Global test2_dialogue_take ()

Prueba la funcion que construye el dialogo para el comando TAKE

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

Global test2_dialogue_turn_off ()

Prueba la funcion que construye el dialogo para el comando TURN_OFF

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

Global test2_dialogue_turn_on ()

Prueba la funcion que construye el dialogo para el comando TURN_ON

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

Global test2_dialogue_unknown ()

Prueba la funcion que construye el dialogo para comandos desconocidos

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

Global test2_game_add_link ()

Prueba la función que añade un link al juego

Precondición

El juego no ha sido inicializado pero el link si

Postcondición

La salida esperada es ERROR

Global test2_game_add_object ()

Prueba la función que añade un objeto al juego

Precondición

El juego no ha sido inicializado pero el objeto si

Postcondición

La salida esperada es ERROR

Global test2_game_add_player ()

Prueba la función que añade un jugador al juego

Precondición

El juego no ha sido inicializado pero el jugador si

Postcondición

La salida esperada es ERROR

Global test2_game_add_space ()

Prueba la función que añade un espacio al juego

Precondición

El juego no ha sido inicializado pero el espacio si

Postcondición

La salida esperada es ERROR

Global test2_game_destroy ()

Prueba la funcon que destruye un juego

Precondición

El juego no se ha creado previamente

Postcondición

La salida esperada es ERROR

Global test2_game_get_dialogue ()

Prueba la función que devuelve el dialogo del juego

Precondición

El juego no ha sido inicializado

Postcondición

La salida debe ser NULL

Global test2_game_get_die ()

Prueba la función que devuelve el dado del juego

Precondición

El juego no ha sido inicializado

Postcondición

La salida debe ser NULL

Global test2_game_get_last_inspected_object ()

Prueba la función que devuelve el ultimo objeto inspeccionado

Precondición

El objeto está marcado como oculto

Postcondición

La salida esperada es NULL

Global test2_game_get_last_inspected_space ()

Prueba la función que devuelve el ultimo espacio inspeccionado

Precondición

El juego no ha sido creado

Postcondición

La salida esperada es NULL

Global test2_game_get_link ()

Prueba la función que devuelve el link con in id dado

Precondición

El juego no se ha creado pero el link si

Postcondición

La salida debe ser NULL

Global test2_game_get_link_at ()

Prueba la función que devuelve el link en una posición dada

Precondición

La posición en la que se busca es -1

Postcondición

La salida debe ser NULL

Global test2_game_get_obj_list_as_str ()

Prueba la funcion que devuelve la lista de objetos de una casilla como cadena

Precondición

El juego no se ha inicializado

Postcondición

La salida esperada es NULL

Global test2_game_get_object_at ()

Prueba la función que devuelve el objeto en una posición dada

Precondición

La posición en la que se busca es -1

Postcondición

La salida debe ser NULL

Global test2_game_get_object_location ()

Prueba la funcion que devuelve la localizacion de un objeto

Precondición

El objeto no se ha añadido al juego

Postcondición

La salida es NO_ID - localizacion nula

Global test2_game_get_player ()

Prueba la función que devuelve el jugador del juego

Precondición

El juego ha sido inicializado pero no se ha añadido ningun jugador

Postcondición

La salida debe ser NULL

Global test2_game_get_player_location ()

Prueba la funcion que devuelve la localizacion del jugador

Precondición

El jugador no se ha añadido al juego

Postcondición

La salida es NO_ID, localizacion nula

Global test2_game_get_space ()

Prueba la función que devuelve el espacio con in id dado

Precondición

El juego no se ha creado pero el espacio si

Postcondición

La salida debe ser NULL

Global test2_game_get_space_at ()

Prueba la función que devuelve el espacio en una posición dada

Precondición

La posición en la que se busca es -1

Postcondición

La salida debe ser NULL

Global test2_game_is_over ()

Prueba la función devuelve si el juego ha acabado o no

Precondición

El juego no ha sido inicializado

Postcondición

La salida esperada es TRUE, de esa forma en gameloop no se sigue jugando

Global test2_game_management_load ()

Prueba la función que carga los datos del juego desde un archivo

Precondición

El juego está sin inicializar

Postcondición

La salida esperada es ERROR

Global test2_game_management_save ()

Prueba que la función guarde los datos de la partida con el mismo formato que los ficheros de carga

Precondición

El juego no ha sido inicializado y el nombre del archivo es nulo

Postcondición

La salida esperada es ERROR

Global test2_game_management_start_from_file ()

Prueba la función que carga los datos del juego desde un archivo

Precondición

No se lee de forma correcta del archivo

Postcondición

La salida esperada es ERROR

Global test2_game_rules_run_random_rule ()

Prueba la funcion que ejecuta una regla aleatoria

Precondición

El juego no se ha inicializado previamente

Postcondición

La salida esperada es ERROR

Global test2_game_update ()

Prueba la función que actualiza el juego

Precondición

El juego no se ha creado. El comando ejecutado es valido

Postcondición

La salida esperada es ERROR

Global test2_inventory_add_object ()

Prueba la función que añade un objeto a un inventario

Precondición

El inventario es un puntero a NULL

Postcondición

La salida esperada es ERROR

Global test2_inventory_get_max ()

Prueba la función que devuelve el número máximo de objetos de un inventario

Precondición

El inventario es un puntero a NULL

Postcondición

La salida esperada es -1

Global test2_inventory_get_set ()

Prueba la función que devuelve el set de objetos de un inventario

Precondición

El inventario es un puntero a NULL

Postcondición

La salida esperada es NULL

Global test2_inventory_remove_object ()

Prueba la función que elimina un objeto de un inventario

Precondición

El objeto no ha sido añadido previamente

Postcondición

La salida esperada es ERROR

Global test2_inventory_set_max ()

Prueba la función que establece el número máximo de objetos de un inventario

Precondición

El número máximo de objetos es invalido (-5)

Postcondición

La salida esperada es ERROR

Global test2_link_get_dest_from ()

Prueba la función que devuelve la casilla destino desde otra casilla

Precondición

El enlace se inicializa a (1, 2) y se establece que el id de origen es 2

Postcondición

La salida tiene que ser 1

Global test2_link_get_id ()

Prueba la función para obtener el id de un link

Precondición

El enlace es un puntero a NULL

Postcondición

La salida debe ser NO_ID

Global test2_link_get_name ()

Prueba la función para obtener el nombre de un link

Precondición

El enlace es un puntero a NULL

Postcondición

La salida debe ser NULL

Global test2_link_get_space1 ()

Prueba la función para obtener el space1 de un link

Precondición

El enlace es un puntero a NULL

Postcondición

La salida debe ser NO_ID

Global test2_link_get_space2 ()

Prueba la función para obtener el space2 de un link

Precondición

El enlace es un puntero a NULL

Postcondición

La salida debe ser NO_ID

Global test2_link_get_state ()

Prueba la función para obtener el estado de un link

Precondición

El link es un puntero a NULL

Postcondición

La salida debe ser -1

Global test2_link_set_id ()

Prueba la función para establecer el id de un link

Precondición

El enlace es un puntero a NULL

Postcondición

La salida debe ser ERROR

Global test2_link_set_name ()

Prueba la función para establecer el nombre de un link

Precondición

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

Postcondición

La salida debe ser ERROR

Global test2_link_set_space1 ()

Prueba la función para establecer el space1 de un link

Precondición

El enlace es un puntero a NULL

Postcondición

La salida debe ser ERROR

Global test2_link_set_space2 ()

Prueba la función para establecer el space2 de un link

Precondición

El enlace es un puntero a NULL

Postcondición

La salida debe ser ERROR

Global test2_link_set_state ()

Prueba la función para establecer el estado de un link

Precondición

El enlace es un puntero a NULL

Postcondición

La salida debe ser ERROR

Global test2_object_Get_Description ()

Prueba leer la descripcion de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es NULL

Global test2_object_Get_Description2 ()

Prueba leer la descripcion de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es NULL

Global test2_object_Get_Graphics ()

Prueba leer los graficos de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es NULL

Global test2_object_Get_Hidden ()

Prueba leer la situacion de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es FALSE

Global test2_object_Get_Id ()

Prueba leer el id de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es NO_ID

Global test2_object_Get_Illuminates ()

Prueba leer la situacion de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es FALSE

Global test2_object_Get_Light ()

Prueba leer la situacion de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es FALSE

Global test2_object_Get_Mobile ()

Prueba leer la movilidad de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es FALSE

Global test2_object_Get_Moved ()

Prueba leer la movilidad de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es FALSE

Global test2_object_Get_Name ()

Prueba leer el nombre de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es NULL

Global test2_object_Get_Open ()

Prueba leer la situacion del link a un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es NO_ID

Global test2_object_set_description ()

Prueba asignar una descripcion a un objeto sin inicializar

Precondición

El objeto al que establecer la descripcion es un puntero a NULL

Postcondición

La salida debe ser NULL

Global test2_object_set_description2 ()

Prueba asignar una descripcion2 a un objeto sin inicializar

Precondición

El objeto al que establecer la descripcion2 es un puntero a NULL

Postcondición

La salida debe ser NULL

Global test2_object_set_graphics ()

Prueba asignar graficos a un objeto sin inicializar

Precondición

El objeto al que establecer los graficos es un puntero a NULL

Postcondición

La salida debe ser NULL

Global test2_object_set_Hidden ()

Prueba asignar una situacion a un objeto sin inicializar

Precondición

El objeto al que establecer la situacion es un puntero a NULL

Postcondición

La salida debe ser NULL

Global test2_object_set_Id ()

Prueba asignar un Id a un objeto sin inicializar

Precondición

El objeto al que establecer el Id es un puntero a NULL

Postcondición

La salida debe ser NULL

Global test2_object_set_Illuminates ()

Prueba asignar una iluminacion a un objeto sin inicializar

Precondición

El objeto al que establecer la iluminacion es un puntero a NULL

Postcondición

La salida debe ser NULL

Global test2_object_set_Light ()

Prueba asignar una iluminacion a un objeto sin inicializar

Precondición

El objeto al que establecer la iluminacion es un puntero a NULL

Postcondición

La salida debe ser NULL

Global test2_object_set_Mobile ()

Prueba asignar una movilidad a un objeto sin inicializar

Precondición

El objeto al que establecer la movilidad es un puntero a NULL

Postcondición

La salida debe ser NULL

Global test2_object_set_Moved ()

Prueba asignar una movilidad a un objeto sin inicializar

Precondición

El objeto al que establecer la movilidad es un puntero a NULL

Postcondición

La salida debe ser NULL

Global test2_object_set_name ()

Prueba asignar un nombre a un objeto sin inicializar

Precondición

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

Postcondición

La salida debe ser NULL

Global test2_object_set_Open ()

Prueba asignar una situacion a un objeto sin inicializar

Precondición

El objeto al que establecer la situacion es un puntero a NULL

Postcondición

La salida debe ser NULL

Global test2_player_add_object ()

Prueba añadir un objeto a un jugador en condiciones normales

Precondición

El jugador ha sido correctamente inicializado y el id del objeto es valido

Postcondición

La salida debe ser OK

Global test2_player_get_id ()

Prueba leer el id de un jugador

Precondición

El jugador no ha sido inicializado previamente

Postcondición

La salida esperada es NO_ID

Global test2_player_get_location ()

Prueba leer la localización del jugador

Precondición

El jugador no ha sido inicializado

Postcondición

La salida esperada es NO_ID

Global test2_player_get_name ()

Prueba leer el nombre de un jugador

Precondición

El jugador está sin inicializar

Postcondición

La salida debe ser NULL

Global test2_player_has_object ()

Prueba si el jugador tiene un objeto

Precondición

El jugador está sin inicializar

Postcondición

La salida debe ser FALSE

Global test2_player_remove_object ()

Prueba quitar un objeto a un jugador

Precondición

Al jugador no se le ha añadido todavía ningun objeto

Postcondición

La salida que se espera es ERROR

Global test2_player_set_id ()

Prueba asignar un id a un jugador sin inicializar

Precondición

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

Postcondición

La salida esperada es NULL

Global test2_player_set_location ()

Prueba asignarle una localización a un jugador sin inicializar.

Precondición

El jugador no ha sido inicializado previamente

Postcondición

La salida que se espera es NULL

Global test2_player_set_max_objects ()

Prueba poner el número máximo de objetos de un jugador

Precondición

El número de objetos está por encima del limite permitido

Postcondición

La salida debe ser NULL

Global test2_player_set_name ()

Prueba asignar un nombre a un jugador sin inicializar

Precondición

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

Postcondición

La salida debe ser NULL

Global test2_screen_area_init ()

Prueba la función que inicializa un area

Precondición

La posición x es negativa

Postcondición

La salida esperada es NULL

Global test2_space_add_object ()

Prueba la función que añade un objeto a una casilla

Precondición

Se añade un objeto con id 10

Postcondición

La salida debe ser OK.

Global test2_space_contains_object ()

Prueba la función que comprueba si una casilla contiene un objeto

Precondición

El espacio no se ha inicializado.

Postcondición

La salida esperada es FALSE

Global test2_space_create ()

Prueba la función de creación de un espacio

Precondición

Un identificador como parámetro

Postcondición

El identificador del espacio es el introducido

Global test2_space_get_description ()

Prueba la función que devuelve la descripción de una casilla

Precondición

El espacio es un puntero a NULL

Postcondición

La salda debe ser un puntero a NULL

Global test2_space_get_down ()

Prueba la función que devuelve el enlace inferior de una casilla

Precondición

El espacio es un puntero a NULL.

Postcondición

La salida debe ser NO_ID.

Global test2_space_get_east ()

Prueba la función que devuelve el enlace este de una casilla

Precondición

El espacio es un puntero a NULL.

Postcondición

La salida debe ser NO_ID.

Global test2_space_get_graphics ()

Prueba la función que devuelve los graficos de una casilla

Precondición

El espacio ha sido inicializado pero no se le han establecido graficos

Postcondición

La salida debe ser una matriz de 3 cadenas de 7 caracteres llenas de espacios

Global test2_space_get_id ()

Prueba la función que devuelve el id de una casilla

Precondición

El espacio es un puntero a NULL

Postcondición

La salida esperada es NO_ID

Global test2_space_get_iluminated ()

Prueba la funcion que devuelve el estado de la iluminacion en una casilla

Precondición

No se ha creado la casilla

Postcondición

La salida esperada es FALSE

Global test2_space_get_long_description ()

Prueba la función que devuelve la descripción larga de una casilla

Precondición

El espacio es un puntero a NULL

Postcondición

La salda debe ser un puntero a NULL

Global test2_space_get_name ()

Prueba la función que devuelve el nombre de una casilla

Precondición

El espacio es un puntero a NULL

Postcondición

La salda debe ser un puntero a NULL

Global test2_space_get_north ()

Prueba la función que devuelve el enlace norte de una casilla

Precondición

El espacio es un puntero a NULL.

Postcondición

La salida debe ser NO_ID.

Global test2_space_get_objects ()

Prueba la función que devuelve el set de objetos de una casilla

Precondición

El espacio es un puntero a NULL

Postcondición

La salida debe ser NULL

Global test2_space_get_south ()

Prueba la función que devuelve el enlace sur de una casilla

Precondición

El espacio es un puntero a NULL.

Postcondición

La salida debe ser NO_ID.

Global test2_space_get_up ()

Prueba la función que devuelve el enlace superior de una casilla

Precondición

El espacio es un puntero a NULL.

Postcondición

La salida debe ser NO_ID.

Global test2_space_get_west ()

Prueba la función que devuelve el enlace oeste de una casilla

Precondición

El espacio es un puntero a NULL.

Postcondición

La salida debe ser NO_ID.

Global test2_space_graphics_areEmpty ()

Prueba la función que comprueba si los graficos de una casilla están vacios.

Precondición

El espacio no se ha inicializado

Postcondición

La salida esperada es TRUE.

Global test2_space_remove_object ()

Prueba la función que quita un objeto de una casilla

Precondición

Todavia no he ha añadido ningun objeto

Postcondición

La salida debe ser ERROR.

Global test2_space_set_description ()

Prueba la función para establecer la descripción de un espacio

Precondición

El espacio al que establecer la descripción es un puntero a NULL

Postcondición

La salida debe ser ERROR

Global test2_space_set_down ()

Prueba la función que establece el enlace inferior de un espacio

Precondición

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

Postcondición

La salida esperada es ERROR

Global test2_space_set_east ()

Prueba la función que establece el enlace este de un espacio

Precondición

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

Postcondición

La saida esperada es ERROR

Global test2_space_set_graphics ()

Prueba la función que establece los graficos de una casilla

Precondición

El espacio es un puntero a NULL

Postcondición

La salida esperada es un puntero a NULL

Global test2_space_set_iluminated ()

Prueba la funcion que pone el estado de la iluminacion en una casilla

Precondición

No se ha creado la casilla

Postcondición

La salida esperada es ERROR

Global test2_space_set_long_description ()

Prueba la función para establecer la descripción larga de un espacio

Precondición

El espacio al que establecer la descripción es un puntero a NULL

Postcondición

La salida debe ser ERROR

Global test2_space_set_name ()

Prueba la función para establecer el nombre de un espacio

Precondición

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

Postcondición

La salida debe ser ERROR

Global test2_space_set_north ()

Prueba la función que establece el enlace norte de un espacio

Precondición

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

Postcondición

La salida esperada es ERROR

Global test2_space_set_south ()

Prueba la función que establece el enlace sur de un espacio

Precondición

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

Postcondición

Global test2_space_set_up ()

Prueba la función que establece el enlace superior de un espacio

Precondición

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

Postcondición

La salida esperada es ERROR

Global test2_space_set_west ()

Prueba la función que establece el enlace oeste de un espacio

Precondición

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

Postcondición

La salida esperada es ERROR

Global test3_command_get_cmd ()

Prueba la función que devuelve el tipo de comando guardado en un gestor de comandos

Precondición

El gestor de comandos ha sido inicializado previamente pero no se le ha asignado un comando

Postcondición

La salida esperada es NO_CMD

Global test3_command_get_cmd_arg ()

Prueba la función que devuelve un argumento de un comando guardado en un gestor de comandos

Precondición

El gestor de comandos ha sido inicializado previamente pero no se le ha asignado un argumento

Postcondición

La salida esperada es una cadena vacia

Global test3_command_set_cmd ()

Prueba la función que guarda el tipo de comando en un gestor de comandos

Precondición

El gestor de comandos no ha sido inicializado previamente

Postcondición

La salida esperada es ERROR

Global test3_command_set_cmd_arg ()

Prueba la función que guarda un argumento de un comando en un gestor de comandos

Precondición

El gestor de comandos no ha sido inicializado previamente

Postcondición

La salida esperada es ERROR

Global test3_game_add_link ()

Prueba la función que añade un link al juego

Precondición

El juego ha sido inicializado pero el link no

Postcondición

La salida esperada es ERROR

Global test3_game_add_object ()

Prueba la función que añade un objeto al juego

Precondición

El juego ha sido inicializado pero el objeto no

Postcondición

La salida esperada es ERROR

Global test3_game_add_player ()

Prueba la función que añade un jugador al juego

Precondición

El juego ha sido inicializado pero el jugador no

Postcondición

La salida esperada es ERROR

Global test3_game_add_space ()

Prueba la función que añade un espacio al juego

Precondición

El juego ha sido inicializado pero el espacio no

Postcondición

La salida esperada es ERROR

Global test3_game_get_last_inspected_object ()

Prueba la función que devuelve el ultimo objeto inspeccionado

Precondición

El nombre del objeto no se corresponde con el del objeto del juego

Postcondición

La salida esperada es NULL

Global test3_game_get_link ()

Prueba la función que devuelve el link con in id dado

Precondición

El link buscado no coresponde con el id del link añadido

Postcondición

La salida debe ser NULL

Global test3_game_get_link_at ()

Prueba la función que devuelve el link en una posición dada

Precondición

El juego no se ha inicializado

Postcondición

La salida debe ser NULL

Global test3_game_get_object_at ()

Prueba la función que devuelve el objeto en una posición dada

Precondición

El juego no se ha inicializado

Postcondición

La salida debe ser NULL

Global test3_game_get_object_location ()

Prueba la funcion que devuelve la localizacion de un objeto

Precondición

El objeto está marcado como oculto

Postcondición

La salida es NO_ID - localizacion nula

Global test3_game_get_space ()

Prueba la función que devuelve el espacio con in id dado

Precondición

El espacio buscado no coresponde con el id del espacio añadido

Postcondición

La salida debe ser NULL

Global test3_game_get_space_at ()

Prueba la función que devuelve el espacio en una posición dada

Precondición

El juego no se ha inicializado

Postcondición

La salida debe ser NULL

Global test3_game_update ()

Prueba la función que actualiza el juego

Precondición

El juego se ha creado pero el comando ejecutado es invalido

Postcondición

La salida esperada es ERROR

Global test3_inventory_add_object ()

Prueba la función que añade un objeto a un inventario

Precondición

El id del objeto que se quiere añadir es NO_ID

Postcondición

La salida esperada es ERROR

Global test3_inventory_remove_object ()

Prueba la función que elimina un objeto de un inventario

Precondición

El inventario es un puntero a NULL

Postcondición

La salida esperada es ERROR

Global test3_inventory_set_max ()

Prueba la función que establece el número máximo de objetos de un inventario

Precondición

El inventario es un puntero a NULL

Postcondición

La salida esperada es ERROR

Global test3_link_get_dest_from ()

Prueba la función que devuelve la casilla destino desde otra casilla

Precondición

El enlace es un puntero a NULL

Postcondición

La salida tiene que ser NO_ID

Global test3_link_set_id ()

Prueba la función para establecer el id de un link

Precondición

El id que se quiere establecer es NO_ID

Postcondición

La salida debe ser ERROR

Global test3_link_set_name ()

Prueba la función para establecer el nombre de un link

Precondición

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

Postcondición

La salida debe ser ERROR

Global test3_link_set_space1 ()

Prueba la función para establecer el space1 de un link

Precondición

El id que se quiere establecer es NO_ID

Postcondición

La salida debe ser ERROR

Global test3_link_set_space2 ()

Prueba la función para establecer el space2 de un link

Precondición

El id que se quiere establecer es NO_ID

Postcondición

La salida debe ser ERROR

Global test3_link_set_state ()

Prueba la función para establecer el estado de un link

Precondición

El estado que se le quiere poner al link es invalido

Postcondición

La salida debe ser ERROR

Global test3_player_add_object ()

Prueba añadir un objeto al inventario del jugador

Precondición

El objeto tiene id -1

Postcondición

La salida que se espera es ERROR

Global test3_player_get_id ()

Intenta leer el id del jugador

Precondición

Al jugador no se le ha asignado ninguna id todavia

Postcondición

La salida esperada es NO_ID

Global test3_player_get_location ()

Prueba leer la localización del jugador

Precondición

Al jugador no se le ha asignado ninguna localización todavía

Postcondición

La salida que se espera es NO_ID

Global test3_player_get_name ()

Prueba leer el nombre de un jugador

Precondición

Al jugador no se le ha dado ningun nombre previamente

Postcondición

La salida esperada es una cadena vacía

Global test3_player_has_object ()

Prueba si el jugador tiene un objeto

Precondición

El jugador está inicializado pero no tiene ningun objeto

Postcondición

La salida debe ser FALSE

Global test3_player_remove_object ()

Prueba quitar un objeto a un jugador

Precondición

El jugador está sin inicializar

Postcondición

La salida que se espera es ERROR

Global test3_player_set_id ()

Prueba asignar un id a un jugador

Precondición

El id que se quiere asignar es NO_ID

Postcondición

La salida que se espera es NULL

Global test3_player_set_location ()

Prueba asignarle una localización invalida a un jugador.

Precondición

La localización que se quiere asignar al jugador es es negativa

Postcondición

La salida esperada es NULL

Global test3_player_set_max_objects ()

Prueba poner el número máximo de objetos de un jugador

Precondición

El jugador está sin inicializar

Postcondición

La salida debe ser NULL

Global test3_player_set_name ()

Prueba asignarle un nombre nulo a un jugador

Precondición

La cadena que se quiere asignar es un puntero a NULL

Postcondición

La salida tiene que ser NULL

Global test3_screen_area_init ()

Prueba la función que inicializa un area

Precondición

La posición y es negativa

Postcondición

La salida esperada es NULL

Global test3_space_add_object ()

Prueba la función que añade un objeto a una casilla

Precondición

El id del objeto es -1 (NO_ID)

Postcondición

La salida debe ser ERROR.

Global test3_space_contains_object ()

Prueba la función que comprueba si una casilla contiene un objeto

Precondición

El espacio se ha creado pero no se ha añadido ningun objeto

Postcondición

La salida esperada es FALSE

Global test3_space_create ()

Prueba la función de creación de un espacio

Precondición

El identificador del espacio es NO_ID

Postcondición

La salida esperada es un puntero a NULL

Global test3_space_get_description ()

Prueba la función que devuelve la descripción de una casilla

Precondición

El espacio ha sido inicializado pero no se le ha establecido ninguna descripción

Postcondición

La salida debe ser una cadena vacila

Global test3_space_get_down ()

Prueba la función que devuelve el enlace inferior de una casilla

Precondición

No se ha establecido ningun enlace previamente

Postcondición

La salida debe ser NO_ID.

Global test3_space_get_east ()

Prueba la función que devuelve el enlace este de una casilla

Precondición

No se ha establecido ningun enlace previamente

Postcondición

La salida debe ser NO_ID.

Global test3_space_get_long_description ()

Prueba la función que devuelve la descripción larga de una casilla

Precondición

El espacio ha sido inicializado pero no se le ha establecido ninguna descripción

Postcondición

La salida debe ser una cadena vacila

Global test3_space_get_name ()

Prueba la función que devuelve el nombre de una casilla

Precondición

El espacio ha sido inicializado pero no se le ha establecido ningun nombre

Postcondición

La salida debe ser una cadena vacila

Global test3_space_get_north ()

Prueba la función que devuelve el enlace norte de una casilla

Precondición

No se ha establecido ningun enlace previamente

Postcondición

La salida debe ser NO_ID.

Global test3_space_get_south ()

Prueba la función que devuelve el enlace sur de una casilla

Precondición

No se ha establecido ningun enlace previamente

Postcondición

La salida debe ser NO_ID.

Global test3_space_get_up ()

Prueba la función que devuelve el enlace superior de una casilla

Precondición

No se ha establecido ningun enlace previamente

Postcondición

La salida debe ser NO_ID.

Global test3_space_get_west ()

Prueba la función que devuelve el enlace oeste de una casilla

Precondición

No se ha establecido ningun enlace previamente

Postcondición

La salida debe ser NO_ID.

Global test3_space_graphics_areEmpty ()

Prueba la función que comprueba si los graficos de una casilla están vacios.

Precondición

El espacio se ha creado pero no se le han establecido graficos

Postcondición

La salida esperada es TRUE

Global test3_space_remove_object ()

Prueba la función que quita un objeto de una casilla

Precondición

El espacio no se ha inicializado previamente

Postcondición

La salida debe ser ERROR.

Global test3_space_set_description ()

Prueba la función para establecer la descripción de un espacio

Precondición

El espacio es un puntero no NULL, pero la descripción a establecer es NULL

Postcondición

La salida debe ser ERROR

Global test3_space_set_down ()

Prueba la función que establece el enlace inferior de un espacio

Precondición

El id que se le pasa a la función es NO_ID

Postcondición

La salida esperada es ERROR

Global test3_space_set_east ()

Prueba la función que establece el enlace este de un espacio

Precondición

El id que se le pasa a la función es NO_ID

Postcondición

La salida esperada es ERROR

Global test3_space_set_long_description ()

Prueba la función para establecer la descripción larga de un espacio

Precondición

El espacio es un puntero no NULL, pero la descripción a establecer es NULL

Postcondición

La salida debe ser ERROR

Global test3_space_set_name ()

Prueba la función para establecer el nombre de un espacio

Precondición

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

Postcondición

La salida debe ser ERROR

Global test3_space_set_north ()

Prueba la función que establece el enlace norte de un espacio

Precondición

El id que se le pasa a la función es NO_ID

Postcondición

La salida esperada es ERROR

Global test3_space_set_south ()

Prueba la función que establece el enlace sur de un espacio

Precondición

El id que se le pasa a la función es NO_ID

Postcondición

La salida esperada es ERROR

Global test3_space_set_up ()

Prueba la función que establece el enlace superior de un espacio

Precondición

El id que se le pasa a la función es NO_ID

Postcondición

La salida esperada es ERROR

Global test3_space_set_west ()

Prueba la función que establece el enlace oeste de un espacio

Precondición

El id que se le pasa a la función es NO_ID

Postcondición

La salida esperada es ERROR

Global test4_command_set_cmd_arg ()

Prueba la función que guarda un argumento de un comando en un gestor de comandos

Precondición

El argumento es un puntero a NULL

Postcondición

La salida esperada es ERROR

Global test4_link_get_dest_from ()

Prueba la función que devuelve la casilla destino desde otra casilla

Precondición

El id de origen no está en el enlace

Postcondición

La salida tiene que ser NO_ID

Global test4_player_remove_object ()

Prueba quitar un objeto al jugador

Precondición

El objeto ya ha sido borrado anteriormente

Postcondición

La salida que se espera es ERROR

Global test4_player_set_max_objects ()

Prueba poner el numero máximo de objetos de un jugador

Precondición

El número de objetos es un número negativo

Postcondición

La salida debe ser NULL

Global test4_screen_area_init ()

Prueba la función que inicializa un area

Precondición

El ancho del area es negativo

Postcondición

La salida esperada es NULL

Global test4_space_remove_object ()

Prueba la función que quita un objeto de una casilla

Precondición

El objeto ya ha sido eliminado previamente

Postcondición

La salida debe ser ERROR.

Global test5_screen_area_init ()

Prueba la función que inicializa un area

Precondición

El alto del area es negativo

Postcondición

La salida esperada es NULL

Capítulo 3

Indice de archivos

3.1. Lista de archivos

Lista de todos los archivos documentados y con descripciones breves:

include/command.h
: Implementa el interpreatador de comandos. Lee los comandos que introduce el usuario parseandolos al tipo de datos T_Command, para poder trabajar facilmente con estos comandos . 61

include/dialogue.h
: Implementa el dialogo del juego 65

include/die.h
: Primitivas del TAD Die en el juego 72

include/game.h
: It defines the game interface for each command 75

include/game_management.h
: Define los prototipos de las funciones necesarias para cargar los datos del juego desde un archivo asi como guardar una partida 84

include/game_rules.h
: Define los prototipos de las funciones necesarias para el funcionamiento de las funciones de game_rules.c 86

include/graphic_engine.h
: It defines a textual graphic engine 87

include/inventory.h
: Define la interfaz pública del TAD Inventory 90

include/link.h
: Implementa el nuevo TAD Link (Enlace). Sirve para enlazar de forma más potente las casillas 93

include/object.h
: Define la interfaz pública del TAD Objeto 99

include/player.h
: Define el TAD player y los prototipos de las funciones necesarias para trabajar con este TAD . 107

include/screen.h
: Define la intefaz pública del TAD screen 112

include/set.h
: Definicion de los prototipos de las primitivas del TAD Set 114

include/space.h
: Define un espacio 117

include/types.h
: Define tipos de datos comunes 206

include/tests/command_test.h
: Define las funciones para la prueba del modulo Command 128

include/tests/dialogue_test.h
: Define las funciones para la prueba del modulo dialogue 133

include/tests/game_management_test.h
Define las funciones para la prueba del modulo Game_Management . . . 140
include/tests/game_rules_test.h
Define las funciones para la prueba del modulo game_rules . . . 142
include/tests/game_test.h
Define las funciones para la prueba del modulo Game . . . 143
include/tests/graphic_engine_test.h
Define las funciones para la prueba del modulo graphic_engine . . . 155
include/tests/inventory_test.h
Define las funciones para la prueba del modulo Inventory . . . 156
include/tests/link_test.h
Pruebas para el modulo Link . . . 160
include/tests/object_test.h
Pruebas para el modulo Object . . . 167
include/tests/player_test.h
It declares the tests for the player module . . . 177
include/tests/screen_test.h
Define las funciones para la prueba del modulo Screen . . . 185
include/tests/space_test.h
It declares the tests for the space module . . . 186
include/tests/test.h
Define macros útiles para las pruebas unitarias . . . 205
src/command.c
Implementa el interpreatador de comandos Lee los comandos que introduce el usuario parseandolos al tipo de datos T_Command, para poder trabajar facilmente con estos comandos . . . 207
src/command_test.c
Programa para probar el modulo Command. Progama para probar la correcta funcionalidad del nuevo TAD Command . . . 210
src/dialogue.c
Implementa Los dialogos del juego . . . 215
src/dialogue_test.c
Programa para probar el modulo dialogue. Progama para probar la correcta funcionalidad del nuevo TAD dialogue . . . 220
src/die.c
Implememtacion del dado del juego . . . 228
src/die_test.c
Implementacion del dado en el juego Progama para probar la correcta funcionalidad del nuevo TAD Die . . . 231
src/game.c
It implements the game interface and all the associated callbacks for each command . . . 231
src/game_loop.c
It defines the game loop . . . 240
src/game_management.c
Este modulo se encarga de cargar los datos del juego. Carga las casillas los enlaces, los objetos y los datos del jugador. Tambié incluye la función de guardado . . . 240
src/game_management_test.c
Programa para probar el modulo game_management. Progama para probar la correcta funcionalidad del nuevo TAD Game_management . . . 242
src/game_rules.c
Implementacion de las reglas del juego . . . 244
src/game_rules_test.c
Programa para probar el modulo game_management. Progama para probar la correcta funcionalidad del nuevo TAD Game_management . . . 244
src/game_test.c
Programa para probar el modulo game. Progama para probar la correcta funcionalidad del nuevo TAD Game . . . 246

src/graphic_engine_test.c
Programa para probar el modulo graphic_engine. Progama para probar la correcta funcionalidad del nuevo TAD graphic_engine . . . 258
src/inventory.c
Implememtacion del inventario del juego . . . 259
src/inventory_test.c
Programa para probar el modulo Inventory. Progama para probar la correcta funcionalidad del nuevo TAD Inventory . . . 261
src/link.c
Módulo que define el TAD Enlace así como las primitivas encargadas de trabajar con este TAD 265
src/link_test.c
Programa que prueba la funcionalidad del TAD link . . . 270
src/object.c
Imlementación del TAD Objeto . . . 277
src/object_test.c
Prueba el módulo Object . . . 286
src/player.c
Define el TAD player . . . 297
src/player_test.c
Prueba el módulo Player . . . 301
src/screen.c
Modulo necesario para imprimir por pantalla las areas del juego . . . 309
src/screen_test.c
Programa para probar el modulo screen. Progama para probar la correcta funcionalidad del nuevo TAD screen . . . 311
src/set.c
Implememtacion del set del juego . . . 313
src/set_test.c
Implementacion del conjuntos del juego Progama para probar la correcta funcionalidad del nuevo TAD Set . . . 316
src/space.c
Implementa el TAD space Define los campos de la estructura Space. Tambien define todas las funciones necesarias para trabajar con este tipo de datos . . . 317
src/space_test.c
It tests space module . . . 328

Capítulo 4

Documentación de archivos

4.1. Referencia del Archivo include/command.h

Implementa el interpreatador de comandos. Lee los comandos que introduce el usuario parseandolos al tipo de datos T_Command, para poder trabajar facilmente con estos comandos.

```
#include "types.h"
```

'defines'

- #define CMD_LENGHT 70

  Extension del comando.

'typedefs'

- typedef struct _Command Command

  Definicion de la estructura del comando.
- typedef enum enum_Command T_Command

  Define los tipos de comandos que existen en el juego.

Enumeraciones

- enum enum_Command {
  NO_CMD = -1, UNKNOWN, QUIT, TAKE,
  LEAVE, ROLL, GO, INSPECT,
  TURNON, TURNOFF, OPEN, **SAVE**,
  LOAD, **DIR**, **HELP**, **ATTACK** }

  Define los tipos de comandos que existen en el juego.

Funciones

- Command * Command_ini ()

  Crea e inicializa una estructura tipo Command.
- STATUS Command_destroy (Command *cmdManager)

  Libera memoria usada por una estructura Command.
- STATUS Command_set_cmd (Command *cmdManager, T_Command cmd)

  Guarda un comando en la estructura.

- STATUS Command_set_cmd_arg (Command *cmdManager, char *arg, int index)

  Guarda un parametro de un comando en la estructura en una posición dada.
- T_Command Command_get_cmd (Command *cmdManager)

  Devuelve el comando guardado en la estructura.
- char * Command_get_cmd_arg (Command *cmdManager, int index)

  Devuelve un parametro del comando guardado en la estructura en una posición dada.
- STATUS Command_clear (Command *cmdManager)

  Limpia los datos de una estructura Command.
- STATUS get_user_input (Command *cmdManager)

  Lee los comandos de introduce el usuario.

4.1.1. Descripción detallada

Implementa el interpreatador de comandos. Lee los comandos que introduce el usuario parseandolos al tipo de datos T_Command, para poder trabajar facilmente con estos comandos.

Autor

Profesores PPROG
Javier Bernardo y Mihai Blidaru

Versión

3.0

Fecha

09/03/2017

Copyright

GNU Public License

4.1.2. Documentación de los 'typedefs'

4.1.2.1. typedef struct _Command Command

Definicion de la estructura del comando.

Tad Comando. Gestiona la entrada de comandos en el juego

4.1.3. Documentación de las enumeraciones

4.1.3.1. enum enum_Command

Define los tipos de comandos que existen en el juego.

Valores de enumeraciones

NO_CMD Sin Comando

UNKNOWN Comando desconocido

QUIT Comando salir

TAKE Comando coger

LEAVE Comando dejar

ROLL Comando lanzar

GO Comando ir

INSPECT Comando inspeccionar

TURNON Comando encender

TURNOFF Comando apagar

OPEN Comando abrir

4.1.4. Documentación de las funciones

4.1.4.1. STATUS Command_clear (Command * *cmdManager*)

Limpia los datos de una estructura Command.

Autor

Mihai Blidaru

Parámetros

cmdManager	Una estructura Command

Devuelve

OK si se ha hecho correctamente o ERROR en caso contrario

4.1.4.2. STATUS Command_destroy (Command * *cmdManager*)

Libera memoria usada por una estructura Command.

Autor

Mihai Blidaru

Parámetros

cmdManager	La estructura command que se tiene que liberar

Devuelve

OK si se ha realizado correctamente o ERROR en caso contrario

4.1.4.3. T_Command Command_get_cmd (Command * *cmdManager*)

Devuelve el comando guardado en la estructura.

Autor

Mihai Blidaru

Parámetros

cmdManager	La estructura Command de la cual leer los datos

Devuelve

El comando guardado

4.1.4.4. char* Command_get_cmd_arg (Command * *cmdManager,* int *index*)

Devuelve un parametro del comando guardado en la estructura en una posición dada.

Autor

Mihai Blidaru

Parámetros

cmdManager	La estructura Command de la cual leer los datos
index	El indice del parametro que se quiere leer

Devuelve

El parametro del comando guardado

4.1.4.5. Command* Command_ini ( )

Crea e inicializa una estructura tipo Command.

Autor

Mihai Blidaru

Devuelve

Una estructura command inicializada o NULL si hay algun error.

4.1.4.6. STATUS Command_set_cmd (Command * *cmdManager,* T_Command *cmd*)

Guarda un comando en la estructura.

Autor

Mihai Blidaru

Parámetros

cmdManager	Una estructura tipo command
cmd	El comando a guardar

Devuelve

OK si se ha guardado correctamente o error en caso contrario

4.1.4.7. STATUS Command_set_cmd_arg (Command * *cmdManager,* char * *arg,* int *index*)

Guarda un parametro de un comando en la estructura en una posición dada.

Autor

Mihai Blidaru

Parámetros

cmdManager	Una estructura tipo command
arg	El parametro a guardar
index	La posición donde se quiere guardar el parametro

Devuelve

OK si se ha guardado correctamente o error en caso contrario

4.1.4.8. STATUS get_user_input (Command * *cmdManager*)

Lee los comandos de introduce el usuario.

Lee por teclado el comando que introduce el usuario y comprueba que esta en la lista de los comandos permitidos. Guarda en la estructura Command los datos introducidos: en cmd guarda el tipo de comando y en arg guarda el argumento que tiene un comando

Autor

Profesores PPROG
Javier Bernardo

Parámetros

cmdManager	Una estructura tipo Command donde guardar los datos leidos por teclado

Devuelve

OK si ha leido bien el comando o ERROR en caso contrario

4.2. Referencia del Archivo include/dialogue.h

Implementa el dialogo del juego.

```
#include "space.h"
#include "link.h"
#include "object.h"
```

'defines'

- #define GLOBAL_NO_ARGS-20

'typedefs'

- typedef struct _Dialogue Dialogue

Enumeraciones

- enum TAKE_STATUS {
  NO_OBJ, **NOT_IN_SPACE**, **NOT_MOBILE**, **INVENTORY_FULL**,
  TAKE_OK }

  Codigos de estado del comando TAKE.
- enum LEAVE_STATUS { **NOT_IN_INV**, **LEAVE_OK** }

  Codigos de estado del comando LEAVE.
- enum DIALOGUE_SAVE_STATUS { SAVE_PROTECTED_FILE, SAVE_SAVE_OK, SAVE_WRITE_FAILED }

  Codigos de estado del comando SAVE.
- enum INSPECT_STATUS { **INSPECT_NO_OBJ**, **INSPECT_OK** }

  Codigos de estado del comando INSPECT.
- enum TURN_STATUS { **TURN_NOT_IN_INV**, **TURN_NO_LIGHT**, **TURN_ALREADY**, **TURN_OK** }

  Codigos de estado de los comandos TURNON TURNOFF.
- enum DIALOGUE_LOAD_STATUS { **LOAD_OK**, **LOAD_ERROR** }

  Codigos de estado del comando LOAD.
- enum GAME_RULES_LIST {
  NO_RULE, **LIGHT_OBJECT**, **LIGHT_SPACE**, **HIDE_OBJECT**,
  LOSE_OBJECT, **CLOSE_LINK**, **CHANGE_LAST_LINK** }

  Codigos de idenficación de las reglas.
- enum ATTACK_STATUS { **NOTHING_TO_ATTACK**, **BOSS1_OK**, **BOSS2_OK**, **WRONG_WEAPON** }

  Codigos de estado del comando ATTACK.
- enum OPEN_STATUS {
  WRONG_SYNTAX, **NO_OBJECT**, **NO_LINK**, **NOT_SAME_ID**,
  OPEN_OK }

  Codigos de estado del comando OPEN.

Funciones

- Dialogue ∗ dialogue_ini ()

  Inicializa un modulo dialogo.
- void dialogue_destroy (Dialogue ∗d)

  Libera la memoria usasa por un dialogo.
- char ∗ dialogue_get_text (Dialogue ∗d)

  Devuelve el texto del dialogo.
- STATUS dialogue_go (Dialogue ∗d, DIRECTION direction, Space ∗space, STATUS status, char ∗dir_name, Link ∗link)

  Contruye el texto del dialogo para el comando GO.
- STATUS dialogue_unknown (Dialogue ∗d)

  Contruye el texto del dialogo para comandos desconocidos.
- STATUS dialogue_dir (Dialogue ∗d)

  Contruye el texto del dialogo para el comando DIR.
- STATUS dialogue_take (Dialogue ∗d, Object ∗object, char ∗name, TAKE_STATUS status)

  Contruye el texto del dialogo para el comando TAKE.
- STATUS dialogue_leave (Dialogue ∗d, Object ∗object, char ∗name, LEAVE_STATUS status)

  Contruye el texto del dialogo para el comando LEAVE.
- STATUS dialogue_save (Dialogue ∗d, char ∗name, DIALOGUE_SAVE_STATUS status)

  Contruye el texto del dialogo para el comando SAVE.
- STATUS dialogue_load (Dialogue ∗d, char ∗name, DIALOGUE_LOAD_STATUS status)

  Contruye el texto del dialogo para el comando LOAD.
- STATUS dialogue_attack (Dialogue ∗d, ATTACK_STATUS status)

Contruye el texto del dialogo para el comando ATTACK.

- STATUS dialogue_inspect (Dialogue *d, Object *object, Space *space, char *name, INSPECT_STATUS status)

  Contruye el texto del dialogo para el comando TAKE.

- STATUS dialogue_open (Dialogue *d, char *objName, char *linkName, OPEN_STATUS status)

  Contruye el texto del dialogo para el comando TAKE.

- STATUS dialogue_turn_on (Dialogue *d, Object *object, char *name, TURN_STATUS status)

  Contruye el texto del dialogo para el comando TURN_ON.

- STATUS dialogue_turn_off (Dialogue *d, Object *object, char *name, TURN_STATUS status)

  Contruye el texto del dialogo para el comando TURN_ON.

- STATUS dialogue_help (Dialogue *d)

  Contruye el texto del dialogo para el comando HELP.

- STATUS dialogue_game_rule (Dialogue *d, int rule)

  Contruye el texto del dialogo para las reglas dej juego.

4.2.1. Descripción detallada

Implementa el dialogo del juego.

Autor

Profesores PPROG
Mihai Blidaru
Sandra Benítez

Versión

3.0

Fecha

09/03/2017

Copyright

GNU Public License

4.2.2. Documentación de los 'defines'

4.2.2.1. #define GLOBAL_NO_ARGS-20

Codigo de error global para acciones a los que no se han pasado los argumentos necesarios

4.2.3. Documentación de los 'typedefs'

4.2.3.1. typedef struct _Dialogue Dialogue

Declaración del tipo dialogo

4.2.4. Documentación de las enumeraciones

4.2.4.1. enum DIALOGUE_SAVE_STATUS

Codigos de estado del comando SAVE.

Valores de enumeraciones

SAVE_PROTECTED_FILE EL archivo donde se quiere guardar es un archivo protegido

SAVE_SAVE_OK Se ha gaurdado todo sin problemas

SAVE_WRITE_FAILED No se puede escribir en ese archivo

4.2.5. Documentación de las funciones

4.2.5.1. STATUS dialogue_attack (Dialogue * *d,* ATTACK_STATUS *status*)

Contruye el texto del dialogo para el comando ATTACK.

Parámetros

d	Una estructura dialogo
status	El codigo de error del comando

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.2.5.2. void dialogue_destroy (Dialogue * *d*)

Libera la memoria usasa por un dialogo.

Autor

Javier Bernardo

Parámetros

d	El dialogo que se quiere destruir

4.2.5.3. STATUS dialogue_dir (Dialogue * *d*)

Contruye el texto del dialogo para el comando DIR.

Parámetros

d	Una estructura dialogo

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.2.5.4. STATUS dialogue_game_rule (Dialogue * *d,* int *rule*)

Contruye el texto del dialogo para las reglas dej juego.

Parámetros

d	Una estructura dialogo
rule	Codigo de la regla que se ha ejecutado

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.2.5.5. char* dialogue_get_text (Dialogue * *d*)

Devuelve el texto del dialogo.

Parámetros

d	El dialogo del que se quiere leer el texto

Devuelve

El texto del dialogo

4.2.5.6. STATUS dialogue_go (Dialogue * *d*, DIRECTION *direction*, Space * *space*, STATUS *status*, char * *dir_name*, Link * *link*)

Contruye el texto del dialogo para el comando GO.

Parámetros

d	Una estructura dialogo
direction	La direccion en la que se quiere mover
space	La casilla destino
status	Si se ha llevado a cabo con exito o no
dir_name	la dirrecion introducida por teclado
link	El enlace por el que se quiere pasar

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.2.5.7. STATUS dialogue_help (Dialogue * *d*)

Contruye el texto del dialogo para el comando HELP.

Parámetros

d	Una estructura dialogo

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.2.5.8. Dialogue* dialogue_ini ( )

Inicializa un modulo dialogo.

Autor

Javier Bernardo

Devuelve

Un dialogo inicializado

4.2.5.9. STATUS dialogue_inspect (Dialogue * *d,* Object * *object,* Space * *space,* char * *name,* INSPECT_STATUS *status*)

Contruye el texto del dialogo para el comando TAKE.

Parámetros

d	Una estructura dialogo
object	El objeto que se quiere inspeccionar si existe
space	El espacio que se quiere inspeccionar si existe
name	El nombre introducido por teclado
status	El codigo de error del comando

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.2.5.10. STATUS dialogue_leave (Dialogue * *d,* Object * *object,* char * *name,* LEAVE_STATUS *status*)

Contruye el texto del dialogo para el comando LEAVE.

Parámetros

d	Una estructura dialogo
object	El objeto que se quiere dejar si existe
name	El nombre introducido por teclado
status	El codigo de error del comando

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.2.5.11. STATUS dialogue_load (Dialogue * *d,* char * *name,* DIALOGUE_LOAD_STATUS *status*)

Contruye el texto del dialogo para el comando LOAD.

Parámetros

d	Una estructura dialogo
name	El nombre del fichero desde el cual se quiere cargar
status	El codigo de error del comando

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.2.5.12. STATUS dialogue_open (Dialogue * *d,* char * *objName,* char * *linkName,* OPEN_STATUS *status*)

Contruye el texto del dialogo para el comando TAKE.

Parámetros

d	Una estructura dialogo
objName	Nombre del objeto con el que se quiere abrir
linkName	Nombre del link que se quiere abris
status	Codigo de error de la operacion

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.2.5.13. STATUS dialogue_save (Dialogue * *d*, char * *name*, DIALOGUE_SAVE_STATUS *status*)

Contruye el texto del dialogo para el comando SAVE.

Parámetros

d	Una estructura dialogo
name	El nombre del ficheri donde se quiere guardar
status	El codigo de error del comando

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.2.5.14. STATUS dialogue_take (Dialogue * *d*, Object * *object*, char * *name*, TAKE_STATUS *status*)

Contruye el texto del dialogo para el comando TAKE.

Parámetros

d	Una estructura dialogo
object	El objeto que se quiere llevar si existe
name	El nombre introducido por teclado
status	El codigo de error del comando

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.2.5.15. STATUS dialogue_turn_off (Dialogue * *d*, Object * *object*, char * *name*, TURN_STATUS *status*)

Contruye el texto del dialogo para el comando TURN_ON.

Parámetros

d	Una estructura dialogo
object	El objeto que se quiere apagar
name	El nombre introducido por teclado
status	El codigo de error del comando

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.2.5.16. STATUS dialogue_turn_on (Dialogue * *d*, Object * *object*, char * *name*, TURN_STATUS *status*)

Contruye el texto del dialogo para el comando TURN_ON.

Parámetros

d	Una estructura dialogo
object	El objeto que se quiere encender
name	El nombre introducido por teclado
status	El codigo de error del comando

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.2.5.17. STATUS dialogue_unknown (Dialogue * *d*)

Contruye el texto del dialogo para comandos desconocidos.

Parámetros

d	Una estructura dialogo

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.3. Referencia del Archivo include/die.h

Primitivas del TAD Die en el juego.

```
#include "types.h"
```

'defines'

- #define DIE_ID 1

'typedefs'

- typedef struct _Die Die

Funciones

- Die * die_create (Id die_id)
  Crea un nuevo dado.
- STATUS die_destroy (Die *die)
  Destruir un nuevo dado.
- unsigned short die_roll (Die *die)
  Lanza el dado.
- int die_print (Die *die)
  Imprimir el dado.
- int die_get_number (Die *die)
  Obtiene el ultimo valor del dado.
- STATUS die_set_faces (Die *die, int faces)
  Coloca un numero de caras al dado.
- int die_get_faces (Die *die)
  Coloca un numero de caras al dado.

4.3.1. Descripción detallada

Primitivas del TAD Die en el juego.

Autor

Javier Bernando

Versión

1.0

Fecha

20-02-2017

4.3.2. Documentación de los 'defines'

4.3.2.1. #define DIE_ID 1

Id del dado por defecto

4.3.3. Documentación de los 'typedefs'

4.3.3.1. typedef struct _Die Die

Definición del tipo

4.3.4. Documentación de las funciones

4.3.4.1. Die* die_create (Id *die_id*)

Crea un nuevo dado.

Autor

Javier Bernardo

Parámetros

die_id	Id que define el dado

Devuelve

dado creado

4.3.4.2. STATUS die_destroy (Die * *die*)

Destruir un nuevo dado.

Autor

Javier Bernardo

Parámetros

die	Die que quieres destruir

Devuelve

OK si se elimina, ERROR en caso contrario

4.3.4.3. int die_get_faces (Die * *die*)

Coloca un numero de caras al dado.

Autor

Javier Bernardo

Parámetros

die	Die que quieres lanzar

Devuelve

El numero de caras del dado

4.3.4.4. int die_get_number (Die * *die*)

Obtiene el ultimo valor del dado.

Autor

Javier Bernardo

Parámetros

die	Die del que quieres saber su valor

Devuelve

Devuelve el valor en int de la estructura numero del dado

4.3.4.5. int die_print (Die * *die*)

Imprimir el dado.

Autor

Javier Bernardo

Parámetros

die	Die que quieres imprimir

Devuelve

Impresion por pantalla del dado y su valor

4.3.4.6. unsigned short die_roll (Die * *die*)

Lanza el dado.

Autor

Javier Bernardo

Parámetros

die	Die que quieres lanzar

Devuelve

Devuelve el valor del dado lanzado con valores entre 1 y 6

4.3.4.7. STATUS die_set_faces (Die * *die,* int *faces*)

Coloca un numero de caras al dado.

Autor

Javier Bernardo

Parámetros

die	Die que quieres lanzar
faces	Numero de caras del dado

Devuelve

OK si lo hace bien o ERROR si da algun problema

4.4. Referencia del Archivo include/game.h

It defines the game interface for each command.

```
#include "command.h"
#include "space.h"
#include "player.h"
#include "object.h"
#include "types.h"
#include "die.h"
#include "set.h"
#include "link.h"
#include "dialogue.h"
```

'defines'

- #define MAX_OBJECTS 20
- #define MAX_SPACES 10000

'typedefs'

- typedef struct _Game Game

Funciones

- Game * game_create ()

  Inicializa los datos del juego Esta función inicializa los espacios, crea e inicializa el jugador, los objetos y el dado.
- STATUS game_destroy (Game *game)

  Destruye la estructura game. Se encarga de liberar la memoria reservada por los elementos del juego spaces, player, objects y die.
- STATUS game_update (Game *game, Command *command)

  Ejecuta una de las funciones callback en funcion del comando recibido.
- BOOL game_is_over (Game *game)

  Devuelve si el juego a acabado o no.
- STATUS game_add_object (Game *game, Object *object)

  Añade un objeto al juego.
- STATUS game_add_space (Game *game, Space *space)

  Añade un espacio al juego.
- STATUS game_add_link (Game *game, Link *link)

  Añade un objeto al juego.
- STATUS game_add_player (Game *game, Player *player)

  Añade un jugador al juego. Si el jugador ya está inicializado, lo sustituye por otro.
- Link * game_get_link (Game *game, Id link_id)

  Devuelve un puntero al link que tiene el Id igual al Id pasado como parametro.
- Link * game_get_link_at (Game *game, int index)

  Develve el link en una posición dada.
- Space * game_get_space (Game *game, Id space_id)

  Devuelve un puntero al space que tiene el Id igual al Id pasado como parametro.
- Space * game_get_space_at (Game *game, int index)

  Develve el espacop una posición dada.
- Object * game_get_object_at (Game *game, int index)

  Develve el objeto una posición dada.
- Die * game_get_die (Game *game)

  Devuelve el dado del juego.
- Player * game_get_player (Game *game)

  Devuelve el jugador del juego.
- char * game_get_obj_list_as_str (Game *game, Space *space)

  Devuelve la lista de objetos de una casilla como cadena.
- Space * game_get_last_inspected_space (Game *game)

  Devuelve el último espacio inspeccionado.
- Object * game_get_last_inspected_object (Game *game)

  Devuelve el ultimo objeto inspeccionado.
- Id game_get_player_location (Game *game)

  Devuelve la localización del jugador.
- Id game_get_object_location (Game *game, Object *object)

  Obtiene la localización de un objeto.
- void game_print_data (Game *game)

  Imprime la información del juego.
- Dialogue * game_get_dialogue (Game *game)

  Devuelve un puntero al modulo dialogo del juego.

4.4.1. Descripción detallada

It defines the game interface for each command.

Autor

Profesores PPROG
Javier Bernardo
Mihai Blidaru

Versión

2.0

Fecha

20-02-2017

Copyright

GNU Public License

4.4.2. Documentación de los 'defines'

4.4.2.1. #define MAX_OBJECTS 20

Numero maximo de objetos

4.4.2.2. #define MAX_SPACES 10000

Número máximo de casillas permitidas en el juegos

4.4.3. Documentación de los 'typedefs'

4.4.3.1. typedef struct _Game Game

Estructura game Guarda los datos del juego: Jugador, Objetos, Casillas, Dado, ultimo comando ejecutado el resultado de este comando.

4.4.4. Documentación de las funciones

4.4.4.1. STATUS game_add_link (Game * *game,* Link * *link*)

Añade un objeto al juego.

Autor

Mihai Blidaru

Parámetros

game	El puntero al juego
link	Un puntero al objeto

Devuelve

OK si se ha añaido correctamente, sino ERROR

4.4.4.2. STATUS game_add_object (Game * *game,* Object * *object*)

Añade un objeto al juego.

Autor

Javier Bernardo

Parámetros

game	El puntero al juego
object	Un puntero al objeto

Devuelve

OK si se ha añadido correctamente, sino ERROR

4.4.4.3. STATUS game_add_player (Game * *game,* Player * *player*)

Añade un jugador al juego. Si el jugador ya está inicializado, lo sustituye por otro.

Autor

Mihai Blidaru

Parámetros

game	El puntero al juego
player	Un puntero al jugador

Devuelve

OK si se ha añaido correctamente, sino ERROR

4.4.4.4. STATUS game_add_space (Game * *game,* Space * *space*)

Añade un espacio al juego.

Autor

Profesores PPROG

Parámetros

game	El puntero al juego
space	Un puntero al espacio

Devuelve

OK si se ha añadido correctamente, sino ERROR

4.4.4.5. Game* game_create ( )

Inicializa los datos del juego Esta función inicializa los espacios, crea e inicializa el jugador, los objetos y el dado.

Autor

Profesores PPROG
Mihai Blidaru

Devuelve

OK si todo ha ido bien. En caso contrario ERROR

4.4.4.6. STATUS game_destroy (Game * *game*)

Destruye la estructura game. Se encarga de liberar la memoria reservada por los elementos del juego spaces, player, objects y die.

Autor

Profesores PPROG
Mihai Blidaru

Parámetros

game	Puntero a la estructura del juego.

Devuelve

OK si se ha podido hacer todo correctamente. ERROR en caso contrario.

4.4.4.7. Dialogue* game_get_dialogue (Game * *game*)

Devuelve un puntero al modulo dialogo del juego.

Autor

Mihai Blidaru

Parámetros

game	Puntero a la estructura del juego.

Devuelve

Un puntero al modulo dialogo del juego o NULL si hay algun error

4.4.4.8. Die* game_get_die (Game * *game*)

Devuelve el dado del juego.

Autor

Mihai Blidaru

Parámetros

game	Puntero a la estructura del juego

Devuelve

Dado del juego.

4.4.4.9. Object* game_get_last_inspected_object (Game * *game*)

Devuelve el ultimo objeto inspeccionado.

Autor

Laura Bernal y Sandra Benitez

Parámetros

game	Un puntero a la estructura game

Devuelve

El ultimo objeto inspeccionado

4.4.4.10. Space* game_get_last_inspected_space (Game * *game*)

Devuelve el último espacio inspeccionado.

Autor

Laura Bernal y Sandra Benitez

Parámetros

game	Un puntero a la estructura game

Devuelve

El ultimo espacio inspeccionado

4.4.4.11. Link* game_get_link (Game * *game,* Id *link_id*)

Devuelve un puntero al link que tiene el Id igual al Id pasado como parametro.

Autor

Profesores PPROG

Parámetros

game	Puntero a la estructura del juego
link_id	Id del link

Devuelve

Puntero al link que tiene el Id igual al Id pasado como parametro. NULL si no existe.

4.4.4.12. Link* game_get_link_at (Game * *game*, int *index*)

Develve el link en una posición dada.

Autor

Mihai Blidaru

Parámetros

game	Puntero a la estructura del juego
index	La posición de la cual se quiere encontrar el link

Devuelve

El espacio en la posición dada.Null si la posición es invalida.

4.4.4.13. char* game_get_obj_list_as_str (Game * *game*, Space * *space*)

Devuelve la lista de objetos de una casilla como cadena.

Autor

Mihai Blidaru

Parámetros

game	Un puntero a la estructura game
space	Un puntero a la casilla desde la cual leer los objetos

Devuelve

Una cadena con la lista de los objetos. Quien use esta funciónse tiene que encargar de liberar la memoria usada.

4.4.4.14. Object* game_get_object_at (Game * *game*, int *index*)

Develve el objeto una posición dada.

Autor

Mihai Blidaru

Parámetros

game	Puntero a la estructura del juego
index	La posición de la cual se quiere encontrar el objeto

Devuelve

El objeto en la posición dada. Null si la posición es invalida.

4.4.4.15. Id game_get_object_location (Game * *game,* Object * *object*)

Obtiene la localización de un objeto.

Autor

Javier Bernardo

Parámetros

game	Puntero a la estructura de el juego.
object	El objeto del que quieres obtener la posicion.

Devuelve

La posicion del objeto o NO_ID en caso de error

4.4.4.16. Player* game_get_player (Game * *game*)

Devuelve el jugador del juego.

Autor

Mihai Blidaru

Parámetros

game	Puntero a la estructura del juego

Devuelve

Jugador del juego.

4.4.4.17. Id game_get_player_location (Game * *game*)

Devuelve la localización del jugador.

Autor

Mihai Blidaru

Parámetros

game	Puntero a la estructura de l juego.

Devuelve

La localización del jugador. NO_ID si hay algun error.

4.4.4.18. Space* game_get_space (Game * *game,* Id *space_id*)

Devuelve un puntero al space que tiene el Id igual al Id pasado como parametro.

Autor

Profesores PPROG

Parámetros

game	Puntero a la estructura del juego
space_id	Id del espacio

Devuelve

Puntero al espacio que tiene el Id igual al Id pasado como parametro. NULL si no existe.

4.4.4.19. Space* game_get_space_at (Game * *game,* int *index*)

Develve el espacop una posición dada.

Autor

Mihai Blidaru

Parámetros

game	Puntero a la estructura del juego
index	La posición de la cual se quiere encontrar el espacio

Devuelve

El espacio en la posición dada.Null si la posición es invalida.

4.4.4.20. BOOL game_is_over (Game * *game*)

Devuelve si el juego a acabado o no.

En esta iteración la función devuelve siempre FALSE ya que no hay suficiente funcionalidad en el juego como para poder decidir si el juego ha acabado o no.

Autor

Profesores PPROG

Parámetros

game	Puntero a la estructura del juego.

Devuelve

FALSE

4.4.4.21. void game_print_data (Game * *game*)

Imprime la información del juego.

Autor

Profesores PPROG
Javier Bernardo

Parámetros

game	Puntero a la estructura del juego.

Devuelve

Nada

4.4.4.22. STATUS game_update (Game * *game*, Command * *command*)

Ejecuta una de las funciones callback en funcion del comando recibido.

Autor

Profesores PPROG

Parámetros

game	Puntero a la estructura del juego
command	Comando que se tiene que ejecutar

Devuelve

4.5. Referencia del Archivo include/game_management.h

Define los prototipos de las funciones necesarias para cargar los datos del juego desde un archivo asi como guardar una partida.

Enumeraciones

- enum SAVE_STATUS { PROTECTED_FILE, SAVE_OK, WRITE_FAILED, BAD_ARGUMENTS }

  Define los estados que puede devolver la función Save.

Funciones

- SAVE_STATUS game_management_save (Game *game, char *filename)

  Guarda los datos de la partida con el mismo formato que los ficheros de carga.
- STATUS game_management_load (Game *game, char *filename)

  Carga los datos del juego desde un archivo.
- STATUS game_management_start_from_file (Game *game, char *spacesFile, char *objectsFile, char *linksfile, char *playersFile)

  Carga los datos del juego desde un archivo.

4.5.1. Descripción detallada

Define los prototipos de las funciones necesarias para cargar los datos del juego desde un archivo asi como guardar una partida.

Autor

Javier Bernardo
Mihai Blidaru

Versión

1.0

Fecha

23-01-2017

Copyright

GNU Public License

4.5.2. Documentación de las enumeraciones

4.5.2.1. enum SAVE_STATUS

Define los estados que puede devolver la función Save.

Valores de enumeraciones

PROTECTED_FILE EL archivo donde se quiere guardar es un archivo protegido

SAVE_OK Se ha gaurdado todo sin problemas

WRITE_FAILED No se puede escribir en ese archivo

BAD_ARGUMENTS Los argumentos de la función no son validos

4.5.3. Documentación de las funciones

4.5.3.1. STATUS game_management_load (Game * *game*, char * *filename*)

Carga los datos del juego desde un archivo.

Parámetros

game	Un puntero al juego donde cargar los datos
filename	nombre del archivo desde donde cargar los datos

Devuelve

OK si se ha cargado correctamente o ERROR en caso contrario

4.5.3.2. SAVE_STATUS game_management_save (Game * *game*, char * *filename*)

Guarda los datos de la partida con el mismo formato que los ficheros de carga.

Parámetros

game	Un puntero al game desde donde se quieren guardar los datos
filename	Archivo donde guardar los datos

Devuelve

OK si ha guardado correctamente o ERROR en caso contrario

4.5.3.3. STATUS game_management_start_from_file (Game * *game*, char * *spacesFile*, char * *objectsFile*, char * *linksfile*, char * *playersFile*)

Carga los datos del juego desde un archivo.

Autor

Javier Bernardo

Parámetros

game	Puntero a la estructura del juego.
spacesFile	Nombre del archivo desde donde hay que cargar los espacios
objectsFile	Nombre del archivo desde donde hay que cargar los objetos
linksfile	Nombre del archivo desde donde hay que cargar los links
playersFile	Nombre del archivo desde donde hay que cargar los datos del jugador

Devuelve

OK si todo ha ido bien. ERROR en caso contrario.

4.6. Referencia del Archivo include/game_rules.h

Define los prototipos de las funciones necesarias para el funcionamiento de las funciones de game_rules.c.

Funciones

- STATUS game_rules_run_random_rule (Game *game)

  Ejecuta una regla aleatoria que está en la lista de reglas.

4.6.1. Descripción detallada

Define los prototipos de las funciones necesarias para el funcionamiento de las funciones de game_rules.c.

Autor

Laura Bernal
Sandra Benítez

Versión

1.0

Fecha

28-04-2017

Copyright

GNU Public License

4.6.2. Documentación de las funciones

4.6.2.1. STATUS game_rules_run_random_rule (Game * *game*)

Ejecuta una regla aleatoria que está en la lista de reglas.

Autor

Laura Bernal

Parámetros

game	Un puntero a la estructura del juego

Devuelve

OK si todo ha ido bien o ERROR en caso contrario

4.7. Referencia del Archivo include/graphic_engine.h

It defines a textual graphic engine.

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

'typedefs'

- typedef struct _Graphic_engine Graphic_engine

  Definicion de la estructura de graphic engine.

Funciones

- Graphic_engine * graphic_engine_create ()

  Crea el motor del juego e inicializa las areas donde se va a imprimir por pantalla.
- void graphic_engine_destroy (Graphic_engine *ge)

  Libera la memoria ocupada por un graphic_engine.
- void graphic_engine_paint_game (Graphic_engine *ge, Game *game)

  Imprime el juego por pantalla.
- void graphic_engine_paint_directions (FILE *fp, Game *game)

  Imprime las direcciones adyacentes de una casilla.
- void graphic_engine_paint_help (FILE *fp)

  Imprime la pantalla de ayuda.
- void graphic_engine_play_intro (FILE *fp)

  Imprime la intro del juego.
- void graphic_engine_game_over (FILE *fp)

  Imprime la panatlla de final de juego.

4.7.1. Descripción detallada

It defines a textual graphic engine.

Autor

Profesores PPROG
Javier Bernardo
Mihai Blidaru

Versión

2.0

Fecha

13-03-2017

Copyright

GNU Public License

Autor

Profesores PPROG

Versión

1.0

Fecha

18-01-2017

Copyright

GNU Public License

4.7.2. Documentación de las funciones

4.7.2.1. Graphic_engine* graphic_engine_create ( )

Crea el motor del juego e inicializa las areas donde se va a imprimir por pantalla.

Autor

Profesores PPROG

Devuelve

Un puntero al graphic_engine creado

Cabeceras Libc Cabeceras propias

4.7.2.2. void graphic_engine_destroy (Graphic_engine * *ge*)

Libera la memoria ocupada por un graphic_engine.

Autor

Profesores PPROG

Parámetros

ge	El graphic_engine que se tiene que destruir

Devuelve

Nada

4.7.2.3. void graphic_engine_game_over (FILE * *fp*)

Imprime la panatlla de final de juego.

Autor

Mihai Blidaru

Parámetros

fp	Descriptor del fichero donde imprimir

4.7.2.4. void graphic_engine_paint_directions (FILE * *fp,* Game * *game*)

Imprime las direcciones adyacentes de una casilla.

Autor

Mihai Blidaru

Parámetros

fp	Descriptor del fichero donde imprimir
game	Un puntero a la estructura del juego

4.7.2.5. void graphic_engine_paint_game (Graphic_engine * *ge*, Game * *game*)

Imprime el juego por pantalla.

Autor

Profesores PPROG
Javier Bernardo

Parámetros

ge	Puntero a un graphic_engine
game	Puntero a un juego

4.7.2.6. void graphic_engine_paint_help (FILE * *fp*)

Imprime la pantalla de ayuda.

Autor

Javier Bernardo

Parámetros

fp	Descriptor del fichero donde imprimir

4.7.2.7. void graphic_engine_play_intro (FILE * *fp*)

Imprime la intro del juego.

Autor

Mihai Blidaru

Parámetros

fp	Descriptor del fichero donde imprimir

4.8. Referencia del Archivo include/inventory.h

Define la interfaz pública del TAD Inventory.

```
#include "set.h"
#include "types.h"
```

'typedefs'

- typedef struct _Inventory Inventory

  Definicion de la estructura del inventario.

Funciones

- Inventory * inventory_create ()
  Crea el inventario e inicializa sus campos.
- STATUS inventory_destroy (Inventory *inventory)
  Destruye un inventario.
- STATUS inventory_add_object (Inventory *inventory, Id id)
  Añade un objeto al inventario.
- STATUS inventory_set_max (Inventory *inventory, int NumIds)
  Modifica el numero de objetos del inventario.
- int inventory_get_max (Inventory *inventory)
  Obtiene el número de objetos de un inventario.
- STATUS inventory_remove_object (Inventory *inventory, Id id)
  Elimina un objeto del inventario.
- Set * inventory_get_set (Inventory *inventory)
  Devuelve el conjunto de identificadores.
- STATUS inventory_print (FILE *fp, Inventory *inventory)
  Imprime los datos de un inventario.

4.8.1. Descripción detallada

Define la interfaz pública del TAD Inventory.

Autor

Sandra Benítez y Laura Bernal

Versión

1.0

Fecha

14-03-2017

Copyright

GNU Public License

4.8.2. Documentación de las funciones

4.8.2.1. STATUS inventory_add_object (Inventory * *inventory*, Id *id*)

Añade un objeto al inventario.

Parámetros

inventory	Inventario al que se desea añadir objeto
id	Id del objeto a añadir

Devuelve

Un array de Ids de los objetos que tiene el jugador

4.8.2.2. Inventory* inventory_create ( )

Crea el inventario e inicializa sus campos.

Autor

Laura Bernal y Sandra Benitez

Devuelve

Puntero a inventory

4.8.2.3. STATUS inventory_destroy (Inventory * *inventory*)

Destruye un inventario.

Parámetros

inventory	Inventario que se desea eliminar

Devuelve

OK si se ha liberado correctamente, ERROR en caso contrario

4.8.2.4. int inventory_get_max (Inventory * *inventory*)

Obtiene el número de objetos de un inventario.

Parámetros

inventory	Inventario del que se obtiene el numero maximo de objetos

Devuelve

Número de objetos de un inventario

4.8.2.5. Set* inventory_get_set (Inventory * *inventory*)

Devuelve el conjunto de identificadores.

Parámetros

inventory	Inventario del que se desea obtener el conjunto de identificadores

Devuelve

Conjunto de identificadores

4.8.2.6. STATUS inventory_print (FILE * *fp,* Inventory * *inventory*)

Imprime los datos de un inventario.

Parámetros

inventory	
fp	Archivo

Devuelve

OK si se ha realizado correctamente, ERROR en caso contrario

4.8.2.7. STATUS inventory_remove_object (Inventory * *inventory*, Id *id*)

Elimina un objeto del inventario.

Parámetros

inventory	Inventario de que eliminar el objeto
id	Id del objeto a eliminar

Devuelve

OK si se ha realizado correctamente, ERROR en caso contrario

4.8.2.8. STATUS inventory_set_max (Inventory * *inventory*, int *NumIds*)

Modifica el numero de objetos del inventario.

Parámetros

inventory	Inventario en el que se quiere poner el numero maximo de ids
NumIds	El numero maximo de ids a colocar

Devuelve

OK si se ha realizado correctamente, ERROR en caso contrario

4.9. Referencia del Archivo include/link.h

Implementa el nuevo TAD Link (Enlace). Sirve para enlazar de forma más potente las casillas.

```
#include "types.h"
```

'defines'

- #define MAX_LINK 10000

'typedefs'

- typedef struct _Link Link

  Estructura de link.

Enumeraciones

- enum State { OPENED, CLOSED }

  Define los posibles estados de un enlace.

Funciones

- Link * link_create ()
  Crea un nuevo link.
- STATUS link_destroy (Link *link)
  Destruye el link.
- STATUS link_set_id (Link *link, Id id)
  Coloca la id del link.
- Id link_get_id (Link *link)
  Obtiene la id del link.
- STATUS link_set_name (Link *link, char *name)
  Coloca el nombre del link.
- char * link_get_name (Link *link)
  Obtiene el nombre del link.
- STATUS link_set_space1 (Link *link, Id id)
  Coloca la id del spacio 1 del link.
- Id link_get_space1 (Link *link)
  Obtiene la id del spacio 1 del link.
- STATUS link_set_space2 (Link *link, Id id)
  Coloca la id del spacio 2 del link.
- Id link_get_space2 (Link *link)
  Obtiene la id del spacio 2 del link.
- STATUS link_set_state (Link *link, int state)
  Coloca el estado del link.
- State link_get_state (Link *link)
  Obtiene el estado del link.
- Id link_get_dest_from (Link *link, Id from)
  Obtiene el id de destino de un link desde una casilla dada.
- int link_print (Link *link)
  Imprime el link.

4.9.1. Descripción detallada

Implementa el nuevo TAD Link (Enlace). Sirve para enlazar de forma más potente las casillas.

Versión

1.0

4.9.2. Documentación de los 'defines'

4.9.2.1. #define MAX_LINK 10000

Numero maximo de enlaces permitidos

4.9.3. Documentación de las enumeraciones

4.9.3.1. enum State

Define los posibles estados de un enlace.

Valores de enumeraciones

OPENED Enlace abierto
CLOSED Enlace cerrado

4.9.4. Documentación de las funciones

4.9.4.1. Link* link_create ( )

Crea un nuevo link.

Autor

Javier Bernardo

Devuelve

El nuevo link creado

4.9.4.2. STATUS link_destroy (Link * *link*)

Destruye el link.

Autor

Javier Bernardo

Parámetros

link	Puntero al link.

Devuelve

OK si se destruido, ERROR en caso contrario

4.9.4.3. Id link_get_dest_from (Link * *link,* Id *from*)

Obtiene el id de destino de un link desde una casilla dada.

Parámetros

link	Puntero a link.
from	Id de la casilla de origen.

Devuelve

El id de destino si se ha podido encontrar o NO_ID en caso contrario.

4.9.4.4. Id link_get_id (Link * *link*)

Obtiene la id del link.

Autor

Javier Bernardo

Parámetros

link	Puntero al link.

Devuelve

La id del link

4.9.4.5. char* link_get_name (Link * *link*)

Obtiene el nombre del link.

Autor

Javier Bernardo

Parámetros

link	Puntero al link.

Devuelve

El nombre del link

4.9.4.6. Id link_get_space1 (Link * *link*)

Obtiene la id del spacio 1 del link.

Autor

Javier Bernardo

Parámetros

link	Puntero al link.

Devuelve

El id del spacio 1

4.9.4.7. Id link_get_space2 (Link * *link*)

Obtiene la id del spacio 2 del link.

Autor

Javier Bernardo

Parámetros

link	Puntero al link.

Devuelve

Id del spacio 2 del link

4.9.4.8. State link_get_state (Link * link)

Obtiene el estado del link.

Autor

Javier Bernardo

Parámetros

link	Puntero al link.

Devuelve

El estado del link

4.9.4.9. int link_print (Link * link)

Imprime el link.

Autor

Javier Bernardo

Parámetros

link	Puntero al link.

Devuelve

Impresion del link

4.9.4.10. STATUS link_set_id (Link * link, Id id)

Coloca la id del link.

Autor

Javier Bernardo

Parámetros

link	Puntero al link.
id	La id que se desea colocar

Devuelve

OK si se ha colocado bien la id, ERROR en caso contrario

4.9.4.11. STATUS link_set_name (Link * link, char * name)

Coloca el nombre del link.

Autor

Javier Bernardo

Parámetros

link	Puntero al link.
name	El nombre que se desea colocar

Devuelve

OK si se ha colocado bien el nombre, ERROR en caso contrario

4.9.4.12. STATUS link_set_space1 (Link * *link*, Id *id*)

Coloca la id del spacio 1 del link.

Autor

Javier Bernardo

Parámetros

link	Puntero al link.
id	La id que se desea colocar

Devuelve

OK si se ha colocado bien la id, ERROR en caso contrario

4.9.4.13. STATUS link_set_space2 (Link * *link*, Id *id*)

Coloca la id del spacio 2 del link.

Autor

Javier Bernardo

Parámetros

link	Puntero al link.
id	La id que se desea colocar

Devuelve

OK si se ha colocado bien la id, ERROR en caso contrario

4.9.4.14. STATUS link_set_state (Link * *link*, int *state*)

Coloca el estado del link.

Autor

Javier Bernardo

Parámetros

link	Puntero al link.
state	El estado al que se quiere poner el link

Devuelve

OK si se ha colocado bien el estado, ERROR en caso contrario

4.10. Referencia del Archivo include/object.h

Define la interfaz pública del TAD Objeto.

```
#include "types.h"
```

'typedefs'

- typedef struct _Object Object

  Definicion de la estructura objecto.

Funciones

- Object * object_create ()

  Reservamos memoria para un nuevo objeto.
- STATUS object_destroy (Object *object)

  Destruye un objeto.
- Object * object_Set_Id (Object *object, Id object_id)

  Pone el id del objeto.
- Object * object_Set_Name (Object *object, char *name)

  Pone el nombre del objeto.
- Object * object_Set_Description (Object *object, char *description)

  Pone la descripción del objeto.
- Object * object_Set_Description2 (Object *object, char *desc2)

  Pone la descripción dos del objeto.
- STATUS object_Set_Mobile (Object *object, const BOOL mobile)

  Pone la disponibilidad de movimiento del objeto.
- STATUS object_Set_Moved (Object *object, const BOOL moved)

  Pone si se ha mpvido el objeto.
- STATUS object_Set_Hidden (Object *object, BOOL hidden)

  Pone la situacion del objeto, oculto o no oculto.
- STATUS object_Set_Open (Object *object, Id open)

  Pone la situacion del link del objeto.
- STATUS object_Set_Illuminates (Object *object, BOOL illuminates)

  Pone la iluminacion del objeto.
- STATUS object_Set_Light (Object *object, BOOL on)

  Pone el encendido del objeto.
- Id object_Get_Id (Object *object)

  Obtiene el id del objeto.
- char * object_Get_Name (Object *object)

  Obtiene el nombre del objeto.

- char * object_Get_Description (Object *object)
  Obtiene la descripción del objeto.
- char * object_Get_Description2 (Object *object)
  Obtiene la descripción dos del objeto.
- BOOL object_Get_Mobile (const Object *object)
  Obtiene la movibilidad del objeto.
- BOOL object_Get_Moved (const Object *object)
  Obtiene si se ha movido el objeto.
- BOOL object_Get_Hidden (const Object *object)
  Obtiene si se ha movido el objeto.
- Id object_Get_Open (const Object *object)
  Obtiene si esta abierto el objeto.
- BOOL object_Get_Illuminates (const Object *object)
  Obtiene si se ha iluminado el objeto.
- BOOL object_Get_Light (const Object *object)
  Obtiene si se ha encendido el objeto.
- STATUS object_print (Object *object)
  Imprime por pantalla el nombre del objeto y el id.
- Object * object_Set_Graphics (Object *object, char *graphics)
  Le asigna al objeto descripción grafica del objeto.
- char * object_Get_Graphics (Object *object)
  Obtiene la descripción grafica del objeto.

4.10.1. Descripción detallada

Define la interfaz pública del TAD Objeto.

Autor

Javier Bernardo

Versión

1.0

Fecha

31-01-2017

Copyright

GNU Public License

4.10.2. Documentación de las funciones

4.10.2.1. Object* object_create ( )

Reservamos memoria para un nuevo objeto.

Devuelve

Devuelve el objeto creado

Reservamos memoria para un nuevo objeto.

Cabeceras Libc Cabeceras propias

4.10.2.2. STATUS object_destroy (Object * *object*)

Destruye un objeto.

Fecha

12-12-2005

Parámetros

object	El puntero del objeto.

Devuelve

Error si se ha hecho mal o Ok si se ha liberado correctamente.

4.10.2.3. char* object_Get_Description (Object * *object*)

Obtiene la descripción del objeto.

Parámetros

object	El puntero del objeto.

Devuelve

La descripción del objeto

4.10.2.4. char* object_Get_Description2 (Object * *object*)

Obtiene la descripción dos del objeto.

Parámetros

object	El puntero del objeto.

Devuelve

La descripción del objeto

4.10.2.5. char* object_Get_Graphics (Object * *object*)

Obtiene la descripción grafica del objeto.

Parámetros

object	El puntero del objeto.

Devuelve

La descripción grafica del objeto

4.10.2.6. BOOL object_Get_Hidden (const Object * *object*)

Obtiene si se ha movido el objeto.

Parámetros

object	El puntero del objeto.

Devuelve

Si se ha movido o no el objeto

4.10.2.7. Id object_Get_Id (Object * *object*)

Obtiene el id del objeto.

Parámetros

object	El puntero del objeto.

Devuelve

El id del objeto

4.10.2.8. BOOL object_Get_Illuminates (const Object * *object*)

Obtiene si se ha iluminado el objeto.

Parámetros

object	El puntero del objeto.

Devuelve

Si se ha iluminado o no el objeto

4.10.2.9. BOOL object_Get_Light (const Object * *object*)

Obtiene si se ha encendido el objeto.

Parámetros

object	El puntero del objeto.

Devuelve

Si se ha encendido o no el objeto

4.10.2.10. BOOL object_Get_Mobile (const Object * *object*)

Obtiene la movibilidad del objeto.

Parámetros

object	El puntero del objeto.

Devuelve

La movibilidad del objeto

4.10.2.11. BOOL object_Get_Moved (const Object * *object*)

Obtiene si se ha movido el objeto.

Parámetros

object	El puntero del objeto.

Devuelve

Si se ha movido o no el objeto

4.10.2.12. char* object_Get_Name (Object * *object*)

Obtiene el nombre del objeto.

Parámetros

object	El puntero del objeto.

Devuelve

El nombre del objeto

4.10.2.13. Id object_Get_Open (const Object * *object*)

Obtiene si esta abierto el objeto.

Parámetros

object	El puntero del objeto.

Devuelve

Si se ha abierto o no el link del objeto

4.10.2.14. STATUS object_print (Object * *object*)

Imprime por pantalla el nombre del objeto y el id.

Parámetros

object	El puntero del objeto.

Devuelve

Imprime por pantalla si OK y da ERROR si lo contrario

4.10.2.15. Object* object_Set_Description (Object * *object,* char * *description*)

Pone la descripción del objeto.

Parámetros

object	El puntero del objeto.

description	Descripcion que recibe para el objeto

Devuelve

El objeto con la nueva descripción

4.10.2.16. Object* object_Set_Description2 (Object * *object,* char * *desc2*)

Pone la descripción dos del objeto.

Parámetros

object	El puntero del objeto.
desc2	Descripcion que recibe para el objeto

Devuelve

El objeto con la nueva descripción

4.10.2.17. Object* object_Set_Graphics (Object * *object,* char * *graphics*)

Le asigna al objeto descripción grafica del objeto.

Parámetros

object	El puntero del objeto.
graphics	La descripción grafica del objeto

Devuelve

El objeto actualizado o NULL si se produce algun error

4.10.2.18. STATUS object_Set_Hidden (Object * *object,* BOOL *hidden*)

Pone la situacion del objeto, oculto o no oculto.

Parámetros

object	El puntero del objeto.
hidden	Estado que recibe para el objeto

Devuelve

Ok si se hace, ERROR en caso contrario

4.10.2.19. Object* object_Set_Id (Object * *object,* Id *object_id*)

Pone el id del objeto.

Parámetros

object	El puntero del objeto.
object_id	Id que recibe para el objeto

Devuelve

El objeto con el nuevo id

4.10.2.20. STATUS object_Set_Illuminates (Object * *object*, BOOL *illuminates*)

Pone la iluminacion del objeto.

Parámetros

object	El puntero del objeto.
illuminates	Estado que recibe para el objeto

Devuelve

Ok si se hace, ERROR en caso contrario

4.10.2.21. STATUS object_Set_Light (Object * *object*, BOOL *on*)

Pone el encendido del objeto.

Parámetros

object	El puntero del objeto.
on	Estado que recibe para el objeto

Devuelve

Ok si se hace, ERROR en caso contrario

4.10.2.22. STATUS object_Set_Mobile (Object * *object*, const BOOL *mobile*)

Pone la disponibilidad de movimiento del objeto.

Parámetros

object	El puntero del objeto.
mobile	Estado que recibe para el objeto

Devuelve

Ok si se hace, ERROR en caso contrario

4.10.2.23. STATUS object_Set_Moved (Object * *object*, const BOOL *moved*)

Pone si se ha mpvido el objeto.

Parámetros

object	El puntero del objeto.
moved	Estado que recibe para el objeto

Devuelve

Ok si se hace, ERROR en caso contrario

4.10.2.24. Object* object_Set_Name (Object * *object,* char * *name*)

Pone el nombre del objeto.

Parámetros

object	El puntero del objeto.
name	Nombre que recibe para el objeto

Devuelve

El objeto con el nuevo nombre

4.10.2.25. STATUS object_Set_Open (Object * *object,* Id *open*)

Pone la situacion del link del objeto.

Parámetros

object	El puntero del objeto.
open	Estado que recibe para el objeto

Devuelve

Ok si se hace, ERROR en caso contrario

4.11. Referencia del Archivo include/player.h

Define el TAD player y los prototipos de las funciones necesarias para trabajar con este TAD.

```
#include "types.h"
#include "set.h"
#include "inventory.h"
```

'defines'

- #define PLAYER_INV_LOCATION-65535

  Localizacion ficticia que corresponde al inventario del jugador.

'typedefs'

- typedef struct _Player Player

  Definicion de la estructura player.

Funciones

- Player * player_create ()

  Crea un nuevo jugador e inicializa sus campos.
- STATUS player_destroy (Player *player)

  Destruye un jugador.
- Player * player_Set_Id (Player *player, Id player_id)

  Pone el id del jugador.
- Player * player_Set_Name (Player *player, char *name)

  Pone el nombre del jugador.
- Player * player_Set_Location (Player *player, Id location)

  Pone la localizacion del jugador.
- Player * player_Set_Max_Objects (Player *player, int max_objects)

  Asigna el número maximo de objetos que puede llevar el jugador.
- Id player_Get_Id (Player *player)

  Obtiene el id del jugador.
- char * player_Get_Name (Player *player)

  Obtiene el nombre del jugador.
- Id player_Get_Location (Player *player)

  Obtiene la localizacion del jugador.
- int player_Get_Max_Objects (Player *player)

  Devuelve el número maximo de objetos que puede llevar el jugador.
- STATUS player_Add_Object (Player *player, Id object_id)

  Añade un objeto al Inventory del jugador.
- STATUS player_Remove_Object (Player *player, Id object_id)

  Borra un objeto del Inventory del jugador.
- BOOL player_Has_Object (Player *player, Id object_id)

  Comprueba si el jugador tiene un objeto determinado.
- STATUS player_Print (Player *player)

  Imprime por pantalla los datos del jugador.

4.11.1. Descripción detallada

Define el TAD player y los prototipos de las funciones necesarias para trabajar con este TAD.

Autor

Mihai Blidaru Pareja 7

Versión

1.0

Fecha

31-01-2017

Copyright

GNU Public License

4.11.2. Documentación de las funciones

4.11.2.1. STATUS player_Add_Object (Player * player, Id object_id)

Añade un objeto al Inventory del jugador.

Autor

Mihai Blidaru

Parámetros

player	El jugador al que se quiere añadir un objeto
object_id	El id del objeto

Devuelve

OK si se hace bien, ERROR en caso contrario

4.11.2.2. Player* player_create ( )

Crea un nuevo jugador e inicializa sus campos.

Devuelve

El nuevo jugador

4.11.2.3. STATUS player_destroy (Player * player)

Destruye un jugador.

Parámetros

player	El puntero del jugador.

Devuelve

Error si se ha hecho mal o Ok si se ha liberado correctamente.

4.11.2.4. Id player_Get_Id (Player * player)

Obtiene el id del jugador.

Parámetros

player	El puntero del jugador.

Devuelve

El id del jugador

4.11.2.5. Id player_Get_Location (Player * player)

Obtiene la localizacion del jugador.

Parámetros

player	El puntero del jugador.

Devuelve

La localizacion del jugador

4.11.2.6. int player_Get_Max_Objects (Player * *player*)

Devuelve el número maximo de objetos que puede llevar el jugador.

Autor

Sandra Benitez

Parámetros

player	El jugador del que se quiere conocer el numero máximo de objetos

Devuelve

Devuelve el número máximo de objetos

4.11.2.7. char* player_Get_Name (Player * *player*)

Obtiene el nombre del jugador.

Parámetros

player	El puntero del jugador.

Devuelve

El nombre del jugador

4.11.2.8. BOOL player_Has_Object (Player * *player,* Id *object_id*)

Comprueba si el jugador tiene un objeto determinado.

Parámetros

player	El jugador que se quiere comprobar
object_id	El id del objeto

Devuelve

TRUE si el jugador tiene el objeto o FALSE en caso contrario o ERROR

4.11.2.9. STATUS player_Print (Player * *player*)

Imprime por pantalla los datos del jugador.

Parámetros

player	El puntero del jugador.

Devuelve

Imprime por pantalla si hay jugador y da ERROR si lo contrario

4.11.2.10. STATUS player_Remove_Object (Player * *player,* Id *object_id*)

Borra un objeto del Inventory del jugador.

Autor

Mihai Blidaru

Parámetros

player	El jugador al que se quiere quitar un objeto
object_id	El id del objeto

4.11.2.11. Player* player_Set_Id (Player * *player,* Id *player_id*)

Pone el id del jugador.

Parámetros

player	El puntero del jugador.
player_id	Id del jugador

Devuelve

El jugador

4.11.2.12. Player* player_Set_Location (Player * *player,* Id *location*)

Pone la localizacion del jugador.

Parámetros

player	El puntero del jugador.
location	La localizacion del jugador

Devuelve

El jugador

4.11.2.13. Player* player_Set_Max_Objects (Player * *player,* int *max_objects*)

Asigna el número maximo de objetos que puede llevar el jugador.

Autor

Mihai Blidaru

Parámetros

player	El jugador al que se le quiere cambiar el numero máximo de objetos
max_objects	Número máximo de objetos que puede llevar el jugador

Devuelve

Devuelve un puntero al jugador si todo ha ido bien o NULL en caso contrario.

4.11.2.14. Player* player_Set_Name (Player * *player*, char * *name*)

Pone el nombre del jugador.

Parámetros

player	El puntero del jugador.
name	Nombre del jugador

Devuelve

El jugador

4.12. Referencia del Archivo include/screen.h

Define la intefaz pública del TAD screen.

'defines'

- #define SCREEN_MAX_STR 80

'typedefs'

- typedef struct _Area Area

Funciones

- void screen_init ()
  Inicializa los datos para imprimir el juego en pantalla.
- void screen_destroy ()
  Destruye los datos usados para imprimir el juego en pantalla.
- void screen_paint ()
  Imprime el juego en pantalla.
- void screen_gets (char *str)
  Obtiene una pantalla.
- Area * screen_area_init (int x, int y, int width, int height)
  Crea un area.
- void screen_area_destroy (Area *area)
  Destruye un area.
- void screen_area_clear (Area *area)
  Despeja un area.
- void screen_area_reset_cursor (Area *area)
  Resetea el cursor en un area.
- void screen_area_puts (Area *area, char *str)
  Escribe texto en un area.

4.12.1. Descripción detallada

Define la intefaz pública del TAD screen.

Autor

Profesores PPROG

Versión

1.0

Fecha

11-01-2017

Copyright

GNU Public License

4.12.2. Documentación de los 'defines'

4.12.2.1. #define SCREEN_MAX_STR 80

Definicion de la variable SCREEN_MAX_STR

4.12.3. Documentación de los 'typedefs'

4.12.3.1. typedef struct _Area Area

Definicion de la estructura Area

4.12.4. Documentación de las funciones

4.12.4.1. void screen_area_clear (Area * *area*)

Despeja un area.

Parámetros

area	Area del cual tiene que quitar el texto

4.12.4.2. void screen_area_destroy (Area * *area*)

Destruye un area.

Parámetros

area	Area que tiene que destruir

4.12.4.3. Area* screen_area_init (int *x,* int *y,* int *width,* int *height*)

Crea un area.

Parámetros

x	Posición x del area
y	Posición y del area
width	Ancho del area
height	Alto del area

Devuelve

area creada

4.12.4.4. void screen_area_puts (Area * *area*, char * *str*)

Escribe texto en un area.

Parámetros

area	Area en el cual escribir
str	Cadena de texto que se quiere imprimir en el area

4.12.4.5. void screen_area_reset_cursor (Area * *area*)

Resetea el cursor en un area.

Parámetros

area	Area del cual se quiere resetear el cursor

4.12.4.6. void screen_gets (char * *str*)

Obtiene una pantalla.

Imprime con colores

Parámetros

str	Puntero a cadena para indicar que pantalla cojer

4.12.4.7. void screen_init ( )

Inicializa los datos para imprimir el juego en pantalla.

Cabeceras Libc

4.13. Referencia del Archivo include/set.h

Definicion de los prototipos de las primitivas del TAD Set.

```
#include <stdio.h>
#include "types.h"
```

'defines'

- #define MAX_SET 15

'typedefs'

- typedef struct _Set Set

  Definicion de la estructura set.

Funciones

- Set * set_create ()

  Crea un nuevo conjunto.
- STATUS set_destroy (Set *set)

  Libera memoria usada por un connjunto.
- STATUS set_addId (Set *set, Id id)

  Añade un nuevo id al conjunto.
- Set * set_delId (Set *set, Id id)

  Borra un id del Set.
- BOOL set_Id_is_in (Set *set, Id id)

  Comprueba si un id ya existe en el conjunto;.
- int set_getNumberOfIds (Set *set)

  Obtiene el numero de ids del set.
- int set_print (FILE *fp, Set *set)

  Imprime el set.

4.13.1. Descripción detallada

Definicion de los prototipos de las primitivas del TAD Set.

Autor

Mihai Blidaru

Versión

1.0

Fecha

20/03/2017

4.13.2. Documentación de los 'defines'

4.13.2.1. #define MAX_SET 15

Número máximo de ids permitidos en un set

4.13.3. Documentación de las funciones

4.13.3.1. STATUS set_addId (Set * *set,* Id *id*)

Añade un nuevo id al conjunto.

Autor

Mihai Blidaru

Parámetros

set	Conjunto donde añadir el nuevo id.
id	Id que se tiene que añadir al conjunto.

Devuelve

OK si se ha añadido correctamente el Id o ERROR en caso contrario.

4.13.3.2. Set* set_create ( )

Crea un nuevo conjunto.

Autor

Mihai Blidaru

Devuelve

El nuevo conjunto creadoo o NULL si no se ha podido cread

4.13.3.3. Set* set_delId (Set * *set,* Id *id*)

Borra un id del Set.

Autor

Mihai Blidaru

Parámetros

set	Conjunto donde borrar el id pasado como parametro
id	Id que se tiene que borrar del conjunto

Devuelve

Conjunto.

4.13.3.4. STATUS set_destroy (Set * *set*)

Libera memoria usada por un connjunto.

Autor

Mihai Blidaru

Parámetros

set	Conjunto que se tiene que destruir

Devuelve

OK si se ha realizado la operación correctamente o ERROR en caso contrario.

4.13.3.5. int set_getNumberOfIds (Set * set)

Obtiene el numero de ids del set.

Autor

Mihai Blidaru

Parámetros

set	Conjunto del que se quiere saber su numero de ids.

Devuelve

Numero de ids.

4.13.3.6. BOOL set_Id_is_in (Set * set, Id id)

Comprueba si un id ya existe en el conjunto;.

Autor

Mihai Blidaru

Parámetros

set	Conjunto en el que buscar el Id
id	Id que se tiene que comprobar si existe

Devuelve

TRUE si el Id se encuentra en el conjuto o FALSE en caso contrario.

4.13.3.7. int set_print (FILE * fp, Set * set)

Imprime el set.

Autor

Mihai Blidaru

Parámetros

fp	Archivo donde imprimir
set	Conjunto a imprimir

Devuelve

El número de caracteres imprimidos

4.14. Referencia del Archivo include/space.h

Define un espacio.

```
#include "types.h"
#include "set.h"
```

'defines'

- #define G_ROWS 6
- #define G_COLUMNS 24

'typedefs'

- typedef struct _Space Space

Funciones

- Space * space_create (Id space_id)

  Crea un nuevo espacio y le asigna un Id y unos valores por defecto.
- STATUS space_destroy (Space *space)

  Libera la memoria de un espacio.
- STATUS space_set_name (Space *space, char *name)

  Pone nombre a un espacio.
- STATUS space_set_description (Space *space, char *description)

  Pone nombre una descripcion al espacio.
- STATUS space_set_long_description (Space *space, char *description)

  Pone una descripcion detallada al espacio.
- Space * space_set_graphics (Space *space, char graphics[][G_COLUMNS])

  Pone la descripcion de la casilla.
- STATUS space_set_north (Space *space, Id link_id)

  Pone un norte en el espacio.
- STATUS space_set_south (Space *space, Id link_id)

  Pone un sur en el espacio.
- STATUS space_set_east (Space *space, Id link_id)

  Pone un este en el espacio.
- STATUS space_set_west (Space *space, Id link_id)

  Pone un oeste en el espacio.
- STATUS space_set_up (Space *space, Id link_id)

  Pone un up en el espacio.
- STATUS space_set_down (Space *space, Id link_id)

  Pone un down en el espacio.
- Id space_get_id (Space *space)

  Obtienes el id del espacio.
- const char * space_get_name (Space *space)

  Obtienes el nombre del espacio.
- const char * space_get_description (Space *space)

  Obtiene la descripcion de un espacio.
- const char * space_get_long_description (Space *space)

  Obtiene la descripción detallada de un espacio.
- Space * space_get_graphics (Space *space, char dest[][G_COLUMNS])

  Devuelve la descripción grafica de la casilla.
- Set * space_get_objects (Space *space)

  Obtienes la lista de los objetos del espacio.
- Id space_get_north (Space *space)

  Obtienes el norte del espacio.
- Id space_get_south (Space *space)

  Obtienes el sur del espacio.

- Id space_get_east (Space *space)

  Obtienes el este del espacio.
- Id space_get_west (Space *space)

  Obtienes el oeste del espacio.
- Id space_get_up (Space *space)

  Obtienes la dirección hacia arriba del espacio.
- Id space_get_down (Space *space)

  Obtienes la dirección hacia abajo del espacio.
- STATUS space_add_object (Space *space, Id object_id)

  Añades el objeto al espacio.
- STATUS space_remove_object (Space *space, Id object_id)

  Eliminas el objeto del espacio.
- BOOL space_contains_object (Space *space, Id object_id)

  Compruebas si el espacio tiene objeto.
- BOOL space_graphics_areEmpty (Space *space)

  Devuelve si el espacio tiene descripcion grafica o no.
- STATUS space_set_iluminated (Space *space, BOOL iluminated)

  Asigna la iluminación al espacio.
- BOOL space_get_iluminated (Space *space)

  Devuelve si el espacio está iluminado.
- int space_print_graphics (Space *space)

  Imprime el espacio.
- STATUS space_print (Space *space)

  Imprimes lo que hay en el espacio.

4.14.1. Descripción detallada

Define un espacio.

Autor

Profesores PPROG
Javier Bernardo

Versión

1.0

Fecha

13-01-2015

Copyright

GNU Public License

4.14.2. Documentación de los 'defines'

4.14.2.1. #define G_COLUMNS 24

Columnas que ocupa la descripción grafica

4.14.2.2. #define G_ROWS 6

Filas que ocupa la descripción grafica

4.14.3. Documentación de los 'typedefs'

4.14.3.1. typedef struct _Space Space

Definición del tipo de dato Space

4.14.4. Documentación de las funciones

4.14.4.1. STATUS space_add_object (Space * *space,* Id *object_id*)

Añades el objeto al espacio.

Parámetros

space	Un puntero al espacio y una id del objeto.
object_id	Id del objeto.

Devuelve

Ok si se añade el objeto o ERROR si no.

4.14.4.2. BOOL space_contains_object (Space * *space,* Id *object_id*)

Compruebas si el espacio tiene objeto.

Parámetros

space	Un puntero al espacio.
object_id	Id del objeto.

Devuelve

funcion que coloca objeto.

4.14.4.3. Space* space_create (Id *space_id*)

Crea un nuevo espacio y le asigna un Id y unos valores por defecto.

Parámetros

space_id	Un Id del nuevo espacio.

Devuelve

Un puntero al nuevo espacio. Devuelve NULL si no se ha podido crear.

4.14.4.4. STATUS space_destroy (Space * *space*)

Libera la memoria de un espacio.

Parámetros

space	Un puntero al espacio que se quiere destruir

Devuelve

OK si se ha liberado correctamente, si no ERROR

4.14.4.5. const char* space_get_description (Space * *space*)

Obtiene la descripcion de un espacio.

Parámetros

space	Un puntero al espacio al que se le quiere poner nombre.

Devuelve

La descripcion del espacio.

4.14.4.6. Id space_get_down (Space * *space*)

Obtienes la dirección hacia abajo del espacio.

Parámetros

space	Un puntero al espacio.

Devuelve

La dirección hacia abajo del espacio.

4.14.4.7. Id space_get_east (Space * *space*)

Obtienes el este del espacio.

Parámetros

space	Un puntero al espacio.

Devuelve

El este del espacio.

4.14.4.8. Space* space_get_graphics (Space * *space,* char *dest[][G_COLUMNS]*)

Devuelve la descripción grafica de la casilla.

Parámetros

in	*space*	Un puntero al espacio.

out	*dest*	Matriz donde guardar la descripción grafica.

Devuelve

Un puntero a la descripción grafica del espacio o NULL si hay algun error.

4.14.4.9. Id space_get_id (Space * *space*)

Obtienes el id del espacio.

Parámetros

space	Un puntero al espacio.

Devuelve

El id del espacio.

4.14.4.10. BOOL space_get_iluminated (Space * *space*)

Devuelve si el espacio está iluminado.

Parámetros

space	Un puntero al espacio.

Devuelve

Devuelve el espacio iluminado

4.14.4.11. const char* space_get_long_description (Space * *space*)

Obtiene la descripción detallada de un espacio.

Parámetros

space	Un puntero al espacio al que se le quiere poner nombre.

Devuelve

La descripción detallada del espacio.

4.14.4.12. const char* space_get_name (Space * *space*)

Obtienes el nombre del espacio.

Parámetros

space	Un puntero al espacio.

Devuelve

El nombre del espacio.

4.14.4.13. Id space_get_north (Space * *space*)

Obtienes el norte del espacio.

Parámetros

space	Un puntero al espacio.

Devuelve

El norte del espacio.

4.14.4.14. Set* space_get_objects (Space * *space*)

Obtienes la lista de los objetos del espacio.

Parámetros

space	Un puntero al espacio.

Devuelve

La lista de los objetos del espacio.

4.14.4.15. Id space_get_south (Space * *space*)

Obtienes el sur del espacio.

Parámetros

space	Un puntero al espacio.

Devuelve

El sur del espacio.

4.14.4.16. Id space_get_up (Space * *space*)

Obtienes la dirección hacia arriba del espacio.

Parámetros

space	Un puntero al espacio.

Devuelve

La dirección hacia arriba del espacio.

4.14.4.17. Id space_get_west (Space * *space*)

Obtienes el oeste del espacio.

Parámetros

space	Un puntero al espacio.

Devuelve

El oeste del espacio.

4.14.4.18. BOOL space_graphics_areEmpty (Space * *space*)

Devuelve si el espacio tiene descripcion grafica o no.

Parámetros

space	Un puntero al espacio.

Devuelve

TRUE si los graficos están vacios o FALSE en caso contrario.

4.14.4.19. STATUS space_print (Space * *space*)

Imprimes lo que hay en el espacio.

Parámetros

space	Un puntero al espacio.

Devuelve

El contenido del espacio.

4.14.4.20. int space_print_graphics (Space * *space*)

Imprime el espacio.

Parámetros

space	Un puntero al espacio.

Devuelve

Impresion del objeto.

4.14.4.21. STATUS space_remove_object (Space * *space,* Id *object_id*)

Eliminas el objeto del espacio.

Parámetros

space	Un puntero al espacio.
object_id	Identificador del objeto.

Devuelve

Ok si se elimina o ERROR si no.

4.14.4.22. STATUS space_set_description (Space * *space,* char * *description*)

Pone nombre una descripcion al espacio.

Parámetros

space	Un puntero al espacio al que se le quiere poner nombre.

description	La descripcion que se le quiere poner.

Devuelve

OK si se ha asignado correctamente el nombre, si no ERROR.

4.14.4.23. STATUS space_set_down (Space * *space,* Id *link_id*)

Pone un down en el espacio.

Parámetros

space	Un puntero al espacio.
link_id	Un Id que determina el espacio.

Devuelve

OK si se ha realizado correctamente. Sino devuelve ERROR.

4.14.4.24. STATUS space_set_east (Space * *space,* Id *link_id*)

Pone un este en el espacio.

Parámetros

space	Un puntero al espacio.
link_id	Un Id que determina el espacio.

Devuelve

OK si se ha realizado correctamente. Sino devuelve ERROR.

4.14.4.25. Space* space_set_graphics (Space * *space,* char *graphics[][G_COLUMNS]*)

Pone la descripcion de la casilla.

Parámetros

space	Un puntero al espacio.
graphics	Una matriz que contiene la descripción grafica.

Devuelve

El espacio.

4.14.4.26. STATUS space_set_iluminated (Space * *space,* BOOL *iluminated*)

Asigna la iluminación al espacio.

Parámetros

space	Un puntero al espacio
iluminated	variable de tipo BOOL

Devuelve

Devuelve OK si se realiza correctamente

4.14.4.27. STATUS space_set_long_description (Space * *space*, char * *description*)

Pone una descripcion detallada al espacio.

Parámetros

space	Un puntero al espacio al que se le quiere poner nombre.
description	La descripcion que se le quiere poner.

Devuelve

OK si se ha asignado correctamente el nombre, si no ERROR.

4.14.4.28. STATUS space_set_name (Space * *space*, char * *name*)

Pone nombre a un espacio.

Parámetros

space	Un puntero al espacio al que se le quiere poner nombre.
name	El nombre que se le quiere poner.

Devuelve

OK si se ha asignado correctamente el nombre, si no ERROR.

4.14.4.29. STATUS space_set_north (Space * *space*, Id *link_id*)

Pone un norte en el espacio.

Parámetros

space	Un puntero al espacio
link_id	Un Id que determina el espacio

Devuelve

OK si se ha realizado correctamente. Sino devuelve ERROR

4.14.4.30. STATUS space_set_south (Space * *space*, Id *link_id*)

Pone un sur en el espacio.

Parámetros

space	Un puntero al espacio
link_id	Un Id que determina el espacio

Devuelve

OK si se ha realizado correctamente. Sino devuelve ERROR

4.14.4.31. STATUS space_set_up (Space * *space*, Id *link_id*)

Pone un up en el espacio.

Parámetros

space	Un puntero al espacio.
link_id	Un Id que determina el espacio.

Devuelve

OK si se ha realizado correctamente. Sino devuelve ERROR.

4.14.4.32. STATUS space_set_west (Space * *space*, Id *link_id*)

Pone un oeste en el espacio.

Parámetros

space	Un puntero al espacio
link_id	Un Id que determina el espacio

Devuelve

OK si se ha realizado correctamente. Sino devuelve ERROR

4.15. Referencia del Archivo include/tests/command_test.h

Define las funciones para la prueba del modulo Command.

Funciones

- void test1_command_ini ()
- void test1_command_destroy ()
- void test2_command_destroy ()
- void test1_command_set_cmd ()
- void test2_command_set_cmd ()
- void test3_command_set_cmd ()
- void test1_command_set_cmd_arg ()
- void test2_command_set_cmd_arg ()
- void test3_command_set_cmd_arg ()
- void test4_command_set_cmd_arg ()
- void test1_command_get_cmd ()
- void test2_command_get_cmd ()
- void test3_command_get_cmd ()

- void test1_command_get_cmd_arg ()
- void test2_command_get_cmd_arg ()
- void test3_command_get_cmd_arg ()
- void test1_command_clear ()
- void test2_command_clear ()

4.15.1. Descripción detallada

Define las funciones para la prueba del modulo Command.

Autor

Mihai Blidaru

Versión

1.0

Fecha

14-03-2017

Copyright

GNU Public License

4.15.2. Documentación de las funciones

4.15.2.1. void test1_command_clear ( )

Prueba Prueba la función que limpia los campos de un commando

Precondición

El gestor de comandos ha sido inicializado previamente y se ha guardado un comando dentro

Postcondición

La salida esperada es NO_CMD y una cadena vacia

4.15.2.2. void test1_command_destroy ( )

Prueba Pruba la función que libera memoria usada por un gestor de comandos

Precondición

El gestor de comandos ha sido previamente inicializado

Postcondición

La salida esperada es OK

4.15.2.3. void test1_command_get_cmd ( )

Prueba Prueba la función que devuelve el tipo de comando guardado en un gestor de comandos

Precondición

El gestor de comandos ha sido inicializado previamente y se le ha asignado el comando GO

Postcondición

La salida esperada es GO, el comando previamente asignado

4.15.2.4. void test1_command_get_cmd_arg ( )

Prueba Prueba la función que devuelve un argumento de un comando guardado en un gestor de comandos

Precondición

El gestor de comandos ha sido inicializado previamente y se le ha asignado un argumento "test"

Postcondición

La salida esperada es "test", el argumento añadido previamente

4.15.2.5. void test1_command_ini ( )

Prueba Prueba la función de creación de un gestor de comandos

Precondición

Condiciones normales para la prueba

Postcondición

Un puntero no nulo al al gestor de comandos creado

Número maximo de tests

4.15.2.6. void test1_command_set_cmd ( )

Prueba Prueba la función que guarda el tipo de comando en un gestor de comandos

Precondición

El gestor de comandos ha sido inicializado previamente

Postcondición

La salida esperada es OK

4.15.2.7. void test1_command_set_cmd_arg ( )

Prueba Prueba la función que guarda un argumento de un comando en un gestor de comandos

Precondición

El gestor de comandos ha sido inicializado previamente

Postcondición

La salida esperada es OK

4.15.2.8. void test2_command_clear ( )

Prueba Prueba la función que limpia los campos de un commando

Precondición

El gestor de comandos no ha sido inicializado previamente

Postcondición

La salida esperada es ERROR

4.15.2.9. void test2_command_destroy ( )

Prueba Pruba la función que libera memoria usada por un gestor de comandos

Precondición

El gestor de comandos no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.15.2.10. void test2_command_get_cmd ( )

Prueba Prueba la función que devuelve el tipo de comando guardado en un gestor de comandos

Precondición

El gestor de comandos no ha sido inicializado previamente

Postcondición

La salida esperada es UNKNOWN, comando desconocido

4.15.2.11. void test2_command_get_cmd_arg ( )

Prueba Prueba la función que devuelve un argumento de un comando guardado en un gestor de comandos

Precondición

El gestor de comandos no ha sido inicializado previamente

Postcondición

La salida esperada es NULL

4.15.2.12. void test2_command_set_cmd ( )

Prueba Prueba la función que guarda el tipo de comando en un gestor de comandos

Precondición

El gestor de comandos ha sido inicializado previamente pero el comando no es valido

Postcondición

La salida esperada es ERROR

4.15.2.13. void test2_command_set_cmd_arg ( )

Prueba Prueba la función que guarda un argumento de un comando en un gestor de comandos

Precondición

La posición en la que se quiere guardar es invalida

Postcondición

La salida esperada es ERROR

4.15.2.14. void test3_command_get_cmd ( )

Prueba Prueba la función que devuelve el tipo de comando guardado en un gestor de comandos

Precondición

El gestor de comandos ha sido inicializado previamente pero no se le ha asignado un comando

Postcondición

La salida esperada es NO_CMD

4.15.2.15. void test3_command_get_cmd_arg ( )

Prueba Prueba la función que devuelve un argumento de un comando guardado en un gestor de comandos

Precondición

El gestor de comandos ha sido inicializado previamente pero no se le ha asignado un argumento

Postcondición

La salida esperada es una cadena vacia

4.15.2.16. void test3_command_set_cmd ( )

Prueba Prueba la función que guarda el tipo de comando en un gestor de comandos

Precondición

El gestor de comandos no ha sido inicializado previamente

Postcondición

La salida esperada es ERROR

4.15.2.17. void test3_command_set_cmd_arg ( )

Prueba Prueba la función que guarda un argumento de un comando en un gestor de comandos

Precondición

El gestor de comandos no ha sido inicializado previamente

Postcondición

La salida esperada es ERROR

4.15.2.18. void test4_command_set_cmd_arg ( )

Prueba Prueba la función que guarda un argumento de un comando en un gestor de comandos

Precondición

El argumento es un puntero a NULL

Postcondición

La salida esperada es ERROR

4.16. Referencia del Archivo include/tests/dialogue_test.h

Define las funciones para la prueba del modulo dialogue.

Funciones

- void test1_dialogue_ini ()
- void test1_dialogue_get_text ()
- void test2_dialogue_get_text ()
- void test1_dialogue_go ()
- void test2_dialogue_go ()
- void test1_dialogue_unknown ()
- void test2_dialogue_unknown ()
- void test1_dialogue_dir ()
- void test2_dialogue_dir ()
- void test1_dialogue_take ()
- void test2_dialogue_take ()
- void test1_dialogue_leave ()
- void test2_dialogue_leave ()
- void test1_dialogue_save ()
- void test2_dialogue_save ()
- void test1_dialogue_load ()
- void test2_dialogue_load ()
- void test1_dialogue_attack ()
- void test2_dialogue_attack ()
- void test1_dialogue_inspect ()
- void test2_dialogue_inspect ()
- void test1_dialogue_open ()
- void test2_dialogue_open ()
- void test1_dialogue_turn_on ()
- void test2_dialogue_turn_on ()
- void test1_dialogue_turn_off ()
- void test2_dialogue_turn_off ()
- void test1_dialogue_help ()
- void test2_dialogue_help ()
- void test1_dialogue_game_rule ()
- void test2_dialogue_game_rule ()

4.16.1. Descripción detallada

Define las funciones para la prueba del modulo dialogue.

Autor

Mihai Blidaru

Versión

1.0

Fecha

14-03-2017

Copyright

GNU Public License

4.16.2. Documentación de las funciones

4.16.2.1. void test1_dialogue_attack ( )

Prueba Prueba la funcion que construye el dialogo para el comando ATTACK

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

4.16.2.2. void test1_dialogue_dir ( )

Prueba Prueba la funcion que construye el dialogo para el comando DIR

Precondición

El dialogo ha sido inicializado

Postcondición

La salida esperada es OK

4.16.2.3. void test1_dialogue_game_rule ( )

Prueba Prueba la funcion que construye el dialogo para las reglas del juego

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

4.16.2.4. void test1_dialogue_get_text ( )

Prueba Prueba la funcion que devuelve el texto del dialogo

Precondición

El dialogo se ha inicializado antes

Postcondición

La salida es un texto no nulo

4.16.2.5. void test1_dialogue_go ( )

Prueba Prueba la funcion que construye el dialogo para el comando GO

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

4.16.2.6. void test1_dialogue_help ( )

Prueba Prueba la funcion que construye el dialogo para el comando HELP

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

4.16.2.7. void test1_dialogue_ini ( )

Prueba Prueba la función que crea un dialogo

Precondición

Se reserva memoria para el dialogo

Postcondición

La salida que se espera es el dialogo inicializado

Número maximo de tests

4.16.2.8. void test1_dialogue_inspect ( )

Prueba Prueba la funcion que construye el dialogo para el comando INSPECT

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

4.16.2.9. void test1_dialogue_leave ( )

Prueba Prueba la funcion que construye el dialogo para el comando LEAVE

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

4.16.2.10. void test1_dialogue_load ( )

Prueba Prueba la funcion que construye el dialogo para el comando LOAD

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

4.16.2.11. void test1_dialogue_open ( )

Prueba Prueba la funcion que construye el dialogo para el comando OPEN

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

4.16.2.12. void test1_dialogue_save ( )

Prueba Prueba la funcion que construye el dialogo para el comando SAVE

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

4.16.2.13. void test1_dialogue_take ( )

Prueba Prueba la funcion que construye el dialogo para el comando TAKE

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

4.16.2.14. void test1_dialogue_turn_off ( )

Prueba Prueba la funcion que construye el dialogo para el comando TURN_OFF

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

4.16.2.15. void test1_dialogue_turn_on ( )

Prueba Prueba la funcion que construye el dialogo para el comando TURN_ON

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

4.16.2.16. void test1_dialogue_unknown ( )

Prueba Prueba la funcion que construye el dialogo para comandos desconocidos

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

4.16.2.17. void test2_dialogue_attack ( )

Prueba Prueba la funcion que construye el dialogo para el comando ATTACK

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.16.2.18. void test2_dialogue_dir ( )

Prueba Prueba la funcion que construye el dialogo para el comando DIR

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.16.2.19. void test2_dialogue_game_rule ( )

Prueba Prueba la funcion que construye el dialogo para las reglas del juego

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.16.2.20. void test2_dialogue_get_text ( )

Prueba Prueba la funcion que devuelve el texto del dialogo

Precondición

El dialogo no se ha inicializado antes

Postcondición

La salida esperada es NULL

4.16.2.21. void test2_dialogue_go ( )

Prueba Prueba la funcion que construye el dialogo para el comando GO

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.16.2.22. void test2_dialogue_help ( )

Prueba Prueba la funcion que construye el dialogo para el comando HELP

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.16.2.23. void test2_dialogue_inspect ( )

Prueba Prueba la funcion que construye el dialogo para el comando INSPECT

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.16.2.24. void test2_dialogue_leave ( )

Prueba Prueba la funcion que construye el dialogo para el comando LEAVE

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.16.2.25. void test2_dialogue_load ( )

Prueba Prueba la funcion que construye el dialogo para el comando LOAD

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.16.2.26. void test2_dialogue_open ( )

Prueba Prueba la funcion que construye el dialogo para el comando OPEN

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.16.2.27. void test2_dialogue_save ( )

Prueba Prueba la funcion que construye el dialogo para el comando SAVE

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.16.2.28. void test2_dialogue_take ( )

Prueba Prueba la funcion que construye el dialogo para el comando TAKE

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.16.2.29. void test2_dialogue_turn_off ( )

Prueba Prueba la funcion que construye el dialogo para el comando TURN_OFF

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.16.2.30. void test2_dialogue_turn_on ( )

Prueba Prueba la funcion que construye el dialogo para el comando TURN_ON

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.16.2.31. void test2_dialogue_unknown ( )

Prueba Prueba la funcion que construye el dialogo para comandos desconocidos

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.17. Referencia del Archivo include/tests/game_management_test.h

Define las funciones para la prueba del modulo Game_Management.

Funciones

- void test1_game_management_start_from_file ()
- void test2_game_management_start_from_file ()
- void test1_game_management_save ()
- void test2_game_management_save ()
- void test1_game_management_load ()
- void test2_game_management_load ()

4.17.1. Descripción detallada

Define las funciones para la prueba del modulo Game_Management.

Autor

Sandra Benítez
Laura Bernal

Versión

1.0

Fecha

24-04-2017

Copyright

GNU Public License

4.17.2. Documentación de las funciones

4.17.2.1. void test1_game_management_load ( )

Prueba Prueba la función que carga los datos del juego desde un archivo

Precondición

El nombre del fichero desde donde cargar los datos

Postcondición

La salida esperada es OK

4.17.2.2. void test1_game_management_save ( )

Prueba Prueba que la función guarde los datos de la partida con el mismo formato que los ficheros de carga

Precondición

El juego se ha creado correctamente y el nombre del archivo es valido

Postcondición

La salida esperada es OK

4.17.2.3. void test1_game_management_start_from_file ( )

Prueba Prueba la función que carga los datos del juego desde un archivo

Precondición

Se lee de forma correcta los archivos

Postcondición

La salida esperada es OK

Número maximo de tests

4.17.2.4. void test2_game_management_load ( )

Prueba Prueba la función que carga los datos del juego desde un archivo

Precondición

El juego está sin inicializar

Postcondición

La salida esperada es ERROR

4.17.2.5. void test2_game_management_save ( )

Prueba Prueba que la función guarde los datos de la partida con el mismo formato que los ficheros de carga

Precondición

El juego no ha sido inicializado y el nombre del archivo es nulo

Postcondición

La salida esperada es ERROR

4.17.2.6. void test2_game_management_start_from_file ( )

Prueba Prueba la función que carga los datos del juego desde un archivo

Precondición

No se lee de forma correcta del archivo

Postcondición

La salida esperada es ERROR

4.18. Referencia del Archivo include/tests/game_rules_test.h

Define las funciones para la prueba del modulo game_rules.

Funciones

- void test1_game_rules_run_random_rule ()
- void test2_game_rules_run_random_rule ()

4.18.1. Descripción detallada

Define las funciones para la prueba del modulo game_rules.

Autor

Mihai Blidaru

Versión

1.0

Fecha

14-03-2017

Copyright

GNU Public License

4.18.2. Documentación de las funciones

4.18.2.1. void test1_game_rules_run_random_rule ( )

Prueba Prueba la funcion que ejecuta una regla aleatoria

Precondición

El juego se ha inicializado previamente

Postcondición

La salida esperada es OK

Número maximo de tests

4.18.2.2. void test2_game_rules_run_random_rule ( )

Prueba Prueba la funcion que ejecuta una regla aleatoria

Precondición

El juego no se ha inicializado previamente

Postcondición

La salida esperada es ERROR

4.19. Referencia del Archivo include/tests/game_test.h

Define las funciones para la prueba del modulo Game.

Funciones

- void test1_game_create ()
- void test1_game_destroy ()
- void test2_game_destroy ()
- void test1_game_update ()
- void test2_game_update ()
- void test3_game_update ()
- void test1_game_is_over ()
- void test2_game_is_over ()
- void test1_game_add_object ()
- void test2_game_add_object ()
- void test3_game_add_object ()
- void test1_game_add_space ()
- void test2_game_add_space ()
- void test3_game_add_space ()

- void test1_game_add_link ()
- void test2_game_add_link ()
- void test3_game_add_link ()
- void test1_game_add_player ()
- void test2_game_add_player ()
- void test3_game_add_player ()
- void test1_game_get_link ()
- void test2_game_get_link ()
- void test3_game_get_link ()
- void test1_game_get_link_at ()
- void test2_game_get_link_at ()
- void test3_game_get_link_at ()
- void test1_game_get_space ()
- void test2_game_get_space ()
- void test3_game_get_space ()
- void test1_game_get_space_at ()
- void test2_game_get_space_at ()
- void test3_game_get_space_at ()
- void test1_game_get_object_at ()
- void test2_game_get_object_at ()
- void test3_game_get_object_at ()
- void test1_game_get_die ()
- void test2_game_get_die ()
- void test1_game_get_player ()
- void test2_game_get_player ()
- void test1_game_get_obj_list_as_str ()
- void test2_game_get_obj_list_as_str ()
- void test1_game_get_last_inspected_space ()
- void test2_game_get_last_inspected_space ()
- void test1_game_get_last_inspected_object ()
- void test2_game_get_last_inspected_object ()
- void test3_game_get_last_inspected_object ()
- void test1_game_get_player_location ()
- void test2_game_get_player_location ()
- void test1_game_get_object_location ()
- void test2_game_get_object_location ()
- void test3_game_get_object_location ()
- void test1_game_get_dialogue ()
- void test2_game_get_dialogue ()

4.19.1. Descripción detallada

Define las funciones para la prueba del modulo Game.

Autor

Mihai Blidaru

Versión

1.0

Fecha

14-03-2017

Copyright

GNU Public License

4.19.2. Documentación de las funciones

4.19.2.1. void test1_game_add_link ( )

Prueba Prueba la función que añade un link al juego

Precondición

El juego y el link han sido inicializados correctamente

Postcondición

La salida esperada es OK

4.19.2.2. void test1_game_add_object ( )

Prueba Prueba la función que añade un objeto al juego

Precondición

El juego y el objeto han sido inicializados correctamente

Postcondición

La salida esperada es OK

4.19.2.3. void test1_game_add_player ( )

Prueba Prueba la función que añade un jugador al juego

Precondición

El juego y el jugador han sido inicializados correctamente

Postcondición

La salida esperada es OK

4.19.2.4. void test1_game_add_space ( )

Prueba Prueba la función que añade un espacio al juego

Precondición

El juego y el espacio han sido inicializados correctamente

Postcondición

La salida esperada es OK

4.19.2.5. void test1_game_create ( )

Prueba Prueba la funcion que reserva memoria para la estructra del jueg

Precondición

Se reserva memoria normalmente

Postcondición

La salida tiene que ser el juego inicializado

Número maximo de tests

4.19.2.6. void test1_game_destroy ( )

Prueba Prueba la funcon que destruye un juego

Precondición

El juego se ha creado previamente

Postcondición

La salida esperada es OK

4.19.2.7. void test1_game_get_dialogue ( )

Prueba Prueba la función que devuelve el dialogo del juego

Precondición

El juego ha sido creado correctamente

Postcondición

La salida debe ser un puntero distinto de NULL

4.19.2.8. void test1_game_get_die ( )

Prueba Prueba la función que devuelve el dado del juego

Precondición

El juego ha sido creado correctamente

Postcondición

La salida debe ser un puntero distinto de NULL

4.19.2.9. void test1_game_get_last_inspected_object ( )

Prueba Prueba la función que devuelve el ultimo objeto inspeccionado

Precondición

El nombre del objeto añadido es "obj"

Postcondición

La salida esperada es el objeto con nombre "obj"

4.19.2.10. void test1_game_get_last_inspected_space ( )

Prueba Prueba la función que devuelve el ultimo espacio inspeccionado

Precondición

El juego ha sido creado, se ha añadido un espacio y se ha colocado al jugador dentro

Postcondición

La salida esperada es la casilla en la que esta el jugador

4.19.2.11. void test1_game_get_link ( )

Prueba Prueba la función que devuelve el link con in id dado

Precondición

El juego y el link han sido creados correctamente

Postcondición

La salida debe ser el link creado anteriormente

4.19.2.12. void test1_game_get_link_at ( )

Prueba Prueba la función que devuelve el link en una posición dada

Precondición

El link ha sido añadido correctamente

Postcondición

La salida debe ser el link añadido anteriormente

4.19.2.13. void test1_game_get_obj_list_as_str ( )

Prueba Prueba la funcion que devuelve la lista de objetos de una casilla como cadena

Precondición

El juego, una casilla, y un objeto han sido añadidos correctamente

Postcondición

La salida esperada es la lista de objetos de la casilla

4.19.2.14. void test1_game_get_object_at ( )

Prueba Prueba la función que devuelve el objeto en una posición dada

Precondición

El objeto ha sido añadido correctamente

Postcondición

La salida debe ser el objeto añadido anteriormente

4.19.2.15. void test1_game_get_object_location ( )

Prueba Prueba la funcion que devuelve la localizacion de un objeto

Precondición

El objeto se ha añadido en la casilla con id 2

Postcondición

La salida es 2, la localización del objeto

4.19.2.16. void test1_game_get_player ( )

Prueba Prueba la función que devuelve el jugador del juego

Precondición

El juego y el jugador han sido creado correctamente

Postcondición

La salida debe ser un puntero distinto de NULL

4.19.2.17. void test1_game_get_player_location ( )

Prueba Prueba la funcion que devuelve la localizacion del jugador

Precondición

El jugador se ha colocado en la casilla con id 2

Postcondición

La salida es 2, la localización del jugador

4.19.2.18. void test1_game_get_space ( )

Prueba Prueba la función que devuelve el espacio con in id dado

Precondición

El juego y el espacio han sido creados correctamente

Postcondición

La salida debe ser el espacio creado anteriormente

4.19.2.19. void test1_game_get_space_at ( )

Prueba Prueba la función que devuelve el espacio en una posición dada

Precondición

El espacio ha sido añadido correctamente

Postcondición

La salida debe ser el espacio añadido anteriormente

4.19.2.20. void test1_game_is_over ( )

Prueba Prueba la función devuelve si el juego ha acabado o no

Precondición

El juego ha sido inicializado

Postcondición

La salida esperada es FALSE

4.19.2.21. void test1_game_update ( )

Prueba Prueba la función que actualiza el juego

Precondición

El juego se ha creado y el comando ejecutado es valido

Postcondición

La salida esperada es OK

4.19.2.22. void test2_game_add_link ( )

Prueba Prueba la función que añade un link al juego

Precondición

El juego no ha sido inicializado pero el link si

Postcondición

La salida esperada es ERROR

4.19.2.23. void test2_game_add_object ( )

Prueba Prueba la función que añade un objeto al juego

Precondición

El juego no ha sido inicializado pero el objeto si

Postcondición

La salida esperada es ERROR

4.19.2.24. void test2_game_add_player ( )

Prueba Prueba la función que añade un jugador al juego

Precondición

El juego no ha sido inicializado pero el jugador si

Postcondición

La salida esperada es ERROR

4.19.2.25. void test2_game_add_space ( )

Prueba Prueba la función que añade un espacio al juego

Precondición

El juego no ha sido inicializado pero el espacio si

Postcondición

La salida esperada es ERROR

4.19.2.26. void test2_game_destroy ( )

Prueba Prueba la funcon que destruye un juego

Precondición

El juego no se ha creado previamente

Postcondición

La salida esperada es ERROR

4.19.2.27. void test2_game_get_dialogue ( )

Prueba Prueba la función que devuelve el dialogo del juego

Precondición

El juego no ha sido inicializado

Postcondición

La salida debe ser NULL

4.19.2.28. void test2_game_get_die ( )

Prueba Prueba la función que devuelve el dado del juego

Precondición

El juego no ha sido inicializado

Postcondición

La salida debe ser NULL

4.19.2.29. void test2_game_get_last_inspected_object ( )

Prueba Prueba la función que devuelve el ultimo objeto inspeccionado

Precondición

El objeto está marcado como oculto

Postcondición

La salida esperada es NULL

4.19.2.30. void test2_game_get_last_inspected_space ( )

Prueba Prueba la función que devuelve el ultimo espacio inspeccionado

Precondición

El juego no ha sido creado

Postcondición

La salida esperada es NULL

4.19.2.31. void test2_game_get_link ( )

Prueba Prueba la función que devuelve el link con in id dado

Precondición

El juego no se ha creado pero el link si

Postcondición

La salida debe ser NULL

4.19.2.32. void test2_game_get_link_at ( )

Prueba Prueba la función que devuelve el link en una posición dada

Precondición

La posición en la que se busca es -1

Postcondición

La salida debe ser NULL

4.19.2.33. void test2_game_get_obj_list_as_str ( )

Prueba Prueba la funcion que devuelve la lista de objetos de una casilla como cadena

Precondición

El juego no se ha inicializado

Postcondición

La salida esperada es NULL

4.19.2.34. void test2_game_get_object_at ( )

Prueba Prueba la función que devuelve el objeto en una posición dada

Precondición

La posición en la que se busca es -1

Postcondición

La salida debe ser NULL

4.19.2.35. void test2_game_get_object_location ( )

Prueba Prueba la funcion que devuelve la localizacion de un objeto

Precondición

El objeto no se ha añadido al juego

Postcondición

La salida es NO_ID - localizacion nula

4.19.2.36. void test2_game_get_player ( )

Prueba Prueba la función que devuelve el jugador del juego

Precondición

El juego ha sido inicializado pero no se ha añadido ningun jugador

Postcondición

La salida debe ser NULL

4.19.2.37. void test2_game_get_player_location ( )

Prueba Prueba la funcion que devuelve la localizacion del jugador

Precondición

El jugador no se ha añadido al juego

Postcondición

La salida es NO_ID, localizacion nula

4.19.2.38. void test2_game_get_space ( )

Prueba Prueba la función que devuelve el espacio con in id dado

Precondición

El juego no se ha creado pero el espacio si

Postcondición

La salida debe ser NULL

4.19.2.39. void test2_game_get_space_at ( )

Prueba Prueba la función que devuelve el espacio en una posición dada

Precondición

La posición en la que se busca es -1

Postcondición

La salida debe ser NULL

4.19.2.40. void test2_game_is_over ( )

Prueba Prueba la función devuelve si el juego ha acabado o no

Precondición

El juego no ha sido inicializado

Postcondición

La salida esperada es TRUE, de esa forma en gameloop no se sigue jugando

4.19.2.41. void test2_game_update ( )

Prueba Prueba la función que actualiza el juego

Precondición

El juego no se ha creado. El comando ejecutado es valido

Postcondición

La salida esperada es ERROR

4.19.2.42. void test3_game_add_link ( )

Prueba Prueba la función que añade un link al juego

Precondición

El juego ha sido inicializado pero el link no

Postcondición

La salida esperada es ERROR

4.19.2.43. void test3_game_add_object ( )

Prueba Prueba la función que añade un objeto al juego

Precondición

El juego ha sido inicializado pero el objeto no

Postcondición

La salida esperada es ERROR

4.19.2.44. void test3_game_add_player ( )

Prueba Prueba la función que añade un jugador al juego

Precondición

El juego ha sido inicializado pero el jugador no

Postcondición

La salida esperada es ERROR

4.19.2.45. void test3_game_add_space ( )

Prueba Prueba la función que añade un espacio al juego

Precondición

El juego ha sido inicializado pero el espacio no

Postcondición

La salida esperada es ERROR

4.19.2.46. void test3_game_get_last_inspected_object ( )

Prueba Prueba la función que devuelve el ultimo objeto inspeccionado

Precondición

El nombre del objeto no se corresponde con el del objeto del juego

Postcondición

La salida esperada es NULL

4.19.2.47. void test3_game_get_link ( )

Prueba Prueba la función que devuelve el link con in id dado

Precondición

El link buscado no coresponde con el id del link añadido

Postcondición

La salida debe ser NULL

4.19.2.48. void test3_game_get_link_at ( )

Prueba Prueba la función que devuelve el link en una posición dada

Precondición

El juego no se ha inicializado

Postcondición

La salida debe ser NULL

4.19.2.49. void test3_game_get_object_at ( )

Prueba Prueba la función que devuelve el objeto en una posición dada

Precondición

El juego no se ha inicializado

Postcondición

La salida debe ser NULL

4.19.2.50. void test3_game_get_object_location ( )

Prueba Prueba la funcion que devuelve la localizacion de un objeto

Precondición

El objeto está marcado como oculto

Postcondición

La salida es NO_ID - localizacion nula

4.19.2.51. void test3_game_get_space ( )

Prueba Prueba la función que devuelve el espacio con in id dado

Precondición

El espacio buscado no coresponde con el id del espacio añadido

Postcondición

La salida debe ser NULL

4.19.2.52. void test3_game_get_space_at ( )

Prueba Prueba la función que devuelve el espacio en una posición dada

Precondición

El juego no se ha inicializado

Postcondición

La salida debe ser NULL

4.19.2.53. void test3_game_update ( )

Prueba Prueba la función que actualiza el juego

Precondición

El juego se ha creado pero el comando ejecutado es invalido

Postcondición

La salida esperada es ERROR

4.20. Referencia del Archivo include/tests/graphic_engine_test.h

Define las funciones para la prueba del modulo graphic_engine.

Funciones

- void test1_graphic_engine_create ()

4.20.1. Descripción detallada

Define las funciones para la prueba del modulo graphic_engine.

Autor

Mihai Blidaru

Versión

1.0

Fecha

14-03-2017

Copyright

GNU Public License

4.20.2. Documentación de las funciones

4.20.2.1. void test1_graphic_engine_create ( )

Prueba Prueba la función que crea el motor grafico

Precondición

El motor grafico se reserva

Postcondición

La salida esperada es el motor grafico inicializado

Número maximo de tests

4.21. Referencia del Archivo include/tests/inventory_test.h

Define las funciones para la prueba del modulo Inventory.

Funciones

- void test1_inventory_create ()
- void test1_inventory_add_object ()
- void test2_inventory_add_object ()
- void test3_inventory_add_object ()
- void test1_inventory_set_max ()
- void test2_inventory_set_max ()
- void test3_inventory_set_max ()
- void test1_inventory_get_max ()
- void test2_inventory_get_max ()
- void test1_inventory_remove_object ()
- void test2_inventory_remove_object ()
- void test3_inventory_remove_object ()
- void test1_inventory_get_set ()
- void test2_inventory_get_set ()

4.21.1. Descripción detallada

Define las funciones para la prueba del modulo Inventory.

Autor

Sandra Benítez
Laura Bernal

Versión

1.0

Fecha

14-03-2017

Copyright

GNU Public License

4.21.2. Documentación de las funciones

4.21.2.1. void test1_inventory_add_object ( )

Prueba Prueba la función que añade un objeto a un inventario

Precondición

El inventario ha sido inicializado previamente

Postcondición

La salida esperada es ERROR

4.21.2.2. void test1_inventory_create ( )

Prueba Prueba la función que crea un inventario

Postcondición

La salida debe ser un inventario inicializado (diferente de NULL)

Número maximo de tests

4.21.2.3. void test1_inventory_get_max ( )

Prueba Prueba la función que devuelve el número máximo de objetos de un inventario

Precondición

Se establece previamente el número máximo de objetos a 20

Postcondición

La salida esperada es 20

4.21.2.4. void test1_inventory_get_set ( )

Prueba Prueba la función que devuelve el set de objetos de un inventario

Precondición

El inventario ha sido inicializado previamente

Postcondición

La salida esperada es diferente de NULL

4.21.2.5. void test1_inventory_remove_object ( )

Prueba Prueba la función que elimina un objeto de un inventario

Precondición

Se añade previamente un objeto con el id 10

Postcondición

La salida esperada es OK

4.21.2.6. void test1_inventory_set_max ( )

Prueba Prueba la función que establece el número máximo de objetos de un inventario

Precondición

El inventario ha sido creado previamente

Postcondición

La salida esperada es OK

4.21.2.7. void test2_inventory_add_object ( )

Prueba Prueba la función que añade un objeto a un inventario

Precondición

El inventario es un puntero a NULL

Postcondición

La salida esperada es ERROR

4.21.2.8. void test2_inventory_get_max ( )

Prueba Prueba la función que devuelve el número máximo de objetos de un inventario

Precondición

El inventario es un puntero a NULL

Postcondición

La salida esperada es -1

4.21.2.9. void test2_inventory_get_set ( )

Prueba Prueba la función que devuelve el set de objetos de un inventario

Precondición

El inventario es un puntero a NULL

Postcondición

La salida esperada es NULL

4.21.2.10. void test2_inventory_remove_object ( )

Prueba Prueba la función que elimina un objeto de un inventario

Precondición

El objeto no ha sido añadido previamente

Postcondición

La salida esperada es ERROR

4.21.2.11. void test2_inventory_set_max ( )

Prueba Prueba la función que establece el número máximo de objetos de un inventario

Precondición

El número máximo de objetos es invalido (-5)

Postcondición

La salida esperada es ERROR

4.21.2.12. void test3_inventory_add_object ( )

Prueba Prueba la función que añade un objeto a un inventario

Precondición

El id del objeto que se quiere añadir es NO_ID

Postcondición

La salida esperada es ERROR

4.21.2.13. void test3_inventory_remove_object ( )

Prueba Prueba la función que elimina un objeto de un inventario

Precondición

El inventario es un puntero a NULL

Postcondición

La salida esperada es ERROR

4.21.2.14. void test3_inventory_set_max ( )

Prueba Prueba la función que establece el número máximo de objetos de un inventario

Precondición

El inventario es un puntero a NULL

Postcondición

La salida esperada es ERROR

4.22. Referencia del Archivo include/tests/link_test.h

Pruebas para el modulo Link.

Funciones

- void test1_link_create ()
- void test1_link_set_id ()
- void test2_link_set_id ()
- void test3_link_set_id ()
- void test1_link_set_name ()
- void test2_link_set_name ()
- void test3_link_set_name ()
- void test1_link_set_space1 ()
- void test2_link_set_space1 ()
- void test3_link_set_space1 ()
- void test1_link_set_space2 ()
- void test2_link_set_space2 ()
- void test3_link_set_space2 ()
- void test1_link_set_state ()
- void test2_link_set_state ()
- void test3_link_set_state ()
- void test1_link_get_name ()
- void test2_link_get_name ()
- void test1_link_get_id ()
- void test2_link_get_id ()
- void test1_link_get_space1 ()
- void test2_link_get_space1 ()
- void test1_link_get_space2 ()
- void test2_link_get_space2 ()
- void test1_link_get_state ()
- void test2_link_get_state ()
- void test1_link_get_dest_from ()
- void test2_link_get_dest_from ()
- void test3_link_get_dest_from ()
- void test4_link_get_dest_from ()

4.22.1. Descripción detallada

Pruebas para el modulo Link.

Autor

Javier Bernardo

Fecha

27/03/2017

4.22.2. Documentación de las funciones

4.22.2.1. void test1_link_create ( )

Prueba Prueba la función de creación de un link

Precondición

Un identificador como parámetro

Postcondición

Un puntero no nulo al espacio creado

Número máximo de tests

4.22.2.2. void test1_link_get_dest_from ( )

Prueba Prueba la función que devuelve la casilla destino desde otra casilla

Precondición

El enlace se inicializa a (1, 2) y se establece que el id de origen es 1

Postcondición

La salida tiene que ser 2

4.22.2.3. void test1_link_get_id ( )

Prueba Prueba la función para obtener el id de un link

Precondición

Se establece previamente el id del link a 25

Postcondición

La salida debe ser el mismo id: 25

4.22.2.4. void test1_link_get_name ( )

Prueba Prueba la función para obtener el nombre de un link

Precondición

Al link se le pone un nombre previamente

Postcondición

La salida debe ser el mismo nombre establecido

4.22.2.5. void test1_link_get_space1 ( )

Prueba Prueba la función para obtener el space1 de un link

Precondición

Se ha establecido el space1 del link a 26

Postcondición

La salida debe ser el mismo id: 26

4.22.2.6. void test1_link_get_space2 ( )

Prueba Prueba la función para obtener el space2 de un link

Precondición

Se ha establecido el space2 del link a 27

Postcondición

La salida debe ser 27

4.22.2.7. void test1_link_get_state ( )

Prueba Prueba la función para obtener el estado de un link

Precondición

Se establece el estado del enlace como CLOSED

Postcondición

La salida debe ser CLOSED

4.22.2.8. void test1_link_set_id ( )

Prueba Prueba la función para establecer el id de un link

Precondición

El enlace ha sido inicializado previamente

Postcondición

La salida debe ser OK

4.22.2.9. void test1_link_set_name ( )

Prueba Prueba la función para establecer el nombre de un link

Precondición

El link ha sido creado previamente y se le pone un nombre

Postcondición

La salida debe ser OK

4.22.2.10. void test1_link_set_space1 ( )

Prueba Prueba la función para establecer el space1 de un link

Precondición

El enlace ha sido inicializado previamente

Postcondición

La salida debe ser OK

4.22.2.11. void test1_link_set_space2 ( )

Prueba Prueba la función para establecer el space2 de un link

Precondición

El enlace ha sido creado previamente

Postcondición

La salida debe ser OK

4.22.2.12. void test1_link_set_state ( )

Prueba Prueba la función para establecer el estado de un link

Precondición

El link se crea previamente

Postcondición

La salida debe ser OK

4.22.2.13. void test2_link_get_dest_from ( )

Prueba Prueba la función que devuelve la casilla destino desde otra casilla

Precondición

El enlace se inicializa a (1, 2) y se establece que el id de origen es 2

Postcondición

La salida tiene que ser 1

4.22.2.14. void test2_link_get_id ( )

Prueba Prueba la función para obtener el id de un link

Precondición

El enlace es un puntero a NULL

Postcondición

La salida debe ser NO_ID

4.22.2.15. void test2_link_get_name ( )

Prueba Prueba la función para obtener el nombre de un link

Precondición

El enlace es un puntero a NULL

Postcondición

La salida debe ser NULL

4.22.2.16. void test2_link_get_space1 ( )

Prueba Prueba la función para obtener el space1 de un link

Precondición

El enlace es un puntero a NULL

Postcondición

La salida debe ser NO_ID

4.22.2.17. void test2_link_get_space2 ( )

Prueba Prueba la función para obtener el space2 de un link

Precondición

El enlace es un puntero a NULL

Postcondición

La salida debe ser NO_ID

4.22.2.18. void test2_link_get_state ( )

Prueba Prueba la función para obtener el estado de un link

Precondición

El link es un puntero a NULL

Postcondición

La salida debe ser -1

4.22.2.19. void test2_link_set_id ( )

Prueba Prueba la función para establecer el id de un link

Precondición

El enlace es un puntero a NULL

Postcondición

La salida debe ser ERROR

4.22.2.20. void test2_link_set_name ( )

Prueba Prueba la función para establecer el nombre de un link

Precondición

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

Postcondición

La salida debe ser ERROR

4.22.2.21. void test2_link_set_space1 ( )

Prueba Prueba la función para establecer el space1 de un link

Precondición

El enlace es un puntero a NULL

Postcondición

La salida debe ser ERROR

4.22.2.22. void test2_link_set_space2 ( )

Prueba Prueba la función para establecer el space2 de un link

Precondición

El enlace es un puntero a NULL

Postcondición

La salida debe ser ERROR

4.22.2.23. void test2_link_set_state ( )

Prueba Prueba la función para establecer el estado de un link

Precondición

El enlace es un puntero a NULL

Postcondición

La salida debe ser ERROR

4.22.2.24. void test3_link_get_dest_from ( )

Prueba Prueba la función que devuelve la casilla destino desde otra casilla

Precondición

El enlace es un puntero a NULL

Postcondición

La salida tiene que ser NO_ID

4.22.2.25. void test3_link_set_id ( )

Prueba Prueba la función para establecer el id de un link

Precondición

El id que se quiere establecer es NO_ID

Postcondición

La salida debe ser ERROR

4.22.2.26. void test3_link_set_name ( )

Prueba Prueba la función para establecer el nombre de un link

Precondición

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

Postcondición

La salida debe ser ERROR

4.22.2.27. void test3_link_set_space1 ( )

Prueba Prueba la función para establecer el space1 de un link

Precondición

El id que se quiere establecer es NO_ID

Postcondición

La salida debe ser ERROR

4.22.2.28. void test3_link_set_space2 ( )

Prueba Prueba la función para establecer el space2 de un link

Precondición

El id que se quiere establecer es NO_ID

Postcondición

La salida debe ser ERROR

4.22.2.29. void test3_link_set_state ( )

Prueba Prueba la función para establecer el estado de un link

Precondición

El estado que se le quiere poner al link es invalido

Postcondición

La salida debe ser ERROR

4.22.2.30. void test4_link_get_dest_from ( )

Prueba Prueba la función que devuelve la casilla destino desde otra casilla

Precondición

El id de origen no está en el enlace

Postcondición

La salida tiene que ser NO_ID

4.23. Referencia del Archivo include/tests/object_test.h

Pruebas para el modulo Object.

Funciones

- void test1_object_create ()
- void test1_object_set_name ()
- void test2_object_set_name ()
- void test1_object_set_graphics ()
- void test2_object_set_graphics ()
- void test1_object_set_Id ()
- void test2_object_set_Id ()
- void test1_object_set_description ()
- void test2_object_set_description ()
- void test1_object_set_description2 ()
- void test2_object_set_description2 ()
- void test1_object_set_Mobile ()
- void test2_object_set_Mobile ()
- void test1_object_set_Moved ()
- void test2_object_set_Moved ()
- void test1_object_set_Hidden ()
- void test2_object_set_Hidden ()
- void test1_object_set_Open ()
- void test2_object_set_Open ()
- void test1_object_set_Illuminates ()
- void test2_object_set_Illuminates ()
- void test1_object_set_Light ()
- void test2_object_set_Light ()
- void test1_object_Get_Name ()
- void test2_object_Get_Name ()
- void test1_object_Get_Graphics ()
- void test2_object_Get_Graphics ()
- void test1_object_Get_Id ()
- void test2_object_Get_Id ()
- void test1_object_Get_Description ()
- void test2_object_Get_Description ()
- void test1_object_Get_Description2 ()
- void test2_object_Get_Description2 ()
- void test1_object_Get_Mobile ()
- void test2_object_Get_Mobile ()
- void test1_object_Get_Moved ()

- void test2_object_Get_Moved ()
- void test1_object_Get_Hidden ()
- void test2_object_Get_Hidden ()
- void test1_object_Get_Open ()
- void test2_object_Get_Open ()
- void test1_object_Get_Illuminates ()
- void test2_object_Get_Illuminates ()
- void test1_object_Get_Light ()
- void test2_object_Get_Light ()

4.23.1. Descripción detallada

Pruebas para el modulo Object.

Autor

Javier Bernardo

Fecha

24/04/2017

4.23.2. Documentación de las funciones

4.23.2.1. void test1_object_create ( )

Prueba Prueba si se crea correctamente un objeto

Postcondición

Un puntero no nulo al objeto creado

4.23.2.2. void test1_object_Get_Description ( )

Prueba Prueba leer la descripcion de un objeto

Precondición

Al objeto se le ha asignado previamente la descripcion

Postcondición

La salida esperada es la descripcion asignado antes

4.23.2.3. void test1_object_Get_Description2 ( )

Prueba Prueba leer la descripcion de un objeto

Precondición

Al objeto se le ha asignado previamente la descripcion

Postcondición

La salida esperada es la descripcion asignado antes

4.23.2.4. void test1_object_Get_Graphics ( )

Prueba Prueba leer los graficos de un objeto

Precondición

Al nombre se le ha asignado previamente los graficos "Bryan"

Postcondición

La salida esperada son los graficos asignados antes "Bryan"

4.23.2.5. void test1_object_Get_Hidden ( )

Prueba Prueba leer si se ha escondido un objeto

Precondición

Al objeto se le ha asignado previamente la invibisibilidad

Postcondición

La salida esperada es la movilidad asignada antes

4.23.2.6. void test1_object_Get_Id ( )

Prueba Prueba leer el id de un objeto

Precondición

Al objeto se le ha asignado previamente el id 12

Postcondición

La salida esperada es el id asignado antes (12)

4.23.2.7. void test1_object_Get_Illuminates ( )

Prueba Prueba leer si se ha iluminado un objeto

Precondición

Al objeto se le ha asignado previamente la posibilidad de iluminar

Postcondición

La salida esperada es la movilidad asignada antes

4.23.2.8. void test1_object_Get_Light ( )

Prueba Prueba leer si se puede iluminar un objeto

Precondición

Al objeto se le ha asignado previamente la posibilidad de iluminar

Postcondición

La salida esperada es la movilidad asignada antes

4.23.2.9. void test1_object_Get_Mobile ( )

Prueba Prueba leer la movilidad de un objeto

Precondición

Al objeto se le ha asignado previamente la movilidad

Postcondición

La salida esperada es la movilidad asignada antes

4.23.2.10. void test1_object_Get_Moved ( )

Prueba Prueba leer si se ha movido un objeto

Precondición

Al objeto se le ha asignado previamente la movilidad

Postcondición

La salida esperada es la movilidad asignada antes

4.23.2.11. void test1_object_Get_Name ( )

Prueba Prueba leer el nombre de un objeto

Precondición

Al nombre se le ha asignado previamente el nombre Bryan

Postcondición

La salida esperada es el id asignado antes "Bryan"

4.23.2.12. void test1_object_Get_Open ( )

Prueba Prueba leer si se ha abierto el link a un objeto

Precondición

Al objeto se le ha asignado previamente la disponibilidad

Postcondición

La salida esperada es la movilidad asignada antes

4.23.2.13. void test1_object_set_description ( )

Prueba Prueba si se le asigna correctamente una descripcion a un objeto

Precondición

La descripcion del objeto

Postcondición

La salida tiene que ser el puntero al objeto

4.23.2.14. void test1_object_set_description2 ( )

Prueba Prueba si se le asigna correctamente una descripcion2 a un objeto

Precondición

La descripcion2 del objeto

Postcondición

La salida tiene que ser el puntero al objeto

4.23.2.15. void test1_object_set_graphics ( )

Prueba Prueba si se le asigna correctamente los graficos a un objeto

Precondición

El nombre del objeto

Postcondición

La salida tiene que ser el puntero al objeto

4.23.2.16. void test1_object_set_Hidden ( )

Prueba Prueba si se le asigna correctamente una situacion a un objeto

Precondición

La situacion del objeto

Postcondición

La salida tiene que ser el puntero al objeto

4.23.2.17. void test1_object_set_Id ( )

Prueba Prueba si se le asigna correctamente un Id a un objeto

Precondición

El Id del objeto

Postcondición

La salida tiene que ser el puntero al objeto

4.23.2.18. void test1_object_set_Illuminates ( )

Prueba Prueba si se le asigna correctamente una iluminacion a un objeto

Precondición

La iluminacion del objeto

Postcondición

La salida tiene que ser el puntero al objeto

4.23.2.19. void test1_object_set_Light ( )

Prueba Prueba si se le asigna correctamente una iluminacion a un objeto

Precondición

La iluminacion del objeto

Postcondición

La salida tiene que ser el puntero al objeto

4.23.2.20. void test1_object_set_Mobile ( )

Prueba Prueba si se le asigna correctamente una movilidad a un objeto

Precondición

La movilidad del objeto

Postcondición

La salida tiene que ser el puntero al objeto

4.23.2.21. void test1_object_set_Moved ( )

Prueba Prueba si se le asigna correctamente una movilidad a un objeto

Precondición

La movilidad del objeto

Postcondición

La salida tiene que ser el puntero al objeto

4.23.2.22. void test1_object_set_name ( )

Prueba Prueba si se le asigna correctamente un nombre a un objeto

Precondición

El nombre del objeto

Postcondición

La salida tiene que ser el puntero al objeto

4.23.2.23. void test1_object_set_Open ( )

Prueba Prueba si se le asigna correctamente una situacion a un objeto

Precondición

La situacion del objeto

Postcondición

La salida tiene que ser el puntero al objeto

4.23.2.24. void test2_object_Get_Description ( )

Prueba Prueba leer la descripcion de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es NULL

4.23.2.25. void test2_object_Get_Description2 ( )

Prueba Prueba leer la descripcion de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es NULL

4.23.2.26. void test2_object_Get_Graphics ( )

Prueba Prueba leer los graficos de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es NULL

4.23.2.27. void test2_object_Get_Hidden ( )

Prueba Prueba leer la situacion de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es FALSE

4.23.2.28. void test2_object_Get_Id ( )

Prueba Prueba leer el id de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es NO_ID

4.23.2.29. void test2_object_Get_Illuminates ( )

Prueba Prueba leer la situacion de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es FALSE

4.23.2.30. void test2_object_Get_Light ( )

Prueba Prueba leer la situacion de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es FALSE

4.23.2.31. void test2_object_Get_Mobile ( )

Prueba Prueba leer la movilidad de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es FALSE

4.23.2.32. void test2_object_Get_Moved ( )

Prueba Prueba leer la movilidad de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es FALSE

4.23.2.33. void test2_object_Get_Name ( )

Prueba Prueba leer el nombre de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es NULL

4.23.2.34. void test2_object_Get_Open ( )

Prueba Prueba leer la situacion del link a un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es NO_ID

4.23.2.35. void test2_object_set_description ( )

Prueba Prueba asignar una descripcion a un objeto sin inicializar

Precondición

El objeto al que establecer la descripcion es un puntero a NULL

Postcondición

La salida debe ser NULL

4.23.2.36. void test2_object_set_description2 ( )

Prueba Prueba asignar una descripcion2 a un objeto sin inicializar

Precondición

El objeto al que establecer la descripcion2 es un puntero a NULL

Postcondición

La salida debe ser NULL

4.23.2.37. void test2_object_set_graphics ( )

Prueba Prueba asignar graficos a un objeto sin inicializar

Precondición

El objeto al que establecer los graficos es un puntero a NULL

Postcondición

La salida debe ser NULL

4.23.2.38. void test2_object_set_Hidden ( )

Prueba Prueba asignar una situacion a un objeto sin inicializar

Precondición

El objeto al que establecer la situacion es un puntero a NULL

Postcondición

La salida debe ser NULL

4.23.2.39. void test2_object_set_Id ( )

Prueba Prueba asignar un Id a un objeto sin inicializar

Precondición

El objeto al que establecer el Id es un puntero a NULL

Postcondición

La salida debe ser NULL

4.23.2.40. void test2_object_set_Illuminates ( )

Prueba Prueba asignar una iluminacion a un objeto sin inicializar

Precondición

El objeto al que establecer la iluminacion es un puntero a NULL

Postcondición

La salida debe ser NULL

4.23.2.41. void test2_object_set_Light ( )

Prueba Prueba asignar una iluminacion a un objeto sin inicializar

Precondición

El objeto al que establecer la iluminacion es un puntero a NULL

Postcondición

La salida debe ser NULL

4.23.2.42. void test2_object_set_Mobile ( )

Prueba Prueba asignar una movilidad a un objeto sin inicializar

Precondición

El objeto al que establecer la movilidad es un puntero a NULL

Postcondición

La salida debe ser NULL

4.23.2.43. void test2_object_set_Moved ( )

Prueba Prueba asignar una movilidad a un objeto sin inicializar

Precondición

El objeto al que establecer la movilidad es un puntero a NULL

Postcondición

La salida debe ser NULL

4.23.2.44. void test2_object_set_name ( )

Prueba Prueba asignar un nombre a un objeto sin inicializar

Precondición

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

Postcondición

La salida debe ser NULL

4.23.2.45. void test2_object_set_Open ( )

Prueba Prueba asignar una situacion a un objeto sin inicializar

Precondición

El objeto al que establecer la situacion es un puntero a NULL

Postcondición

La salida debe ser NULL

4.24. Referencia del Archivo include/tests/player_test.h

It declares the tests for the player module.

Funciones

- void test1_player_create ()
- void test1_player_set_name ()
- void test2_player_set_name ()
- void test3_player_set_name ()
- void test1_player_set_id ()
- void test2_player_set_id ()
- void test3_player_set_id ()
- void test1_player_set_location ()
- void test2_player_set_location ()
- void test3_player_set_location ()
- void test1_player_get_id ()
- void test2_player_get_id ()
- void test3_player_get_id ()
- void test1_player_get_name ()
- void test2_player_get_name ()
- void test3_player_get_name ()
- void test1_player_get_location ()
- void test2_player_get_location ()
- void test3_player_get_location ()
- void test1_player_add_object ()
- void test2_player_add_object ()
- void test3_player_add_object ()
- void test1_player_remove_object ()
- void test2_player_remove_object ()
- void test3_player_remove_object ()

- void test4_player_remove_object ()
- void test1_player_set_max_objects ()
- void test2_player_set_max_objects ()
- void test3_player_set_max_objects ()
- void test4_player_set_max_objects ()
- void test1_player_has_object ()
- void test2_player_has_object ()
- void test3_player_has_object ()

4.24.1. Descripción detallada

It declares the tests for the player module.

Autor

Mihai Blidaru

Versión

1.0

Fecha

19-01-2016

Copyright

GNU Public License

4.24.2. Documentación de las funciones

4.24.2.1. void test1_player_add_object ( )

Prueba Prueba añadir un objeto a un jugador

Precondición

El jugador está sin inicializar

Postcondición

La salida que se espera es ERROR

4.24.2.2. void test1_player_create ( )

Prueba Prueba si se crea correctamente un jugador

Postcondición

Un puntero no nulo al jugador creado

4.24.2.3. void test1_player_get_id ( )

Prueba Prueba leer el id de un jugador

Precondición

Al jugador se le ha asignado previamente el id 12

Postcondición

La salida esperada es el id asignado antes (12)

4.24.2.4. void test1_player_get_location ( )

Prueba Leer la localización de un jugador

Precondición

Al jugador se le ha asignado previamente la localización 12

Postcondición

La salida esperada es la localización previamente asignada (12)

4.24.2.5. void test1_player_get_name ( )

Prueba Intenta leer el nombre de un jugador

Precondición

Al jugador se le ha asignado previamente el nombre "Bob"

Postcondición

La salida esperada es el nombre asignado antes : "Bob"

4.24.2.6. void test1_player_has_object ( )

Prueba Prubea si el jugador tiene un objeto en condiciones normales

Precondición

Se le añade un objeto al jugador

Postcondición

La salida debe ser TRUE

4.24.2.7. void test1_player_remove_object ( )

Prueba Prueba quitar un objeto al jugador en condiciones normales

Precondición

Al jugador se le añade un objeto

Postcondición

La salida que se espera es OK

4.24.2.8. void test1_player_set_id ( )

Prueba Prueba asignar un id a un jugador en condiciones normales

Precondición

Al jugador se le asigna un id cualquiera mayor que cero

Postcondición

La salida tiene que ser el puntero al jugador

4.24.2.9. void test1_player_set_location ( )

Prueba Prueba asignarle una localización a un jugador en condiciones

Precondición

El jugador ha sido previamente inicializado y la localización que se le asigna es valida

Postcondición

La salida es el puntero al jugador

4.24.2.10. void test1_player_set_max_objects ( )

Prueba Prueba poner el número máximo de objetos de un jugador

Precondición

Las condiciones son normales: jugador inicializado, número de objeto dentro de limites.

Postcondición

La salida debe ser el puntero al jugador

4.24.2.11. void test1_player_set_name ( )

Prueba Prueba si se le asigna correctamente un nombre a un jugador

Precondición

El nombre del jugador

Postcondición

La salida tiene que ser el puntero al jugador

4.24.2.12. void test2_player_add_object ( )

Prueba Prueba añadir un objeto a un jugador en condiciones normales

Precondición

El jugador ha sido correctamente inicializado y el id del objeto es valido

Postcondición

La salida debe ser OK

4.24.2.13. void test2_player_get_id ( )

Prueba Prueba leer el id de un jugador

Precondición

El jugador no ha sido inicializado previamente

Postcondición

La salida esperada es NO_ID

4.24.2.14. void test2_player_get_location ( )

Prueba Prueba leer la localización del jugador

Precondición

El jugador no ha sido inicializado

Postcondición

La salida esperada es NO_ID

4.24.2.15. void test2_player_get_name ( )

Prueba Prueba leer el nombre de un jugador

Precondición

El jugador está sin inicializar

Postcondición

La salida debe ser NULL

4.24.2.16. void test2_player_has_object ( )

Prueba Prueba si el jugador tiene un objeto

Precondición

El jugador está sin inicializar

Postcondición

La salida debe ser FALSE

4.24.2.17. void test2_player_remove_object ( )

Prueba Prueba quitar un objeto a un jugador

Precondición

Al jugador no se le ha añadido todavía ningun objeto

Postcondición

La salida que se espera es ERROR

4.24.2.18. void test2_player_set_id ( )

Prueba Prueba asignar un id a un jugador sin inicializar

Precondición

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

Postcondición

La salida esperada es NULL

4.24.2.19. void test2_player_set_location ( )

Prueba Prueba asignarle una localización a un jugador sin inicializar.

Precondición

El jugador no ha sido inicializado previamente

Postcondición

La salida que se espera es NULL

4.24.2.20. void test2_player_set_max_objects ( )

Prueba Prueba poner el número máximo de objetos de un jugador

Precondición

El número de objetos está por encima del limite permitido

Postcondición

La salida debe ser NULL

4.24.2.21. void test2_player_set_name ( )

Prueba Prueba asignar un nombre a un jugador sin inicializar

Precondición

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

Postcondición

La salida debe ser NULL

4.24.2.22. void test3_player_add_object ( )

Prueba Prueba añadir un objeto al inventario del jugador

Precondición

El objeto tiene id -1

Postcondición

La salida que se espera es ERROR

4.24.2.23. void test3_player_get_id ( )

Prueba Intenta leer el id del jugador

Precondición

Al jugador no se le ha asignado ninguna id todavia

Postcondición

La salida esperada es NO_ID

4.24.2.24. void test3_player_get_location ( )

Prueba Prueba leer la localización del jugador

Precondición

Al jugador no se le ha asignado ninguna localización todavía

Postcondición

La salida que se espera es NO_ID

4.24.2.25. void test3_player_get_name ( )

Prueba Prueba leer el nombre de un jugador

Precondición

Al jugador no se le ha dado ningun nombre previamente

Postcondición

La salida esperada es una cadena vacía

4.24.2.26. void test3_player_has_object ( )

Prueba Prueba si el jugador tiene un objeto

Precondición

El jugador está inicializado pero no tiene ningun objeto

Postcondición

La salida debe ser FALSE

4.24.2.27. void test3_player_remove_object ( )

Prueba Prueba quitar un objeto a un jugador

Precondición

El jugador está sin inicializar

Postcondición

La salida que se espera es ERROR

4.24.2.28. void test3_player_set_id ( )

Prueba Prueba asignar un id a un jugador

Precondición

El id que se quiere asignar es NO_ID

Postcondición

La salida que se espera es NULL

4.24.2.29. void test3_player_set_location ( )

Prueba Prueba asignarle una localización invalida a un jugador.

Precondición

La localización que se quiere asignar al jugador es es negativa

Postcondición

La salida esperada es NULL

4.24.2.30. void test3_player_set_max_objects ( )

Prueba Prueba poner el número máximo de objetos de un jugador

Precondición

El jugador está sin inicializar

Postcondición

La salida debe ser NULL

4.24.2.31. void test3_player_set_name ( )

Prueba Prueba asignarle un nombre nulo a un jugador

Precondición

La cadena que se quiere asignar es un puntero a NULL

Postcondición

La salida tiene que ser NULL

4.24.2.32. void test4_player_remove_object ( )

Prueba Prueba quitar un objeto al jugador

Precondición

El objeto ya ha sido borrado anteriormente

Postcondición

La salida que se espera es ERROR

4.24.2.33. void test4_player_set_max_objects ( )

Prueba Prueba poner el numero máximo de objetos de un jugador

Precondición

El número de objetos es un número negativo

Postcondición

La salida debe ser NULL

4.25. Referencia del Archivo include/tests/screen_test.h

Define las funciones para la prueba del modulo Screen.

Funciones

- void test1_screen_area_init ()
- void test2_screen_area_init ()
- void test3_screen_area_init ()
- void test4_screen_area_init ()
- void test5_screen_area_init ()

4.25.1. Descripción detallada

Define las funciones para la prueba del modulo Screen.

Autor

Javier Bernardo

Versión

1.0

Fecha

24-04-2017

Copyright

GNU Public License

4.25.2. Documentación de las funciones

4.25.2.1. void test1_screen_area_init ( )

Prueba Prueba la función que inicializa un area

Precondición

Todos los parametros son correctos

Postcondición

La salida esperada es el area inicializado

Número maximo de tests

4.25.2.2. void test2_screen_area_init ( )

Prueba Prueba la función que inicializa un area

Precondición

La posición x es negativa

Postcondición

La salida esperada es NULL

4.25.2.3. void test3_screen_area_init ( )

Prueba Prueba la función que inicializa un area

Precondición

La posición y es negativa

Postcondición

La salida esperada es NULL

4.25.2.4. void test4_screen_area_init ( )

Prueba Prueba la función que inicializa un area

Precondición

El ancho del area es negativo

Postcondición

La salida esperada es NULL

4.25.2.5. void test5_screen_area_init ( )

Prueba Prueba la función que inicializa un area

Precondición

El alto del area es negativo

Postcondición

La salida esperada es NULL

4.26. Referencia del Archivo include/tests/space_test.h

It declares the tests for the space module.

Funciones

- void test1_space_create ()
- void test2_space_create ()
- void test3_space_create ()
- void test1_space_set_name ()
- void test2_space_set_name ()
- void test3_space_set_name ()
- void test1_space_set_description ()
- void test2_space_set_description ()
- void test3_space_set_description ()
- void test1_space_set_long_description ()
- void test2_space_set_long_description ()
- void test3_space_set_long_description ()
- void test1_space_set_graphics ()
- void test2_space_set_graphics ()
- void test1_space_set_north ()
- void test2_space_set_north ()
- void test3_space_set_north ()
- void test1_space_set_south ()
- void test2_space_set_south ()
- void test3_space_set_south ()
- void test1_space_set_east ()
- void test2_space_set_east ()
- void test3_space_set_east ()
- void test1_space_set_west ()
- void test2_space_set_west ()
- void test3_space_set_west ()
- void test1_space_set_up ()
- void test2_space_set_up ()
- void test3_space_set_up ()
- void test1_space_set_down ()
- void test2_space_set_down ()
- void test3_space_set_down ()
- void test1_space_get_id ()
- void test2_space_get_id ()
- void test1_space_get_name ()
- void test2_space_get_name ()
- void test3_space_get_name ()
- void test1_space_get_description ()
- void test2_space_get_description ()
- void test3_space_get_description ()
- void test1_space_get_long_description ()
- void test2_space_get_long_description ()
- void test3_space_get_long_description ()
- void test1_space_get_graphics ()
- void test2_space_get_graphics ()
- void test1_space_get_objects ()
- void test2_space_get_objects ()
- void test1_space_get_north ()
- void test2_space_get_north ()
- void test3_space_get_north ()
- void test1_space_get_south ()
- void test2_space_get_south ()
- void test3_space_get_south ()

- void test1_space_get_east ()
- void test2_space_get_east ()
- void test3_space_get_east ()
- void test1_space_get_west ()
- void test2_space_get_west ()
- void test3_space_get_west ()
- void test1_space_get_up ()
- void test2_space_get_up ()
- void test3_space_get_up ()
- void test1_space_get_down ()
- void test2_space_get_down ()
- void test3_space_get_down ()
- void test1_space_add_object ()
- void test2_space_add_object ()
- void test3_space_add_object ()
- void test1_space_remove_object ()
- void test2_space_remove_object ()
- void test3_space_remove_object ()
- void test4_space_remove_object ()
- void test1_space_contains_object ()
- void test2_space_contains_object ()
- void test3_space_contains_object ()
- void test1_space_graphics_areEmpty ()
- void test2_space_graphics_areEmpty ()
- void test3_space_graphics_areEmpty ()
- void test1_space_set_iluminated ()
- void test2_space_set_iluminated ()
- void test1_space_get_iluminated ()
- void test2_space_get_iluminated ()

4.26.1. Descripción detallada

It declares the tests for the space module.

Autor

Javier Bernardo
Sandra Benitez

Versión

2.0

Fecha

29-03-2016

Copyright

GNU Public License

4.26.2. Documentación de las funciones

4.26.2.1. void test1_space_add_object ( )

Prueba Prueba la función que añade un objeto a una casilla

Precondición

El espacio es un puntero a NULL.

Postcondición

La salida debe ser ERROR.

4.26.2.2. void test1_space_contains_object ( )

Prueba Prueba la función que comprueba si una casilla contiene un objeto

Precondición

El espacio se ha inicializado y se ha añadido un objeto

Postcondición

La salida esperada es TRUE

4.26.2.3. void test1_space_create ( )

Prueba Prueba la función de creación de un espacio

Precondición

Un identificador como parámetro

Postcondición

Un puntero no nulo al espacio creado

Número máximo de pruebas para el módulo space

4.26.2.4. void test1_space_get_description ( )

Prueba Prueba la función que devuelve la descripción de una casilla

Precondición

Al espacio se le ha establecido una descripción previamente

Postcondición

La salda debe ser la descripción previamente establecida

4.26.2.5. void test1_space_get_down ( )

Prueba Prueba la función que devuelve el enlace inferior de una casilla

Precondición

Se ha añadido previamente un enlace on id 7

Postcondición

La salida debe ser el id del enlace 7

4.26.2.6. void test1_space_get_east ( )

Prueba Prueba la función que devuelve el enlace este de una casilla

Precondición

Se ha añadido previamente un enlace on id 7

Postcondición

La salida debe ser el id del enlace 7

4.26.2.7. void test1_space_get_graphics ( )

Prueba Prueba la función que devuelve los graficos de una casilla

Precondición

A la casilla se le han establecido graficos previamente

Postcondición

La salida tiene que ser igual a los graficos establecidos previamente

4.26.2.8. void test1_space_get_id ( )

Prueba Prueba la función que devuelve el id de una casilla

Precondición

Al espacio se le ha establecido un id (12)

Postcondición

La salida esperada es el id establecido 12

4.26.2.9. void test1_space_get_iluminated ( )

Prueba Prueba la funcion que devuelve el estado de la iluminacion en una casilla

Precondición

Se ha creado e iluminado la casilla

Postcondición

La salida esperada es TRUE

4.26.2.10. void test1_space_get_long_description ( )

Prueba Prueba la función que devuelve la descripción larga de una casilla

Precondición

Al espacio se le ha establecido una descripción previamente

Postcondición

La salda debe ser la descripción previamente establecida

4.26.2.11. void test1_space_get_name ( )

Prueba Prueba la función que devuelve el nombre de una casilla

Precondición

Al espacio se le ha establecido un nombre previamente

Postcondición

La salda debe ser el nombre previamente establecido

4.26.2.12. void test1_space_get_north ( )

Prueba Prueba la función que devuelve el enlace norte de una casilla

Precondición

Se ha añadido previamente un enlace on id 7

Postcondición

La salida debe ser el id del enlace 7

4.26.2.13. void test1_space_get_objects ( )

Prueba Prueba la función que devuelve el set de objetos de una casilla

Precondición

El espacio ha sido previamente inicializado.

Postcondición

La salida debe ser diferente de NULL.

4.26.2.14. void test1_space_get_south ( )

Prueba Prueba la función que devuelve el enlace sur de una casilla

Precondición

Se ha añadido previamente un enlace on id 7

Postcondición

La salida debe ser el id del enlace 7

4.26.2.15. void test1_space_get_up ( )

Prueba Prueba la función que devuelve el enlace superior de una casilla

Precondición

Se ha añadido previamente un enlace on id 7

Postcondición

La salida debe ser el id del enlace 7

4.26.2.16. void test1_space_get_west ( )

Prueba Prueba la función que devuelve el enlace oeste de una casilla

Precondición

Se ha añadido previamente un enlace on id 7

Postcondición

La salida debe ser el id del enlace 7

4.26.2.17. void test1_space_graphics_areEmpty ( )

Prueba Prueba la función que comprueba si los graficos de una casilla están vacios.

Precondición

El espacio se ha creado y se le han establecido unos graficos

Postcondición

La salida esperada es FALSE

4.26.2.18. void test1_space_remove_object ( )

Prueba Prueba la función que quita un objeto de una casilla

Precondición

Se añade un objeto previamente

Postcondición

La salida debe ser OK.

4.26.2.19. void test1_space_set_description ( )

Prueba Prueba la función para establecer la descripción de un espacio

Precondición

Nombre que establecer al espacio

Postcondición

La salida debe ser OK

4.26.2.20. void test1_space_set_down ( )

Prueba Prueba la función que establece el enlace inferior de un espacio

Precondición

A la función se le pasa el identificador 4

Postcondición

La salida esperada es OK

4.26.2.21. void test1_space_set_east ( )

Prueba Prueba la función que establece el enlace este de un espacio

Precondición

A la función se le pasa el identificador 4

Postcondición

La salida esperada es OK

4.26.2.22. void test1_space_set_graphics ( )

Prueba Prueba la función que establece los graficos de una casilla

Precondición

Unos graficos aleatorios

Postcondición

La salida esperada es un puntero al espacio

4.26.2.23. void test1_space_set_iluminated ( )

Prueba Prueba la funcion que pone el estado de la iluminacion en una casilla

Precondición

Se ha creado una casiilla con id 5

Postcondición

La salida esperada es OK

4.26.2.24. void test1_space_set_long_description ( )

Prueba Prueba la función para establecer la descripción larga de un espacio

Precondición

Nombre que establecer al espacio

Postcondición

La salida debe ser OK

4.26.2.25. void test1_space_set_name ( )

Prueba Prueba la función para establecer el nombre de un espacio

Precondición

Nombre que establecer al espacio

Postcondición

La salida debe ser OK

4.26.2.26. void test1_space_set_north ( )

Prueba Prueba la función que establece el enlace norte de un espacio

Precondición

Id del enlace

Postcondición

La salida esperada es OK

4.26.2.27. void test1_space_set_south ( )

Prueba Prueba la función que establece el enlace sur de un espacio

Precondición

A la función se le pasa el identificador 4

Postcondición

La salida esperada es OK

4.26.2.28. void test1_space_set_up ( )

Prueba Prueba la función que establece el enlace superior de un espacio

Precondición

A la función se le pasa el identificador 4

Postcondición

La salida esperada es OK

4.26.2.29. void test1_space_set_west ( )

Prueba Prueba la función que establece el enlace oeste de un espacio

Precondición

A la función se le pasa el identificador 4

Postcondición

La salida esperada es OK

4.26.2.30. void test2_space_add_object ( )

Prueba Prueba la función que añade un objeto a una casilla

Precondición

Se añade un objeto con id 10

Postcondición

La salida debe ser OK.

4.26.2.31. void test2_space_contains_object ( )

Prueba Prueba la función que comprueba si una casilla contiene un objeto

Precondición

El espacio no se ha inicializado.

Postcondición

La salida esperada es FALSE

4.26.2.32. void test2_space_create ( )

Prueba Prueba la función de creación de un espacio

Precondición

Un identificador como parámetro

Postcondición

El identificador del espacio es el introducido

4.26.2.33. void test2_space_get_description ( )

Prueba Prueba la función que devuelve la descripción de una casilla

Precondición

El espacio es un puntero a NULL

Postcondición

La salda debe ser un puntero a NULL

4.26.2.34. void test2_space_get_down ( )

Prueba Prueba la función que devuelve el enlace inferior de una casilla

Precondición

El espacio es un puntero a NULL.

Postcondición

La salida debe ser NO_ID.

4.26.2.35. void test2_space_get_east ( )

Prueba Prueba la función que devuelve el enlace este de una casilla

Precondición

El espacio es un puntero a NULL.

Postcondición

La salida debe ser NO_ID.

4.26.2.36. void test2_space_get_graphics ( )

Prueba Prueba la función que devuelve los graficos de una casilla

Precondición

El espacio ha sido inicializado pero no se le han establecido graficos

Postcondición

La salida debe ser una matriz de 3 cadenas de 7 caracteres llenas de espacios

4.26.2.37. void test2_space_get_id ( )

Prueba Prueba la función que devuelve el id de una casilla

Precondición

El espacio es un puntero a NULL

Postcondición

La salida esperada es NO_ID

4.26.2.38. void test2_space_get_iluminated ( )

Prueba Prueba la funcion que devuelve el estado de la iluminacion en una casilla

Precondición

No se ha creado la casilla

Postcondición

La salida esperada es FALSE

4.26.2.39. void test2_space_get_long_description ( )

Prueba Prueba la función que devuelve la descripción larga de una casilla

Precondición

El espacio es un puntero a NULL

Postcondición

La salda debe ser un puntero a NULL

4.26.2.40. void test2_space_get_name ( )

Prueba Prueba la función que devuelve el nombre de una casilla

Precondición

El espacio es un puntero a NULL

Postcondición

La salda debe ser un puntero a NULL

4.26.2.41. void test2_space_get_north ( )

Prueba Prueba la función que devuelve el enlace norte de una casilla

Precondición

El espacio es un puntero a NULL.

Postcondición

La salida debe ser NO_ID.

4.26.2.42. void test2_space_get_objects ( )

Prueba Prueba la función que devuelve el set de objetos de una casilla

Precondición

El espacio es un puntero a NULL

Postcondición

La salida debe ser NULL

4.26.2.43. void test2_space_get_south ( )

Prueba Prueba la función que devuelve el enlace sur de una casilla

Precondición

El espacio es un puntero a NULL.

Postcondición

La salida debe ser NO_ID.

4.26.2.44. void test2_space_get_up ( )

Prueba Prueba la función que devuelve el enlace superior de una casilla

Precondición

El espacio es un puntero a NULL.

Postcondición

La salida debe ser NO_ID.

4.26.2.45. void test2_space_get_west ( )

Prueba Prueba la función que devuelve el enlace oeste de una casilla

Precondición

El espacio es un puntero a NULL.

Postcondición

La salida debe ser NO_ID.

4.26.2.46. void test2_space_graphics_areEmpty ( )

Prueba Prueba la función que comprueba si los graficos de una casilla están vacios.

Precondición

El espacio no se ha inicializado

Postcondición

La salida esperada es TRUE.

4.26.2.47. void test2_space_remove_object ( )

Prueba Prueba la función que quita un objeto de una casilla

Precondición

Todavia no he ha añadido ningun objeto

Postcondición

La salida debe ser ERROR.

4.26.2.48. void test2_space_set_description ( )

Prueba Prueba la función para establecer la descripción de un espacio

Precondición

El espacio al que establecer la descripción es un puntero a NULL

Postcondición

La salida debe ser ERROR

4.26.2.49. void test2_space_set_down ( )

Prueba Prueba la función que establece el enlace inferior de un espacio

Precondición

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

Postcondición

La salida esperada es ERROR

4.26.2.50. void test2_space_set_east ( )

Prueba Prueba la función que establece el enlace este de un espacio

Precondición

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

Postcondición

La saida esperada es ERROR

4.26.2.51. void test2_space_set_graphics ( )

Prueba Prueba la función que establece los graficos de una casilla

Precondición

El espacio es un puntero a NULL

Postcondición

La salida esperada es un puntero a NULL

4.26.2.52. void test2_space_set_iluminated ( )

Prueba Prueba la funcion que pone el estado de la iluminacion en una casilla

Precondición

No se ha creado la casilla

Postcondición

La salida esperada es ERROR

4.26.2.53. void test2_space_set_long_description ( )

Prueba Prueba la función para establecer la descripción larga de un espacio

Precondición

El espacio al que establecer la descripción es un puntero a NULL

Postcondición

La salida debe ser ERROR

4.26.2.54. void test2_space_set_name ( )

Prueba Prueba la función para establecer el nombre de un espacio

Precondición

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

Postcondición

La salida debe ser ERROR

4.26.2.55. void test2_space_set_north ( )

Prueba Prueba la función que establece el enlace norte de un espacio

Precondición

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

Postcondición

La salida esperada es ERROR

4.26.2.56. void test2_space_set_south ( )

Prueba Prueba la función que establece el enlace sur de un espacio

Precondición

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

Postcondición

4.26.2.57. void test2_space_set_up ( )

Prueba Prueba la función que establece el enlace superior de un espacio

Precondición

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

Postcondición

La salida esperada es ERROR

4.26.2.58. void test2_space_set_west ( )

Prueba Prueba la función que establece el enlace oeste de un espacio

Precondición

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

Postcondición

La salida esperada es ERROR

4.26.2.59. void test3_space_add_object ( )

Prueba Prueba la función que añade un objeto a una casilla

Precondición

El id del objeto es -1 (NO_ID)

Postcondición

La salida debe ser ERROR.

4.26.2.60. void test3_space_contains_object ( )

Prueba Prueba la función que comprueba si una casilla contiene un objeto

Precondición

El espacio se ha creado pero no se ha añadido ningun objeto

Postcondición

La salida esperada es FALSE

4.26.2.61. void test3_space_create ( )

Prueba Prueba la función de creación de un espacio

Precondición

El identificador del espacio es NO_ID

Postcondición

La salida esperada es un puntero a NULL

4.26.2.62. void test3_space_get_description ( )

Prueba Prueba la función que devuelve la descripción de una casilla

Precondición

El espacio ha sido inicializado pero no se le ha establecido ninguna descripción

Postcondición

La salida debe ser una cadena vacila

4.26.2.63. void test3_space_get_down ( )

Prueba Prueba la función que devuelve el enlace inferior de una casilla

Precondición

No se ha establecido ningun enlace previamente

Postcondición

La salida debe ser NO_ID.

4.26.2.64. void test3_space_get_east ( )

Prueba Prueba la función que devuelve el enlace este de una casilla

Precondición

No se ha establecido ningun enlace previamente

Postcondición

La salida debe ser NO_ID.

4.26.2.65. void test3_space_get_long_description ( )

Prueba Prueba la función que devuelve la descripción larga de una casilla

Precondición

El espacio ha sido inicializado pero no se le ha establecido ninguna descripción

Postcondición

La salida debe ser una cadena vacila

4.26.2.66. void test3_space_get_name ( )

Prueba Prueba la función que devuelve el nombre de una casilla

Precondición

El espacio ha sido inicializado pero no se le ha establecido ningun nombre

Postcondición

La salida debe ser una cadena vacila

4.26.2.67. void test3_space_get_north ( )

Prueba Prueba la función que devuelve el enlace norte de una casilla

Precondición

No se ha establecido ningun enlace previamente

Postcondición

La salida debe ser NO_ID.

4.26.2.68. void test3_space_get_south ( )

Prueba Prueba la función que devuelve el enlace sur de una casilla

Precondición

No se ha establecido ningun enlace previamente

Postcondición

La salida debe ser NO_ID.

4.26.2.69. void test3_space_get_up ( )

Prueba Prueba la función que devuelve el enlace superior de una casilla

Precondición

No se ha establecido ningun enlace previamente

Postcondición

La salida debe ser NO_ID.

4.26.2.70. void test3_space_get_west ( )

Prueba Prueba la función que devuelve el enlace oeste de una casilla

Precondición

No se ha establecido ningun enlace previamente

Postcondición

La salida debe ser NO_ID.

4.26.2.71. void test3_space_graphics_areEmpty ( )

Prueba Prueba la función que comprueba si los graficos de una casilla están vacios.

Precondición

El espacio se ha creado pero no se le han establecido graficos

Postcondición

La salida esperada es TRUE

4.26.2.72. void test3_space_remove_object ( )

Prueba Prueba la función que quita un objeto de una casilla

Precondición

El espacio no se ha inicializado previamente

Postcondición

La salida debe ser ERROR.

4.26.2.73. void test3_space_set_description ( )

Prueba Prueba la función para establecer la descripción de un espacio

Precondición

El espacio es un puntero no NULL, pero la descripción a establecer es NULL

Postcondición

La salida debe ser ERROR

4.26.2.74. void test3_space_set_down ( )

Prueba Prueba la función que establece el enlace inferior de un espacio

Precondición

El id que se le pasa a la función es NO_ID

Postcondición

La salida esperada es ERROR

4.26.2.75. void test3_space_set_east ( )

Prueba Prueba la función que establece el enlace este de un espacio

Precondición

El id que se le pasa a la función es NO_ID

Postcondición

La salida esperada es ERROR

4.26.2.76. void test3_space_set_long_description ( )

Prueba Prueba la función para establecer la descripción larga de un espacio

Precondición

El espacio es un puntero no NULL, pero la descripción a establecer es NULL

Postcondición

La salida debe ser ERROR

4.26.2.77. void test3_space_set_name ( )

Prueba Prueba la función para establecer el nombre de un espacio

Precondición

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

Postcondición

La salida debe ser ERROR

4.26.2.78. void test3_space_set_north ( )

Prueba Prueba la función que establece el enlace norte de un espacio

Precondición

El id que se le pasa a la función es NO_ID

Postcondición

La salida esperada es ERROR

4.26.2.79. void test3_space_set_south ( )

Prueba Prueba la función que establece el enlace sur de un espacio

Precondición

El id que se le pasa a la función es NO_ID

Postcondición

La salida esperada es ERROR

4.26.2.80. void test3_space_set_up ( )

Prueba Prueba la función que establece el enlace superior de un espacio

Precondición

El id que se le pasa a la función es NO_ID

Postcondición

La salida esperada es ERROR

4.26.2.81. void test3_space_set_west ( )

Prueba Prueba la función que establece el enlace oeste de un espacio

Precondición

El id que se le pasa a la función es NO_ID

Postcondición

La salida esperada es ERROR

4.26.2.82. void test4_space_remove_object ( )

Prueba Prueba la función que quita un objeto de una casilla

Precondición

El objeto ya ha sido eliminado previamente

Postcondición

La salida debe ser ERROR.

4.27. Referencia del Archivo include/tests/test.h

Define macros útiles para las pruebas unitarias.

4.27.1. Descripción detallada

Define macros útiles para las pruebas unitarias.

Autor

Profesores PPROG

Versión

1.0

Fecha

11-01-2017

Copyright

GNU Public License

4.28. Referencia del Archivo include/types.h

Define tipos de datos comunes.

'defines'

- #define WORD_SIZE 1000

  Definicion del tamaño global de cadenas de textp.
- #define NO_ID-1

  Definicion de la constante NO_ID (Id no valido)
- #define GENERIC_ID 1

  Definicion de un id generico.

'typedefs'

- typedef long Id

  Definicion de Id.

Enumeraciones

- enum BOOL { **FALSE**, **TRUE** }

  Definicion de la enumeración BOOL.
- enum STATUS { **ERROR**, **OK**, **UNDEFINED** }

  Definicion de la enumeración STATUS.
- enum DIRECTION {
  NORTH, **SOUTH**, **EAST**, **WEST**,
  UP, **DOWN**, **UNKNOWN_DIRECTION** }

  Direciones en las que puede ir en jugador.

4.28.1. Descripción detallada

Define tipos de datos comunes.

Autor

Profesores PPROG

Versión

1.0

Fecha

13-01-2015

Copyright

GNU Public License

4.29. Referencia del Archivo src/command.c

Implementa el interpreatador de comandos Lee los comandos que introduce el usuario parseandolos al tipo de datos T_Command, para poder trabajar facilmente con estos comandos.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#include "command.h"
```

Funciones

- Command * Command_ini ()

  Crea e inicializa una estructura tipo Command.
- STATUS Command_destroy (Command *cmdManager)

  Libera memoria usada por una estructura Command.
- STATUS Command_set_cmd (Command *cmdManager, T_Command cmd)

  Guarda un comando en la estructura.
- STATUS Command_set_cmd_arg (Command *cmdManager, char *arg, int index)

  Guarda un parametro de un comando en la estructura en una posición dada.
- T_Command Command_get_cmd (Command *cmdManager)

  Devuelve el comando guardado en la estructura.
- char * Command_get_cmd_arg (Command *cmdManager, int index)

  Devuelve un parametro del comando guardado en la estructura en una posición dada.
- STATUS Command_clear (Command *cmdManager)

  Limpia los datos de una estructura Command.
- STATUS get_user_input (Command *cmdManager)

  Lee los comandos de introduce el usuario.

4.29.1. Descripción detallada

Implementa el interpreatador de comandos Lee los comandos que introduce el usuario parseandolos al tipo de datos T_Command, para poder trabajar facilmente con estos comandos.

Autor

Profesores PPROG
Javier Bernardo

Versión

2.0

Fecha

19-12-2014

Copyright

GNU Public License

4.29.2. Documentación de las funciones

4.29.2.1. STATUS Command_clear (Command * *cmdManager*)

Limpia los datos de una estructura Command.

Autor

Mihai Blidaru

Parámetros

cmdManager	Una estructura Command

Devuelve

OK si se ha hecho correctamente o ERROR en caso contrario

4.29.2.2. STATUS Command_destroy (Command * *cmdManager*)

Libera memoria usada por una estructura Command.

Autor

Mihai Blidaru

Parámetros

cmdManager	La estructura command que se tiene que liberar

Devuelve

OK si se ha realizado correctamente o ERROR en caso contrario

4.29.2.3. T_Command Command_get_cmd (Command * *cmdManager*)

Devuelve el comando guardado en la estructura.

Autor

Mihai Blidaru

Parámetros

cmdManager	La estructura Command de la cual leer los datos

Devuelve

El comando guardado

4.29.2.4. char* Command_get_cmd_arg (Command * *cmdManager,* int *index*)

Devuelve un parametro del comando guardado en la estructura en una posición dada.

Autor

Mihai Blidaru

Parámetros

cmdManager	La estructura Command de la cual leer los datos
index	El indice del parametro que se quiere leer

Devuelve

El parametro del comando guardado

4.29.2.5. Command* Command_ini ( )

Crea e inicializa una estructura tipo Command.

Autor

Mihai Blidaru

Devuelve

Una estructura command inicializada o NULL si hay algun error.

4.29.2.6. STATUS Command_set_cmd (Command * *cmdManager,* T_Command *cmd*)

Guarda un comando en la estructura.

Autor

Mihai Blidaru

Parámetros

cmdManager	Una estructura tipo command
cmd	El comando a guardar

Devuelve

OK si se ha guardado correctamente o error en caso contrario

4.29.2.7. STATUS Command_set_cmd_arg (Command * *cmdManager,* char * *arg,* int *index*)

Guarda un parametro de un comando en la estructura en una posición dada.

Autor

Mihai Blidaru

Parámetros

cmdManager	Una estructura tipo command
arg	El parametro a guardar
index	La posición donde se quiere guardar el parametro

Devuelve

OK si se ha guardado correctamente o error en caso contrario

4.29.2.8. STATUS get_user_input (Command * *cmdManager*)

Lee los comandos de introduce el usuario.

Lee por teclado el comando que introduce el usuario y comprueba que esta en la lista de los comandos permitidos. Guarda en la estructura Command los datos introducidos: en cmd guarda el tipo de comando y en arg guarda el argumento que tiene un comando

Autor

Profesores PPROG
Javier Bernardo

Parámetros

cmdManager	Una estructura tipo Command donde guardar los datos leidos por teclado

Devuelve

OK si ha leido bien el comando o ERROR en caso contrario

4.30. Referencia del Archivo src/command_test.c

Programa para probar el modulo Command. Progama para probar la correcta funcionalidad del nuevo TAD Command.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "tests/test.h"
#include "tests/command_test.h"
#include "types.h"
#include "command.h"
```

Funciones

- void test1_command_ini ()
- void test1_command_destroy ()
- void test2_command_destroy ()
- void test1_command_set_cmd ()
- void test2_command_set_cmd ()
- void test3_command_set_cmd ()
- void test1_command_set_cmd_arg ()
- void test2_command_set_cmd_arg ()
- void test3_command_set_cmd_arg ()
- void test4_command_set_cmd_arg ()
- void test1_command_get_cmd ()
- void test2_command_get_cmd ()
- void test3_command_get_cmd ()
- void test1_command_get_cmd_arg ()
- void test2_command_get_cmd_arg ()
- void test3_command_get_cmd_arg ()
- void test1_command_clear ()
- void test2_command_clear ()

4.30.1. Descripción detallada

Programa para probar el modulo Command. Progama para probar la correcta funcionalidad del nuevo TAD Command.

Autor

Mihai Blidaru

Versión

1.0

Fecha

01-04-2017

4.30.2. Documentación de las funciones

4.30.2.1. void test1_command_clear ( )

Prueba Prueba la función que limpia los campos de un commando

Precondición

El gestor de comandos ha sido inicializado previamente y se ha guardado un comando dentro

Postcondición

La salida esperada es NO_CMD y una cadena vacia

4.30.2.2. void test1_command_destroy ( )

Prueba Pruba la función que libera memoria usada por un gestor de comandos

Precondición

El gestor de comandos ha sido previamente inicializado

Postcondición

La salida esperada es OK

4.30.2.3. void test1_command_get_cmd ( )

Prueba Prueba la función que devuelve el tipo de comando guardado en un gestor de comandos

Precondición

El gestor de comandos ha sido inicializado previamente y se le ha asignado el comando GO

Postcondición

La salida esperada es GO, el comando previamente asignado

4.30.2.4. void test1_command_get_cmd_arg ( )

Prueba Prueba la función que devuelve un argumento de un comando guardado en un gestor de comandos

Precondición

El gestor de comandos ha sido inicializado previamente y se le ha asignado un argumento "test"

Postcondición

La salida esperada es "test", el argumento añadido previamente

4.30.2.5. void test1_command_ini ( )

Número maximo de tests

4.30.2.6. void test1_command_set_cmd ( )

Prueba Prueba la función que guarda el tipo de comando en un gestor de comandos

Precondición

El gestor de comandos ha sido inicializado previamente

Postcondición

La salida esperada es OK

4.30.2.7. void test1_command_set_cmd_arg ( )

Prueba Prueba la función que guarda un argumento de un comando en un gestor de comandos

Precondición

El gestor de comandos ha sido inicializado previamente

Postcondición

La salida esperada es OK

4.30.2.8. void test2_command_clear ( )

Prueba Prueba la función que limpia los campos de un commando

Precondición

El gestor de comandos no ha sido inicializado previamente

Postcondición

La salida esperada es ERROR

4.30.2.9. void test2_command_destroy ( )

Prueba Pruba la función que libera memoria usada por un gestor de comandos

Precondición

El gestor de comandos no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.30.2.10. void test2_command_get_cmd ( )

Prueba Prueba la función que devuelve el tipo de comando guardado en un gestor de comandos

Precondición

El gestor de comandos no ha sido inicializado previamente

Postcondición

La salida esperada es UNKNOWN, comando desconocido

4.30.2.11. void test2_command_get_cmd_arg ( )

Prueba Prueba la función que devuelve un argumento de un comando guardado en un gestor de comandos

Precondición

El gestor de comandos no ha sido inicializado previamente

Postcondición

La salida esperada es NULL

4.30.2.12. void test2_command_set_cmd ( )

Prueba Prueba la función que guarda el tipo de comando en un gestor de comandos

Precondición

El gestor de comandos ha sido inicializado previamente pero el comando no es valido

Postcondición

La salida esperada es ERROR

4.30.2.13. void test2_command_set_cmd_arg ( )

Prueba Prueba la función que guarda un argumento de un comando en un gestor de comandos

Precondición

La posición en la que se quiere guardar es invalida

Postcondición

La salida esperada es ERROR

4.30.2.14. void test3_command_get_cmd ( )

Prueba Prueba la función que devuelve el tipo de comando guardado en un gestor de comandos

Precondición

El gestor de comandos ha sido inicializado previamente pero no se le ha asignado un comando

Postcondición

La salida esperada es NO_CMD

4.30.2.15. void test3_command_get_cmd_arg ( )

Prueba Prueba la función que devuelve un argumento de un comando guardado en un gestor de comandos

Precondición

El gestor de comandos ha sido inicializado previamente pero no se le ha asignado un argumento

Postcondición

La salida esperada es una cadena vacia

4.30.2.16. void test3_command_set_cmd ( )

Prueba Prueba la función que guarda el tipo de comando en un gestor de comandos

Precondición

El gestor de comandos no ha sido inicializado previamente

Postcondición

La salida esperada es ERROR

4.30.2.17. void test3_command_set_cmd_arg ( )

Prueba Prueba la función que guarda un argumento de un comando en un gestor de comandos

Precondición

El gestor de comandos no ha sido inicializado previamente

Postcondición

La salida esperada es ERROR

4.30.2.18. void test4_command_set_cmd_arg ( )

Prueba Prueba la función que guarda un argumento de un comando en un gestor de comandos

Precondición

El argumento es un puntero a NULL

Postcondición

La salida esperada es ERROR

4.31. Referencia del Archivo src/dialogue.c

Implementa Los dialogos del juego.

```
#include <string.h>
#include <stdlib.h>
#include "types.h"
#include "command.h"
#include "dialogue.h"
#include "game.h"
#include "space.h"
#include "link.h"
#include "game_management.h"
#include "object.h"
#include "game_rules.h"
```

Funciones

- Dialogue * dialogue_ini ()

  Inicializa un modulo dialogo.
- void dialogue_destroy (Dialogue *d)

  Libera la memoria usasa por un dialogo.
- char * dialogue_get_text (Dialogue *d)

  Devuelve el texto del dialogo.
- STATUS dialogue_go (Dialogue *d, DIRECTION direction, Space *space, STATUS status, char *dir_name, Link *link)

  Contruye el texto del dialogo para el comando GO.
- STATUS dialogue_unknown (Dialogue *d)

  Contruye el texto del dialogo para comandos desconocidos.
- STATUS dialogue_dir (Dialogue *d)

  Contruye el texto del dialogo para el comando DIR.
- STATUS dialogue_take (Dialogue *d, Object *object, char *name, TAKE_STATUS status)

  Contruye el texto del dialogo para el comando TAKE.
- STATUS dialogue_leave (Dialogue *d, Object *object, char *name, LEAVE_STATUS status)

  Contruye el texto del dialogo para el comando LEAVE.
- STATUS dialogue_save (Dialogue *d, char *name, DIALOGUE_SAVE_STATUS status)

  Contruye el texto del dialogo para el comando SAVE.
- STATUS dialogue_load (Dialogue *d, char *name, DIALOGUE_LOAD_STATUS status)

  Contruye el texto del dialogo para el comando LOAD.
- STATUS dialogue_attack (Dialogue *d, ATTACK_STATUS status)

  Contruye el texto del dialogo para el comando ATTACK.
- STATUS dialogue_inspect (Dialogue *d, Object *object, Space *space, char *name, INSPECT_STATUS status)

  Contruye el texto del dialogo para el comando TAKE.
- STATUS dialogue_open (Dialogue *d, char *objName, char *linkName, OPEN_STATUS status)

  Contruye el texto del dialogo para el comando TAKE.
- STATUS dialogue_turn_on (Dialogue *d, Object *object, char *name, TURN_STATUS status)

  Contruye el texto del dialogo para el comando TURN_ON.
- STATUS dialogue_turn_off (Dialogue *d, Object *object, char *name, TURN_STATUS status)

  Contruye el texto del dialogo para el comando TURN_ON.
- STATUS dialogue_help (Dialogue *d)

  Contruye el texto del dialogo para el comando HELP.
- STATUS dialogue_game_rule (Dialogue *d, int rule)

  Contruye el texto del dialogo para las reglas dej juego.

4.31.1. Descripción detallada

Implementa Los dialogos del juego.

Autor

Mihai Blidaru
Sandra Benítez

Versión

2.0

Fecha

19-12-2014

Copyright

GNU Public License

4.31.2. Documentación de las funciones

4.31.2.1. STATUS dialogue_attack (Dialogue * d, ATTACK_STATUS status)

Contruye el texto del dialogo para el comando ATTACK.

Parámetros

d	Una estructura dialogo
status	El codigo de error del comando

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.31.2.2. void dialogue_destroy (Dialogue * d)

Libera la memoria usasa por un dialogo.

Autor

Javier Bernardo

Parámetros

d	El dialogo que se quiere destruir

4.31.2.3. STATUS dialogue_dir (Dialogue * d)

Contruye el texto del dialogo para el comando DIR.

Parámetros

d	Una estructura dialogo

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.31.2.4. STATUS dialogue_game_rule (Dialogue * *d,* int *rule*)

Contruye el texto del dialogo para las reglas dej juego.

Parámetros

d	Una estructura dialogo
rule	Codigo de la regla que se ha ejecutado

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.31.2.5. char* dialogue_get_text (Dialogue * *d*)

Devuelve el texto del dialogo.

Parámetros

d	El dialogo del que se quiere leer el texto

Devuelve

El texto del dialogo

4.31.2.6. STATUS dialogue_go (Dialogue * *d,* DIRECTION *direction,* Space * *space,* STATUS *status,* char * *dir_name,* Link * *link*)

Contruye el texto del dialogo para el comando GO.

Parámetros

d	Una estructura dialogo
direction	La direccion en la que se quiere mover
space	La casilla destino
status	Si se ha llevado a cabo con exito o no
dir_name	la dirrecion introducida por teclado
link	El enlace por el que se quiere pasar

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.31.2.7. STATUS dialogue_help (Dialogue * *d*)

Contruye el texto del dialogo para el comando HELP.

Parámetros

d	Una estructura dialogo

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.31.2.8. Dialogue* dialogue_ini ( )

Inicializa un modulo dialogo.

Autor

Javier Bernardo

Devuelve

Un dialogo inicializado

4.31.2.9. STATUS dialogue_inspect (Dialogue * *d*, Object * *object*, Space * *space*, char * *name*, INSPECT_STATUS *status*)

Contruye el texto del dialogo para el comando TAKE.

Parámetros

d	Una estructura dialogo
object	El objeto que se quiere inspeccionar si existe
space	El espacio que se quiere inspeccionar si existe
name	El nombre introducido por teclado
status	El codigo de error del comando

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.31.2.10. STATUS dialogue_leave (Dialogue * *d*, Object * *object*, char * *name*, LEAVE_STATUS *status*)

Contruye el texto del dialogo para el comando LEAVE.

Parámetros

d	Una estructura dialogo
object	El objeto que se quiere dejar si existe
name	El nombre introducido por teclado
status	El codigo de error del comando

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.31.2.11. STATUS dialogue_load (Dialogue * *d*, char * *name*, DIALOGUE_LOAD_STATUS *status*)

Contruye el texto del dialogo para el comando LOAD.

Parámetros

d	Una estructura dialogo
name	El nombre del fichero desde el cual se quiere cargar
status	El codigo de error del comando

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.31.2.12. STATUS dialogue_open (Dialogue * *d*, char * *objName*, char * *linkName*, OPEN_STATUS *status*)

Contruye el texto del dialogo para el comando TAKE.

Parámetros

d	Una estructura dialogo
objName	Nombre del objeto con el que se quiere abrir
linkName	Nombre del link que se quiere abris
status	Codigo de error de la operacion

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.31.2.13. STATUS dialogue_save (Dialogue * *d*, char * *name*, DIALOGUE_SAVE_STATUS *status*)

Contruye el texto del dialogo para el comando SAVE.

Parámetros

d	Una estructura dialogo
name	El nombre del ficheri donde se quiere guardar
status	El codigo de error del comando

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.31.2.14. STATUS dialogue_take (Dialogue * *d*, Object * *object*, char * *name*, TAKE_STATUS *status*)

Contruye el texto del dialogo para el comando TAKE.

Parámetros

d	Una estructura dialogo
object	El objeto que se quiere llevar si existe
name	El nombre introducido por teclado
status	El codigo de error del comando

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.31.2.15. STATUS dialogue_turn_off (Dialogue * *d*, Object * *object*, char * *name*, TURN_STATUS *status*)

Contruye el texto del dialogo para el comando TURN_ON.

Parámetros

d	Una estructura dialogo
object	El objeto que se quiere apagar
name	El nombre introducido por teclado
status	El codigo de error del comando

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.31.2.16. STATUS dialogue_turn_on (Dialogue * *d*, Object * *object*, char * *name*, TURN_STATUS *status*)

Contruye el texto del dialogo para el comando TURN_ON.

Parámetros

d	Una estructura dialogo
object	El objeto que se quiere encender
name	El nombre introducido por teclado
status	El codigo de error del comando

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.31.2.17. STATUS dialogue_unknown (Dialogue * *d*)

Contruye el texto del dialogo para comandos desconocidos.

Parámetros

d	Una estructura dialogo

Devuelve

OK si el texto se ha asignado con exito o ERROR en caso contrario

4.32. Referencia del Archivo src/dialogue_test.c

Programa para probar el modulo dialogue. Progama para probar la correcta funcionalidad del nuevo TAD dialogue.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "game.h"
#include "space.h"
#include "dialogue.h"
#include "tests/dialogue_test.h"
#include "tests/test.h"
```

Funciones

- void test1_dialogue_ini ()

- void test1_dialogue_get_text ()
- void test2_dialogue_get_text ()
- void test1_dialogue_go ()
- void test2_dialogue_go ()
- void test1_dialogue_unknown ()
- void test2_dialogue_unknown ()
- void test1_dialogue_dir ()
- void test2_dialogue_dir ()
- void test1_dialogue_take ()
- void test2_dialogue_take ()
- void test1_dialogue_leave ()
- void test2_dialogue_leave ()
- void test1_dialogue_save ()
- void test2_dialogue_save ()
- void test1_dialogue_load ()
- void test2_dialogue_load ()
- void test1_dialogue_attack ()
- void test2_dialogue_attack ()
- void test1_dialogue_inspect ()
- void test2_dialogue_inspect ()
- void test1_dialogue_open ()
- void test2_dialogue_open ()
- void test1_dialogue_turn_on ()
- void test2_dialogue_turn_on ()
- void test1_dialogue_turn_off ()
- void test2_dialogue_turn_off ()
- void test1_dialogue_help ()
- void test2_dialogue_help ()
- void test1_dialogue_game_rule ()
- void test2_dialogue_game_rule ()

4.32.1. Descripción detallada

Programa para probar el modulo dialogue. Progama para probar la correcta funcionalidad del nuevo TAD dialogue.

Autor

Javier Bernardo

Versión

1.0

Fecha

24-04-2017

4.32.2. Documentación de las funciones

4.32.2.1. void test1_dialogue_attack ( )

Prueba Prueba la funcion que construye el dialogo para el comando ATTACK

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

4.32.2.2. void test1_dialogue_dir ( )

Prueba Prueba la funcion que construye el dialogo para el comando DIR

Precondición

El dialogo ha sido inicializado

Postcondición

La salida esperada es OK

4.32.2.3. void test1_dialogue_game_rule ( )

Prueba Prueba la funcion que construye el dialogo para las reglas del juego

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

4.32.2.4. void test1_dialogue_get_text ( )

Prueba Prueba la funcion que devuelve el texto del dialogo

Precondición

El dialogo se ha inicializado antes

Postcondición

La salida es un texto no nulo

4.32.2.5. void test1_dialogue_go ( )

Prueba Prueba la funcion que construye el dialogo para el comando GO

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

4.32.2.6. void test1_dialogue_help ( )

Prueba Prueba la funcion que construye el dialogo para el comando HELP

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

4.32.2.7. void test1_dialogue_ini ( )

Número maximo de tests

4.32.2.8. void test1_dialogue_inspect ( )

Prueba Prueba la funcion que construye el dialogo para el comando INSPECT

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

4.32.2.9. void test1_dialogue_leave ( )

Prueba Prueba la funcion que construye el dialogo para el comando LEAVE

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

4.32.2.10. void test1_dialogue_load ( )

Prueba Prueba la funcion que construye el dialogo para el comando LOAD

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

4.32.2.11. void test1_dialogue_open ( )

Prueba Prueba la funcion que construye el dialogo para el comando OPEN

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

4.32.2.12. void test1_dialogue_save ( )

Prueba Prueba la funcion que construye el dialogo para el comando SAVE

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

4.32.2.13. void test1_dialogue_take ( )

Prueba Prueba la funcion que construye el dialogo para el comando TAKE

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

4.32.2.14. void test1_dialogue_turn_off ( )

Prueba Prueba la funcion que construye el dialogo para el comando TURN_OFF

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

4.32.2.15. void test1_dialogue_turn_on ( )

Prueba Prueba la funcion que construye el dialogo para el comando TURN_ON

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

4.32.2.16. void test1_dialogue_unknown ( )

Prueba Prueba la funcion que construye el dialogo para comandos desconocidos

Precondición

Todos los parametros han sido inicializados

Postcondición

La salida esperada es OK

4.32.2.17. void test2_dialogue_attack ( )

Prueba Prueba la funcion que construye el dialogo para el comando ATTACK

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.32.2.18. void test2_dialogue_dir ( )

Prueba Prueba la funcion que construye el dialogo para el comando DIR

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.32.2.19. void test2_dialogue_game_rule ( )

Prueba Prueba la funcion que construye el dialogo para las reglas del juego

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.32.2.20. void test2_dialogue_get_text ( )

Prueba Prueba la funcion que devuelve el texto del dialogo

Precondición

El dialogo no se ha inicializado antes

Postcondición

La salida esperada es NULL

4.32.2.21. void test2_dialogue_go ( )

Prueba Prueba la funcion que construye el dialogo para el comando GO

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.32.2.22. void test2_dialogue_help ( )

Prueba Prueba la funcion que construye el dialogo para el comando HELP

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.32.2.23. void test2_dialogue_inspect ( )

Prueba Prueba la funcion que construye el dialogo para el comando INSPECT

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.32.2.24. void test2_dialogue_leave ( )

Prueba Prueba la funcion que construye el dialogo para el comando LEAVE

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.32.2.25. void test2_dialogue_load ( )

Prueba Prueba la funcion que construye el dialogo para el comando LOAD

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.32.2.26. void test2_dialogue_open ( )

Prueba Prueba la funcion que construye el dialogo para el comando OPEN

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.32.2.27. void test2_dialogue_save ( )

Prueba Prueba la funcion que construye el dialogo para el comando SAVE

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.32.2.28. void test2_dialogue_take ( )

Prueba Prueba la funcion que construye el dialogo para el comando TAKE

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.32.2.29. void test2_dialogue_turn_off ( )

Prueba Prueba la funcion que construye el dialogo para el comando TURN_OFF

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.32.2.30. void test2_dialogue_turn_on ( )

Prueba Prueba la funcion que construye el dialogo para el comando TURN_ON

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.32.2.31. void test2_dialogue_unknown ( )

Prueba Prueba la funcion que construye el dialogo para comandos desconocidos

Precondición

El dialogo no ha sido inicializado

Postcondición

La salida esperada es ERROR

4.33. Referencia del Archivo src/die.c

Implememtacion del dado del juego.

```
#include <stdlib.h>
#include <time.h>
#include <stdio.h>
#include "die.h"
```

Funciones

- Die * die_create (Id die_id)
  Crea un nuevo dado.
- STATUS die_destroy (Die *die)
  Destruir un nuevo dado.
- unsigned short die_roll (Die *die)
  Lanza el dado.
- int die_get_number (Die *die)
  Obtiene el ultimo valor del dado.
- int die_print (Die *die)
  Imprimir el dado.
- STATUS die_set_faces (Die *die, int faces)
  Coloca un numero de caras al dado.
- int die_get_faces (Die *die)
  Coloca un numero de caras al dado.

4.33.1. Descripción detallada

Implememtacion del dado del juego.

Autor

Javier Bernando

Versión

1.0

Fecha

20-02-2017

4.33.2. Documentación de las funciones

4.33.2.1. Die* die_create (Id *die_id*)

Crea un nuevo dado.

Autor

Javier Bernardo

Parámetros

die_id	Id que define el dado

Devuelve

dado creado

4.33.2.2. STATUS die_destroy (Die * *die*)

Destruir un nuevo dado.

Autor

Javier Bernardo

Parámetros

die	Die que quieres destruir

Devuelve

OK si se elimina, ERROR en caso contrario

4.33.2.3. int die_get_faces (Die * *die*)

Coloca un numero de caras al dado.

Autor

Javier Bernardo

Parámetros

die	Die que quieres lanzar

Devuelve

El numero de caras del dado

4.33.2.4. int die_get_number (Die * *die*)

Obtiene el ultimo valor del dado.

Autor

Javier Bernardo

Parámetros

die	Die del que quieres saber su valor

Devuelve

Devuelve el valor en int de la estructura numero del dado

4.33.2.5. int die_print (Die * *die*)

Imprimir el dado.

Autor

Javier Bernardo

Parámetros

die	Die que quieres imprimir

Devuelve

Impresion por pantalla del dado y su valor

4.33.2.6. unsigned short die_roll (Die * *die*)

Lanza el dado.

Autor

Javier Bernardo

Parámetros

die	Die que quieres lanzar

Devuelve

Devuelve el valor del dado lanzado con valores entre 1 y 6

4.33.2.7. STATUS die_set_faces (Die * *die,* int *faces*)

Coloca un numero de caras al dado.

Autor

Javier Bernardo

Parámetros

die	Die que quieres lanzar
faces	Numero de caras del dado

Devuelve

OK si lo hace bien o ERROR si da algun problema

4.34. Referencia del Archivo src/die_test.c

Implementacion del dado en el juego Progama para probar la correcta funcionalidad del nuevo TAD Die.

```
#include <stdio.h>
#include <stdlib.h>
#include "die.h"
```

Funciones

- int main ()

  Funcion main del die_test.c.

4.34.1. Descripción detallada

Implementacion del dado en el juego Progama para probar la correcta funcionalidad del nuevo TAD Die.

Autor

Javier Bernando

Versión

1.0

Fecha

11-03-2017

4.35. Referencia del Archivo src/game.c

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

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include "die.h"
#include "game.h"
#include "command.h"
#include "space.h"
#include "player.h"
#include "object.h"
#include "dialogue.h"
#include "game_management.h"
```

Funciones

- Game * game_create ()

  Inicializa los datos del juego Esta función inicializa los espacios, crea e inicializa el jugador, los objetos y el dado.

- STATUS game_destroy (Game *game)

  Destruye la estructura game. Se encarga de liberar la memoria reservada por los elementos del juego spaces, player, objects y die.

- STATUS game_add_space (Game *game, Space *space)

  Añade un espacio al juego.
- STATUS game_add_object (Game *game, Object *object)

  Añade un objeto al juego.
- STATUS game_add_link (Game *game, Link *link)

  Añade un objeto al juego.
- STATUS game_add_player (Game *game, Player *player)

  Añade un jugador al juego. Si el jugador ya está inicializado, lo sustituye por otro.
- Space * game_get_space (Game *game, Id space_id)

  Devuelve un puntero al space que tiene el Id igual al Id pasado como parametro.
- Space * game_get_space_at (Game *game, int index)

  Develve el espacop una posición dada.
- Object * game_get_object_at (Game *game, int index)

  Develve el objeto una posición dada.
- Link * game_get_link (Game *game, Id link_id)

  Devuelve un puntero al link que tiene el Id igual al Id pasado como parametro.
- Link * game_get_link_at (Game *game, int index)

  Develve el link en una posición dada.
- Die * game_get_die (Game *game)

  Devuelve el dado del juego.
- Id game_get_player_location (Game *game)

  Devuelve la localización del jugador.
- Player * game_get_player (Game *game)

  Devuelve el jugador del juego.
- Dialogue * game_get_dialogue (Game *game)

  Devuelve un puntero al modulo dialogo del juego.
- Id game_get_object_location (Game *game, Object *object)

  Obtiene la localización de un objeto.
- char * game_get_obj_list_as_str (Game *game, Space *space)

  Devuelve la lista de objetos de una casilla como cadena.
- Space * game_get_last_inspected_space (Game *game)

  Devuelve el último espacio inspeccionado.
- Object * game_get_last_inspected_object (Game *game)

  Devuelve el ultimo objeto inspeccionado.
- void game_print_data (Game *game)

  Imprime la información del juego.
- BOOL game_is_over (Game *game)

  Devuelve si el juego a acabado o no.
- STATUS game_update (Game *game, Command *command)

  Ejecuta una de las funciones callback en funcion del comando recibido.

4.35.1. Descripción detallada

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

Autor

Profesores PPROG
Javier Bernardo
Mihai Blidaru
Laura Bernal
Sandra Benitez

Versión

2.0

Fecha

13-01-2015

Copyright

GNU Public License

4.35.2. Documentación de las funciones

4.35.2.1. STATUS game_add_link (Game * *game,* Link * *link*)

Añade un objeto al juego.

Autor

Mihai Blidaru

Parámetros

game	El puntero al juego
link	Un puntero al objeto

Devuelve

OK si se ha añaido correctamente, sino ERROR

4.35.2.2. STATUS game_add_object (Game * *game,* Object * *object*)

Añade un objeto al juego.

Autor

Javier Bernardo

Parámetros

game	El puntero al juego
object	Un puntero al objeto

Devuelve

OK si se ha añadido correctamente, sino ERROR

4.35.2.3. STATUS game_add_player (Game * *game,* Player * *player*)

Añade un jugador al juego. Si el jugador ya está inicializado, lo sustituye por otro.

Autor

Mihai Blidaru

Parámetros

game	El puntero al juego
player	Un puntero al jugador

Devuelve

OK si se ha añaido correctamente, sino ERROR

4.35.2.4. STATUS game_add_space (Game * *game,* Space * *space*)

Añade un espacio al juego.

Autor

Profesores PPROG

Parámetros

game	El puntero al juego
space	Un puntero al espacio

Devuelve

OK si se ha añadido correctamente, sino ERROR

4.35.2.5. Game* game_create ( )

Inicializa los datos del juego Esta función inicializa los espacios, crea e inicializa el jugador, los objetos y el dado.

Autor

Profesores PPROG
Mihai Blidaru

Devuelve

OK si todo ha ido bien. En caso contrario ERROR

4.35.2.6. STATUS game_destroy (Game * *game*)

Destruye la estructura game. Se encarga de liberar la memoria reservada por los elementos del juego spaces, player, objects y die.

Autor

Profesores PPROG
Mihai Blidaru

Parámetros

game	Puntero a la estructura del juego.

Devuelve

OK si se ha podido hacer todo correctamente. ERROR en caso contrario.

4.35.2.7. Dialogue* game_get_dialogue (Game * *game*)

Devuelve un puntero al modulo dialogo del juego.

Autor

Mihai Blidaru

Parámetros

game	Puntero a la estructura del juego.

Devuelve

Un puntero al modulo dialogo del juego o NULL si hay algun error

4.35.2.8. Die* game_get_die (Game * *game*)

Devuelve el dado del juego.

Autor

Mihai Blidaru

Parámetros

game	Puntero a la estructura del juego

Devuelve

Dado del juego.

4.35.2.9. Object* game_get_last_inspected_object (Game * *game*)

Devuelve el ultimo objeto inspeccionado.

Autor

Laura Bernal y Sandra Benitez

Parámetros

game	Un puntero a la estructura game

Devuelve

El ultimo objeto inspeccionado

4.35.2.10. Space* game_get_last_inspected_space (Game * *game*)

Devuelve el último espacio inspeccionado.

Autor

Laura Bernal y Sandra Benitez

Parámetros

game	Un puntero a la estructura game

Devuelve

El ultimo espacio inspeccionado

4.35.2.11. Link* game_get_link (Game * *game,* Id *link_id*)

Devuelve un puntero al link que tiene el Id igual al Id pasado como parametro.

Autor

Profesores PPROG

Parámetros

game	Puntero a la estructura del juego
link_id	Id del link

Devuelve

Puntero al link que tiene el Id igual al Id pasado como parametro. NULL si no existe.

4.35.2.12. Link* game_get_link_at (Game * *game,* int *index*)

Develve el link en una posición dada.

Autor

Mihai Blidaru

Parámetros

game	Puntero a la estructura del juego
index	La posición de la cual se quiere encontrar el link

Devuelve

El espacio en la posición dada.Null si la posición es invalida.

4.35.2.13. char* game_get_obj_list_as_str (Game * *game,* Space * *space*)

Devuelve la lista de objetos de una casilla como cadena.

Autor

Mihai Blidaru

Parámetros

game	Un puntero a la estructura game
space	Un puntero a la casilla desde la cual leer los objetos

Devuelve

Una cadena con la lista de los objetos. Quien use esta funciónse tiene que encargar de liberar la memoria usada.

4.35.2.14. Object* game_get_object_at (Game * *game,* int *index*)

Develve el objeto una posición dada.

Autor

Mihai Blidaru

Parámetros

game	Puntero a la estructura del juego
index	La posición de la cual se quiere encontrar el objeto

Devuelve

El objeto en la posición dada. Null si la posición es invalida.

4.35.2.15. Id game_get_object_location (Game * *game,* Object * *object*)

Obtiene la localización de un objeto.

Autor

Javier Bernardo

Parámetros

game	Puntero a la estructura de el juego.
object	El objeto del que quieres obtener la posicion.

Devuelve

La posicion del objeto o NO_ID en caso de error

4.35.2.16. Player* game_get_player (Game * *game*)

Devuelve el jugador del juego.

Autor

Mihai Blidaru

Parámetros

game	Puntero a la estructura del juego

Devuelve

Jugador del juego.

4.35.2.17. Id game_get_player_location (Game * *game*)

Devuelve la localización del jugador.

Autor

Mihai Blidaru

Parámetros

game	Puntero a la estructura de l juego.

Devuelve

La localización del jugador. NO_ID si hay algun error.

4.35.2.18. Space* game_get_space (Game * *game,* Id *space_id*)

Devuelve un puntero al space que tiene el Id igual al Id pasado como parametro.

Autor

Profesores PPROG

Parámetros

game	Puntero a la estructura del juego
space_id	Id del espacio

Devuelve

Puntero al espacio que tiene el Id igual al Id pasado como parametro. NULL si no existe.

4.35.2.19. Space* game_get_space_at (Game * *game,* int *index*)

Develve el espacop una posición dada.

Autor

Mihai Blidaru

Parámetros

game	Puntero a la estructura del juego
index	La posición de la cual se quiere encontrar el espacio

Devuelve

El espacio en la posición dada.Null si la posición es invalida.

4.35.2.20. BOOL game_is_over (Game * *game*)

Devuelve si el juego a acabado o no.

En esta iteración la función devuelve siempre FALSE ya que no hay suficiente funcionalidad en el juego como para poder decidir si el juego ha acabado o no.

Autor

Profesores PPROG

Parámetros

game	Puntero a la estructura del juego.

Devuelve

FALSE

4.35.2.21. void game_print_data (Game * *game*)

Imprime la información del juego.

Autor

Profesores PPROG
Javier Bernardo

Parámetros

game	Puntero a la estructura del juego.

Devuelve

Nada

4.35.2.22. STATUS game_update (Game * *game,* Command * *command*)

Ejecuta una de las funciones callback en funcion del comando recibido.

Autor

Profesores PPROG

Parámetros

game	Puntero a la estructura del juego
command	Comando que se tiene que ejecutar

Devuelve

4.36. Referencia del Archivo src/game_loop.c

It defines the game loop.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <limits.h>
#include "graphic_engine.h"
#include "game.h"
#include "game_management.h"
#include "command.h"
#include "game_rules.h"
```

4.36.1. Descripción detallada

It defines the game loop.

Autor

Mihai Blidaru
Javier Bernardo

Versión

1.0

Fecha

31-01-2017

Copyright

GNU Public License

4.37. Referencia del Archivo src/game_management.c

Este modulo se encarga de cargar los datos del juego. Carga las casillas los enlaces, los objetos y los datos del jugador. Tambié incluye la función de guardado.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "game.h"
#include "space.h"
#include "game_management.h"
```

Funciones

- STATUS game_management_start_from_file (Game *game, char *spacesFile, char *objectsFile, char *linksfile, char *playersFile)

  Carga los datos del juego desde un archivo.
- SAVE_STATUS game_management_save (Game *game, char *filename)

  Guarda los datos de la partida con el mismo formato que los ficheros de carga.
- STATUS game_management_load (Game *game, char *filename)

  Carga los datos del juego desde un archivo.

4.37.1. Descripción detallada

Este modulo se encarga de cargar los datos del juego. Carga las casillas los enlaces, los objetos y los datos del jugador. Tambié incluye la función de guardado.

Autor

Javier Bernardo
Mihai Blidaru

Versión

2.2

Fecha

29-04-2017

Copyright

GNU Public License

4.37.2. Documentación de las funciones

4.37.2.1. STATUS game_management_load (Game * *game,* char * *filename*)

Carga los datos del juego desde un archivo.

Parámetros

game	Un puntero al juego donde cargar los datos
filename	nombre del archivo desde donde cargar los datos

Devuelve

OK si se ha cargado correctamente o ERROR en caso contrario

4.37.2.2. SAVE_STATUS game_management_save (Game * *game,* char * *filename*)

Guarda los datos de la partida con el mismo formato que los ficheros de carga.

Parámetros

game	Un puntero al game desde donde se quieren guardar los datos
filename	Archivo donde guardar los datos

Devuelve

OK si ha guardado correctamente o ERROR en caso contrario

4.37.2.3. STATUS game_management_start_from_file (Game * *game*, char * *spacesFile*, char * *objectsFile*, char * *linksfile*, char * *playersFile*)

Carga los datos del juego desde un archivo.

Autor

Javier Bernardo

Parámetros

game	Puntero a la estructura del juego.
spacesFile	Nombre del archivo desde donde hay que cargar los espacios
objectsFile	Nombre del archivo desde donde hay que cargar los objetos
linksfile	Nombre del archivo desde donde hay que cargar los links
playersFile	Nombre del archivo desde donde hay que cargar los datos del jugador

Devuelve

OK si todo ha ido bien. ERROR en caso contrario.

4.38. Referencia del Archivo src/game_management_test.c

Programa para probar el modulo game_management. Progama para probar la correcta funcionalidad del nuevo TAD Game_management.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "game.h"
#include "space.h"
#include "game_management.h"
#include "tests/game_management_test.h"
#include "tests/test.h"
```

Funciones

- void test1_game_management_start_from_file ()
- void test2_game_management_start_from_file ()
- void test1_game_management_save ()
- void test2_game_management_save ()
- void test1_game_management_load ()
- void test2_game_management_load ()

4.38.1. Descripción detallada

Programa para probar el modulo game_management. Progama para probar la correcta funcionalidad del nuevo TAD Game_management.

Autor

Sandra Benítez
Laura Bernal

Versión

1.0

Fecha

24-04-2017

4.38.2. Documentación de las funciones

4.38.2.1. void test1_game_management_load ( )

Prueba Prueba la función que carga los datos del juego desde un archivo

Precondición

El nombre del fichero desde donde cargar los datos

Postcondición

La salida esperada es OK

4.38.2.2. void test1_game_management_save ( )

Prueba Prueba que la función guarde los datos de la partida con el mismo formato que los ficheros de carga

Precondición

El juego se ha creado correctamente y el nombre del archivo es valido

Postcondición

La salida esperada es OK

4.38.2.3. void test1_game_management_start_from_file ( )

Número maximo de tests

4.38.2.4. void test2_game_management_load ( )

Prueba Prueba la función que carga los datos del juego desde un archivo

Precondición

El juego está sin inicializar

Postcondición

La salida esperada es ERROR

4.38.2.5. void test2_game_management_save ( )

Prueba Prueba que la función guarde los datos de la partida con el mismo formato que los ficheros de carga

Precondición

El juego no ha sido inicializado y el nombre del archivo es nulo

Postcondición

La salida esperada es ERROR

4.38.2.6. void test2_game_management_start_from_file ( )

Prueba Prueba la función que carga los datos del juego desde un archivo

Precondición

No se lee de forma correcta del archivo

Postcondición

La salida esperada es ERROR

4.39. Referencia del Archivo src/game_rules.c

Implementacion de las reglas del juego.

```
#include "types.h"
#include "game.h"
#include "dialogue.h"
```

4.39.1. Descripción detallada

Implementacion de las reglas del juego.

Autor

Laura Bernal
Sandra Benitez

Versión

1.0

Fecha

28-04-2017

4.40. Referencia del Archivo src/game_rules_test.c

Programa para probar el modulo game_management. Progama para probar la correcta funcionalidad del nuevo TAD Game_management.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "game.h"
#include "space.h"
#include "game_rules.h"
#include "game_management.h"
#include "tests/game_rules_test.h"
#include "tests/test.h"
```

Funciones

- void test1_game_rules_run_random_rule ()
- void test2_game_rules_run_random_rule ()

4.40.1. Descripción detallada

Programa para probar el modulo game_management. Progama para probar la correcta funcionalidad del nuevo TAD Game_management.

Autor

Mihai Blidaru
Laura Bernal

Versión

1.0

Fecha

24-04-2017

4.40.2. Documentación de las funciones

4.40.2.1. void test1_game_rules_run_random_rule ( )

Número maximo de tests

4.40.2.2. void test2_game_rules_run_random_rule ( )

Prueba Prueba la funcion que ejecuta una regla aleatoria

Precondición

El juego no se ha inicializado previamente

Postcondición

La salida esperada es ERROR

4.41. Referencia del Archivo src/game_test.c

Programa para probar el modulo game. Progama para probar la correcta funcionalidad del nuevo TAD Game.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "game.h"
#include "command.h"
#include "tests/game_test.h"
#include "tests/test.h"
```

Funciones

- void test1_game_create ()
- void test1_game_destroy ()
- void test2_game_destroy ()
- void test1_game_update ()
- void test2_game_update ()
- void test3_game_update ()
- void test1_game_is_over ()
- void test2_game_is_over ()
- void test1_game_add_object ()
- void test2_game_add_object ()
- void test3_game_add_object ()
- void test1_game_add_space ()
- void test2_game_add_space ()
- void test3_game_add_space ()
- void test1_game_add_link ()
- void test2_game_add_link ()
- void test3_game_add_link ()
- void test1_game_add_player ()
- void test2_game_add_player ()
- void test3_game_add_player ()
- void test1_game_get_link ()
- void test2_game_get_link ()
- void test3_game_get_link ()
- void test1_game_get_link_at ()
- void test2_game_get_link_at ()
- void test3_game_get_link_at ()
- void test1_game_get_space ()
- void test2_game_get_space ()
- void test3_game_get_space ()
- void test1_game_get_space_at ()
- void test2_game_get_space_at ()
- void test3_game_get_space_at ()
- void test1_game_get_object_at ()
- void test2_game_get_object_at ()
- void test3_game_get_object_at ()
- void test1_game_get_die ()
- void test2_game_get_die ()
- void test1_game_get_player ()
- void test2_game_get_player ()
- void test1_game_get_obj_list_as_str ()

- void test2_game_get_obj_list_as_str ()
- void test1_game_get_last_inspected_space ()
- void test2_game_get_last_inspected_space ()
- void test1_game_get_last_inspected_object ()
- void test2_game_get_last_inspected_object ()
- void test3_game_get_last_inspected_object ()
- void test1_game_get_player_location ()
- void test2_game_get_player_location ()
- void test1_game_get_object_location ()
- void test2_game_get_object_location ()
- void test3_game_get_object_location ()
- void test1_game_get_dialogue ()
- void test2_game_get_dialogue ()

4.41.1. Descripción detallada

Programa para probar el modulo game. Progama para probar la correcta funcionalidad del nuevo TAD Game.

Autor

Mihai Blidaru

Versión

1.0

Fecha

24-04-2017

4.41.2. Documentación de las funciones

4.41.2.1. void test1_game_add_link ( )

Prueba Prueba la función que añade un link al juego

Precondición

El juego y el link han sido inicializados correctamente

Postcondición

La salida esperada es OK

4.41.2.2. void test1_game_add_object ( )

Prueba Prueba la función que añade un objeto al juego

Precondición

El juego y el objeto han sido inicializados correctamente

Postcondición

La salida esperada es OK

4.41.2.3. void test1_game_add_player ( )

Prueba Prueba la función que añade un jugador al juego

Precondición

El juego y el jugador han sido inicializados correctamente

Postcondición

La salida esperada es OK

4.41.2.4. void test1_game_add_space ( )

Prueba Prueba la función que añade un espacio al juego

Precondición

El juego y el espacio han sido inicializados correctamente

Postcondición

La salida esperada es OK

4.41.2.5. void test1_game_create ( )

Número maximo de tests

4.41.2.6. void test1_game_destroy ( )

Prueba Prueba la funcon que destruye un juego

Precondición

El juego se ha creado previamente

Postcondición

La salida esperada es OK

4.41.2.7. void test1_game_get_dialogue ( )

Prueba Prueba la función que devuelve el dialogo del juego

Precondición

El juego ha sido creado correctamente

Postcondición

La salida debe ser un puntero distinto de NULL

4.41.2.8. void test1_game_get_die ( )

Prueba Prueba la función que devuelve el dado del juego

Precondición

El juego ha sido creado correctamente

Postcondición

La salida debe ser un puntero distinto de NULL

4.41.2.9. void test1_game_get_last_inspected_object ( )

Prueba Prueba la función que devuelve el ultimo objeto inspeccionado

Precondición

El nombre del objeto añadido es "obj"

Postcondición

La salida esperada es el objeto con nombre "obj"

4.41.2.10. void test1_game_get_last_inspected_space ( )

Prueba Prueba la función que devuelve el ultimo espacio inspeccionado

Precondición

El juego ha sido creado, se ha añadido un espacio y se ha colocado al jugador dentro

Postcondición

La salida esperada es la casilla en la que esta el jugador

4.41.2.11. void test1_game_get_link ( )

Prueba Prueba la función que devuelve el link con in id dado

Precondición

El juego y el link han sido creados correctamente

Postcondición

La salida debe ser el link creado anteriormente

4.41.2.12. void test1_game_get_link_at ( )

Prueba Prueba la función que devuelve el link en una posición dada

Precondición

El link ha sido añadido correctamente

Postcondición

La salida debe ser el link añadido anteriormente

4.41.2.13. void test1_game_get_obj_list_as_str ( )

Prueba Prueba la funcion que devuelve la lista de objetos de una casilla como cadena

Precondición

El juego, una casilla, y un objeto han sido añadidos correctamente

Postcondición

La salida esperada es la lista de objetos de la casilla

4.41.2.14. void test1_game_get_object_at ( )

Prueba Prueba la función que devuelve el objeto en una posición dada

Precondición

El objeto ha sido añadido correctamente

Postcondición

La salida debe ser el objeto añadido anteriormente

4.41.2.15. void test1_game_get_object_location ( )

Prueba Prueba la funcion que devuelve la localizacion de un objeto

Precondición

El objeto se ha añadido en la casilla con id 2

Postcondición

La salida es 2, la localización del objeto

4.41.2.16. void test1_game_get_player ( )

Prueba Prueba la función que devuelve el jugador del juego

Precondición

El juego y el jugador han sido creado correctamente

Postcondición

La salida debe ser un puntero distinto de NULL

4.41.2.17. void test1_game_get_player_location ( )

Prueba Prueba la funcion que devuelve la localizacion del jugador

Precondición

El jugador se ha colocado en la casilla con id 2

Postcondición

La salida es 2, la localización del jugador

4.41.2.18. void test1_game_get_space ( )

Prueba Prueba la función que devuelve el espacio con in id dado

Precondición

El juego y el espacio han sido creados correctamente

Postcondición

La salida debe ser el espacio creado anteriormente

4.41.2.19. void test1_game_get_space_at ( )

Prueba Prueba la función que devuelve el espacio en una posición dada

Precondición

El espacio ha sido añadido correctamente

Postcondición

La salida debe ser el espacio añadido anteriormente

4.41.2.20. void test1_game_is_over ( )

Prueba Prueba la función devuelve si el juego ha acabado o no

Precondición

El juego ha sido inicializado

Postcondición

La salida esperada es FALSE

4.41.2.21. void test1_game_update ( )

Prueba Prueba la función que actualiza el juego

Precondición

El juego se ha creado y el comando ejecutado es valido

Postcondición

La salida esperada es OK

4.41.2.22. void test2_game_add_link ( )

Prueba Prueba la función que añade un link al juego

Precondición

El juego no ha sido inicializado pero el link si

Postcondición

La salida esperada es ERROR

4.41.2.23. void test2_game_add_object ( )

Prueba Prueba la función que añade un objeto al juego

Precondición

El juego no ha sido inicializado pero el objeto si

Postcondición

La salida esperada es ERROR

4.41.2.24. void test2_game_add_player ( )

Prueba Prueba la función que añade un jugador al juego

Precondición

El juego no ha sido inicializado pero el jugador si

Postcondición

La salida esperada es ERROR

4.41.2.25. void test2_game_add_space ( )

Prueba Prueba la función que añade un espacio al juego

Precondición

El juego no ha sido inicializado pero el espacio si

Postcondición

La salida esperada es ERROR

4.41.2.26. void test2_game_destroy ( )

Prueba Prueba la funcon que destruye un juego

Precondición

El juego no se ha creado previamente

Postcondición

La salida esperada es ERROR

4.41.2.27. void test2_game_get_dialogue ( )

Prueba Prueba la función que devuelve el dialogo del juego

Precondición

El juego no ha sido inicializado

Postcondición

La salida debe ser NULL

4.41.2.28. void test2_game_get_die ( )

Prueba Prueba la función que devuelve el dado del juego

Precondición

El juego no ha sido inicializado

Postcondición

La salida debe ser NULL

4.41.2.29. void test2_game_get_last_inspected_object ( )

Prueba Prueba la función que devuelve el ultimo objeto inspeccionado

Precondición

El objeto está marcado como oculto

Postcondición

La salida esperada es NULL

4.41.2.30. void test2_game_get_last_inspected_space ( )

Prueba Prueba la función que devuelve el ultimo espacio inspeccionado

Precondición

El juego no ha sido creado

Postcondición

La salida esperada es NULL

4.41.2.31. void test2_game_get_link ( )

Prueba Prueba la función que devuelve el link con in id dado

Precondición

El juego no se ha creado pero el link si

Postcondición

La salida debe ser NULL

4.41.2.32. void test2_game_get_link_at ( )

Prueba Prueba la función que devuelve el link en una posición dada

Precondición

La posición en la que se busca es -1

Postcondición

La salida debe ser NULL

4.41.2.33. void test2_game_get_obj_list_as_str ( )

Prueba Prueba la funcion que devuelve la lista de objetos de una casilla como cadena

Precondición

El juego no se ha inicializado

Postcondición

La salida esperada es NULL

4.41.2.34. void test2_game_get_object_at ( )

Prueba Prueba la función que devuelve el objeto en una posición dada

Precondición

La posición en la que se busca es -1

Postcondición

La salida debe ser NULL

4.41.2.35. void test2_game_get_object_location ( )

Prueba Prueba la funcion que devuelve la localizacion de un objeto

Precondición

El objeto no se ha añadido al juego

Postcondición

La salida es NO_ID - localizacion nula

4.41.2.36. void test2_game_get_player ( )

Prueba Prueba la función que devuelve el jugador del juego

Precondición

El juego ha sido inicializado pero no se ha añadido ningun jugador

Postcondición

La salida debe ser NULL

4.41.2.37. void test2_game_get_player_location ( )

Prueba Prueba la funcion que devuelve la localizacion del jugador

Precondición

El jugador no se ha añadido al juego

Postcondición

La salida es NO_ID, localizacion nula

4.41.2.38. void test2_game_get_space ( )

Prueba Prueba la función que devuelve el espacio con in id dado

Precondición

El juego no se ha creado pero el espacio si

Postcondición

La salida debe ser NULL

4.41.2.39. void test2_game_get_space_at ( )

Prueba Prueba la función que devuelve el espacio en una posición dada

Precondición

La posición en la que se busca es -1

Postcondición

La salida debe ser NULL

4.41.2.40. void test2_game_is_over ( )

Prueba Prueba la función devuelve si el juego ha acabado o no

Precondición

El juego no ha sido inicializado

Postcondición

La salida esperada es TRUE, de esa forma en gameloop no se sigue jugando

4.41.2.41. void test2_game_update ( )

Prueba Prueba la función que actualiza el juego

Precondición

El juego no se ha creado. El comando ejecutado es valido

Postcondición

La salida esperada es ERROR

4.41.2.42. void test3_game_add_link ( )

Prueba Prueba la función que añade un link al juego

Precondición

El juego ha sido inicializado pero el link no

Postcondición

La salida esperada es ERROR

4.41.2.43. void test3_game_add_object ( )

Prueba Prueba la función que añade un objeto al juego

Precondición

El juego ha sido inicializado pero el objeto no

Postcondición

La salida esperada es ERROR

4.41.2.44. void test3_game_add_player ( )

Prueba Prueba la función que añade un jugador al juego

Precondición

El juego ha sido inicializado pero el jugador no

Postcondición

La salida esperada es ERROR

4.41.2.45. void test3_game_add_space ( )

Prueba Prueba la función que añade un espacio al juego

Precondición

El juego ha sido inicializado pero el espacio no

Postcondición

La salida esperada es ERROR

4.41.2.46. void test3_game_get_last_inspected_object ( )

Prueba Prueba la función que devuelve el ultimo objeto inspeccionado

Precondición

El nombre del objeto no se corresponde con el del objeto del juego

Postcondición

La salida esperada es NULL

4.41.2.47. void test3_game_get_link ( )

Prueba Prueba la función que devuelve el link con in id dado

Precondición

El link buscado no coresponde con el id del link añadido

Postcondición

La salida debe ser NULL

4.41.2.48. void test3_game_get_link_at ( )

Prueba Prueba la función que devuelve el link en una posición dada

Precondición

El juego no se ha inicializado

Postcondición

La salida debe ser NULL

4.41.2.49. void test3_game_get_object_at ( )

Prueba Prueba la función que devuelve el objeto en una posición dada

Precondición

El juego no se ha inicializado

Postcondición

La salida debe ser NULL

4.41.2.50. void test3_game_get_object_location ( )

Prueba Prueba la funcion que devuelve la localizacion de un objeto

Precondición

El objeto está marcado como oculto

Postcondición

La salida es NO_ID - localizacion nula

4.41.2.51. void test3_game_get_space ( )

Prueba Prueba la función que devuelve el espacio con in id dado

Precondición

El espacio buscado no coresponde con el id del espacio añadido

Postcondición

La salida debe ser NULL

4.41.2.52. void test3_game_get_space_at ( )

Prueba Prueba la función que devuelve el espacio en una posición dada

Precondición

El juego no se ha inicializado

Postcondición

La salida debe ser NULL

4.41.2.53. void test3_game_update ( )

Prueba Prueba la función que actualiza el juego

Precondición

El juego se ha creado pero el comando ejecutado es invalido

Postcondición

La salida esperada es ERROR

4.42. Referencia del Archivo src/graphic_engine_test.c

Programa para probar el modulo graphic_engine. Progama para probar la correcta funcionalidad del nuevo TAD graphic_engine.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "game.h"
#include "space.h"
#include "graphic_engine.h"
#include "tests/graphic_engine_test.h"
#include "tests/test.h"
```

Funciones

- void test1_graphic_engine_create ()

4.42.1. Descripción detallada

Programa para probar el modulo graphic_engine. Progama para probar la correcta funcionalidad del nuevo TAD graphic_engine.

Autor

Mihai Blidaru

Versión

1.0

Fecha

24-04-2017

4.42.2. Documentación de las funciones

4.42.2.1. void test1_graphic_engine_create ( )

Número maximo de tests

4.43. Referencia del Archivo src/inventory.c

Implememtacion del inventario del juego.

```
#include <stdlib.h>
#include <stdio.h>
#include "set.h"
#include "inventory.h"
#include "types.h"
```

Funciones

- Inventory * inventory_create ()
  Crea el inventario e inicializa sus campos.
- STATUS inventory_destroy (Inventory *inventory)
  Destruye un inventario.
- STATUS inventory_add_object (Inventory *inventory, Id id)
  Añade un objeto al inventario.
- STATUS inventory_set_max (Inventory *inventory, int NumIds)
  Modifica el numero de objetos del inventario.
- int inventory_get_max (Inventory *inventory)
  Obtiene el número de objetos de un inventario.
- STATUS inventory_remove_object (Inventory *inventory, Id id)
  Elimina un objeto del inventario.
- Set * inventory_get_set (Inventory *inventory)
  Devuelve el conjunto de identificadores.
- STATUS inventory_print (FILE *fp, Inventory *inventory)
  Imprime los datos de un inventario.

4.43.1. Descripción detallada

Implememtacion del inventario del juego.

Autor

Laura Bernal y Sandra Benítez

Versión

1.0

Fecha

14-03-2017

4.43.2. Documentación de las funciones

4.43.2.1. STATUS inventory_add_object (Inventory * *inventory,* Id *id*)

Añade un objeto al inventario.

Parámetros

inventory	Inventario al que se desea añadir objeto
id	Id del objeto a añadir

Devuelve

Un array de Ids de los objetos que tiene el jugador

4.43.2.2. Inventory* inventory_create ( )

Crea el inventario e inicializa sus campos.

Autor

Laura Bernal y Sandra Benitez

Devuelve

Puntero a inventory

4.43.2.3. STATUS inventory_destroy (Inventory * *inventory*)

Destruye un inventario.

Parámetros

inventory	Inventario que se desea eliminar

Devuelve

OK si se ha liberado correctamente, ERROR en caso contrario

4.43.2.4. int inventory_get_max (Inventory * *inventory*)

Obtiene el número de objetos de un inventario.

Parámetros

inventory	Inventario del que se obtiene el numero maximo de objetos

Devuelve

Número de objetos de un inventario

4.43.2.5. Set* inventory_get_set (Inventory * *inventory*)

Devuelve el conjunto de identificadores.

Parámetros

inventory	Inventario del que se desea obtener el conjunto de identificadores

Devuelve

Conjunto de identificadores

4.43.2.6. STATUS inventory_print (FILE * *fp,* Inventory * *inventory*)

Imprime los datos de un inventario.

Parámetros

inventory	
fp	Archivo

Devuelve

OK si se ha realizado correctamente, ERROR en caso contrario

4.43.2.7. STATUS inventory_remove_object (Inventory * *inventory,* Id *id*)

Elimina un objeto del inventario.

Parámetros

inventory	Inventario de que eliminar el objeto
id	Id del objeto a eliminar

Devuelve

OK si se ha realizado correctamente, ERROR en caso contrario

4.43.2.8. STATUS inventory_set_max (Inventory * *inventory,* int *NumIds*)

Modifica el numero de objetos del inventario.

Parámetros

inventory	Inventario en el que se quiere poner el numero maximo de ids
NumIds	El numero maximo de ids a colocar

Devuelve

OK si se ha realizado correctamente, ERROR en caso contrario

4.44. Referencia del Archivo src/inventory_test.c

Programa para probar el modulo Inventory. Progama para probar la correcta funcionalidad del nuevo TAD Inventory.

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

Funciones

- void test1_inventory_create ()
- void test1_inventory_add_object ()
- void test2_inventory_add_object ()
- void test3_inventory_add_object ()
- void test1_inventory_set_max ()
- void test2_inventory_set_max ()
- void test3_inventory_set_max ()
- void test1_inventory_get_max ()
- void test2_inventory_get_max ()
- void test1_inventory_remove_object ()
- void test2_inventory_remove_object ()
- void test3_inventory_remove_object ()
- void test1_inventory_get_set ()
- void test2_inventory_get_set ()

4.44.1. Descripción detallada

Programa para probar el modulo Inventory. Progama para probar la correcta funcionalidad del nuevo TAD Inventory.

Autor

Sandra Benítez
Laura Bernal

Versión

1.0

Fecha

01-04-2017

4.44.2. Documentación de las funciones

4.44.2.1. void test1_inventory_add_object ( )

Prueba Prueba la función que añade un objeto a un inventario

Precondición

El inventario ha sido inicializado previamente

Postcondición

La salida esperada es ERROR

4.44.2.2. void test1_inventory_create ( )

Número maximo de tests

4.44.2.3. void test1_inventory_get_max ( )

Prueba Prueba la función que devuelve el número máximo de objetos de un inventario

Precondición

Se establece previamente el número máximo de objetos a 20

Postcondición

La salida esperada es 20

4.44.2.4. void test1_inventory_get_set ( )

Prueba Prueba la función que devuelve el set de objetos de un inventario

Precondición

El inventario ha sido inicializado previamente

Postcondición

La salida esperada es diferente de NULL

4.44.2.5. void test1_inventory_remove_object ( )

Prueba Prueba la función que elimina un objeto de un inventario

Precondición

Se añade previamente un objeto con el id 10

Postcondición

La salida esperada es OK

4.44.2.6. void test1_inventory_set_max ( )

Prueba Prueba la función que establece el número máximo de objetos de un inventario

Precondición

El inventario ha sido creado previamente

Postcondición

La salida esperada es OK

4.44.2.7. void test2_inventory_add_object ( )

Prueba Prueba la función que añade un objeto a un inventario

Precondición

El inventario es un puntero a NULL

Postcondición

La salida esperada es ERROR

4.44.2.8. void test2_inventory_get_max ( )

Prueba Prueba la función que devuelve el número máximo de objetos de un inventario

Precondición

El inventario es un puntero a NULL

Postcondición

La salida esperada es -1

4.44.2.9. void test2_inventory_get_set ( )

Prueba Prueba la función que devuelve el set de objetos de un inventario

Precondición

El inventario es un puntero a NULL

Postcondición

La salida esperada es NULL

4.44.2.10. void test2_inventory_remove_object ( )

Prueba Prueba la función que elimina un objeto de un inventario

Precondición

El objeto no ha sido añadido previamente

Postcondición

La salida esperada es ERROR

4.44.2.11. void test2_inventory_set_max ( )

Prueba Prueba la función que establece el número máximo de objetos de un inventario

Precondición

El número máximo de objetos es invalido (-5)

Postcondición

La salida esperada es ERROR

4.44.2.12. void test3_inventory_add_object ( )

Prueba Prueba la función que añade un objeto a un inventario

Precondición

El id del objeto que se quiere añadir es NO_ID

Postcondición

La salida esperada es ERROR

4.44.2.13. void test3_inventory_remove_object ( )

Prueba Prueba la función que elimina un objeto de un inventario

Precondición

El inventario es un puntero a NULL

Postcondición

La salida esperada es ERROR

4.44.2.14. void test3_inventory_set_max ( )

Prueba Prueba la función que establece el número máximo de objetos de un inventario

Precondición

El inventario es un puntero a NULL

Postcondición

La salida esperada es ERROR

4.45. Referencia del Archivo src/link.c

Módulo que define el TAD Enlace así como las primitivas encargadas de trabajar con este TAD.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "link.h"
#include "types.h"
```

Funciones

- Link * link_create ()
  Crea un nuevo link.
- STATUS link_destroy (Link *link)
  Destruye el link.
- STATUS link_set_id (Link *link, Id id)
  Coloca la id del link.
- Id link_get_id (Link *link)
  Obtiene la id del link.
- STATUS link_set_name (Link *link, char *name)
  Coloca el nombre del link.
- Id link_get_dest_from (Link *link, Id from)
  Obtiene el id de destino de un link desde una casilla dada.
- char * link_get_name (Link *link)
  Obtiene el nombre del link.
- STATUS link_set_space1 (Link *link, Id id)
  Coloca la id del spacio 1 del link.
- Id link_get_space1 (Link *link)
  Obtiene la id del spacio 1 del link.

- STATUS link_set_space2 (Link *link, Id id)

  Coloca la id del spacio 2 del link.
- Id link_get_space2 (Link *link)

  Obtiene la id del spacio 2 del link.
- STATUS link_set_state (Link *link, int state)

  Coloca el estado del link.
- State link_get_state (Link *link)

  Obtiene el estado del link.
- int link_print (Link *link)

  Imprime el link.

4.45.1. Descripción detallada

Módulo que define el TAD Enlace así como las primitivas encargadas de trabajar con este TAD.

Autor

Javier Bernardo

Versión

1.0

Fecha

14-03-2017

4.45.2. Documentación de las funciones

4.45.2.1. Link* link_create ( )

Crea un nuevo link.

Autor

Javier Bernardo

Devuelve

El nuevo link creado

4.45.2.2. STATUS link_destroy (Link * *link*)

Destruye el link.

Autor

Javier Bernardo

Parámetros

link	Puntero al link.

Devuelve

OK si se destruido, ERROR en caso contrario

4.45.2.3. Id link_get_dest_from (Link * *link,* Id *from*)

Obtiene el id de destino de un link desde una casilla dada.

Parámetros

link	Puntero a link.
from	Id de la casilla de origen.

Devuelve

El id de destino si se ha podido encontrar o NO_ID en caso contrario.

4.45.2.4. Id link_get_id (Link * *link*)

Obtiene la id del link.

Autor

Javier Bernardo

Parámetros

link	Puntero al link.

Devuelve

La id del link

4.45.2.5. char* link_get_name (Link * *link*)

Obtiene el nombre del link.

Autor

Javier Bernardo

Parámetros

link	Puntero al link.

Devuelve

El nombre del link

4.45.2.6. Id link_get_space1 (Link * *link*)

Obtiene la id del spacio 1 del link.

Autor

Javier Bernardo

Parámetros

link	Puntero al link.

Devuelve

El id del spacio 1

4.45.2.7. Id link_get_space2 (Link * *link*)

Obtiene la id del spacio 2 del link.

Autor

Javier Bernardo

Parámetros

link	Puntero al link.

Devuelve

Id del spacio 2 del link

4.45.2.8. State link_get_state (Link * *link*)

Obtiene el estado del link.

Autor

Javier Bernardo

Parámetros

link	Puntero al link.

Devuelve

El estado del link

4.45.2.9. int link_print (Link * *link*)

Imprime el link.

Autor

Javier Bernardo

Parámetros

link	Puntero al link.

Devuelve

Impresion del link

4.45.2.10. STATUS link_set_id (Link * *link,* Id *id*)

Coloca la id del link.

Autor

Javier Bernardo

Parámetros

link	Puntero al link.
id	La id que se desea colocar

Devuelve

OK si se ha colocado bien la id, ERROR en caso contrario

4.45.2.11. STATUS link_set_name (Link * *link,* char * *name*)

Coloca el nombre del link.

Autor

Javier Bernardo

Parámetros

link	Puntero al link.
name	El nombre que se desea colocar

Devuelve

OK si se ha colocado bien el nombre, ERROR en caso contrario

4.45.2.12. STATUS link_set_space1 (Link * *link,* Id *id*)

Coloca la id del spacio 1 del link.

Autor

Javier Bernardo

Parámetros

link	Puntero al link.
id	La id que se desea colocar

Devuelve

OK si se ha colocado bien la id, ERROR en caso contrario

4.45.2.13. STATUS link_set_space2 (Link * *link,* Id *id*)

Coloca la id del spacio 2 del link.

Autor

Javier Bernardo

Parámetros

link	Puntero al link.
id	La id que se desea colocar

Devuelve

OK si se ha colocado bien la id, ERROR en caso contrario

4.45.2.14. STATUS link_set_state (Link * *link*, int *state*)

Coloca el estado del link.

Autor

Javier Bernardo

Parámetros

link	Puntero al link.
state	El estado al que se quiere poner el link

Devuelve

OK si se ha colocado bien el estado, ERROR en caso contrario

4.46. Referencia del Archivo src/link_test.c

Programa que prueba la funcionalidad del TAD link.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "link.h"
#include "tests/link_test.h"
#include "tests/test.h"
```

Funciones

- void test1_link_create ()
- void test1_link_set_id ()
- void test2_link_set_id ()
- void test3_link_set_id ()
- void test1_link_set_name ()
- void test2_link_set_name ()
- void test3_link_set_name ()
- void test1_link_set_space1 ()
- void test2_link_set_space1 ()
- void test3_link_set_space1 ()
- void test1_link_set_space2 ()
- void test2_link_set_space2 ()
- void test3_link_set_space2 ()
- void test1_link_set_state ()
- void test2_link_set_state ()

- void test3_link_set_state ()
- void test1_link_get_name ()
- void test2_link_get_name ()
- void test1_link_get_id ()
- void test2_link_get_id ()
- void test1_link_get_space1 ()
- void test2_link_get_space1 ()
- void test1_link_get_space2 ()
- void test2_link_get_space2 ()
- void test1_link_get_state ()
- void test2_link_get_state ()
- void test1_link_get_dest_from ()
- void test2_link_get_dest_from ()
- void test3_link_get_dest_from ()
- void test4_link_get_dest_from ()

4.46.1. Descripción detallada

Programa que prueba la funcionalidad del TAD link.

Autor

Javier Bernardo

Fecha

29/03/2017

4.46.2. Documentación de las funciones

4.46.2.1. void test1_link_create ( )

Número máximo de tests

4.46.2.2. void test1_link_get_dest_from ( )

Prueba Prueba la función que devuelve la casilla destino desde otra casilla

Precondición

El enlace se inicializa a (1, 2) y se establece que el id de origen es 1

Postcondición

La salida tiene que ser 2

4.46.2.3. void test1_link_get_id ( )

Prueba Prueba la función para obtener el id de un link

Precondición

Se establece previamente el id del link a 25

Postcondición

La salida debe ser el mismo id: 25

4.46.2.4. void test1_link_get_name ( )

Prueba Prueba la función para obtener el nombre de un link

Precondición

Al link se le pone un nombre previamente

Postcondición

La salida debe ser el mismo nombre establecido

4.46.2.5. void test1_link_get_space1 ( )

Prueba Prueba la función para obtener el space1 de un link

Precondición

Se ha establecido el space1 del link a 26

Postcondición

La salida debe ser el mismo id: 26

4.46.2.6. void test1_link_get_space2 ( )

Prueba Prueba la función para obtener el space2 de un link

Precondición

Se ha establecido el space2 del link a 27

Postcondición

La salida debe ser 27

4.46.2.7. void test1_link_get_state ( )

Prueba Prueba la función para obtener el estado de un link

Precondición

Se establece el estado del enlace como CLOSED

Postcondición

La salida debe ser CLOSED

4.46.2.8. void test1_link_set_id ( )

Prueba Prueba la función para establecer el id de un link

Precondición

El enlace ha sido inicializado previamente

Postcondición

La salida debe ser OK

4.46.2.9. void test1_link_set_name ( )

Prueba Prueba la función para establecer el nombre de un link

Precondición

El link ha sido creado previamente y se le pone un nombre

Postcondición

La salida debe ser OK

4.46.2.10. void test1_link_set_space1 ( )

Prueba Prueba la función para establecer el space1 de un link

Precondición

El enlace ha sido inicializado previamente

Postcondición

La salida debe ser OK

4.46.2.11. void test1_link_set_space2 ( )

Prueba Prueba la función para establecer el space2 de un link

Precondición

El enlace ha sido creado previamente

Postcondición

La salida debe ser OK

4.46.2.12. void test1_link_set_state ( )

Prueba Prueba la función para establecer el estado de un link

Precondición

El link se crea previamente

Postcondición

La salida debe ser OK

4.46.2.13. void test2_link_get_dest_from ( )

Prueba Prueba la función que devuelve la casilla destino desde otra casilla

Precondición

El enlace se inicializa a (1, 2) y se establece que el id de origen es 2

Postcondición

La salida tiene que ser 1

4.46.2.14. void test2_link_get_id ( )

Prueba Prueba la función para obtener el id de un link

Precondición

El enlace es un puntero a NULL

Postcondición

La salida debe ser NO_ID

4.46.2.15. void test2_link_get_name ( )

Prueba Prueba la función para obtener el nombre de un link

Precondición

El enlace es un puntero a NULL

Postcondición

La salida debe ser NULL

4.46.2.16. void test2_link_get_space1 ( )

Prueba Prueba la función para obtener el space1 de un link

Precondición

El enlace es un puntero a NULL

Postcondición

La salida debe ser NO_ID

4.46.2.17. void test2_link_get_space2 ( )

Prueba Prueba la función para obtener el space2 de un link

Precondición

El enlace es un puntero a NULL

Postcondición

La salida debe ser NO_ID

4.46.2.18. void test2_link_get_state ( )

Prueba Prueba la función para obtener el estado de un link

Precondición

El link es un puntero a NULL

Postcondición

La salida debe ser -1

4.46.2.19. void test2_link_set_id ( )

Prueba Prueba la función para establecer el id de un link

Precondición

El enlace es un puntero a NULL

Postcondición

La salida debe ser ERROR

4.46.2.20. void test2_link_set_name ( )

Prueba Prueba la función para establecer el nombre de un link

Precondición

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

Postcondición

La salida debe ser ERROR

4.46.2.21. void test2_link_set_space1 ( )

Prueba Prueba la función para establecer el space1 de un link

Precondición

El enlace es un puntero a NULL

Postcondición

La salida debe ser ERROR

4.46.2.22. void test2_link_set_space2 ( )

Prueba Prueba la función para establecer el space2 de un link

Precondición

El enlace es un puntero a NULL

Postcondición

La salida debe ser ERROR

4.46.2.23. void test2_link_set_state ( )

Prueba Prueba la función para establecer el estado de un link

Precondición

El enlace es un puntero a NULL

Postcondición

La salida debe ser ERROR

4.46.2.24. void test3_link_get_dest_from ( )

Prueba Prueba la función que devuelve la casilla destino desde otra casilla

Precondición

El enlace es un puntero a NULL

Postcondición

La salida tiene que ser NO_ID

4.46.2.25. void test3_link_set_id ( )

Prueba Prueba la función para establecer el id de un link

Precondición

El id que se quiere establecer es NO_ID

Postcondición

La salida debe ser ERROR

4.46.2.26. void test3_link_set_name ( )

Prueba Prueba la función para establecer el nombre de un link

Precondición

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

Postcondición

La salida debe ser ERROR

4.46.2.27. void test3_link_set_space1 ( )

Prueba Prueba la función para establecer el space1 de un link

Precondición

El id que se quiere establecer es NO_ID

Postcondición

La salida debe ser ERROR

4.46.2.28. void test3_link_set_space2 ( )

Prueba Prueba la función para establecer el space2 de un link

Precondición

El id que se quiere establecer es NO_ID

Postcondición

La salida debe ser ERROR

4.46.2.29. void test3_link_set_state ( )

Prueba Prueba la función para establecer el estado de un link

Precondición

El estado que se le quiere poner al link es invalido

Postcondición

La salida debe ser ERROR

4.46.2.30. void test4_link_get_dest_from ( )

Prueba Prueba la función que devuelve la casilla destino desde otra casilla

Precondición

El id de origen no está en el enlace

Postcondición

La salida tiene que ser NO_ID

4.47. Referencia del Archivo src/object.c

Imlementación del TAD Objeto.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "object.h"
#include "types.h"
```

Funciones

- Object * object_create ()
  Estructura del Object.
- STATUS object_destroy (Object *object)
  Destruye un objeto.
- Object * object_Set_Id (Object *object, Id id)
  Pone el id del objeto.
- Object * object_Set_Name (Object *object, char *name)
  Pone el nombre del objeto.
- Object * object_Set_Graphics (Object *object, char *graphics)
  Le asigna al objeto descripción grafica del objeto.
- Object * object_Set_Description (Object *object, char *desc)
  Pone la descripción del objeto.
- Object * object_Set_Description2 (Object *object, char *desc2)
  Pone la descripción dos del objeto.
- STATUS object_Set_Mobile (Object *object, const BOOL mobile)
  Pone la disponibilidad de movimiento del objeto.
- STATUS object_Set_Moved (Object *object, const BOOL moved)
  Pone si se ha mpvido el objeto.

- STATUS object_Set_Hidden (Object *object, BOOL hidden)

  Pone la situacion del objeto, oculto o no oculto.
- STATUS object_Set_Open (Object *object, Id open)

  Pone la situacion del link del objeto.
- STATUS object_Set_Illuminates (Object *object, BOOL illuminates)

  Pone la iluminacion del objeto.
- STATUS object_Set_Light (Object *object, BOOL on)

  Pone el encendido del objeto.
- Id object_Get_Id (Object *object)

  Obtiene el id del objeto.
- char * object_Get_Name (Object *object)

  Obtiene el nombre del objeto.
- char * object_Get_Graphics (Object *object)

  Obtiene la descripción grafica del objeto.
- char * object_Get_Description (Object *object)

  Obtiene la descripción del objeto.
- char * object_Get_Description2 (Object *object)

  Obtiene la descripción dos del objeto.
- BOOL object_Get_Mobile (const Object *object)

  Obtiene la movibilidad del objeto.
- BOOL object_Get_Moved (const Object *object)

  Obtiene si se ha movido el objeto.
- BOOL object_Get_Hidden (const Object *object)

  Obtiene si se ha movido el objeto.
- Id object_Get_Open (const Object *object)

  Obtiene si esta abierto el objeto.
- BOOL object_Get_Illuminates (const Object *object)

  Obtiene si se ha iluminado el objeto.
- BOOL object_Get_Light (const Object *object)

  Obtiene si se ha encendido el objeto.
- STATUS object_print (Object *object)

  Imprime por pantalla el nombre del objeto y el id.

4.47.1. Descripción detallada

Imlementación del TAD Objeto.

Autor

Javier Bernardo Pareja 7

Versión

1.0

Fecha

31-01-2017

Copyright

GNU Public License

4.47.2. Documentación de las funciones

4.47.2.1. Object* object_create ( )

Estructura del Object.

Reservamos memoria para un nuevo objeto.

Cabeceras Libc Cabeceras propias

4.47.2.2. STATUS object_destroy (Object * *object*)

Destruye un objeto.

Fecha

12-12-2005

Parámetros

object	El puntero del objeto.

Devuelve

Error si se ha hecho mal o Ok si se ha liberado correctamente.

4.47.2.3. char* object_Get_Description (Object * *object*)

Obtiene la descripción del objeto.

Parámetros

object	El puntero del objeto.

Devuelve

La descripción del objeto

4.47.2.4. char* object_Get_Description2 (Object * *object*)

Obtiene la descripción dos del objeto.

Parámetros

object	El puntero del objeto.

Devuelve

La descripción del objeto

4.47.2.5. char* object_Get_Graphics (Object * *object*)

Obtiene la descripción grafica del objeto.

Parámetros

object	El puntero del objeto.

Devuelve

La descripción grafica del objeto

4.47.2.6. BOOL object_Get_Hidden (const Object * *object*)

Obtiene si se ha movido el objeto.

Parámetros

object	El puntero del objeto.

Devuelve

Si se ha movido o no el objeto

4.47.2.7. Id object_Get_Id (Object * *object*)

Obtiene el id del objeto.

Parámetros

object	El puntero del objeto.

Devuelve

El id del objeto

4.47.2.8. BOOL object_Get_Illuminates (const Object * *object*)

Obtiene si se ha iluminado el objeto.

Parámetros

object	El puntero del objeto.

Devuelve

Si se ha iluminado o no el objeto

4.47.2.9. BOOL object_Get_Light (const Object * *object*)

Obtiene si se ha encendido el objeto.

Parámetros

object	El puntero del objeto.

Devuelve

Si se ha encendido o no el objeto

4.47.2.10. BOOL object_Get_Mobile (const Object * *object*)

Obtiene la movibilidad del objeto.

Parámetros

object	El puntero del objeto.

Devuelve

La movibilidad del objeto

4.47.2.11. BOOL object_Get_Moved (const Object * *object*)

Obtiene si se ha movido el objeto.

Parámetros

object	El puntero del objeto.

Devuelve

Si se ha movido o no el objeto

4.47.2.12. char* object_Get_Name (Object * *object*)

Obtiene el nombre del objeto.

Parámetros

object	El puntero del objeto.

Devuelve

El nombre del objeto

4.47.2.13. Id object_Get_Open (const Object * *object*)

Obtiene si esta abierto el objeto.

Parámetros

object	El puntero del objeto.

Devuelve

Si se ha abierto o no el link del objeto

4.47.2.14. STATUS object_print (Object * *object*)

Imprime por pantalla el nombre del objeto y el id.

Parámetros

object	El puntero del objeto.

Devuelve

Imprime por pantalla si OK y da ERROR si lo contrario

4.47.2.15. Object* object_Set_Description (Object * *object,* char * *description*)

Pone la descripción del objeto.

Parámetros

object	El puntero del objeto.
description	Descripcion que recibe para el objeto

Devuelve

El objeto con la nueva descripción

4.47.2.16. Object* object_Set_Description2 (Object * *object,* char * *desc2*)

Pone la descripción dos del objeto.

Parámetros

object	El puntero del objeto.
desc2	Descripcion que recibe para el objeto

Devuelve

El objeto con la nueva descripción

4.47.2.17. Object* object_Set_Graphics (Object * *object,* char * *graphics*)

Le asigna al objeto descripción grafica del objeto.

Parámetros

object	El puntero del objeto.
graphics	La descripción grafica del objeto

Devuelve

El objeto actualizado o NULL si se produce algun error

4.47.2.18. STATUS object_Set_Hidden (Object * *object,* BOOL *hidden*)

Pone la situacion del objeto, oculto o no oculto.

Parámetros

object	El puntero del objeto.
hidden	Estado que recibe para el objeto

Devuelve

Ok si se hace, ERROR en caso contrario

4.47.2.19. Object* object_Set_Id (Object * *object,* Id *object_id*)

Pone el id del objeto.

Parámetros

object	El puntero del objeto.
object_id	Id que recibe para el objeto

Devuelve

El objeto con el nuevo id

4.47.2.20. STATUS object_Set_Illuminates (Object * *object*, BOOL *illuminates*)

Pone la iluminacion del objeto.

Parámetros

object	El puntero del objeto.
illuminates	Estado que recibe para el objeto

Devuelve

Ok si se hace, ERROR en caso contrario

4.47.2.21. STATUS object_Set_Light (Object * *object*, BOOL *on*)

Pone el encendido del objeto.

Parámetros

object	El puntero del objeto.
on	Estado que recibe para el objeto

Devuelve

Ok si se hace, ERROR en caso contrario

4.47.2.22. STATUS object_Set_Mobile (Object * *object*, const BOOL *mobile*)

Pone la disponibilidad de movimiento del objeto.

Parámetros

object	El puntero del objeto.
mobile	Estado que recibe para el objeto

Devuelve

Ok si se hace, ERROR en caso contrario

4.47.2.23. STATUS object_Set_Moved (Object * *object*, const BOOL *moved*)

Pone si se ha mpvido el objeto.

Parámetros

object	El puntero del objeto.
moved	Estado que recibe para el objeto

Devuelve

Ok si se hace, ERROR en caso contrario

4.47.2.24. Object* object_Set_Name (Object * *object,* char * *name*)

Pone el nombre del objeto.

Parámetros

object	El puntero del objeto.
name	Nombre que recibe para el objeto

Devuelve

El objeto con el nuevo nombre

4.47.2.25. STATUS object_Set_Open (Object * *object,* Id *open*)

Pone la situacion del link del objeto.

Parámetros

object	El puntero del objeto.
open	Estado que recibe para el objeto

Devuelve

Ok si se hace, ERROR en caso contrario

4.48. Referencia del Archivo src/object_test.c

Prueba el módulo Object.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "tests/test.h"
#include "tests/object_test.h"
#include "object.h"
```

Funciones

- void test1_object_create ()
- void test1_object_set_name ()
- void test2_object_set_name ()
- void test1_object_set_graphics ()
- void test2_object_set_graphics ()
- void test1_object_set_Id ()

- void test2_object_set_Id ()
- void test1_object_set_description ()
- void test2_object_set_description ()
- void test1_object_set_description2 ()
- void test2_object_set_description2 ()
- void test1_object_set_Mobile ()
- void test2_object_set_Mobile ()
- void test1_object_set_Moved ()
- void test2_object_set_Moved ()
- void test1_object_set_Hidden ()
- void test2_object_set_Hidden ()
- void test1_object_set_Open ()
- void test2_object_set_Open ()
- void test1_object_set_Illuminates ()
- void test2_object_set_Illuminates ()
- void test1_object_set_Light ()
- void test2_object_set_Light ()
- void test1_object_Get_Name ()
- void test2_object_Get_Name ()
- void test1_object_Get_Graphics ()
- void test2_object_Get_Graphics ()
- void test1_object_Get_Id ()
- void test2_object_Get_Id ()
- void test1_object_Get_Description ()
- void test2_object_Get_Description ()
- void test1_object_Get_Description2 ()
- void test2_object_Get_Description2 ()
- void test1_object_Get_Mobile ()
- void test2_object_Get_Mobile ()
- void test1_object_Get_Moved ()
- void test2_object_Get_Moved ()
- void test1_object_Get_Hidden ()
- void test2_object_Get_Hidden ()
- void test1_object_Get_Open ()
- void test2_object_Get_Open ()
- void test1_object_Get_Illuminates ()
- void test2_object_Get_Illuminates ()
- void test1_object_Get_Light ()
- void test2_object_Get_Light ()

4.48.1. Descripción detallada

Prueba el módulo Object.

Autor

Javier Bernardo

Versión

2.0

Fecha

24-04-2017

4.48.2. Documentación de las funciones

4.48.2.1. void test1_object_create ( )

Prueba Prueba si se crea correctamente un objeto

Postcondición

Un puntero no nulo al objeto creado

4.48.2.2. void test1_object_Get_Description ( )

Prueba Prueba leer la descripcion de un objeto

Precondición

Al objeto se le ha asignado previamente la descripcion

Postcondición

La salida esperada es la descripcion asignado antes

4.48.2.3. void test1_object_Get_Description2 ( )

Prueba Prueba leer la descripcion de un objeto

Precondición

Al objeto se le ha asignado previamente la descripcion

Postcondición

La salida esperada es la descripcion asignado antes

4.48.2.4. void test1_object_Get_Graphics ( )

Prueba Prueba leer los graficos de un objeto

Precondición

Al nombre se le ha asignado previamente los graficos "Bryan"

Postcondición

La salida esperada son los graficos asignados antes "Bryan"

4.48.2.5. void test1_object_Get_Hidden ( )

Prueba Prueba leer si se ha escondido un objeto

Precondición

Al objeto se le ha asignado previamente la invibisibilidad

Postcondición

La salida esperada es la movilidad asignada antes

4.48.2.6. void test1_object_Get_Id ( )

Prueba Prueba leer el id de un objeto

Precondición

Al objeto se le ha asignado previamente el id 12

Postcondición

La salida esperada es el id asignado antes (12)

4.48.2.7. void test1_object_Get_Illuminates ( )

Prueba Prueba leer si se ha iluminado un objeto

Precondición

Al objeto se le ha asignado previamente la posibilidad de iluminar

Postcondición

La salida esperada es la movilidad asignada antes

4.48.2.8. void test1_object_Get_Light ( )

Prueba Prueba leer si se puede iluminar un objeto

Precondición

Al objeto se le ha asignado previamente la posibilidad de iluminar

Postcondición

La salida esperada es la movilidad asignada antes

4.48.2.9. void test1_object_Get_Mobile ( )

Prueba Prueba leer la movilidad de un objeto

Precondición

Al objeto se le ha asignado previamente la movilidad

Postcondición

La salida esperada es la movilidad asignada antes

4.48.2.10. void test1_object_Get_Moved ( )

Prueba Prueba leer si se ha movido un objeto

Precondición

Al objeto se le ha asignado previamente la movilidad

Postcondición

La salida esperada es la movilidad asignada antes

4.48.2.11. void test1_object_Get_Name ( )

Prueba Prueba leer el nombre de un objeto

Precondición

Al nombre se le ha asignado previamente el nombre Bryan

Postcondición

La salida esperada es el id asignado antes "Bryan"

4.48.2.12. void test1_object_Get_Open ( )

Prueba Prueba leer si se ha abierto el link a un objeto

Precondición

Al objeto se le ha asignado previamente la disponibilidad

Postcondición

La salida esperada es la movilidad asignada antes

4.48.2.13. void test1_object_set_description ( )

Prueba Prueba si se le asigna correctamente una descripcion a un objeto

Precondición

La descripcion del objeto

Postcondición

La salida tiene que ser el puntero al objeto

4.48.2.14. void test1_object_set_description2 ( )

Prueba Prueba si se le asigna correctamente una descripcion2 a un objeto

Precondición

La descripcion2 del objeto

Postcondición

La salida tiene que ser el puntero al objeto

4.48.2.15. void test1_object_set_graphics ( )

Prueba Prueba si se le asigna correctamente los graficos a un objeto

Precondición

El nombre del objeto

Postcondición

La salida tiene que ser el puntero al objeto

4.48.2.16. void test1_object_set_Hidden ( )

Prueba Prueba si se le asigna correctamente una situacion a un objeto

Precondición

La situacion del objeto

Postcondición

La salida tiene que ser el puntero al objeto

4.48.2.17. void test1_object_set_Id ( )

Prueba Prueba si se le asigna correctamente un Id a un objeto

Precondición

El Id del objeto

Postcondición

La salida tiene que ser el puntero al objeto

4.48.2.18. void test1_object_set_Illuminates ( )

Prueba Prueba si se le asigna correctamente una iluminacion a un objeto

Precondición

La iluminacion del objeto

Postcondición

La salida tiene que ser el puntero al objeto

4.48.2.19. void test1_object_set_Light ( )

Prueba Prueba si se le asigna correctamente una iluminacion a un objeto

Precondición

La iluminacion del objeto

Postcondición

La salida tiene que ser el puntero al objeto

4.48.2.20. void test1_object_set_Mobile ( )

Prueba Prueba si se le asigna correctamente una movilidad a un objeto

Precondición

La movilidad del objeto

Postcondición

La salida tiene que ser el puntero al objeto

4.48.2.21. void test1_object_set_Moved ( )

Prueba Prueba si se le asigna correctamente una movilidad a un objeto

Precondición

La movilidad del objeto

Postcondición

La salida tiene que ser el puntero al objeto

4.48.2.22. void test1_object_set_name ( )

Prueba Prueba si se le asigna correctamente un nombre a un objeto

Precondición

El nombre del objeto

Postcondición

La salida tiene que ser el puntero al objeto

4.48.2.23. void test1_object_set_Open ( )

Prueba Prueba si se le asigna correctamente una situacion a un objeto

Precondición

La situacion del objeto

Postcondición

La salida tiene que ser el puntero al objeto

4.48.2.24. void test2_object_Get_Description ( )

Prueba Prueba leer la descripcion de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es NULL

4.48.2.25. void test2_object_Get_Description2 ( )

Prueba Prueba leer la descripcion de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es NULL

4.48.2.26. void test2_object_Get_Graphics ( )

Prueba Prueba leer los graficos de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es NULL

4.48.2.27. void test2_object_Get_Hidden ( )

Prueba Prueba leer la situacion de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es FALSE

4.48.2.28. void test2_object_Get_Id ( )

Prueba Prueba leer el id de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es NO_ID

4.48.2.29. void test2_object_Get_Illuminates ( )

Prueba Prueba leer la situacion de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es FALSE

4.48.2.30. void test2_object_Get_Light ( )

Prueba Prueba leer la situacion de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es FALSE

4.48.2.31. void test2_object_Get_Mobile ( )

Prueba Prueba leer la movilidad de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es FALSE

4.48.2.32. void test2_object_Get_Moved ( )

Prueba Prueba leer la movilidad de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es FALSE

4.48.2.33. void test2_object_Get_Name ( )

Prueba Prueba leer el nombre de un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es NULL

4.48.2.34. void test2_object_Get_Open ( )

Prueba Prueba leer la situacion del link a un objeto

Precondición

El objeto no ha sido inicializado previamente

Postcondición

La salida esperada es NO_ID

4.48.2.35. void test2_object_set_description ( )

Prueba Prueba asignar una descripcion a un objeto sin inicializar

Precondición

El objeto al que establecer la descripcion es un puntero a NULL

Postcondición

La salida debe ser NULL

4.48.2.36. void test2_object_set_description2 ( )

Prueba Prueba asignar una descripcion2 a un objeto sin inicializar

Precondición

El objeto al que establecer la descripcion2 es un puntero a NULL

Postcondición

La salida debe ser NULL

4.48.2.37. void test2_object_set_graphics ( )

Prueba Prueba asignar graficos a un objeto sin inicializar

Precondición

El objeto al que establecer los graficos es un puntero a NULL

Postcondición

La salida debe ser NULL

4.48.2.38. void test2_object_set_Hidden ( )

Prueba Prueba asignar una situacion a un objeto sin inicializar

Precondición

El objeto al que establecer la situacion es un puntero a NULL

Postcondición

La salida debe ser NULL

4.48.2.39. void test2_object_set_Id ( )

Prueba Prueba asignar un Id a un objeto sin inicializar

Precondición

El objeto al que establecer el Id es un puntero a NULL

Postcondición

La salida debe ser NULL

4.48.2.40. void test2_object_set_Illuminates ( )

Prueba Prueba asignar una iluminacion a un objeto sin inicializar

Precondición

El objeto al que establecer la iluminacion es un puntero a NULL

Postcondición

La salida debe ser NULL

4.48.2.41. void test2_object_set_Light ( )

Prueba Prueba asignar una iluminacion a un objeto sin inicializar

Precondición

El objeto al que establecer la iluminacion es un puntero a NULL

Postcondición

La salida debe ser NULL

4.48.2.42. void test2_object_set_Mobile ( )

Prueba Prueba asignar una movilidad a un objeto sin inicializar

Precondición

El objeto al que establecer la movilidad es un puntero a NULL

Postcondición

La salida debe ser NULL

4.48.2.43. void test2_object_set_Moved ( )

Prueba Prueba asignar una movilidad a un objeto sin inicializar

Precondición

El objeto al que establecer la movilidad es un puntero a NULL

Postcondición

La salida debe ser NULL

4.48.2.44. void test2_object_set_name ( )

Prueba Prueba asignar un nombre a un objeto sin inicializar

Precondición

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

Postcondición

La salida debe ser NULL

4.48.2.45. void test2_object_set_Open ( )

Prueba Prueba asignar una situacion a un objeto sin inicializar

Precondición

El objeto al que establecer la situacion es un puntero a NULL

Postcondición

La salida debe ser NULL

4.49. Referencia del Archivo src/player.c

Define el TAD player.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "player.h"
#include "inventory.h"
#include "types.h"
#include "set.h"
```

Funciones

- Player * player_create ()

  Crea un nuevo jugador e inicializa sus campos.
- STATUS player_destroy (Player *player)

  Destruye un jugador.
- Player * player_Set_Id (Player *player, Id player_id)

  Pone el id del jugador.
- Player * player_Set_Name (Player *player, char *name)

  Pone el nombre del jugador.
- Player * player_Set_Location (Player *player, Id location)

  Pone la localizacion del jugador.
- Player * player_Set_Max_Objects (Player *player, int max_objects)

  Asigna el número maximo de objetos que puede llevar el jugador.
- Id player_Get_Id (Player *player)

  Obtiene el id del jugador.
- char * player_Get_Name (Player *player)

  Obtiene el nombre del jugador.
- Id player_Get_Location (Player *player)

  Obtiene la localizacion del jugador.
- int player_Get_Max_Objects (Player *player)

  Devuelve el número maximo de objetos que puede llevar el jugador.
- STATUS player_Add_Object (Player *player, Id object_id)

  Añade un objeto al Inventory del jugador.
- STATUS player_Remove_Object (Player *player, Id object_id)

  Borra un objeto del Inventory del jugador.
- BOOL player_Has_Object (Player *player, Id object_id)

  Comprueba si el jugador tiene un objeto determinado.
- STATUS player_Print (Player *player)

  Imprime por pantalla los datos del jugador.

4.49.1. Descripción detallada

Define el TAD player.

Autor

Mihai Blidaru Pareja 7

Versión

1.0

Fecha

31-01-2017

Copyright

GNU Public License

4.49.2. Documentación de las funciones

4.49.2.1. STATUS player_Add_Object (Player * *player,* Id *object_id*)

Añade un objeto al Inventory del jugador.

Autor

Mihai Blidaru

Parámetros

player	El jugador al que se quiere añadir un objeto
object_id	El id del objeto

Devuelve

OK si se hace bien, ERROR en caso contrario

4.49.2.2. Player* player_create ( )

Crea un nuevo jugador e inicializa sus campos.

Devuelve

El nuevo jugador

4.49.2.3. STATUS player_destroy (Player * *player*)

Destruye un jugador.

Parámetros

player	El puntero del jugador.

Devuelve

Error si se ha hecho mal o Ok si se ha liberado correctamente.

4.49.2.4. Id player_Get_Id (Player * *player*)

Obtiene el id del jugador.

Parámetros

player	El puntero del jugador.

Devuelve

El id del jugador

4.49.2.5. Id player_Get_Location (Player * *player*)

Obtiene la localizacion del jugador.

Parámetros

player	El puntero del jugador.

Devuelve

La localizacion del jugador

4.49.2.6. int player_Get_Max_Objects (Player * *player*)

Devuelve el número maximo de objetos que puede llevar el jugador.

Autor

Sandra Benitez

Parámetros

player	El jugador del que se quiere conocer el numero máximo de objetos

Devuelve

Devuelve el número máximo de objetos

4.49.2.7. char* player_Get_Name (Player * *player*)

Obtiene el nombre del jugador.

Parámetros

player	El puntero del jugador.

Devuelve

El nombre del jugador

4.49.2.8. BOOL player_Has_Object (Player * *player,* Id *object_id*)

Comprueba si el jugador tiene un objeto determinado.

Parámetros

player	El jugador que se quiere comprobar
object_id	El id del objeto

Devuelve

TRUE si el jugador tiene el objeto o FALSE en caso contrario o ERROR

4.49.2.9. STATUS player_Print (Player * *player*)

Imprime por pantalla los datos del jugador.

Parámetros

player	El puntero del jugador.

Devuelve

Imprime por pantalla si hay jugador y da ERROR si lo contrario

4.49.2.10. STATUS player_Remove_Object (Player * *player,* Id *object_id*)

Borra un objeto del Inventory del jugador.

Autor

Mihai Blidaru

Parámetros

player	El jugador al que se quiere quitar un objeto
object_id	El id del objeto

4.49.2.11. Player* player_Set_Id (Player * *player,* Id *player_id*)

Pone el id del jugador.

Parámetros

player	El puntero del jugador.
player_id	Id del jugador

Devuelve

El jugador

4.49.2.12. Player* player_Set_Location (Player * *player,* Id *location*)

Pone la localizacion del jugador.

Parámetros

player	El puntero del jugador.
location	La localizacion del jugador

Devuelve

El jugador

4.49.2.13. Player* player_Set_Max_Objects (Player * *player,* int *max_objects*)

Asigna el número maximo de objetos que puede llevar el jugador.

Autor

Mihai Blidaru

Parámetros

player	El jugador al que se le quiere cambiar el numero máximo de objetos
max_objects	Número máximo de objetos que puede llevar el jugador

Devuelve

Devuelve un puntero al jugador si todo ha ido bien o NULL en caso contrario.

4.49.2.14. Player* player_Set_Name (Player * *player,* char * *name*)

Pone el nombre del jugador.

Parámetros

player	El puntero del jugador.
name	Nombre del jugador

Devuelve

El jugador

4.50. Referencia del Archivo src/player_test.c

Prueba el módulo Player.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "tests/test.h"
#include "tests/player_test.h"
#include "player.h"
```

Funciones

- void test1_player_create ()
- void test1_player_set_name ()

- void test2_player_set_name ()
- void test3_player_set_name ()
- void test1_player_set_id ()
- void test2_player_set_id ()
- void test3_player_set_id ()
- void test1_player_set_location ()
- void test2_player_set_location ()
- void test3_player_set_location ()
- void test1_player_get_id ()
- void test2_player_get_id ()
- void test3_player_get_id ()
- void test1_player_get_name ()
- void test2_player_get_name ()
- void test3_player_get_name ()
- void test1_player_get_location ()
- void test2_player_get_location ()
- void test3_player_get_location ()
- void test1_player_add_object ()
- void test2_player_add_object ()
- void test3_player_add_object ()
- void test1_player_remove_object ()
- void test2_player_remove_object ()
- void test3_player_remove_object ()
- void test4_player_remove_object ()
- void test1_player_set_max_objects ()
- void test2_player_set_max_objects ()
- void test3_player_set_max_objects ()
- void test4_player_set_max_objects ()
- void test1_player_has_object ()
- void test2_player_has_object ()
- void test3_player_has_object ()

4.50.1. Descripción detallada

Prueba el módulo Player.

Autor

Mihai Blidaru

Versión

2.0

Fecha

29-03-2017

4.50.2. Documentación de las funciones

4.50.2.1. void test1_player_add_object ( )

Prueba Prueba añadir un objeto a un jugador

Precondición

El jugador está sin inicializar

Postcondición

La salida que se espera es ERROR

4.50.2.2. void test1_player_create ( )

Prueba Prueba si se crea correctamente un jugador

Postcondición

Un puntero no nulo al jugador creado

4.50.2.3. void test1_player_get_id ( )

Prueba Prueba leer el id de un jugador

Precondición

Al jugador se le ha asignado previamente el id 12

Postcondición

La salida esperada es el id asignado antes (12)

4.50.2.4. void test1_player_get_location ( )

Prueba Leer la localización de un jugador

Precondición

Al jugador se le ha asignado previamente la localización 12

Postcondición

La salida esperada es la localización previamente asignada (12)

4.50.2.5. void test1_player_get_name ( )

Prueba Intenta leer el nombre de un jugador

Precondición

Al jugador se le ha asignado previamente el nombre "Bob"

Postcondición

La salida esperada es el nombre asignado antes : "Bob"

4.50.2.6. void test1_player_has_object ( )

Prueba Prubea si el jugador tiene un objeto en condiciones normales

Precondición

Se le añade un objeto al jugador

Postcondición

La salida debe ser TRUE

4.50.2.7. void test1_player_remove_object ( )

Prueba Prueba quitar un objeto al jugador en condiciones normales

Precondición

Al jugador se le añade un objeto

Postcondición

La salida que se espera es OK

4.50.2.8. void test1_player_set_id ( )

Prueba Prueba asignar un id a un jugador en condiciones normales

Precondición

Al jugador se le asigna un id cualquiera mayor que cero

Postcondición

La salida tiene que ser el puntero al jugador

4.50.2.9. void test1_player_set_location ( )

Prueba Prueba asignarle una localización a un jugador en condiciones

Precondición

El jugador ha sido previamente inicializado y la localización que se le asigna es valida

Postcondición

La salida es el puntero al jugador

4.50.2.10. void test1_player_set_max_objects ( )

Prueba Prueba poner el número máximo de objetos de un jugador

Precondición

Las condiciones son normales: jugador inicializado, número de objeto dentro de limites.

Postcondición

La salida debe ser el puntero al jugador

4.50.2.11. void test1_player_set_name ( )

Prueba Prueba si se le asigna correctamente un nombre a un jugador

Precondición

El nombre del jugador

Postcondición

La salida tiene que ser el puntero al jugador

4.50.2.12. void test2_player_add_object ( )

Prueba Prueba añadir un objeto a un jugador en condiciones normales

Precondición

El jugador ha sido correctamente inicializado y el id del objeto es valido

Postcondición

La salida debe ser OK

4.50.2.13. void test2_player_get_id ( )

Prueba Prueba leer el id de un jugador

Precondición

El jugador no ha sido inicializado previamente

Postcondición

La salida esperada es NO_ID

4.50.2.14. void test2_player_get_location ( )

Prueba Prueba leer la localización del jugador

Precondición

El jugador no ha sido inicializado

Postcondición

La salida esperada es NO_ID

4.50.2.15. void test2_player_get_name ( )

Prueba Prueba leer el nombre de un jugador

Precondición

El jugador está sin inicializar

Postcondición

La salida debe ser NULL

4.50.2.16. void test2_player_has_object ( )

Prueba Prueba si el jugador tiene un objeto

Precondición

El jugador está sin inicializar

Postcondición

La salida debe ser FALSE

4.50.2.17. void test2_player_remove_object ( )

Prueba Prueba quitar un objeto a un jugador

Precondición

Al jugador no se le ha añadido todavía ningun objeto

Postcondición

La salida que se espera es ERROR

4.50.2.18. void test2_player_set_id ( )

Prueba Prueba asignar un id a un jugador sin inicializar

Precondición

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

Postcondición

La salida esperada es NULL

4.50.2.19. void test2_player_set_location ( )

Prueba Prueba asignarle una localización a un jugador sin inicializar.

Precondición

El jugador no ha sido inicializado previamente

Postcondición

La salida que se espera es NULL

4.50.2.20. void test2_player_set_max_objects ( )

Prueba Prueba poner el número máximo de objetos de un jugador

Precondición

El número de objetos está por encima del limite permitido

Postcondición

La salida debe ser NULL

4.50.2.21. void test2_player_set_name ( )

Prueba Prueba asignar un nombre a un jugador sin inicializar

Precondición

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

Postcondición

La salida debe ser NULL

4.50.2.22. void test3_player_add_object ( )

Prueba Prueba añadir un objeto al inventario del jugador

Precondición

El objeto tiene id -1

Postcondición

La salida que se espera es ERROR

4.50.2.23. void test3_player_get_id ( )

Prueba Intenta leer el id del jugador

Precondición

Al jugador no se le ha asignado ninguna id todavia

Postcondición

La salida esperada es NO_ID

4.50.2.24. void test3_player_get_location ( )

Prueba Prueba leer la localización del jugador

Precondición

Al jugador no se le ha asignado ninguna localización todavía

Postcondición

La salida que se espera es NO_ID

4.50.2.25. void test3_player_get_name ( )

Prueba Prueba leer el nombre de un jugador

Precondición

Al jugador no se le ha dado ningun nombre previamente

Postcondición

La salida esperada es una cadena vacía

4.50.2.26. void test3_player_has_object ( )

Prueba Prueba si el jugador tiene un objeto

Precondición

El jugador está inicializado pero no tiene ningun objeto

Postcondición

La salida debe ser FALSE

4.50.2.27. void test3_player_remove_object ( )

Prueba Prueba quitar un objeto a un jugador

Precondición

El jugador está sin inicializar

Postcondición

La salida que se espera es ERROR

4.50.2.28. void test3_player_set_id ( )

Prueba Prueba asignar un id a un jugador

Precondición

El id que se quiere asignar es NO_ID

Postcondición

La salida que se espera es NULL

4.50.2.29. void test3_player_set_location ( )

Prueba Prueba asignarle una localización invalida a un jugador.

Precondición

La localización que se quiere asignar al jugador es es negativa

Postcondición

La salida esperada es NULL

4.50.2.30. void test3_player_set_max_objects ( )

Prueba Prueba poner el número máximo de objetos de un jugador

Precondición

El jugador está sin inicializar

Postcondición

La salida debe ser NULL

4.50.2.31. void test3_player_set_name ( )

Prueba Prueba asignarle un nombre nulo a un jugador

Precondición

La cadena que se quiere asignar es un puntero a NULL

Postcondición

La salida tiene que ser NULL

4.50.2.32. void test4_player_remove_object ( )

Prueba Prueba quitar un objeto al jugador

Precondición

El objeto ya ha sido borrado anteriormente

Postcondición

La salida que se espera es ERROR

4.50.2.33. void test4_player_set_max_objects ( )

Prueba Prueba poner el numero máximo de objetos de un jugador

Precondición

El número de objetos es un número negativo

Postcondición

La salida debe ser NULL

4.51. Referencia del Archivo src/screen.c

Modulo necesario para imprimir por pantalla las areas del juego.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "screen.h"
```

Funciones

- void screen_init ()

  Inicializa los datos para imprimir el juego en pantalla.
- void screen_destroy ()

  Destruye los datos usados para imprimir el juego en pantalla.
- void screen_paint ()

  Imprime el juego en pantalla.
- void screen_gets (char ∗str)

  Obtiene una pantalla.

- Area * screen_area_init (int x, int y, int width, int height)

  Crea un area.
- void screen_area_destroy (Area *area)

  Destruye un area.
- void screen_area_clear (Area *area)

  Despeja un area.
- void screen_area_reset_cursor (Area *area)

  Resetea el cursor en un area.
- void screen_area_puts (Area *area, char *str)

  Escribe texto en un area.

4.51.1. Descripción detallada

Modulo necesario para imprimir por pantalla las areas del juego.

Autor

Profesores PPROG

Fecha

11-01-2017

Copyright

GNU Public License

4.51.2. Documentación de las funciones

4.51.2.1. void screen_area_clear (Area * *area*)

Despeja un area.

Parámetros

area	Area del cual tiene que quitar el texto

4.51.2.2. void screen_area_destroy (Area * *area*)

Destruye un area.

Parámetros

area	Area que tiene que destruir

4.51.2.3. Area* screen_area_init (int *x,* int *y,* int *width,* int *height*)

Crea un area.

Parámetros

x	Posición x del area
y	Posición y del area
width	Ancho del area
height	Alto del area

Devuelve

area creada

4.51.2.4. void screen_area_puts (Area * *area*, char * *str*)

Escribe texto en un area.

Parámetros

area	Area en el cual escribir
str	Cadena de texto que se quiere imprimir en el area

4.51.2.5. void screen_area_reset_cursor (Area * *area*)

Resetea el cursor en un area.

Parámetros

area	Area del cual se quiere resetear el cursor

4.51.2.6. void screen_gets (char * *str*)

Obtiene una pantalla.

Imprime con colores

Parámetros

str	Puntero a cadena para indicar que pantalla cojer

4.51.2.7. void screen_init ( )

Inicializa los datos para imprimir el juego en pantalla.

Cabeceras Libc

4.52. Referencia del Archivo src/screen_test.c

Programa para probar el modulo screen. Progama para probar la correcta funcionalidad del nuevo TAD screen.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "screen.h"
#include "tests/screen_test.h"
#include "tests/test.h"
```

Funciones

- void test1_screen_area_init ()
- void test2_screen_area_init ()
- void test3_screen_area_init ()
- void test4_screen_area_init ()
- void test5_screen_area_init ()

4.52.1. Descripción detallada

Programa para probar el modulo screen. Progama para probar la correcta funcionalidad del nuevo TAD screen.

Autor

Javier Bernardo y Mihai Blidaru

Versión

1.0

Fecha

24-04-2017

4.52.2. Documentación de las funciones

4.52.2.1. void test1_screen_area_init ( )

Número maximo de tests

4.52.2.2. void test2_screen_area_init ( )

Prueba Prueba la función que inicializa un area

Precondición

La posición x es negativa

Postcondición

La salida esperada es NULL

4.52.2.3. void test3_screen_area_init ( )

Prueba Prueba la función que inicializa un area

Precondición

La posición y es negativa

Postcondición

La salida esperada es NULL

4.52.2.4. void test4_screen_area_init ( )

Prueba Prueba la función que inicializa un area

Precondición

El ancho del area es negativo

Postcondición

La salida esperada es NULL

4.52.2.5. void test5_screen_area_init ( )

Prueba Prueba la función que inicializa un area

Precondición

El alto del area es negativo

Postcondición

La salida esperada es NULL

4.53. Referencia del Archivo src/set.c

Implememtacion del set del juego.

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

Funciones

- Set * set_create ()

  Crea un nuevo conjunto.
- STATUS set_destroy (Set *set)

  Libera memoria usada por un connjunto.
- BOOL set_Id_is_in (Set *set, Id id)

  Comprueba si un id ya existe en el conjunto;.
- STATUS set_addId (Set *set, Id id)

  Añade un nuevo id al conjunto.
- Set * set_delId (Set *set, Id id)

  Borra un id del Set.
- int set_getNumberOfIds (Set *set)

  Obtiene el numero de ids del set.
- int set_print (FILE *fp, Set *set)

  Imprime el set.

4.53.1. Descripción detallada

Implememtacion del set del juego.

Autor

Mihai Blidaru

Versión

1.0

Fecha

23-02-2017

4.53.2. Documentación de las funciones

4.53.2.1. STATUS set_addId (Set * *set*, Id *id*)

Añade un nuevo id al conjunto.

Autor

Mihai Blidaru

Parámetros

set	Conjunto donde añadir el nuevo id.
id	Id que se tiene que añadir al conjunto.

Devuelve

OK si se ha añadido correctamente el Id o ERROR en caso contrario.

4.53.2.2. Set* set_create ( )

Crea un nuevo conjunto.

Autor

Mihai Blidaru

Devuelve

El nuevo conjunto creadoo o NULL si no se ha podido cread

4.53.2.3. Set* set_delId (Set * *set*, Id *id*)

Borra un id del Set.

Autor

Mihai Blidaru

Parámetros

set	Conjunto donde borrar el id pasado como parametro
id	Id que se tiene que borrar del conjunto

Devuelve

Conjunto.

4.53.2.4. STATUS set_destroy (Set * *set*)

Libera memoria usada por un connjunto.

Autor

Mihai Blidaru

Parámetros

set	Conjunto que se tiene que destruir

Devuelve

OK si se ha realizado la operación correctamente o ERROR en caso contrario.

4.53.2.5. int set_getNumberOfIds (Set * *set*)

Obtiene el numero de ids del set.

Autor

Mihai Blidaru

Parámetros

set	Conjunto del que se quiere saber su numero de ids.

Devuelve

Numero de ids.

4.53.2.6. BOOL set_Id_is_in (Set * *set,* Id *id*)

Comprueba si un id ya existe en el conjunto;.

Autor

Mihai Blidaru

Parámetros

set	Conjunto en el que buscar el Id

id	Id que se tiene que comprobar si existe

Devuelve

TRUE si el Id se encuentra en el conjuto o FALSE en caso contrario.

4.53.2.7. int set_print (FILE * *fp*, Set * *set*)

Imprime el set.

Autor

Mihai Blidaru

Parámetros

fp	Archivo donde imprimir
set	Conjunto a imprimir

Devuelve

El número de caracteres imprimidos

4.54. Referencia del Archivo src/set_test.c

Implementacion del conjuntos del juego Progama para probar la correcta funcionalidad del nuevo TAD Set.

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

Funciones

- int main ()

  Definicion de la funcion main del set_test.

4.54.1. Descripción detallada

Implementacion del conjuntos del juego Progama para probar la correcta funcionalidad del nuevo TAD Set.

Autor

Javier Bernando y Mihai Blidaru

Versión

1.0

Fecha

11-03-2017

4.55. Referencia del Archivo src/space.c

Implementa el TAD space Define los campos de la estructura Space. Tambien define todas las funciones necesarias para trabajar con este tipo de datos.

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

Funciones

- Space * space_create (Id space_id)

  Crea un nuevo espacio y le asigna un Id y unos valores por defecto.
- STATUS space_destroy (Space *space)

  Libera la memoria de un espacio.
- STATUS space_set_name (Space *space, char *name)

  Pone nombre a un espacio.
- STATUS space_set_description (Space *space, char *description)

  Pone nombre una descripcion al espacio.
- STATUS space_set_long_description (Space *space, char *description)

  Pone una descripcion detallada al espacio.
- Space * space_set_graphics (Space *space, char graphics[][G_COLUMNS])

  Pone la descripcion de la casilla.
- STATUS space_set_north (Space *space, Id link_id)

  Pone un norte en el espacio.
- STATUS space_set_south (Space *space, Id link_id)

  Pone un sur en el espacio.
- STATUS space_set_east (Space *space, Id link_id)

  Pone un este en el espacio.
- STATUS space_set_west (Space *space, Id link_id)

  Pone un oeste en el espacio.
- STATUS space_set_up (Space *space, Id link_id)

  Pone un up en el espacio.
- STATUS space_set_down (Space *space, Id link_id)

  Pone un down en el espacio.
- Id space_get_id (Space *space)

  Obtienes el id del espacio.
- const char * space_get_name (Space *space)

  Obtienes el nombre del espacio.
- const char * space_get_description (Space *space)

  Obtiene la descripcion de un espacio.
- const char * space_get_long_description (Space *space)

  Obtiene la descripción detallada de un espacio.
- Space * space_get_graphics (Space *space, char dest[][G_COLUMNS])

  Devuelve la descripción grafica de la casilla.
- Set * space_get_objects (Space *space)

  Obtienes la lista de los objetos del espacio.

- Id space_get_north (Space *space)
  Obtienes el norte del espacio.
- Id space_get_south (Space *space)
  Obtienes el sur del espacio.
- Id space_get_east (Space *space)
  Obtienes el este del espacio.
- Id space_get_west (Space *space)
  Obtienes el oeste del espacio.
- Id space_get_up (Space *space)
  Obtienes la dirección hacia arriba del espacio.
- Id space_get_down (Space *space)
  Obtienes la dirección hacia abajo del espacio.
- STATUS space_add_object (Space *space, Id object_id)
  Añades el objeto al espacio.
- STATUS space_remove_object (Space *space, Id object_id)
  Eliminas el objeto del espacio.
- BOOL space_contains_object (Space *space, Id object_id)
  Compruebas si el espacio tiene objeto.
- BOOL space_graphics_areEmpty (Space *space)
  Devuelve si el espacio tiene descripcion grafica o no.
- STATUS space_set_iluminated (Space *space, BOOL iluminated)
  Asigna la iluminación al espacio.
- BOOL space_get_iluminated (Space *space)
  Devuelve si el espacio está iluminado.
- int space_print_graphics (Space *space)
  Imprime el espacio.
- STATUS space_print (Space *space)
  Imprimes lo que hay en el espacio.

4.55.1. Descripción detallada

Implementa el TAD space Define los campos de la estructura Space. Tambien define todas las funciones necesarias para trabajar con este tipo de datos.

Autor

Profesores PPROG
Javier Bernardo
Mihai Blidaru
Laura Bernal
Sandra Benitez

Versión

2.0

Fecha

20/02/2017

4.55.2. Documentación de las funciones

4.55.2.1. STATUS space_add_object (Space * *space,* Id *object_id*)

Añades el objeto al espacio.

Parámetros

space	Un puntero al espacio y una id del objeto.
object_id	Id del objeto.

Devuelve

Ok si se añade el objeto o ERROR si no.

4.55.2.2. BOOL space_contains_object (Space * *space,* Id *object_id*)

Compruebas si el espacio tiene objeto.

Parámetros

space	Un puntero al espacio.
object_id	Id del objeto.

Devuelve

funcion que coloca objeto.

4.55.2.3. Space* space_create (Id *space_id*)

Crea un nuevo espacio y le asigna un Id y unos valores por defecto.

Parámetros

space_id	Un Id del nuevo espacio.

Devuelve

Un puntero al nuevo espacio. Devuelve NULL si no se ha podido crear.

4.55.2.4. STATUS space_destroy (Space * *space*)

Libera la memoria de un espacio.

Parámetros

space	Un puntero al espacio que se quiere destruir

Devuelve

OK si se ha liberado correctamente, si no ERROR

4.55.2.5. const char* space_get_description (Space * *space*)

Obtiene la descripcion de un espacio.

Parámetros

space	Un puntero al espacio al que se le quiere poner nombre.

Devuelve

La descripcion del espacio.

4.55.2.6. Id space_get_down (Space * *space*)

Obtienes la dirección hacia abajo del espacio.

Parámetros

space	Un puntero al espacio.

Devuelve

La dirección hacia abajo del espacio.

4.55.2.7. Id space_get_east (Space * *space*)

Obtienes el este del espacio.

Parámetros

space	Un puntero al espacio.

Devuelve

El este del espacio.

4.55.2.8. Space* space_get_graphics (Space * *space,* char *dest[][G_COLUMNS]*)

Devuelve la descripción grafica de la casilla.

Parámetros

in	*space*	Un puntero al espacio.
out	*dest*	Matriz donde guardar la descripción grafica.

Devuelve

Un puntero a la descripción grafica del espacio o NULL si hay algun error.

4.55.2.9. Id space_get_id (Space * *space*)

Obtienes el id del espacio.

Parámetros

space	Un puntero al espacio.

Devuelve

El id del espacio.

4.55.2.10. BOOL space_get_iluminated (Space * *space*)

Devuelve si el espacio está iluminado.

Parámetros

space	Un puntero al espacio.

Devuelve

Devuelve el espacio iluminado

4.55.2.11. const char* space_get_long_description (Space * *space*)

Obtiene la descripción detallada de un espacio.

Parámetros

space	Un puntero al espacio al que se le quiere poner nombre.

Devuelve

La descripción detallada del espacio.

4.55.2.12. const char* space_get_name (Space * *space*)

Obtienes el nombre del espacio.

Parámetros

space	Un puntero al espacio.

Devuelve

El nombre del espacio.

4.55.2.13. Id space_get_north (Space * *space*)

Obtienes el norte del espacio.

Parámetros

space	Un puntero al espacio.

Devuelve

El norte del espacio.

4.55.2.14. Set* space_get_objects (Space * *space*)

Obtienes la lista de los objetos del espacio.

Parámetros

space	Un puntero al espacio.

Devuelve

La lista de los objetos del espacio.

4.55.2.15. Id space_get_south (Space * *space*)

Obtienes el sur del espacio.

Parámetros

space	Un puntero al espacio.

Devuelve

El sur del espacio.

4.55.2.16. Id space_get_up (Space * *space*)

Obtienes la dirección hacia arriba del espacio.

Parámetros

space	Un puntero al espacio.

Devuelve

La dirección hacia arriba del espacio.

4.55.2.17. Id space_get_west (Space * *space*)

Obtienes el oeste del espacio.

Parámetros

space	Un puntero al espacio.

Devuelve

El oeste del espacio.

4.55.2.18. BOOL space_graphics_areEmpty (Space * *space*)

Devuelve si el espacio tiene descripcion grafica o no.

Parámetros

space	Un puntero al espacio.

Devuelve

TRUE si los graficos están vacios o FALSE en caso contrario.

4.55.2.19. STATUS space_print (Space * *space*)

Imprimes lo que hay en el espacio.

Parámetros

space	Un puntero al espacio.

Devuelve

El contenido del espacio.

4.55.2.20. int space_print_graphics (Space * *space*)

Imprime el espacio.

Parámetros

space	Un puntero al espacio.

Devuelve

Impresion del objeto.

4.55.2.21. STATUS space_remove_object (Space * *space,* Id *object_id*)

Eliminas el objeto del espacio.

Parámetros

space	Un puntero al espacio.
object_id	Identificador del objeto.

Devuelve

Ok si se elimina o ERROR si no.

4.55.2.22. STATUS space_set_description (Space * *space,* char * *description*)

Pone nombre una descripcion al espacio.

Parámetros

space	Un puntero al espacio al que se le quiere poner nombre.
description	La descripcion que se le quiere poner.

Devuelve

OK si se ha asignado correctamente el nombre, si no ERROR.

4.55.2.23. STATUS space_set_down (Space * *space,* Id *link_id*)

Pone un down en el espacio.

Parámetros

space	Un puntero al espacio.
link_id	Un Id que determina el espacio.

Devuelve

OK si se ha realizado correctamente. Sino devuelve ERROR.

4.55.2.24. STATUS space_set_east (Space * *space,* Id *link_id*)

Pone un este en el espacio.

Parámetros

space	Un puntero al espacio.
link_id	Un Id que determina el espacio.

Devuelve

OK si se ha realizado correctamente. Sino devuelve ERROR.

4.55.2.25. Space* space_set_graphics (Space * *space,* char *graphics[][G_COLUMNS]*)

Pone la descripcion de la casilla.

Parámetros

space	Un puntero al espacio.
graphics	Una matriz que contiene la descripción grafica.

Devuelve

El espacio.

4.55.2.26. STATUS space_set_iluminated (Space * *space,* BOOL *iluminated*)

Asigna la iluminación al espacio.

Parámetros

space	Un puntero al espacio
iluminated	variable de tipo BOOL

Devuelve

Devuelve OK si se realiza correctamente

4.55.2.27. STATUS space_set_long_description (Space * *space,* char * *description*)

Pone una descripcion detallada al espacio.

Parámetros

space	Un puntero al espacio al que se le quiere poner nombre.
description	La descripcion que se le quiere poner.

Devuelve

OK si se ha asignado correctamente el nombre, si no ERROR.

4.55.2.28. STATUS space_set_name (Space * *space,* char * *name*)

Pone nombre a un espacio.

Parámetros

space	Un puntero al espacio al que se le quiere poner nombre.
name	El nombre que se le quiere poner.

Devuelve

OK si se ha asignado correctamente el nombre, si no ERROR.

4.55.2.29. STATUS space_set_north (Space * *space,* Id *link_id*)

Pone un norte en el espacio.

Parámetros

space	Un puntero al espacio
link_id	Un Id que determina el espacio

Devuelve

OK si se ha realizado correctamente. Sino devuelve ERROR

4.55.2.30. STATUS space_set_south (Space * *space,* Id *link_id*)

Pone un sur en el espacio.

Parámetros

space	Un puntero al espacio
link_id	Un Id que determina el espacio

Devuelve

OK si se ha realizado correctamente. Sino devuelve ERROR

4.55.2.31. STATUS space_set_up (Space * *space,* Id *link_id*)

Pone un up en el espacio.

Parámetros

space	Un puntero al espacio.
link_id	Un Id que determina el espacio.

Devuelve

OK si se ha realizado correctamente. Sino devuelve ERROR.

4.55.2.32. STATUS space_set_west (Space * *space,* Id *link_id*)

Pone un oeste en el espacio.

Parámetros

space	Un puntero al espacio
link_id	Un Id que determina el espacio

Devuelve

OK si se ha realizado correctamente. Sino devuelve ERROR

4.56. Referencia del Archivo src/space_test.c

It tests space module.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "space.h"
#include "tests/space_test.h"
#include "tests/test.h"
```

Funciones

- void test1_space_create ()
- void test2_space_create ()
- void test3_space_create ()
- void test1_space_set_name ()
- void test2_space_set_name ()
- void test3_space_set_name ()
- void test1_space_set_description ()
- void test2_space_set_description ()
- void test3_space_set_description ()
- void test1_space_set_long_description ()
- void test2_space_set_long_description ()
- void test3_space_set_long_description ()
- void test1_space_set_graphics ()
- void test2_space_set_graphics ()
- void test1_space_set_north ()
- void test2_space_set_north ()
- void test3_space_set_north ()
- void test1_space_set_south ()
- void test2_space_set_south ()
- void test3_space_set_south ()
- void test1_space_set_east ()
- void test2_space_set_east ()
- void test3_space_set_east ()
- void test1_space_set_west ()
- void test2_space_set_west ()
- void test3_space_set_west ()
- void test1_space_set_up ()
- void test2_space_set_up ()
- void test3_space_set_up ()
- void test1_space_set_down ()
- void test2_space_set_down ()
- void test3_space_set_down ()

- void test1_space_get_id ()
- void test2_space_get_id ()
- void test1_space_get_name ()
- void test2_space_get_name ()
- void test3_space_get_name ()
- void test1_space_get_description ()
- void test2_space_get_description ()
- void test3_space_get_description ()
- void test1_space_get_long_description ()
- void test2_space_get_long_description ()
- void test3_space_get_long_description ()
- void test1_space_get_graphics ()
- void test2_space_get_graphics ()
- void test1_space_get_objects ()
- void test2_space_get_objects ()
- void test1_space_get_north ()
- void test2_space_get_north ()
- void test3_space_get_north ()
- void test1_space_get_south ()
- void test2_space_get_south ()
- void test3_space_get_south ()
- void test1_space_get_east ()
- void test2_space_get_east ()
- void test3_space_get_east ()
- void test1_space_get_west ()
- void test2_space_get_west ()
- void test3_space_get_west ()
- void test1_space_get_up ()
- void test2_space_get_up ()
- void test3_space_get_up ()
- void test1_space_get_down ()
- void test2_space_get_down ()
- void test3_space_get_down ()
- void test1_space_add_object ()
- void test2_space_add_object ()
- void test3_space_add_object ()
- void test1_space_remove_object ()
- void test2_space_remove_object ()
- void test3_space_remove_object ()
- void test4_space_remove_object ()
- void test1_space_contains_object ()
- void test2_space_contains_object ()
- void test3_space_contains_object ()
- void test1_space_graphics_areEmpty ()
- void test2_space_graphics_areEmpty ()
- void test3_space_graphics_areEmpty ()
- void test1_space_set_iluminated ()
- void test2_space_set_iluminated ()
- void test1_space_get_iluminated ()
- void test2_space_get_iluminated ()

4.56.1. Descripción detallada

It tests space module.

Autor

Profesores Pprog

Versión

2.0

Fecha

16-01-2015

Copyright

GNU Public License

4.56.2. Documentación de las funciones

4.56.2.1. void test1_space_add_object ( )

Prueba Prueba la función que añade un objeto a una casilla

Precondición

El espacio es un puntero a NULL.

Postcondición

La salida debe ser ERROR.

4.56.2.2. void test1_space_contains_object ( )

Prueba Prueba la función que comprueba si una casilla contiene un objeto

Precondición

El espacio se ha inicializado y se ha añadido un objeto

Postcondición

La salida esperada es TRUE

4.56.2.3. void test1_space_create ( )

Número máximo de pruebas para el módulo space

4.56.2.4. void test1_space_get_description ( )

Prueba Prueba la función que devuelve la descripción de una casilla

Precondición

Al espacio se le ha establecido una descripción previamente

Postcondición

La salda debe ser la descripción previamente establecida

4.56.2.5. void test1_space_get_down ( )

Prueba Prueba la función que devuelve el enlace inferior de una casilla

Precondición

Se ha añadido previamente un enlace on id 7

Postcondición

La salida debe ser el id del enlace 7

4.56.2.6. void test1_space_get_east ( )

Prueba Prueba la función que devuelve el enlace este de una casilla

Precondición

Se ha añadido previamente un enlace on id 7

Postcondición

La salida debe ser el id del enlace 7

4.56.2.7. void test1_space_get_graphics ( )

Prueba Prueba la función que devuelve los graficos de una casilla

Precondición

A la casilla se le han establecido graficos previamente

Postcondición

La salida tiene que ser igual a los graficos establecidos previamente

4.56.2.8. void test1_space_get_id ( )

Prueba Prueba la función que devuelve el id de una casilla

Precondición

Al espacio se le ha establecido un id (12)

Postcondición

La salida esperada es el id establecido 12

4.56.2.9. void test1_space_get_iluminated ( )

Prueba Prueba la funcion que devuelve el estado de la iluminacion en una casilla

Precondición

Se ha creado e iluminado la casilla

Postcondición

La salida esperada es TRUE

4.56.2.10. void test1_space_get_long_description ( )

Prueba Prueba la función que devuelve la descripción larga de una casilla

Precondición

Al espacio se le ha establecido una descripción previamente

Postcondición

La salda debe ser la descripción previamente establecida

4.56.2.11. void test1_space_get_name ( )

Prueba Prueba la función que devuelve el nombre de una casilla

Precondición

Al espacio se le ha establecido un nombre previamente

Postcondición

La salda debe ser el nombre previamente establecido

4.56.2.12. void test1_space_get_north ( )

Prueba Prueba la función que devuelve el enlace norte de una casilla

Precondición

Se ha añadido previamente un enlace on id 7

Postcondición

La salida debe ser el id del enlace 7

4.56.2.13. void test1_space_get_objects ( )

Prueba Prueba la función que devuelve el set de objetos de una casilla

Precondición

El espacio ha sido previamente inicializado.

Postcondición

La salida debe ser diferente de NULL.

4.56.2.14. void test1_space_get_south ( )

Prueba Prueba la función que devuelve el enlace sur de una casilla

Precondición

Se ha añadido previamente un enlace on id 7

Postcondición

La salida debe ser el id del enlace 7

4.56.2.15. void test1_space_get_up ( )

Prueba Prueba la función que devuelve el enlace superior de una casilla

Precondición

Se ha añadido previamente un enlace on id 7

Postcondición

La salida debe ser el id del enlace 7

4.56.2.16. void test1_space_get_west ( )

Prueba Prueba la función que devuelve el enlace oeste de una casilla

Precondición

Se ha añadido previamente un enlace on id 7

Postcondición

La salida debe ser el id del enlace 7

4.56.2.17. void test1_space_graphics_areEmpty ( )

Prueba Prueba la función que comprueba si los graficos de una casilla están vacios.

Precondición

El espacio se ha creado y se le han establecido unos graficos

Postcondición

La salida esperada es FALSE

4.56.2.18. void test1_space_remove_object ( )

Prueba Prueba la función que quita un objeto de una casilla

Precondición

Se añade un objeto previamente

Postcondición

La salida debe ser OK.

4.56.2.19. void test1_space_set_description ( )

Prueba Prueba la función para establecer la descripción de un espacio

Precondición

Nombre que establecer al espacio

Postcondición

La salida debe ser OK

4.56.2.20. void test1_space_set_down ( )

Prueba Prueba la función que establece el enlace inferior de un espacio

Precondición

A la función se le pasa el identificador 4

Postcondición

La salida esperada es OK

4.56.2.21. void test1_space_set_east ( )

Prueba Prueba la función que establece el enlace este de un espacio

Precondición

A la función se le pasa el identificador 4

Postcondición

La salida esperada es OK

4.56.2.22. void test1_space_set_graphics ( )

Prueba Prueba la función que establece los graficos de una casilla

Precondición

Unos graficos aleatorios

Postcondición

La salida esperada es un puntero al espacio

4.56.2.23. void test1_space_set_iluminated ( )

Prueba Prueba la funcion que pone el estado de la iluminacion en una casilla

Precondición

Se ha creado una casiilla con id 5

Postcondición

La salida esperada es OK

4.56.2.24. void test1_space_set_long_description ( )

Prueba Prueba la función para establecer la descripción larga de un espacio

Precondición

Nombre que establecer al espacio

Postcondición

La salida debe ser OK

4.56.2.25. void test1_space_set_name ( )

Prueba Prueba la función para establecer el nombre de un espacio

Precondición

Nombre que establecer al espacio

Postcondición

La salida debe ser OK

4.56.2.26. void test1_space_set_north ( )

Prueba Prueba la función que establece el enlace norte de un espacio

Precondición

Id del enlace

Postcondición

La salida esperada es OK

4.56.2.27. void test1_space_set_south ( )

Prueba Prueba la función que establece el enlace sur de un espacio

Precondición

A la función se le pasa el identificador 4

Postcondición

La salida esperada es OK

4.56.2.28. void test1_space_set_up ( )

Prueba Prueba la función que establece el enlace superior de un espacio

Precondición

A la función se le pasa el identificador 4

Postcondición

La salida esperada es OK

4.56.2.29. void test1_space_set_west ( )

Prueba Prueba la función que establece el enlace oeste de un espacio

Precondición

A la función se le pasa el identificador 4

Postcondición

La salida esperada es OK

4.56.2.30. void test2_space_add_object ( )

Prueba Prueba la función que añade un objeto a una casilla

Precondición

Se añade un objeto con id 10

Postcondición

La salida debe ser OK.

4.56.2.31. void test2_space_contains_object ( )

Prueba Prueba la función que comprueba si una casilla contiene un objeto

Precondición

El espacio no se ha inicializado.

Postcondición

La salida esperada es FALSE

4.56.2.32. void test2_space_create ( )

Prueba Prueba la función de creación de un espacio

Precondición

Un identificador como parámetro

Postcondición

El identificador del espacio es el introducido

4.56.2.33. void test2_space_get_description ( )

Prueba Prueba la función que devuelve la descripción de una casilla

Precondición

El espacio es un puntero a NULL

Postcondición

La salda debe ser un puntero a NULL

4.56.2.34. void test2_space_get_down ( )

Prueba Prueba la función que devuelve el enlace inferior de una casilla

Precondición

El espacio es un puntero a NULL.

Postcondición

La salida debe ser NO_ID.

4.56.2.35. void test2_space_get_east ( )

Prueba Prueba la función que devuelve el enlace este de una casilla

Precondición

El espacio es un puntero a NULL.

Postcondición

La salida debe ser NO_ID.

4.56.2.36. void test2_space_get_graphics ( )

Prueba Prueba la función que devuelve los graficos de una casilla

Precondición

El espacio ha sido inicializado pero no se le han establecido graficos

Postcondición

La salida debe ser una matriz de 3 cadenas de 7 caracteres llenas de espacios

4.56.2.37. void test2_space_get_id ( )

Prueba Prueba la función que devuelve el id de una casilla

Precondición

El espacio es un puntero a NULL

Postcondición

La salida esperada es NO_ID

4.56.2.38. void test2_space_get_iluminated ( )

Prueba Prueba la funcion que devuelve el estado de la iluminacion en una casilla

Precondición

No se ha creado la casilla

Postcondición

La salida esperada es FALSE

4.56.2.39. void test2_space_get_long_description ( )

Prueba Prueba la función que devuelve la descripción larga de una casilla

Precondición

El espacio es un puntero a NULL

Postcondición

La salda debe ser un puntero a NULL

4.56.2.40. void test2_space_get_name ( )

Prueba Prueba la función que devuelve el nombre de una casilla

Precondición

El espacio es un puntero a NULL

Postcondición

La salda debe ser un puntero a NULL

4.56.2.41. void test2_space_get_north ( )

Prueba Prueba la función que devuelve el enlace norte de una casilla

Precondición

El espacio es un puntero a NULL.

Postcondición

La salida debe ser NO_ID.

4.56.2.42. void test2_space_get_objects ( )

Prueba Prueba la función que devuelve el set de objetos de una casilla

Precondición

El espacio es un puntero a NULL

Postcondición

La salida debe ser NULL

4.56.2.43. void test2_space_get_south ( )

Prueba Prueba la función que devuelve el enlace sur de una casilla

Precondición

El espacio es un puntero a NULL.

Postcondición

La salida debe ser NO_ID.

4.56.2.44. void test2_space_get_up ( )

Prueba Prueba la función que devuelve el enlace superior de una casilla

Precondición

El espacio es un puntero a NULL.

Postcondición

La salida debe ser NO_ID.

4.56.2.45. void test2_space_get_west ( )

Prueba Prueba la función que devuelve el enlace oeste de una casilla

Precondición

El espacio es un puntero a NULL.

Postcondición

La salida debe ser NO_ID.

4.56.2.46. void test2_space_graphics_areEmpty ( )

Prueba Prueba la función que comprueba si los graficos de una casilla están vacios.

Precondición

El espacio no se ha inicializado

Postcondición

La salida esperada es TRUE.

4.56.2.47. void test2_space_remove_object ( )

Prueba Prueba la función que quita un objeto de una casilla

Precondición

Todavia no he ha añadido ningun objeto

Postcondición

La salida debe ser ERROR.

4.56.2.48. void test2_space_set_description ( )

Prueba Prueba la función para establecer la descripción de un espacio

Precondición

El espacio al que establecer la descripción es un puntero a NULL

Postcondición

La salida debe ser ERROR

4.56.2.49. void test2_space_set_down ( )

Prueba Prueba la función que establece el enlace inferior de un espacio

Precondición

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

Postcondición

La salida esperada es ERROR

4.56.2.50. void test2_space_set_east ( )

Prueba Prueba la función que establece el enlace este de un espacio

Precondición

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

Postcondición

La saida esperada es ERROR

4.56.2.51. void test2_space_set_graphics ( )

Prueba Prueba la función que establece los graficos de una casilla

Precondición

El espacio es un puntero a NULL

Postcondición

La salida esperada es un puntero a NULL

4.56.2.52. void test2_space_set_iluminated ( )

Prueba Prueba la funcion que pone el estado de la iluminacion en una casilla

Precondición

No se ha creado la casilla

Postcondición

La salida esperada es ERROR

4.56.2.53. void test2_space_set_long_description ( )

Prueba Prueba la función para establecer la descripción larga de un espacio

Precondición

El espacio al que establecer la descripción es un puntero a NULL

Postcondición

La salida debe ser ERROR

4.56.2.54. void test2_space_set_name ( )

Prueba Prueba la función para establecer el nombre de un espacio

Precondición

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

Postcondición

La salida debe ser ERROR

4.56.2.55. void test2_space_set_north ( )

Prueba Prueba la función que establece el enlace norte de un espacio

Precondición

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

Postcondición

La salida esperada es ERROR

4.56.2.56. void test2_space_set_south ( )

Prueba Prueba la función que establece el enlace sur de un espacio

Precondición

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

Postcondición

4.56.2.57. void test2_space_set_up ( )

Prueba Prueba la función que establece el enlace superior de un espacio

Precondición

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

Postcondición

La salida esperada es ERROR

4.56.2.58. void test2_space_set_west ( )

Prueba Prueba la función que establece el enlace oeste de un espacio

Precondición

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

Postcondición

La salida esperada es ERROR

4.56.2.59. void test3_space_add_object ( )

Prueba Prueba la función que añade un objeto a una casilla

Precondición

El id del objeto es -1 (NO_ID)

Postcondición

La salida debe ser ERROR.

4.56.2.60. void test3_space_contains_object ( )

Prueba Prueba la función que comprueba si una casilla contiene un objeto

Precondición

El espacio se ha creado pero no se ha añadido ningun objeto

Postcondición

La salida esperada es FALSE

4.56.2.61. void test3_space_create ( )

Prueba Prueba la función de creación de un espacio

Precondición

El identificador del espacio es NO_ID

Postcondición

La salida esperada es un puntero a NULL

4.56.2.62. void test3_space_get_description ( )

Prueba Prueba la función que devuelve la descripción de una casilla

Precondición

El espacio ha sido inicializado pero no se le ha establecido ninguna descripción

Postcondición

La salida debe ser una cadena vacila

4.56.2.63. void test3_space_get_down ( )

Prueba Prueba la función que devuelve el enlace inferior de una casilla

Precondición

No se ha establecido ningun enlace previamente

Postcondición

La salida debe ser NO_ID.

4.56.2.64. void test3_space_get_east ( )

Prueba Prueba la función que devuelve el enlace este de una casilla

Precondición

No se ha establecido ningun enlace previamente

Postcondición

La salida debe ser NO_ID.

4.56.2.65. void test3_space_get_long_description ( )

Prueba Prueba la función que devuelve la descripción larga de una casilla

Precondición

El espacio ha sido inicializado pero no se le ha establecido ninguna descripción

Postcondición

La salida debe ser una cadena vacila

4.56.2.66. void test3_space_get_name ( )

Prueba Prueba la función que devuelve el nombre de una casilla

Precondición

El espacio ha sido inicializado pero no se le ha establecido ningun nombre

Postcondición

La salida debe ser una cadena vacila

4.56.2.67. void test3_space_get_north ( )

Prueba Prueba la función que devuelve el enlace norte de una casilla

Precondición

No se ha establecido ningun enlace previamente

Postcondición

La salida debe ser NO_ID.

4.56.2.68. void test3_space_get_south ( )

Prueba Prueba la función que devuelve el enlace sur de una casilla

Precondición

No se ha establecido ningun enlace previamente

Postcondición

La salida debe ser NO_ID.

4.56.2.69. void test3_space_get_up ( )

Prueba Prueba la función que devuelve el enlace superior de una casilla

Precondición

No se ha establecido ningun enlace previamente

Postcondición

La salida debe ser NO_ID.

4.56.2.70. void test3_space_get_west ( )

Prueba Prueba la función que devuelve el enlace oeste de una casilla

Precondición

No se ha establecido ningun enlace previamente

Postcondición

La salida debe ser NO_ID.

4.56.2.71. void test3_space_graphics_areEmpty ( )

Prueba Prueba la función que comprueba si los graficos de una casilla están vacios.

Precondición

El espacio se ha creado pero no se le han establecido graficos

Postcondición

La salida esperada es TRUE

4.56.2.72. void test3_space_remove_object ( )

Prueba Prueba la función que quita un objeto de una casilla

Precondición

El espacio no se ha inicializado previamente

Postcondición

La salida debe ser ERROR.

4.56.2.73. void test3_space_set_description ( )

Prueba Prueba la función para establecer la descripción de un espacio

Precondición

El espacio es un puntero no NULL, pero la descripción a establecer es NULL

Postcondición

La salida debe ser ERROR

4.56.2.74. void test3_space_set_down ( )

Prueba Prueba la función que establece el enlace inferior de un espacio

Precondición

El id que se le pasa a la función es NO_ID

Postcondición

La salida esperada es ERROR

4.56.2.75. void test3_space_set_east ( )

Prueba Prueba la función que establece el enlace este de un espacio

Precondición

El id que se le pasa a la función es NO_ID

Postcondición

La salida esperada es ERROR

4.56.2.76. void test3_space_set_long_description ( )

Prueba Prueba la función para establecer la descripción larga de un espacio

Precondición

El espacio es un puntero no NULL, pero la descripción a establecer es NULL

Postcondición

La salida debe ser ERROR

4.56.2.77. void test3_space_set_name ( )

Prueba Prueba la función para establecer el nombre de un espacio

Precondición

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

Postcondición

La salida debe ser ERROR

4.56.2.78. void test3_space_set_north ( )

Prueba Prueba la función que establece el enlace norte de un espacio

Precondición

El id que se le pasa a la función es NO_ID

Postcondición

La salida esperada es ERROR

4.56.2.79. void test3_space_set_south ( )

Prueba Prueba la función que establece el enlace sur de un espacio

Precondición

El id que se le pasa a la función es NO_ID

Postcondición

La salida esperada es ERROR

4.56.2.80. void test3_space_set_up ( )

Prueba Prueba la función que establece el enlace superior de un espacio

Precondición

El id que se le pasa a la función es NO_ID

Postcondición

La salida esperada es ERROR

4.56.2.81. void test3_space_set_west ( )

Prueba Prueba la función que establece el enlace oeste de un espacio

Precondición

El id que se le pasa a la función es NO_ID

Postcondición

La salida esperada es ERROR

4.56.2.82. void test4_space_remove_object ( )

Prueba Prueba la función que quita un objeto de una casilla

Precondición

El objeto ya ha sido eliminado previamente

Postcondición

La salida debe ser ERROR.

Índice alfabético

Area
    screen.h, 112

BAD_ARGUMENTS
    game_management.h, 85

CLOSED
    link.h, 94
Command
    command.h, 62
command.h
    GO, 63
    INSPECT, 63
    LEAVE, 62
    NO_CMD, 62
    OPEN, 63
    QUIT, 62
    ROLL, 63
    TAKE, 62
    TURNOFF, 63
    TURNON, 63
    UNKNOWN, 62
command.c
    Command_clear, 206
    Command_destroy, 206
    Command_get_cmd, 206
    Command_get_cmd_arg, 206
    Command_ini, 207
    Command_set_cmd, 207
    Command_set_cmd_arg, 207
    get_user_input, 208
command.h
    Command, 62
    Command_clear, 63
    Command_destroy, 63
    Command_get_cmd, 63
    Command_get_cmd_arg, 64
    Command_ini, 64
    Command_set_cmd, 64
    Command_set_cmd_arg, 64
    enum_Command, 62
    get_user_input, 65
Command_clear
    command.c, 206
    command.h, 63
Command_destroy
    command.c, 206
    command.h, 63
Command_get_cmd
    command.c, 206
    command.h, 63
Command_get_cmd_arg
    command.c, 206
    command.h, 64
Command_ini
    command.c, 207
    command.h, 64
Command_set_cmd
    command.c, 207
    command.h, 64
Command_set_cmd_arg
    command.c, 207
    command.h, 64
command_test.c
    test1_command_clear, 209
    test1_command_destroy, 209
    test1_command_get_cmd, 209
    test1_command_get_cmd_arg, 210
    test1_command_ini, 210
    test1_command_set_cmd, 210
    test1_command_set_cmd_arg, 210
    test2_command_clear, 210
    test2_command_destroy, 211
    test2_command_get_cmd, 211
    test2_command_get_cmd_arg, 211
    test2_command_set_cmd, 211
    test2_command_set_cmd_arg, 211
    test3_command_get_cmd, 212
    test3_command_get_cmd_arg, 212
    test3_command_set_cmd, 212
    test3_command_set_cmd_arg, 212
    test4_command_set_cmd_arg, 212
command_test.h
    test1_command_clear, 127
    test1_command_destroy, 127
    test1_command_get_cmd, 127
    test1_command_get_cmd_arg, 128
    test1_command_ini, 128
    test1_command_set_cmd, 128
    test1_command_set_cmd_arg, 128
    test2_command_clear, 128
    test2_command_destroy, 129
    test2_command_get_cmd, 129
    test2_command_get_cmd_arg, 129
    test2_command_set_cmd, 129
    test2_command_set_cmd_arg, 129
    test3_command_get_cmd, 130
    test3_command_get_cmd_arg, 130
    test3_command_set_cmd, 130

test3_command_set_cmd_arg, 130
test4_command_set_cmd_arg, 130

DIE_ID
die.h, 73
Dialogue
dialogue.h, 67
dialogue.h
SAVE_PROTECTED_FILE, 68
SAVE_SAVE_OK, 68
SAVE_WRITE_FAILED, 68
dialogue.c
dialogue_attack, 214
dialogue_destroy, 214
dialogue_dir, 215
dialogue_game_rule, 215
dialogue_get_text, 215
dialogue_go, 215
dialogue_help, 216
dialogue_ini, 216
dialogue_inspect, 216
dialogue_leave, 216
dialogue_load, 217
dialogue_open, 217
dialogue_save, 217
dialogue_take, 217
dialogue_turn_off, 218
dialogue_turn_on, 218
dialogue_unknown, 218
dialogue.h
Dialogue, 67
dialogue_attack, 68
dialogue_destroy, 68
dialogue_dir, 68
dialogue_game_rule, 68
dialogue_get_text, 69
dialogue_go, 69
dialogue_help, 69
dialogue_ini, 69
dialogue_inspect, 70
dialogue_leave, 70
dialogue_load, 70
dialogue_open, 70
dialogue_save, 71
dialogue_take, 71
dialogue_turn_off, 71
dialogue_turn_on, 71
dialogue_unknown, 72
GLOBAL_NO_ARGS, 67
dialogue_attack
dialogue.c, 214
dialogue.h, 68
dialogue_destroy
dialogue.c, 214
dialogue.h, 68
dialogue_dir
dialogue.c, 215
dialogue.h, 68
dialogue_game_rule
dialogue.c, 215
dialogue.h, 68
dialogue_get_text
dialogue.c, 215
dialogue.h, 69
dialogue_go
dialogue.c, 215
dialogue.h, 69
dialogue_help
dialogue.c, 216
dialogue.h, 69
dialogue_ini
dialogue.c, 216
dialogue.h, 69
dialogue_inspect
dialogue.c, 216
dialogue.h, 70
dialogue_leave
dialogue.c, 216
dialogue.h, 70
dialogue_load
dialogue.c, 217
dialogue.h, 70
dialogue_open
dialogue.c, 217
dialogue.h, 70
dialogue_save
dialogue.c, 217
dialogue.h, 71
dialogue_take
dialogue.c, 217
dialogue.h, 71
dialogue_test.c
test1_dialogue_attack, 220
test1_dialogue_dir, 220
test1_dialogue_game_rule, 220
test1_dialogue_get_text, 220
test1_dialogue_go, 220
test1_dialogue_help, 221
test1_dialogue_ini, 221
test1_dialogue_inspect, 221
test1_dialogue_leave, 221
test1_dialogue_load, 221
test1_dialogue_open, 222
test1_dialogue_save, 222
test1_dialogue_take, 222
test1_dialogue_turn_off, 222
test1_dialogue_turn_on, 222
test1_dialogue_unknown, 223
test2_dialogue_attack, 223
test2_dialogue_dir, 223
test2_dialogue_game_rule, 223
test2_dialogue_get_text, 223
test2_dialogue_go, 224
test2_dialogue_help, 224
test2_dialogue_inspect, 224
test2_dialogue_leave, 224
test2_dialogue_load, 224

test2_dialogue_open, 225
test2_dialogue_save, 225
test2_dialogue_take, 225
test2_dialogue_turn_off, 225
test2_dialogue_turn_on, 225
test2_dialogue_unknown, 226
dialogue_test.h
test1_dialogue_attack, 132
test1_dialogue_dir, 132
test1_dialogue_game_rule, 132
test1_dialogue_get_text, 132
test1_dialogue_go, 133
test1_dialogue_help, 133
test1_dialogue_ini, 133
test1_dialogue_inspect, 133
test1_dialogue_leave, 133
test1_dialogue_load, 134
test1_dialogue_open, 134
test1_dialogue_save, 134
test1_dialogue_take, 134
test1_dialogue_turn_off, 134
test1_dialogue_turn_on, 135
test1_dialogue_unknown, 135
test2_dialogue_attack, 135
test2_dialogue_dir, 135
test2_dialogue_game_rule, 135
test2_dialogue_get_text, 136
test2_dialogue_go, 136
test2_dialogue_help, 136
test2_dialogue_inspect, 136
test2_dialogue_leave, 136
test2_dialogue_load, 137
test2_dialogue_open, 137
test2_dialogue_save, 137
test2_dialogue_take, 137
test2_dialogue_turn_off, 137
test2_dialogue_turn_on, 138
test2_dialogue_unknown, 138
dialogue_turn_off
dialogue.c, 218
dialogue.h, 71
dialogue_turn_on
dialogue.c, 218
dialogue.h, 71
dialogue_unknown
dialogue.c, 218
dialogue.h, 72
Die
die.h, 73
die.c
die_create, 227
die_destroy, 227
die_get_faces, 227
die_get_number, 228
die_print, 228
die_roll, 228
die_set_faces, 228
die.h
DIE_ID, 73
Die, 73
die_create, 73
die_destroy, 73
die_get_faces, 74
die_get_number, 74
die_print, 74
die_roll, 74
die_set_faces, 75
die_create
die.c, 227
die.h, 73
die_destroy
die.c, 227
die.h, 73
die_get_faces
die.c, 227
die.h, 74
die_get_number
die.c, 228
die.h, 74
die_print
die.c, 228
die.h, 74
die_roll
die.c, 228
die.h, 74
die_set_faces
die.c, 228
die.h, 75

enum_Command
command.h, 62

GO
command.h, 63
G_COLUMNS
space.h, 118
G_ROWS
space.h, 118
GLOBAL_NO_ARGS
dialogue.h, 67
Game
game.h, 77
game.c
game_add_link, 231
game_add_object, 231
game_add_player, 232
game_add_space, 232
game_create, 232
game_destroy, 233
game_get_dialogue, 233
game_get_die, 233
game_get_last_inspected_object, 234
game_get_last_inspected_space, 234
game_get_link, 234
game_get_link_at, 235
game_get_obj_list_as_str, 235
game_get_object_at, 235

game_get_object_location, 236
game_get_player, 236
game_get_player_location, 236
game_get_space, 237
game_get_space_at, 237
game_is_over, 237
game_print_data, 238
game_update, 238
game.h
Game, 77
game_add_link, 77
game_add_object, 78
game_add_player, 78
game_add_space, 78
game_create, 79
game_destroy, 79
game_get_dialogue, 79
game_get_die, 80
game_get_last_inspected_object, 80
game_get_last_inspected_space, 80
game_get_link, 80
game_get_link_at, 81
game_get_obj_list_as_str, 81
game_get_object_at, 81
game_get_object_location, 82
game_get_player, 82
game_get_player_location, 82
game_get_space, 83
game_get_space_at, 83
game_is_over, 83
game_print_data, 84
game_update, 84
MAX_OBJECTS, 77
MAX_SPACES, 77
game_management.h
BAD_ARGUMENTS, 85
PROTECTED_FILE, 85
SAVE_OK, 85
WRITE_FAILED, 85
game_add_link
game.c, 231
game.h, 77
game_add_object
game.c, 231
game.h, 78
game_add_player
game.c, 232
game.h, 78
game_add_space
game.c, 232
game.h, 78
game_create
game.c, 232
game.h, 79
game_destroy
game.c, 233
game.h, 79
game_get_dialogue
game.c, 233
game.h, 79
game_get_die
game.c, 233
game.h, 80
game_get_last_inspected_object
game.c, 234
game.h, 80
game_get_last_inspected_space
game.c, 234
game.h, 80
game_get_link
game.c, 234
game.h, 80
game_get_link_at
game.c, 235
game.h, 81
game_get_obj_list_as_str
game.c, 235
game.h, 81
game_get_object_at
game.c, 235
game.h, 81
game_get_object_location
game.c, 236
game.h, 82
game_get_player
game.c, 236
game.h, 82
game_get_player_location
game.c, 236
game.h, 82
game_get_space
game.c, 237
game.h, 83
game_get_space_at
game.c, 237
game.h, 83
game_is_over
game.c, 237
game.h, 83
game_management.c
game_management_load, 240
game_management_save, 240
game_management_start_from_file, 240
game_management.h
game_management_load, 85
game_management_save, 86
game_management_start_from_file, 86
SAVE_STATUS, 85
game_management_load
game_management.c, 240
game_management.h, 85
game_management_save
game_management.c, 240
game_management.h, 86
game_management_start_from_file
game_management.c, 240

game_management.h, 86
game_management_test.c
test1_game_management_load, 242
test1_game_management_save, 242
test1_game_management_start_from_file, 242
test2_game_management_load, 242
test2_game_management_save, 242
test2_game_management_start_from_file, 242
game_management_test.h
test1_game_management_load, 139
test1_game_management_save, 139
test1_game_management_start_from_file, 139
test2_game_management_load, 139
test2_game_management_save, 140
test2_game_management_start_from_file, 140
game_print_data
game.c, 238
game.h, 84
game_rules.h
game_rules_run_random_rule, 87
game_rules_run_random_rule
game_rules.h, 87
game_rules_test.c
test1_game_rules_run_random_rule, 244
test2_game_rules_run_random_rule, 244
game_rules_test.h
test1_game_rules_run_random_rule, 141
test2_game_rules_run_random_rule, 141
game_test.c
test1_game_add_link, 246
test1_game_add_object, 246
test1_game_add_player, 246
test1_game_add_space, 246
test1_game_create, 247
test1_game_destroy, 247
test1_game_get_dialogue, 247
test1_game_get_die, 247
test1_game_get_last_inspected_object, 247
test1_game_get_last_inspected_space, 248
test1_game_get_link, 248
test1_game_get_link_at, 248
test1_game_get_obj_list_as_str, 248
test1_game_get_object_at, 248
test1_game_get_object_location, 249
test1_game_get_player, 249
test1_game_get_player_location, 249
test1_game_get_space, 249
test1_game_get_space_at, 249
test1_game_is_over, 250
test1_game_update, 250
test2_game_add_link, 250
test2_game_add_object, 250
test2_game_add_player, 250
test2_game_add_space, 251
test2_game_destroy, 251
test2_game_get_dialogue, 251
test2_game_get_die, 251
test2_game_get_last_inspected_object, 251
test2_game_get_last_inspected_space, 252
test2_game_get_link, 252
test2_game_get_link_at, 252
test2_game_get_obj_list_as_str, 252
test2_game_get_object_at, 252
test2_game_get_object_location, 253
test2_game_get_player, 253
test2_game_get_player_location, 253
test2_game_get_space, 253
test2_game_get_space_at, 253
test2_game_is_over, 254
test2_game_update, 254
test3_game_add_link, 254
test3_game_add_object, 254
test3_game_add_player, 254
test3_game_add_space, 255
test3_game_get_last_inspected_object, 255
test3_game_get_link, 255
test3_game_get_link_at, 255
test3_game_get_object_at, 255
test3_game_get_object_location, 256
test3_game_get_space, 256
test3_game_get_space_at, 256
test3_game_update, 256
game_test.h
test1_game_add_link, 143
test1_game_add_object, 143
test1_game_add_player, 143
test1_game_add_space, 143
test1_game_create, 143
test1_game_destroy, 143
test1_game_get_dialogue, 144
test1_game_get_die, 144
test1_game_get_last_inspected_object, 144
test1_game_get_last_inspected_space, 144
test1_game_get_link, 144
test1_game_get_link_at, 145
test1_game_get_obj_list_as_str, 145
test1_game_get_object_at, 145
test1_game_get_object_location, 145
test1_game_get_player, 145
test1_game_get_player_location, 146
test1_game_get_space, 146
test1_game_get_space_at, 146
test1_game_is_over, 146
test1_game_update, 146
test2_game_add_link, 147
test2_game_add_object, 147
test2_game_add_player, 147
test2_game_add_space, 147
test2_game_destroy, 147
test2_game_get_dialogue, 148
test2_game_get_die, 148
test2_game_get_last_inspected_object, 148
test2_game_get_last_inspected_space, 148
test2_game_get_link, 148
test2_game_get_link_at, 149
test2_game_get_obj_list_as_str, 149

test2_game_get_object_at, 149
test2_game_get_object_location, 149
test2_game_get_player, 149
test2_game_get_player_location, 150
test2_game_get_space, 150
test2_game_get_space_at, 150
test2_game_is_over, 150
test2_game_update, 150
test3_game_add_link, 151
test3_game_add_object, 151
test3_game_add_player, 151
test3_game_add_space, 151
test3_game_get_last_inspected_object, 151
test3_game_get_link, 152
test3_game_get_link_at, 152
test3_game_get_object_at, 152
test3_game_get_object_location, 152
test3_game_get_space, 152
test3_game_get_space_at, 153
test3_game_update, 153
game_update
game.c, 238
game.h, 84
get_user_input
command.c, 208
command.h, 65
graphic_engine.h
graphic_engine_create, 89
graphic_engine_destroy, 89
graphic_engine_game_over, 89
graphic_engine_paint_directions, 89
graphic_engine_paint_game, 90
graphic_engine_paint_help, 90
graphic_engine_play_intro, 90
graphic_engine_create
graphic_engine.h, 89
graphic_engine_destroy
graphic_engine.h, 89
graphic_engine_game_over
graphic_engine.h, 89
graphic_engine_paint_directions
graphic_engine.h, 89
graphic_engine_paint_game
graphic_engine.h, 90
graphic_engine_paint_help
graphic_engine.h, 90
graphic_engine_play_intro
graphic_engine.h, 90
graphic_engine_test.c
test1_graphic_engine_create, 257
graphic_engine_test.h
test1_graphic_engine_create, 154

INSPECT
command.h, 63
include/command.h, 61
include/dialogue.h, 65
include/die.h, 72
include/game.h, 75
include/game_management.h, 84
include/game_rules.h, 86
include/graphic_engine.h, 87
include/inventory.h, 90
include/link.h, 93
include/object.h, 99
include/player.h, 106
include/screen.h, 111
include/set.h, 113
include/space.h, 116
include/tests/command_test.h, 126
include/tests/dialogue_test.h, 131
include/tests/game_management_test.h, 138
include/tests/game_rules_test.h, 140
include/tests/game_test.h, 141
include/tests/graphic_engine_test.h, 153
include/tests/inventory_test.h, 154
include/tests/link_test.h, 158
include/tests/object_test.h, 165
include/tests/player_test.h, 175
include/tests/screen_test.h, 183
include/tests/space_test.h, 184
include/tests/test.h, 203
include/types.h, 204
inventory.c
inventory_add_object, 258
inventory_create, 258
inventory_destroy, 259
inventory_get_max, 259
inventory_get_set, 259
inventory_print, 259
inventory_remove_object, 260
inventory_set_max, 260
inventory.h
inventory_add_object, 91
inventory_create, 91
inventory_destroy, 92
inventory_get_max, 92
inventory_get_set, 92
inventory_print, 92
inventory_remove_object, 93
inventory_set_max, 93
inventory_add_object
inventory.c, 258
inventory.h, 91
inventory_create
inventory.c, 258
inventory.h, 91
inventory_destroy
inventory.c, 259
inventory.h, 92
inventory_get_max
inventory.c, 259
inventory.h, 92
inventory_get_set
inventory.c, 259
inventory.h, 92
inventory_print

inventory.c, 259
inventory.h, 92
inventory_remove_object
inventory.c, 260
inventory.h, 93
inventory_set_max
inventory.c, 260
inventory.h, 93
inventory_test.c
test1_inventory_add_object, 261
test1_inventory_create, 261
test1_inventory_get_max, 261
test1_inventory_get_set, 261
test1_inventory_remove_object, 262
test1_inventory_set_max, 262
test2_inventory_add_object, 262
test2_inventory_get_max, 262
test2_inventory_get_set, 262
test2_inventory_remove_object, 263
test2_inventory_set_max, 263
test3_inventory_add_object, 263
test3_inventory_remove_object, 263
test3_inventory_set_max, 263
inventory_test.h
test1_inventory_add_object, 155
test1_inventory_create, 155
test1_inventory_get_max, 155
test1_inventory_get_set, 155
test1_inventory_remove_object, 155
test1_inventory_set_max, 156
test2_inventory_add_object, 156
test2_inventory_get_max, 156
test2_inventory_get_set, 156
test2_inventory_remove_object, 156
test2_inventory_set_max, 157
test3_inventory_add_object, 157
test3_inventory_remove_object, 157
test3_inventory_set_max, 157

LEAVE
command.h, 62
link.h
CLOSED, 94
OPENED, 94
link.c
link_create, 265
link_destroy, 265
link_get_dest_from, 265
link_get_id, 266
link_get_name, 266
link_get_space1, 266
link_get_space2, 266
link_get_state, 267
link_print, 267
link_set_id, 267
link_set_name, 268
link_set_space1, 268
link_set_space2, 268
link_set_state, 269
link.h
link_create, 95
link_destroy, 95
link_get_dest_from, 95
link_get_id, 95
link_get_name, 96
link_get_space1, 96
link_get_space2, 96
link_get_state, 97
link_print, 97
link_set_id, 97
link_set_name, 97
link_set_space1, 98
link_set_space2, 98
link_set_state, 98
MAX_LINK, 94
State, 94
link_create
link.c, 265
link.h, 95
link_destroy
link.c, 265
link.h, 95
link_get_dest_from
link.c, 265
link.h, 95
link_get_id
link.c, 266
link.h, 95
link_get_name
link.c, 266
link.h, 96
link_get_space1
link.c, 266
link.h, 96
link_get_space2
link.c, 266
link.h, 96
link_get_state
link.c, 267
link.h, 97
link_print
link.c, 267
link.h, 97
link_set_id
link.c, 267
link.h, 97
link_set_name
link.c, 268
link.h, 97
link_set_space1
link.c, 268
link.h, 98
link_set_space2
link.c, 268
link.h, 98
link_set_state
link.c, 269

link.h, 98
link_test.c
test1_link_create, 270
test1_link_get_dest_from, 270
test1_link_get_id, 270
test1_link_get_name, 270
test1_link_get_space1, 271
test1_link_get_space2, 271
test1_link_get_state, 271
test1_link_set_id, 271
test1_link_set_name, 271
test1_link_set_space1, 272
test1_link_set_space2, 272
test1_link_set_state, 272
test2_link_get_dest_from, 272
test2_link_get_id, 272
test2_link_get_name, 273
test2_link_get_space1, 273
test2_link_get_space2, 273
test2_link_get_state, 273
test2_link_set_id, 273
test2_link_set_name, 274
test2_link_set_space1, 274
test2_link_set_space2, 274
test2_link_set_state, 274
test3_link_get_dest_from, 274
test3_link_set_id, 275
test3_link_set_name, 275
test3_link_set_space1, 275
test3_link_set_space2, 275
test3_link_set_state, 275
test4_link_get_dest_from, 276
link_test.h
test1_link_create, 159
test1_link_get_dest_from, 159
test1_link_get_id, 159
test1_link_get_name, 159
test1_link_get_space1, 159
test1_link_get_space2, 160
test1_link_get_state, 160
test1_link_set_id, 160
test1_link_set_name, 160
test1_link_set_space1, 160
test1_link_set_space2, 161
test1_link_set_state, 161
test2_link_get_dest_from, 161
test2_link_get_id, 161
test2_link_get_name, 161
test2_link_get_space1, 162
test2_link_get_space2, 162
test2_link_get_state, 162
test2_link_set_id, 162
test2_link_set_name, 162
test2_link_set_space1, 163
test2_link_set_space2, 163
test2_link_set_state, 163
test3_link_get_dest_from, 163
test3_link_set_id, 163
test3_link_set_name, 164
test3_link_set_space1, 164
test3_link_set_space2, 164
test3_link_set_state, 164
test4_link_get_dest_from, 164

MAX_LINK
link.h, 94
MAX_OBJECTS
game.h, 77
MAX_SET
set.h, 114
MAX_SPACES
game.h, 77

NO_CMD
command.h, 62

OPEN
command.h, 63
OPENED
link.h, 94
object.c
object_Get_Description, 278
object_Get_Description2, 278
object_Get_Graphics, 278
object_Get_Hidden, 279
object_Get_Id, 279
object_Get_Illuminates, 279
object_Get_Light, 279
object_Get_Mobile, 280
object_Get_Moved, 280
object_Get_Name, 280
object_Get_Open, 280
object_Set_Description, 281
object_Set_Description2, 281
object_Set_Graphics, 281
object_Set_Hidden, 281
object_Set_Id, 282
object_Set_Illuminates, 282
object_Set_Light, 282
object_Set_Mobile, 282
object_Set_Moved, 283
object_Set_Name, 283
object_Set_Open, 283
object_create, 278
object_destroy, 278
object_print, 280
object.h
object_Get_Description, 101
object_Get_Description2, 101
object_Get_Graphics, 101
object_Get_Hidden, 101
object_Get_Id, 102
object_Get_Illuminates, 102
object_Get_Light, 102
object_Get_Mobile, 102
object_Get_Moved, 103
object_Get_Name, 103

object_Get_Open, 103
object_Set_Description, 103
object_Set_Description2, 104
object_Set_Graphics, 104
object_Set_Hidden, 104
object_Set_Id, 104
object_Set_Illuminates, 105
object_Set_Light, 105
object_Set_Mobile, 105
object_Set_Moved, 105
object_Set_Name, 106
object_Set_Open, 106
object_create, 100
object_destroy, 100
object_print, 103
object_Get_Description
object.c, 278
object.h, 101
object_Get_Description2
object.c, 278
object.h, 101
object_Get_Graphics
object.c, 278
object.h, 101
object_Get_Hidden
object.c, 279
object.h, 101
object_Get_Id
object.c, 279
object.h, 102
object_Get_Illuminates
object.c, 279
object.h, 102
object_Get_Light
object.c, 279
object.h, 102
object_Get_Mobile
object.c, 280
object.h, 102
object_Get_Moved
object.c, 280
object.h, 103
object_Get_Name
object.c, 280
object.h, 103
object_Get_Open
object.c, 280
object.h, 103
object_Set_Description
object.c, 281
object.h, 103
object_Set_Description2
object.c, 281
object.h, 104
object_Set_Graphics
object.c, 281
object.h, 104
object_Set_Hidden
object.c, 281
object.h, 104
object_Set_Id
object.c, 282
object.h, 104
object_Set_Illuminates
object.c, 282
object.h, 105
object_Set_Light
object.c, 282
object.h, 105
object_Set_Mobile
object.c, 282
object.h, 105
object_Set_Moved
object.c, 283
object.h, 105
object_Set_Name
object.c, 283
object.h, 106
object_Set_Open
object.c, 283
object.h, 106
object_create
object.c, 278
object.h, 100
object_destroy
object.c, 278
object.h, 100
object_print
object.c, 280
object.h, 103
object_test.c
test1_object_Get_Description, 285
test1_object_Get_Description2, 285
test1_object_Get_Graphics, 285
test1_object_Get_Hidden, 285
test1_object_Get_Id, 286
test1_object_Get_Illuminates, 286
test1_object_Get_Light, 286
test1_object_Get_Mobile, 286
test1_object_Get_Moved, 286
test1_object_Get_Name, 287
test1_object_Get_Open, 287
test1_object_create, 285
test1_object_set_Hidden, 288
test1_object_set_Id, 288
test1_object_set_Illuminates, 288
test1_object_set_Light, 288
test1_object_set_Mobile, 288
test1_object_set_Moved, 289
test1_object_set_Open, 289
test1_object_set_description, 287
test1_object_set_description2, 287
test1_object_set_graphics, 287
test1_object_set_name, 289
test2_object_Get_Description, 289
test2_object_Get_Description2, 289

test2_object_Get_Graphics, 290
test2_object_Get_Hidden, 290
test2_object_Get_Id, 290
test2_object_Get_Illuminates, 290
test2_object_Get_Light, 290
test2_object_Get_Mobile, 291
test2_object_Get_Moved, 291
test2_object_Get_Name, 291
test2_object_Get_Open, 291
test2_object_set_Hidden, 292
test2_object_set_Id, 292
test2_object_set_Illuminates, 292
test2_object_set_Light, 293
test2_object_set_Mobile, 293
test2_object_set_Moved, 293
test2_object_set_Open, 293
test2_object_set_description, 291
test2_object_set_description2, 292
test2_object_set_graphics, 292
test2_object_set_name, 293
object_test.h
test1_object_Get_Description, 166
test1_object_Get_Description2, 166
test1_object_Get_Graphics, 166
test1_object_Get_Hidden, 167
test1_object_Get_Id, 167
test1_object_Get_Illuminates, 167
test1_object_Get_Light, 167
test1_object_Get_Mobile, 167
test1_object_Get_Moved, 168
test1_object_Get_Name, 168
test1_object_Get_Open, 168
test1_object_create, 166
test1_object_set_Hidden, 169
test1_object_set_Id, 169
test1_object_set_Illuminates, 169
test1_object_set_Light, 169
test1_object_set_Mobile, 170
test1_object_set_Moved, 170
test1_object_set_Open, 170
test1_object_set_description, 168
test1_object_set_description2, 168
test1_object_set_graphics, 169
test1_object_set_name, 170
test2_object_Get_Description, 170
test2_object_Get_Description2, 171
test2_object_Get_Graphics, 171
test2_object_Get_Hidden, 171
test2_object_Get_Id, 171
test2_object_Get_Illuminates, 171
test2_object_Get_Light, 172
test2_object_Get_Mobile, 172
test2_object_Get_Moved, 172
test2_object_Get_Name, 172
test2_object_Get_Open, 172
test2_object_set_Hidden, 173
test2_object_set_Id, 173
test2_object_set_Illuminates, 174
test2_object_set_Light, 174
test2_object_set_Mobile, 174
test2_object_set_Moved, 174
test2_object_set_Open, 175
test2_object_set_description, 173
test2_object_set_description2, 173
test2_object_set_graphics, 173
test2_object_set_name, 174

PROTECTED_FILE
game_management.h, 85
player.c
player_Add_Object, 295
player_Get_Id, 296
player_Get_Location, 296
player_Get_Max_Objects, 296
player_Get_Name, 296
player_Has_Object, 297
player_Print, 297
player_Remove_Object, 297
player_Set_Id, 297
player_Set_Location, 298
player_Set_Max_Objects, 298
player_Set_Name, 298
player_create, 295
player_destroy, 295
player.h
player_Add_Object, 108
player_Get_Id, 108
player_Get_Location, 108
player_Get_Max_Objects, 109
player_Get_Name, 109
player_Has_Object, 109
player_Print, 109
player_Remove_Object, 110
player_Set_Id, 110
player_Set_Location, 110
player_Set_Max_Objects, 110
player_Set_Name, 111
player_create, 108
player_destroy, 108
player_Add_Object
player.c, 295
player.h, 108
player_Get_Id
player.c, 296
player.h, 108
player_Get_Location
player.c, 296
player.h, 108
player_Get_Max_Objects
player.c, 296
player.h, 109
player_Get_Name
player.c, 296
player.h, 109
player_Has_Object
player.c, 297
player.h, 109

player_Print
player.c, 297
player.h, 109
player_Remove_Object
player.c, 297
player.h, 110
player_Set_Id
player.c, 297
player.h, 110
player_Set_Location
player.c, 298
player.h, 110
player_Set_Max_Objects
player.c, 298
player.h, 110
player_Set_Name
player.c, 298
player.h, 111
player_create
player.c, 295
player.h, 108
player_destroy
player.c, 295
player.h, 108
player_test.c
test1_player_add_object, 300
test1_player_create, 300
test1_player_get_id, 300
test1_player_get_location, 300
test1_player_get_name, 300
test1_player_has_object, 301
test1_player_remove_object, 301
test1_player_set_id, 301
test1_player_set_location, 301
test1_player_set_max_objects, 301
test1_player_set_name, 302
test2_player_add_object, 302
test2_player_get_id, 302
test2_player_get_location, 302
test2_player_get_name, 302
test2_player_has_object, 303
test2_player_remove_object, 303
test2_player_set_id, 303
test2_player_set_location, 303
test2_player_set_max_objects, 303
test2_player_set_name, 304
test3_player_add_object, 304
test3_player_get_id, 304
test3_player_get_location, 304
test3_player_get_name, 304
test3_player_has_object, 305
test3_player_remove_object, 305
test3_player_set_id, 305
test3_player_set_location, 305
test3_player_set_max_objects, 305
test3_player_set_name, 306
test4_player_remove_object, 306
test4_player_set_max_objects, 306
player_test.h
test1_player_add_object, 176
test1_player_create, 176
test1_player_get_id, 176
test1_player_get_location, 177
test1_player_get_name, 177
test1_player_has_object, 177
test1_player_remove_object, 177
test1_player_set_id, 177
test1_player_set_location, 178
test1_player_set_max_objects, 178
test1_player_set_name, 178
test2_player_add_object, 178
test2_player_get_id, 178
test2_player_get_location, 179
test2_player_get_name, 179
test2_player_has_object, 179
test2_player_remove_object, 179
test2_player_set_id, 179
test2_player_set_location, 180
test2_player_set_max_objects, 180
test2_player_set_name, 180
test3_player_add_object, 180
test3_player_get_id, 180
test3_player_get_location, 181
test3_player_get_name, 181
test3_player_has_object, 181
test3_player_remove_object, 181
test3_player_set_id, 181
test3_player_set_location, 182
test3_player_set_max_objects, 182
test3_player_set_name, 182
test4_player_remove_object, 182
test4_player_set_max_objects, 182

QUIT
command.h, 62

ROLL
command.h, 63

SAVE_OK
game_management.h, 85
SAVE_PROTECTED_FILE
dialogue.h, 68
SAVE_SAVE_OK
dialogue.h, 68
SAVE_WRITE_FAILED
dialogue.h, 68
SAVE_STATUS
game_management.h, 85
SCREEN_MAX_STR
screen.h, 112
screen.c
screen_area_clear, 307
screen_area_destroy, 307
screen_area_init, 308
screen_area_puts, 308
screen_area_reset_cursor, 308

screen_gets, 308
screen_init, 308
screen.h
Area, 112
SCREEN_MAX_STR, 112
screen_area_clear, 112
screen_area_destroy, 112
screen_area_init, 112
screen_area_puts, 113
screen_area_reset_cursor, 113
screen_gets, 113
screen_init, 113
screen_area_clear
screen.c, 307
screen.h, 112
screen_area_destroy
screen.c, 307
screen.h, 112
screen_area_init
screen.c, 308
screen.h, 112
screen_area_puts
screen.c, 308
screen.h, 113
screen_area_reset_cursor
screen.c, 308
screen.h, 113
screen_gets
screen.c, 308
screen.h, 113
screen_init
screen.c, 308
screen.h, 113
screen_test.c
test1_screen_area_init, 309
test2_screen_area_init, 309
test3_screen_area_init, 309
test4_screen_area_init, 310
test5_screen_area_init, 310
screen_test.h
test1_screen_area_init, 183
test2_screen_area_init, 183
test3_screen_area_init, 184
test4_screen_area_init, 184
test5_screen_area_init, 184
set.c
set_Id_is_in, 312
set_addId, 311
set_create, 311
set_delId, 311
set_destroy, 312
set_getNumberOfIds, 312
set_print, 313
set.h
MAX_SET, 114
set_Id_is_in, 116
set_addId, 114
set_create, 115
set_delId, 115
set_destroy, 115
set_getNumberOfIds, 115
set_print, 116
set_Id_is_in
set.c, 312
set.h, 116
set_addId
set.c, 311
set.h, 114
set_create
set.c, 311
set.h, 115
set_delId
set.c, 311
set.h, 115
set_destroy
set.c, 312
set.h, 115
set_getNumberOfIds
set.c, 312
set.h, 115
set_print
set.c, 313
set.h, 116
Space
space.h, 119
space.c
space_add_object, 315
space_contains_object, 316
space_create, 316
space_destroy, 316
space_get_description, 316
space_get_down, 317
space_get_east, 317
space_get_graphics, 317
space_get_id, 317
space_get_iluminated, 317
space_get_long_description, 318
space_get_name, 318
space_get_north, 318
space_get_objects, 318
space_get_south, 319
space_get_up, 319
space_get_west, 319
space_graphics_areEmpty, 319
space_print, 319
space_print_graphics, 320
space_remove_object, 320
space_set_description, 320
space_set_down, 320
space_set_east, 321
space_set_graphics, 321
space_set_iluminated, 321
space_set_long_description, 321
space_set_name, 322
space_set_north, 322
space_set_south, 322

space_set_up, 322
space_set_west, 323
space.h
G_COLUMNS, 118
G_ROWS, 118
Space, 119
space_add_object, 119
space_contains_object, 119
space_create, 119
space_destroy, 119
space_get_description, 120
space_get_down, 120
space_get_east, 120
space_get_graphics, 120
space_get_id, 121
space_get_iluminated, 121
space_get_long_description, 121
space_get_name, 121
space_get_north, 121
space_get_objects, 122
space_get_south, 122
space_get_up, 122
space_get_west, 122
space_graphics_areEmpty, 123
space_print, 123
space_print_graphics, 123
space_remove_object, 123
space_set_description, 123
space_set_down, 124
space_set_east, 124
space_set_graphics, 124
space_set_iluminated, 124
space_set_long_description, 125
space_set_name, 125
space_set_north, 125
space_set_south, 125
space_set_up, 126
space_set_west, 126
space_add_object
space.c, 315
space.h, 119
space_contains_object
space.c, 316
space.h, 119
space_create
space.c, 316
space.h, 119
space_destroy
space.c, 316
space.h, 119
space_get_description
space.c, 316
space.h, 120
space_get_down
space.c, 317
space.h, 120
space_get_east
space.c, 317
space.h, 120
space_get_graphics
space.c, 317
space.h, 120
space_get_id
space.c, 317
space.h, 121
space_get_iluminated
space.c, 317
space.h, 121
space_get_long_description
space.c, 318
space.h, 121
space_get_name
space.c, 318
space.h, 121
space_get_north
space.c, 318
space.h, 121
space_get_objects
space.c, 318
space.h, 122
space_get_south
space.c, 319
space.h, 122
space_get_up
space.c, 319
space.h, 122
space_get_west
space.c, 319
space.h, 122
space_graphics_areEmpty
space.c, 319
space.h, 123
space_print
space.c, 319
space.h, 123
space_print_graphics
space.c, 320
space.h, 123
space_remove_object
space.c, 320
space.h, 123
space_set_description
space.c, 320
space.h, 123
space_set_down
space.c, 320
space.h, 124
space_set_east
space.c, 321
space.h, 124
space_set_graphics
space.c, 321
space.h, 124
space_set_iluminated
space.c, 321
space.h, 124

space_set_long_description
space.c, 321
space.h, 125
space_set_name
space.c, 322
space.h, 125
space_set_north
space.c, 322
space.h, 125
space_set_south
space.c, 322
space.h, 125
space_set_up
space.c, 322
space.h, 126
space_set_west
space.c, 323
space.h, 126
space_test.c
test1_space_add_object, 325
test1_space_contains_object, 325
test1_space_create, 325
test1_space_get_description, 326
test1_space_get_down, 326
test1_space_get_east, 326
test1_space_get_graphics, 326
test1_space_get_id, 326
test1_space_get_iluminated, 327
test1_space_get_long_description, 327
test1_space_get_name, 327
test1_space_get_north, 327
test1_space_get_objects, 327
test1_space_get_south, 328
test1_space_get_up, 328
test1_space_get_west, 328
test1_space_graphics_areEmpty, 328
test1_space_remove_object, 328
test1_space_set_description, 329
test1_space_set_down, 329
test1_space_set_east, 329
test1_space_set_graphics, 329
test1_space_set_iluminated, 329
test1_space_set_long_description, 330
test1_space_set_name, 330
test1_space_set_north, 330
test1_space_set_south, 330
test1_space_set_up, 330
test1_space_set_west, 331
test2_space_add_object, 331
test2_space_contains_object, 331
test2_space_create, 331
test2_space_get_description, 331
test2_space_get_down, 332
test2_space_get_east, 332
test2_space_get_graphics, 332
test2_space_get_id, 332
test2_space_get_iluminated, 332
test2_space_get_long_description, 333
test2_space_get_name, 333
test2_space_get_north, 333
test2_space_get_objects, 333
test2_space_get_south, 333
test2_space_get_up, 334
test2_space_get_west, 334
test2_space_graphics_areEmpty, 334
test2_space_remove_object, 334
test2_space_set_description, 334
test2_space_set_down, 335
test2_space_set_east, 335
test2_space_set_graphics, 335
test2_space_set_iluminated, 335
test2_space_set_long_description, 335
test2_space_set_name, 336
test2_space_set_north, 336
test2_space_set_south, 336
test2_space_set_up, 336
test2_space_set_west, 336
test3_space_add_object, 337
test3_space_contains_object, 337
test3_space_create, 337
test3_space_get_description, 337
test3_space_get_down, 337
test3_space_get_east, 338
test3_space_get_long_description, 338
test3_space_get_name, 338
test3_space_get_north, 338
test3_space_get_south, 338
test3_space_get_up, 339
test3_space_get_west, 339
test3_space_graphics_areEmpty, 339
test3_space_remove_object, 339
test3_space_set_description, 339
test3_space_set_down, 340
test3_space_set_east, 340
test3_space_set_long_description, 340
test3_space_set_name, 340
test3_space_set_north, 340
test3_space_set_south, 341
test3_space_set_up, 341
test3_space_set_west, 341
test4_space_remove_object, 341
space_test.h
test1_space_add_object, 187
test1_space_contains_object, 187
test1_space_create, 187
test1_space_get_description, 187
test1_space_get_down, 187
test1_space_get_east, 188
test1_space_get_graphics, 188
test1_space_get_id, 188
test1_space_get_iluminated, 188
test1_space_get_long_description, 188
test1_space_get_name, 189
test1_space_get_north, 189
test1_space_get_objects, 189
test1_space_get_south, 189

test1_space_get_up, 189
test1_space_get_west, 190
test1_space_graphics_areEmpty, 190
test1_space_remove_object, 190
test1_space_set_description, 190
test1_space_set_down, 190
test1_space_set_east, 191
test1_space_set_graphics, 191
test1_space_set_iluminated, 191
test1_space_set_long_description, 191
test1_space_set_name, 191
test1_space_set_north, 192
test1_space_set_south, 192
test1_space_set_up, 192
test1_space_set_west, 192
test2_space_add_object, 192
test2_space_contains_object, 193
test2_space_create, 193
test2_space_get_description, 193
test2_space_get_down, 193
test2_space_get_east, 193
test2_space_get_graphics, 194
test2_space_get_id, 194
test2_space_get_iluminated, 194
test2_space_get_long_description, 194
test2_space_get_name, 194
test2_space_get_north, 195
test2_space_get_objects, 195
test2_space_get_south, 195
test2_space_get_up, 195
test2_space_get_west, 195
test2_space_graphics_areEmpty, 196
test2_space_remove_object, 196
test2_space_set_description, 196
test2_space_set_down, 196
test2_space_set_east, 196
test2_space_set_graphics, 197
test2_space_set_iluminated, 197
test2_space_set_long_description, 197
test2_space_set_name, 197
test2_space_set_north, 197
test2_space_set_south, 198
test2_space_set_up, 198
test2_space_set_west, 198
test3_space_add_object, 198
test3_space_contains_object, 198
test3_space_create, 199
test3_space_get_description, 199
test3_space_get_down, 199
test3_space_get_east, 199
test3_space_get_long_description, 199
test3_space_get_name, 200
test3_space_get_north, 200
test3_space_get_south, 200
test3_space_get_up, 200
test3_space_get_west, 200
test3_space_graphics_areEmpty, 201
test3_space_remove_object, 201
test3_space_set_description, 201
test3_space_set_down, 201
test3_space_set_east, 201
test3_space_set_long_description, 202
test3_space_set_name, 202
test3_space_set_north, 202
test3_space_set_south, 202
test3_space_set_up, 202
test3_space_set_west, 203
test4_space_remove_object, 203
src/command.c, 205
src/command_test.c, 208
src/dialogue.c, 213
src/dialogue_test.c, 219
src/die.c, 226
src/die_test.c, 229
src/game.c, 229
src/game_loop.c, 238
src/game_management.c, 239
src/game_management_test.c, 241
src/game_rules.c, 243
src/game_rules_test.c, 243
src/game_test.c, 244
src/graphic_engine_test.c, 257
src/inventory.c, 257
src/inventory_test.c, 260
src/link.c, 264
src/link_test.c, 269
src/object.c, 276
src/object_test.c, 283
src/player.c, 294
src/player_test.c, 298
src/screen.c, 306
src/screen_test.c, 308
src/set.c, 310
src/set_test.c, 313
src/space.c, 314
src/space_test.c, 323
State
link.h, 94

TAKE
command.h, 62
TURNOFF
command.h, 63
TURNON
command.h, 63
test1_command_clear
command_test.c, 209
command_test.h, 127
test1_command_destroy
command_test.c, 209
command_test.h, 127
test1_command_get_cmd
command_test.c, 209
command_test.h, 127
test1_command_get_cmd_arg
command_test.c, 210
command_test.h, 128

test1_command_ini
command_test.c, 210
command_test.h, 128
test1_command_set_cmd
command_test.c, 210
command_test.h, 128
test1_command_set_cmd_arg
command_test.c, 210
command_test.h, 128
test1_dialogue_attack
dialogue_test.c, 220
dialogue_test.h, 132
test1_dialogue_dir
dialogue_test.c, 220
dialogue_test.h, 132
test1_dialogue_game_rule
dialogue_test.c, 220
dialogue_test.h, 132
test1_dialogue_get_text
dialogue_test.c, 220
dialogue_test.h, 132
test1_dialogue_go
dialogue_test.c, 220
dialogue_test.h, 133
test1_dialogue_help
dialogue_test.c, 221
dialogue_test.h, 133
test1_dialogue_ini
dialogue_test.c, 221
dialogue_test.h, 133
test1_dialogue_inspect
dialogue_test.c, 221
dialogue_test.h, 133
test1_dialogue_leave
dialogue_test.c, 221
dialogue_test.h, 133
test1_dialogue_load
dialogue_test.c, 221
dialogue_test.h, 134
test1_dialogue_open
dialogue_test.c, 222
dialogue_test.h, 134
test1_dialogue_save
dialogue_test.c, 222
dialogue_test.h, 134
test1_dialogue_take
dialogue_test.c, 222
dialogue_test.h, 134
test1_dialogue_turn_off
dialogue_test.c, 222
dialogue_test.h, 134
test1_dialogue_turn_on
dialogue_test.c, 222
dialogue_test.h, 135
test1_dialogue_unknown
dialogue_test.c, 223
dialogue_test.h, 135
test1_game_add_link
game_test.c, 246
game_test.h, 143
test1_game_add_object
game_test.c, 246
game_test.h, 143
test1_game_add_player
game_test.c, 246
game_test.h, 143
test1_game_add_space
game_test.c, 246
game_test.h, 143
test1_game_create
game_test.c, 247
game_test.h, 143
test1_game_destroy
game_test.c, 247
game_test.h, 143
test1_game_get_dialogue
game_test.c, 247
game_test.h, 144
test1_game_get_die
game_test.c, 247
game_test.h, 144
test1_game_get_last_inspected_object
game_test.c, 247
game_test.h, 144
test1_game_get_last_inspected_space
game_test.c, 248
game_test.h, 144
test1_game_get_link
game_test.c, 248
game_test.h, 144
test1_game_get_link_at
game_test.c, 248
game_test.h, 145
test1_game_get_obj_list_as_str
game_test.c, 248
game_test.h, 145
test1_game_get_object_at
game_test.c, 248
game_test.h, 145
test1_game_get_object_location
game_test.c, 249
game_test.h, 145
test1_game_get_player
game_test.c, 249
game_test.h, 145
test1_game_get_player_location
game_test.c, 249
game_test.h, 146
test1_game_get_space
game_test.c, 249
game_test.h, 146
test1_game_get_space_at
game_test.c, 249
game_test.h, 146
test1_game_is_over
game_test.c, 250

game_test.h, 146
test1_game_management_load
game_management_test.c, 242
game_management_test.h, 139
test1_game_management_save
game_management_test.c, 242
game_management_test.h, 139
test1_game_management_start_from_file
game_management_test.c, 242
game_management_test.h, 139
test1_game_rules_run_random_rule
game_rules_test.c, 244
game_rules_test.h, 141
test1_game_update
game_test.c, 250
game_test.h, 146
test1_graphic_engine_create
graphic_engine_test.c, 257
graphic_engine_test.h, 154
test1_inventory_add_object
inventory_test.c, 261
inventory_test.h, 155
test1_inventory_create
inventory_test.c, 261
inventory_test.h, 155
test1_inventory_get_max
inventory_test.c, 261
inventory_test.h, 155
test1_inventory_get_set
inventory_test.c, 261
inventory_test.h, 155
test1_inventory_remove_object
inventory_test.c, 262
inventory_test.h, 155
test1_inventory_set_max
inventory_test.c, 262
inventory_test.h, 156
test1_link_create
link_test.c, 270
link_test.h, 159
test1_link_get_dest_from
link_test.c, 270
link_test.h, 159
test1_link_get_id
link_test.c, 270
link_test.h, 159
test1_link_get_name
link_test.c, 270
link_test.h, 159
test1_link_get_space1
link_test.c, 271
link_test.h, 159
test1_link_get_space2
link_test.c, 271
link_test.h, 160
test1_link_get_state
link_test.c, 271
link_test.h, 160
test1_link_set_id
link_test.c, 271
link_test.h, 160
test1_link_set_name
link_test.c, 271
link_test.h, 160
test1_link_set_space1
link_test.c, 272
link_test.h, 160
test1_link_set_space2
link_test.c, 272
link_test.h, 161
test1_link_set_state
link_test.c, 272
link_test.h, 161
test1_object_Get_Description
object_test.c, 285
object_test.h, 166
test1_object_Get_Description2
object_test.c, 285
object_test.h, 166
test1_object_Get_Graphics
object_test.c, 285
object_test.h, 166
test1_object_Get_Hidden
object_test.c, 285
object_test.h, 167
test1_object_Get_Id
object_test.c, 286
object_test.h, 167
test1_object_Get_Illuminates
object_test.c, 286
object_test.h, 167
test1_object_Get_Light
object_test.c, 286
object_test.h, 167
test1_object_Get_Mobile
object_test.c, 286
object_test.h, 167
test1_object_Get_Moved
object_test.c, 286
object_test.h, 168
test1_object_Get_Name
object_test.c, 287
object_test.h, 168
test1_object_Get_Open
object_test.c, 287
object_test.h, 168
test1_object_create
object_test.c, 285
object_test.h, 166
test1_object_set_Hidden
object_test.c, 288
object_test.h, 169
test1_object_set_Id
object_test.c, 288
object_test.h, 169
test1_object_set_Illuminates

object_test.c, 288
object_test.h, 169
test1_object_set_Light
object_test.c, 288
object_test.h, 169
test1_object_set_Mobile
object_test.c, 288
object_test.h, 170
test1_object_set_Moved
object_test.c, 289
object_test.h, 170
test1_object_set_Open
object_test.c, 289
object_test.h, 170
test1_object_set_description
object_test.c, 287
object_test.h, 168
test1_object_set_description2
object_test.c, 287
object_test.h, 168
test1_object_set_graphics
object_test.c, 287
object_test.h, 169
test1_object_set_name
object_test.c, 289
object_test.h, 170
test1_player_add_object
player_test.c, 300
player_test.h, 176
test1_player_create
player_test.c, 300
player_test.h, 176
test1_player_get_id
player_test.c, 300
player_test.h, 176
test1_player_get_location
player_test.c, 300
player_test.h, 177
test1_player_get_name
player_test.c, 300
player_test.h, 177
test1_player_has_object
player_test.c, 301
player_test.h, 177
test1_player_remove_object
player_test.c, 301
player_test.h, 177
test1_player_set_id
player_test.c, 301
player_test.h, 177
test1_player_set_location
player_test.c, 301
player_test.h, 178
test1_player_set_max_objects
player_test.c, 301
player_test.h, 178
test1_player_set_name
player_test.c, 302
player_test.h, 178
test1_screen_area_init
screen_test.c, 309
screen_test.h, 183
test1_space_add_object
space_test.c, 325
space_test.h, 187
test1_space_contains_object
space_test.c, 325
space_test.h, 187
test1_space_create
space_test.c, 325
space_test.h, 187
test1_space_get_description
space_test.c, 326
space_test.h, 187
test1_space_get_down
space_test.c, 326
space_test.h, 187
test1_space_get_east
space_test.c, 326
space_test.h, 188
test1_space_get_graphics
space_test.c, 326
space_test.h, 188
test1_space_get_id
space_test.c, 326
space_test.h, 188
test1_space_get_iluminated
space_test.c, 327
space_test.h, 188
test1_space_get_long_description
space_test.c, 327
space_test.h, 188
test1_space_get_name
space_test.c, 327
space_test.h, 189
test1_space_get_north
space_test.c, 327
space_test.h, 189
test1_space_get_objects
space_test.c, 327
space_test.h, 189
test1_space_get_south
space_test.c, 328
space_test.h, 189
test1_space_get_up
space_test.c, 328
space_test.h, 189
test1_space_get_west
space_test.c, 328
space_test.h, 190
test1_space_graphics_areEmpty
space_test.c, 328
space_test.h, 190
test1_space_remove_object
space_test.c, 328
space_test.h, 190

test1_space_set_description
space_test.c, 329
space_test.h, 190
test1_space_set_down
space_test.c, 329
space_test.h, 190
test1_space_set_east
space_test.c, 329
space_test.h, 191
test1_space_set_graphics
space_test.c, 329
space_test.h, 191
test1_space_set_iluminated
space_test.c, 329
space_test.h, 191
test1_space_set_long_description
space_test.c, 330
space_test.h, 191
test1_space_set_name
space_test.c, 330
space_test.h, 191
test1_space_set_north
space_test.c, 330
space_test.h, 192
test1_space_set_south
space_test.c, 330
space_test.h, 192
test1_space_set_up
space_test.c, 330
space_test.h, 192
test1_space_set_west
space_test.c, 331
space_test.h, 192
test2_command_clear
command_test.c, 210
command_test.h, 128
test2_command_destroy
command_test.c, 211
command_test.h, 129
test2_command_get_cmd
command_test.c, 211
command_test.h, 129
test2_command_get_cmd_arg
command_test.c, 211
command_test.h, 129
test2_command_set_cmd
command_test.c, 211
command_test.h, 129
test2_command_set_cmd_arg
command_test.c, 211
command_test.h, 129
test2_dialogue_attack
dialogue_test.c, 223
dialogue_test.h, 135
test2_dialogue_dir
dialogue_test.c, 223
dialogue_test.h, 135
test2_dialogue_game_rule
dialogue_test.c, 223
dialogue_test.h, 135
test2_dialogue_get_text
dialogue_test.c, 223
dialogue_test.h, 136
test2_dialogue_go
dialogue_test.c, 224
dialogue_test.h, 136
test2_dialogue_help
dialogue_test.c, 224
dialogue_test.h, 136
test2_dialogue_inspect
dialogue_test.c, 224
dialogue_test.h, 136
test2_dialogue_leave
dialogue_test.c, 224
dialogue_test.h, 136
test2_dialogue_load
dialogue_test.c, 224
dialogue_test.h, 137
test2_dialogue_open
dialogue_test.c, 225
dialogue_test.h, 137
test2_dialogue_save
dialogue_test.c, 225
dialogue_test.h, 137
test2_dialogue_take
dialogue_test.c, 225
dialogue_test.h, 137
test2_dialogue_turn_off
dialogue_test.c, 225
dialogue_test.h, 137
test2_dialogue_turn_on
dialogue_test.c, 225
dialogue_test.h, 138
test2_dialogue_unknown
dialogue_test.c, 226
dialogue_test.h, 138
test2_game_add_link
game_test.c, 250
game_test.h, 147
test2_game_add_object
game_test.c, 250
game_test.h, 147
test2_game_add_player
game_test.c, 250
game_test.h, 147
test2_game_add_space
game_test.c, 251
game_test.h, 147
test2_game_destroy
game_test.c, 251
game_test.h, 147
test2_game_get_dialogue
game_test.c, 251
game_test.h, 148
test2_game_get_die
game_test.c, 251

game_test.h, 148
test2_game_get_last_inspected_object
game_test.c, 251
game_test.h, 148
test2_game_get_last_inspected_space
game_test.c, 252
game_test.h, 148
test2_game_get_link
game_test.c, 252
game_test.h, 148
test2_game_get_link_at
game_test.c, 252
game_test.h, 149
test2_game_get_obj_list_as_str
game_test.c, 252
game_test.h, 149
test2_game_get_object_at
game_test.c, 252
game_test.h, 149
test2_game_get_object_location
game_test.c, 253
game_test.h, 149
test2_game_get_player
game_test.c, 253
game_test.h, 149
test2_game_get_player_location
game_test.c, 253
game_test.h, 150
test2_game_get_space
game_test.c, 253
game_test.h, 150
test2_game_get_space_at
game_test.c, 253
game_test.h, 150
test2_game_is_over
game_test.c, 254
game_test.h, 150
test2_game_management_load
game_management_test.c, 242
game_management_test.h, 139
test2_game_management_save
game_management_test.c, 242
game_management_test.h, 140
test2_game_management_start_from_file
game_management_test.c, 242
game_management_test.h, 140
test2_game_rules_run_random_rule
game_rules_test.c, 244
game_rules_test.h, 141
test2_game_update
game_test.c, 254
game_test.h, 150
test2_inventory_add_object
inventory_test.c, 262
inventory_test.h, 156
test2_inventory_get_max
inventory_test.c, 262
inventory_test.h, 156
test2_inventory_get_set
inventory_test.c, 262
inventory_test.h, 156
test2_inventory_remove_object
inventory_test.c, 263
inventory_test.h, 156
test2_inventory_set_max
inventory_test.c, 263
inventory_test.h, 157
test2_link_get_dest_from
link_test.c, 272
link_test.h, 161
test2_link_get_id
link_test.c, 272
link_test.h, 161
test2_link_get_name
link_test.c, 273
link_test.h, 161
test2_link_get_space1
link_test.c, 273
link_test.h, 162
test2_link_get_space2
link_test.c, 273
link_test.h, 162
test2_link_get_state
link_test.c, 273
link_test.h, 162
test2_link_set_id
link_test.c, 273
link_test.h, 162
test2_link_set_name
link_test.c, 274
link_test.h, 162
test2_link_set_space1
link_test.c, 274
link_test.h, 163
test2_link_set_space2
link_test.c, 274
link_test.h, 163
test2_link_set_state
link_test.c, 274
link_test.h, 163
test2_object_Get_Description
object_test.c, 289
object_test.h, 170
test2_object_Get_Description2
object_test.c, 289
object_test.h, 171
test2_object_Get_Graphics
object_test.c, 290
object_test.h, 171
test2_object_Get_Hidden
object_test.c, 290
object_test.h, 171
test2_object_Get_Id
object_test.c, 290
object_test.h, 171
test2_object_Get_Illuminates

object_test.c, 290
object_test.h, 171
test2_object_Get_Light
object_test.c, 290
object_test.h, 172
test2_object_Get_Mobile
object_test.c, 291
object_test.h, 172
test2_object_Get_Moved
object_test.c, 291
object_test.h, 172
test2_object_Get_Name
object_test.c, 291
object_test.h, 172
test2_object_Get_Open
object_test.c, 291
object_test.h, 172
test2_object_set_Hidden
object_test.c, 292
object_test.h, 173
test2_object_set_Id
object_test.c, 292
object_test.h, 173
test2_object_set_Illuminates
object_test.c, 292
object_test.h, 174
test2_object_set_Light
object_test.c, 293
object_test.h, 174
test2_object_set_Mobile
object_test.c, 293
object_test.h, 174
test2_object_set_Moved
object_test.c, 293
object_test.h, 174
test2_object_set_Open
object_test.c, 293
object_test.h, 175
test2_object_set_description
object_test.c, 291
object_test.h, 173
test2_object_set_description2
object_test.c, 292
object_test.h, 173
test2_object_set_graphics
object_test.c, 292
object_test.h, 173
test2_object_set_name
object_test.c, 293
object_test.h, 174
test2_player_add_object
player_test.c, 302
player_test.h, 178
test2_player_get_id
player_test.c, 302
player_test.h, 178
test2_player_get_location
player_test.c, 302
player_test.h, 179
test2_player_get_name
player_test.c, 302
player_test.h, 179
test2_player_has_object
player_test.c, 303
player_test.h, 179
test2_player_remove_object
player_test.c, 303
player_test.h, 179
test2_player_set_id
player_test.c, 303
player_test.h, 179
test2_player_set_location
player_test.c, 303
player_test.h, 180
test2_player_set_max_objects
player_test.c, 303
player_test.h, 180
test2_player_set_name
player_test.c, 304
player_test.h, 180
test2_screen_area_init
screen_test.c, 309
screen_test.h, 183
test2_space_add_object
space_test.c, 331
space_test.h, 192
test2_space_contains_object
space_test.c, 331
space_test.h, 193
test2_space_create
space_test.c, 331
space_test.h, 193
test2_space_get_description
space_test.c, 331
space_test.h, 193
test2_space_get_down
space_test.c, 332
space_test.h, 193
test2_space_get_east
space_test.c, 332
space_test.h, 193
test2_space_get_graphics
space_test.c, 332
space_test.h, 194
test2_space_get_id
space_test.c, 332
space_test.h, 194
test2_space_get_iluminated
space_test.c, 332
space_test.h, 194
test2_space_get_long_description
space_test.c, 333
space_test.h, 194
test2_space_get_name
space_test.c, 333
space_test.h, 194

test2_space_get_north
space_test.c, 333
space_test.h, 195
test2_space_get_objects
space_test.c, 333
space_test.h, 195
test2_space_get_south
space_test.c, 333
space_test.h, 195
test2_space_get_up
space_test.c, 334
space_test.h, 195
test2_space_get_west
space_test.c, 334
space_test.h, 195
test2_space_graphics_areEmpty
space_test.c, 334
space_test.h, 196
test2_space_remove_object
space_test.c, 334
space_test.h, 196
test2_space_set_description
space_test.c, 334
space_test.h, 196
test2_space_set_down
space_test.c, 335
space_test.h, 196
test2_space_set_east
space_test.c, 335
space_test.h, 196
test2_space_set_graphics
space_test.c, 335
space_test.h, 197
test2_space_set_iluminated
space_test.c, 335
space_test.h, 197
test2_space_set_long_description
space_test.c, 335
space_test.h, 197
test2_space_set_name
space_test.c, 336
space_test.h, 197
test2_space_set_north
space_test.c, 336
space_test.h, 197
test2_space_set_south
space_test.c, 336
space_test.h, 198
test2_space_set_up
space_test.c, 336
space_test.h, 198
test2_space_set_west
space_test.c, 336
space_test.h, 198
test3_command_get_cmd
command_test.c, 212
command_test.h, 130
test3_command_get_cmd_arg
command_test.c, 212
command_test.h, 130
test3_command_set_cmd
command_test.c, 212
command_test.h, 130
test3_command_set_cmd_arg
command_test.c, 212
command_test.h, 130
test3_game_add_link
game_test.c, 254
game_test.h, 151
test3_game_add_object
game_test.c, 254
game_test.h, 151
test3_game_add_player
game_test.c, 254
game_test.h, 151
test3_game_add_space
game_test.c, 255
game_test.h, 151
test3_game_get_last_inspected_object
game_test.c, 255
game_test.h, 151
test3_game_get_link
game_test.c, 255
game_test.h, 152
test3_game_get_link_at
game_test.c, 255
game_test.h, 152
test3_game_get_object_at
game_test.c, 255
game_test.h, 152
test3_game_get_object_location
game_test.c, 256
game_test.h, 152
test3_game_get_space
game_test.c, 256
game_test.h, 152
test3_game_get_space_at
game_test.c, 256
game_test.h, 153
test3_game_update
game_test.c, 256
game_test.h, 153
test3_inventory_add_object
inventory_test.c, 263
inventory_test.h, 157
test3_inventory_remove_object
inventory_test.c, 263
inventory_test.h, 157
test3_inventory_set_max
inventory_test.c, 263
inventory_test.h, 157
test3_link_get_dest_from
link_test.c, 274
link_test.h, 163
test3_link_set_id
link_test.c, 275

link_test.h, 163
test3_link_set_name
link_test.c, 275
link_test.h, 164
test3_link_set_space1
link_test.c, 275
link_test.h, 164
test3_link_set_space2
link_test.c, 275
link_test.h, 164
test3_link_set_state
link_test.c, 275
link_test.h, 164
test3_player_add_object
player_test.c, 304
player_test.h, 180
test3_player_get_id
player_test.c, 304
player_test.h, 180
test3_player_get_location
player_test.c, 304
player_test.h, 181
test3_player_get_name
player_test.c, 304
player_test.h, 181
test3_player_has_object
player_test.c, 305
player_test.h, 181
test3_player_remove_object
player_test.c, 305
player_test.h, 181
test3_player_set_id
player_test.c, 305
player_test.h, 181
test3_player_set_location
player_test.c, 305
player_test.h, 182
test3_player_set_max_objects
player_test.c, 305
player_test.h, 182
test3_player_set_name
player_test.c, 306
player_test.h, 182
test3_screen_area_init
screen_test.c, 309
screen_test.h, 184
test3_space_add_object
space_test.c, 337
space_test.h, 198
test3_space_contains_object
space_test.c, 337
space_test.h, 198
test3_space_create
space_test.c, 337
space_test.h, 199
test3_space_get_description
space_test.c, 337
space_test.h, 199
test3_space_get_down
space_test.c, 337
space_test.h, 199
test3_space_get_east
space_test.c, 338
space_test.h, 199
test3_space_get_long_description
space_test.c, 338
space_test.h, 199
test3_space_get_name
space_test.c, 338
space_test.h, 200
test3_space_get_north
space_test.c, 338
space_test.h, 200
test3_space_get_south
space_test.c, 338
space_test.h, 200
test3_space_get_up
space_test.c, 339
space_test.h, 200
test3_space_get_west
space_test.c, 339
space_test.h, 200
test3_space_graphics_areEmpty
space_test.c, 339
space_test.h, 201
test3_space_remove_object
space_test.c, 339
space_test.h, 201
test3_space_set_description
space_test.c, 339
space_test.h, 201
test3_space_set_down
space_test.c, 340
space_test.h, 201
test3_space_set_east
space_test.c, 340
space_test.h, 201
test3_space_set_long_description
space_test.c, 340
space_test.h, 202
test3_space_set_name
space_test.c, 340
space_test.h, 202
test3_space_set_north
space_test.c, 340
space_test.h, 202
test3_space_set_south
space_test.c, 341
space_test.h, 202
test3_space_set_up
space_test.c, 341
space_test.h, 202
test3_space_set_west
space_test.c, 341
space_test.h, 203
test4_command_set_cmd_arg

command_test.c, 212
command_test.h, 130
test4_link_get_dest_from
link_test.c, 276
link_test.h, 164
test4_player_remove_object
player_test.c, 306
player_test.h, 182
test4_player_set_max_objects
player_test.c, 306
player_test.h, 182
test4_screen_area_init
screen_test.c, 310
screen_test.h, 184
test4_space_remove_object
space_test.c, 341
space_test.h, 203
test5_screen_area_init
screen_test.c, 310
screen_test.h, 184

UNKNOWN
command.h, 62

WRITE_FAILED
game_management.h, 85