

Humanity simulator (Készítette: Funk Gábor)

Készítette Doxygen 1.9.1



<b>1. Névtérmutató</b>	<b>1</b>
1.1. Névtérlista . . . . .	1
<b>2. Hierarchikus mutató</b>	<b>3</b>
2.1. Osztályhierarchia . . . . .	3
<b>3. Osztálymutató</b>	<b>5</b>
3.1. Osztálylista . . . . .	5
<b>4. Fájlmutató</b>	<b>9</b>
4.1. Fájllista . . . . .	9
<b>5. Névterek dokumentációja</b>	<b>13</b>
5.1. creature névtér-referencia . . . . .	13
5.1.1. Részletes leírás . . . . .	14
5.1.2. Enumerációk dokumentációja . . . . .	14
5.1.2.1. ENTITY_GENDER . . . . .	14
5.1.2.2. ENTITY_TYPE . . . . .	14
5.1.2.3. FACING . . . . .	14
5.1.2.4. LIVINGSTATE . . . . .	15
5.2. gtest_lite névtér-referencia . . . . .	15
5.2.1. Részletes leírás . . . . .	16
5.2.2. Függvények dokumentációja . . . . .	16
5.2.2.1. almostEQ() . . . . .	16
5.2.2.2. eq() . . . . .	16
5.2.2.3. eqstr() . . . . .	16
5.2.2.4. eqstrcase() . . . . .	17
5.2.2.5. EXPECT_() [1/2] . . . . .	17
5.2.2.6. EXPECT_() [2/2] . . . . .	17
5.2.2.7. EXPECTSTR() . . . . .	17
5.2.2.8. ge() . . . . .	18
5.2.2.9. gt() . . . . .	18
5.2.2.10. le() . . . . .	18
5.2.2.11. lt() . . . . .	18
5.2.2.12. ne() . . . . .	18
5.2.2.13. nestr() . . . . .	18
5.3. minerals névtér-referencia . . . . .	19
5.3.1. Részletes leírás . . . . .	19
5.3.2. Enumerációk dokumentációja . . . . .	19
5.3.2.1. MINERAL_TYPE . . . . .	19
5.3.3. Függvények dokumentációja . . . . .	20
5.3.3.1. mineral_to_string() . . . . .	20
5.4. sf névtér-referencia . . . . .	20
5.4.1. Enumerációk dokumentációja . . . . .	21

5.4.1.1. BlendMode	21
5.4.2. Függvények dokumentációja	21
5.4.2.1. file_exists_at_path()	21
5.4.2.2. operator<<()	22
5.4.3. Változók dokumentációja	22
5.4.3.1. BlendAdd	22
5.4.3.2. hungarian	22
5.5. tiles névtér-referencia	22
5.5.1. Részletes leírás	22
5.5.2. Enumerációk dokumentációja	22
5.5.2.1. TILETYPE	22
5.6. ui névtér-referencia	23
5.6.1. Részletes leírás	23
<b>6. Osztályok dokumentációja</b>	<b>25</b>
6.1. _Is_Types< F, T > struktúrasablon-referencia	25
6.1.1. Részletes leírás	25
6.1.2. Tagfüggvények dokumentációja	25
6.1.2.1. f() [1/2]	26
6.1.2.2. f() [2/2]	26
6.1.3. Adattagok dokumentációja	26
6.1.3.1. convertible	26
6.2. creature::AnglerMiner osztályreferencia	26
6.2.1. Részletes leírás	27
6.2.2. Konstruktorok és destruktorok dokumentációja	27
6.2.2.1. AnglerMiner()	27
6.2.2.2. ~AnglerMiner()	27
6.2.3. Tagfüggvények dokumentációja	27
6.2.3.1. update_logic()	28
6.3. creature::Bear osztályreferencia	28
6.3.1. Részletes leírás	29
6.3.2. Konstruktorok és destruktorok dokumentációja	29
6.3.2.1. Bear()	29
6.3.2.2. ~Bear()	29
6.3.3. Tagfüggvények dokumentációja	29
6.3.3.1. die()	29
6.3.3.2. draw_logic()	30
6.3.3.3. get_type()	30
6.3.3.4. select_target()	30
6.3.3.5. update_logic()	31
6.4. minerals::BerryBush osztályreferencia	31
6.4.1. Részletes leírás	32

6.4.2.	Konstruktorok és destruktorok dokumentációja	32
6.4.2.1.	BerryBush()	32
6.4.3.	Tagfüggvények dokumentációja	32
6.4.3.1.	get_type()	32
6.4.3.2.	harvest()	32
6.4.3.3.	update_logic()	32
6.5.	sf::Bound osztályreferencia	33
6.5.1.	Adattagok dokumentációja	33
6.5.1.1.	height	33
6.5.1.2.	width	33
6.6.	creature::Builder osztályreferencia	34
6.6.1.	Részletes leírás	34
6.6.2.	Konstruktorok és destruktorok dokumentációja	34
6.6.2.1.	Builder()	34
6.6.2.2.	~Builder()	35
6.6.3.	Tagfüggvények dokumentációja	35
6.6.3.1.	update_logic()	35
6.7.	ui::Button osztályreferencia	35
6.7.1.	Részletes leírás	36
6.7.2.	Konstruktorok és destruktorok dokumentációja	36
6.7.2.1.	Button()	36
6.7.3.	Tagfüggvények dokumentációja	37
6.7.3.1.	draw()	37
6.7.3.2.	onClick()	37
6.7.3.3.	setCallback()	37
6.7.3.4.	setPosition()	38
6.7.3.5.	setTexture()	38
6.7.3.6.	try_hover_animation()	38
6.8.	minerals::CityCenter osztályreferencia	39
6.8.1.	Részletes leírás	39
6.8.2.	Konstruktorok és destruktorok dokumentációja	39
6.8.2.1.	CityCenter()	40
6.8.3.	Tagfüggvények dokumentációja	40
6.8.3.1.	get_settlement_age()	40
6.8.3.2.	get_type()	40
6.8.3.3.	is_there_room_for_housing()	40
6.8.3.4.	register_new_house()	40
6.8.3.5.	update_logic()	40
6.9.	CityCenterException osztályreferencia	41
6.9.1.	Részletes leírás	41
6.9.2.	Konstruktorok és destruktorok dokumentációja	41
6.9.2.1.	CityCenterException()	41

6.10. sf::Clock osztályreferencia	42
6.10.1. Tagfüggvények dokumentációja	42
6.10.1.1. getElapsedTime()	42
6.10.1.2. restart()	42
6.11. sf::ClockTime osztályreferencia	42
6.11.1. Konstruktork és destruktorok dokumentációja	42
6.11.1.1. ClockTime() [1/2]	43
6.11.1.2. ClockTime() [2/2]	43
6.11.2. Tagfüggvények dokumentációja	43
6.11.2.1. asSeconds()	43
6.11.2.2. increment()	43
6.11.2.3. reset()	43
6.12. sf::Color osztályreferencia	43
6.12.1. Konstruktork és destruktorok dokumentációja	44
6.12.1.1. Color() [1/3]	44
6.12.1.2. Color() [2/3]	44
6.12.1.3. Color() [3/3]	44
6.12.2. Adattagok dokumentációja	44
6.12.2.1. a	45
6.12.2.2. b	45
6.12.2.3. Black	45
6.12.2.4. Blue	45
6.12.2.5. g	45
6.12.2.6. Green	45
6.12.2.7. r	45
6.12.2.8. Red	45
6.12.2.9. Transparent	46
6.12.2.10. White	46
6.13. creature::Crocodile osztályreferencia	46
6.13.1. Részletes leírás	47
6.13.2. Konstruktork és destruktorok dokumentációja	47
6.13.2.1. Crocodile()	47
6.13.2.2. ~Crocodile()	47
6.13.3. Tagfüggvények dokumentációja	47
6.13.3.1. die()	47
6.13.3.2. draw_logic()	48
6.13.3.3. get_type()	48
6.13.3.4. select_target()	48
6.13.3.5. update_logic()	49
6.14. sf::Event osztályreferencia	49
6.14.1. Enumeráció-tagok dokumentációja	49
6.14.1.1. EType	49

6.14.2. Konstruktork és destruktorok dokumentációja	50
6.14.2.1. Event()	50
6.14.3. Adattagok dokumentációja	50
6.14.3.1. type	50
6.15. creature::Farmer osztályreferencia	50
6.15.1. Részletes leírás	51
6.15.2. Konstruktork és destruktorok dokumentációja	51
6.15.2.1. Farmer()	51
6.15.2.2. ~Farmer()	51
6.15.3. Tagfüggvények dokumentációja	51
6.15.3.1. update_logic()	52
6.16. creature::Fisherman osztályreferencia	52
6.16.1. Részletes leírás	53
6.16.2. Konstruktork és destruktorok dokumentációja	53
6.16.2.1. Fisherman()	53
6.16.2.2. ~Fisherman()	53
6.16.3. Tagfüggvények dokumentációja	54
6.16.3.1. try_fishing()	54
6.16.3.2. update_logic()	54
6.16.4. Adattagok dokumentációja	54
6.16.4.1. fishing	54
6.17. sf::FloatRect osztályreferencia	55
6.17.1. Konstruktork és destruktorok dokumentációja	55
6.17.1.1. FloatRect() [1/2]	55
6.17.1.2. FloatRect() [2/2]	55
6.17.2. Tagfüggvények dokumentációja	55
6.17.2.1. contains()	55
6.17.3. Adattagok dokumentációja	56
6.17.3.1. height	56
6.17.3.2. left	56
6.17.3.3. top	56
6.17.3.4. width	56
6.18. GameConfig osztályreferencia	56
6.18.1. Részletes leírás	57
6.18.2. Konstruktork és destruktorok dokumentációja	58
6.18.2.1. GameConfig()	58
6.18.3. Tagfüggvények dokumentációja	58
6.18.3.1. get_config_level()	58
6.18.3.2. get_instance()	58
6.18.3.3. get_screen_height()	58
6.18.3.4. get_screen_width()	59
6.18.3.5. get_target_fps()	59

6.18.3.6. <code>get_world_size()</code> . . . . .	59
6.18.3.7. <code>operator=()</code> . . . . .	59
6.18.3.8. <code>set_config_level()</code> . . . . .	59
6.18.3.9. <code>set_world_size()</code> . . . . .	59
6.18.4. Adattagok dokumentációja . . . . .	60
6.18.4.1. <code>berry_growth_time</code> . . . . .	60
6.18.4.2. <code>day_length</code> . . . . .	60
6.18.4.3. <code>hostiles_count</code> . . . . .	60
6.18.4.4. <code>max_city_size</code> . . . . .	60
6.18.4.5. <code>max_spawn_tries</code> . . . . .	60
6.18.4.6. <code>max_tree_age</code> . . . . .	61
6.18.4.7. <code>resource_scarcity</code> . . . . .	61
6.19. GameManager osztályreferencia . . . . .	61
6.19.1. Részletes leírás . . . . .	62
6.19.2. Konstruktork és destruktorok dokumentációja . . . . .	62
6.19.2.1. <code>GameManager()</code> . . . . .	62
6.19.2.2. <code>~GameManager()</code> . . . . .	62
6.19.3. Tagfüggvények dokumentációja . . . . .	62
6.19.3.1. <code>draw_buttons()</code> . . . . .	62
6.19.3.2. <code>game_loop()</code> . . . . .	62
6.19.3.3. <code>get_elapsed_time()</code> . . . . .	63
6.19.3.4. <code>is_valid()</code> . . . . .	63
6.19.3.5. <code>run()</code> . . . . .	63
6.19.3.6. <code>setup_buttons()</code> . . . . .	63
6.19.3.7. <code>simulate_tick()</code> . . . . .	63
6.19.3.8. <code>update_buttons()</code> . . . . .	63
6.20. <code>creature::Goat</code> osztályreferencia . . . . .	64
6.20.1. Részletes leírás . . . . .	64
6.20.2. Konstruktork és destruktorok dokumentációja . . . . .	64
6.20.2.1. <code>Goat()</code> . . . . .	64
6.20.2.2. <code>~Goat()</code> . . . . .	65
6.20.3. Tagfüggvények dokumentációja . . . . .	65
6.20.3.1. <code>die()</code> . . . . .	65
6.20.3.2. <code>draw_logic()</code> . . . . .	65
6.20.3.3. <code>get_type()</code> . . . . .	66
6.20.3.4. <code>update_logic()</code> . . . . .	66
6.21. <code>creature::HostileInterface</code> osztályreferencia . . . . .	66
6.21.1. Részletes leírás . . . . .	67
6.21.2. Konstruktork és destruktorok dokumentációja . . . . .	67
6.21.2.1. <code>~HostileInterface()</code> . . . . .	68
6.21.3. Tagfüggvények dokumentációja . . . . .	68
6.21.3.1. <code>check_aggroed()</code> . . . . .	68



6.21.3.2. hostile_run()	68
6.21.3.3. hostile_walk()	68
6.21.3.4. retarget()	68
6.21.3.5. select_target()	69
6.21.3.6. set_hostile_config()	69
6.21.3.7. try_attack()	69
6.21.4. Adattagok dokumentációja	69
6.21.4.1. attack_speed	70
6.21.4.2. damage	70
6.21.4.3. goal	70
6.21.4.4. target	70
6.22. minerals::House osztályreferencia	70
6.22.1. Részletes leírás	71
6.22.2. Konstruktork és destruktorok dokumentációja	71
6.22.2.1. House()	71
6.22.3. Tagfüggvények dokumentációja	71
6.22.3.1. get_type()	72
6.22.3.2. update_logic()	72
6.22.4. Adattagok dokumentációja	72
6.22.4.1. iron_req	72
6.22.4.2. level	72
6.22.4.3. stone_req	72
6.22.4.4. wood_req	73
6.23. creature::Human osztályreferencia	73
6.23.1. Részletes leírás	74
6.23.2. Konstruktork és destruktorok dokumentációja	74
6.23.2.1. Human() [1/2]	74
6.23.2.2. Human() [2/2]	74
6.23.2.3. ~Human()	75
6.23.3. Tagfüggvények dokumentációja	75
6.23.3.1. die()	75
6.23.3.2. draw_logic()	75
6.23.3.3. get_profession_string()	76
6.23.3.4. get_type()	76
6.23.3.5. initialize()	76
6.23.3.6. select_texture()	76
6.23.3.7. update_logic()	78
6.23.4. Adattagok dokumentációja	78
6.23.4.1. goal	78
6.23.4.2. needs_promotion	78
6.23.4.3. needs_to_be_royal	79
6.23.4.4. profession	79

6.24. ImportErrorException osztályreferencia	79
6.24.1. Részletes leírás	79
6.24.2. Konstruktorok és destruktorok dokumentációja	79
6.24.2.1. ImportErrorException()	80
6.25. ImportErrorHousingLevelException osztályreferencia	80
6.25.1. Részletes leírás	80
6.25.2. Konstruktorok és destruktorok dokumentációja	80
6.25.2.1. ImportErrorHousingLevelException()	80
6.26. ImportErrorHumanProfessionException osztályreferencia	81
6.26.1. Részletes leírás	81
6.26.2. Konstruktorok és destruktorok dokumentációja	81
6.26.2.1. ImportErrorHumanProfessionException()	81
6.27. ImportErrorResourceException osztályreferencia	81
6.27.1. Részletes leírás	82
6.27.2. Konstruktorok és destruktorok dokumentációja	82
6.27.2.1. ImportErrorResourceException()	82
6.28. sf::IntRect osztályreferencia	82
6.28.1. Konstruktorok és destruktorok dokumentációja	82
6.28.1.1. IntRect() [1/2]	83
6.28.1.2. IntRect() [2/2]	83
6.28.2. Adattagok dokumentációja	83
6.28.2.1. height	83
6.28.2.2. left	83
6.28.2.3. top	83
6.28.2.4. width	83
6.29. InvalidBorderSizeException osztályreferencia	84
6.29.1. Részletes leírás	84
6.29.2. Konstruktorok és destruktorok dokumentációja	84
6.29.2.1. InvalidBorderSizeException()	84
6.30. minerals::Iron osztályreferencia	84
6.30.1. Részletes leírás	85
6.30.2. Konstruktorok és destruktorok dokumentációja	85
6.30.2.1. Iron()	85
6.30.3. Tagfüggvények dokumentációja	85
6.30.3.1. get_type()	85
6.30.3.2. harvest()	86
6.30.3.3. update_logic()	86
6.31. sf::Keyboard osztályreferencia	87
6.31.1. Enumeráció-tagok dokumentációja	87
6.31.1.1. Keydowntype	87
6.31.2. Tagfüggvények dokumentációja	87
6.31.2.1. isKeyPressed()	88

6.31.2.2. simulate_key_press()	88
6.31.2.3. simulate_key_release()	88
6.32. creature::KillerRobot osztályreferencia	88
6.32.1. Részletes leírás	89
6.32.2. Konstruktork és destruktorok dokumentációja	89
6.32.2.1. KillerRobot()	89
6.32.2.2. ~KillerRobot()	90
6.32.3. Tagfüggvények dokumentációja	90
6.32.3.1. die()	90
6.32.3.2. draw_logic()	90
6.32.3.3. get_type()	90
6.32.3.4. select_target()	91
6.32.3.5. update_logic()	91
6.33. creature::King osztályreferencia	91
6.33.1. Részletes leírás	92
6.33.2. Konstruktork és destruktorok dokumentációja	92
6.33.2.1. King()	92
6.33.2.2. ~King()	93
6.33.3. Tagfüggvények dokumentációja	93
6.33.3.1. update_logic()	93
6.34. creature::Living osztályreferencia	93
6.34.1. Részletes leírás	96
6.34.2. Konstruktork és destruktorok dokumentációja	96
6.34.2.1. ~Living()	96
6.34.3. Tagfüggvények dokumentációja	96
6.34.3.1. apply_age()	96
6.34.3.2. check_aggroed()	97
6.34.3.3. damage()	97
6.34.3.4. die()	97
6.34.3.5. draw()	97
6.34.3.6. draw_logic()	98
6.34.3.7. get_gender()	98
6.34.3.8. get_state()	98
6.34.3.9. get_type()	99
6.34.3.10. get_width()	99
6.34.3.11. init_spritesheet_data()	99
6.34.3.12. look_left()	99
6.34.3.13. look_right()	100
6.34.3.14. needs_drawn()	100
6.34.3.15. retarget()	100
6.34.3.16. set_attack_texture()	100
6.34.3.17. set_death_texture()	101

6.34.3.18.set_health()	101
6.34.3.19.set_idle_texture()	101
6.34.3.20.set_run_texture()	101
6.34.3.21.set_state()	102
6.34.3.22.set_walk_texture()	102
6.34.3.23.setPosition()	102
6.34.3.24.setTexture()	103
6.34.3.25.setTheShadow()	103
6.34.3.26.shadow_logic()	103
6.34.3.27.update_logic()	104
6.34.3.28.update_spritesheet()	104
6.34.4. Adattagok dokumentációja	104
6.34.4.1. damaged_by	104
6.34.4.2. death_timer	105
6.34.4.3. facing	105
6.34.4.4. gender	105
6.34.4.5. health	105
6.34.4.6. hit_timer	105
6.34.4.7. inner_timer	105
6.34.4.8. max_age	106
6.34.4.9. MAX_CREATURE_SIZE	106
6.34.4.10.posx	106
6.34.4.11.posy	106
6.34.4.12.run_speed_modifier	106
6.34.4.13.save_name	106
6.34.4.14.speed	107
6.34.4.15.state	107
6.35. sf::Mouse osztályreferencia	107
6.35.1. Enumeráció-tagok dokumentációja	107
6.35.1.1. Mousedowntype	107
6.35.2. Tagfüggvények dokumentációja	108
6.35.2.1. getPosition()	108
6.35.2.2. isButtonPressed()	108
6.35.2.3. simulate_key_press()	108
6.35.2.4. simulate_key_release()	108
6.36. sf::Music osztályreferencia	108
6.36.1. Konstruktorkok és destruktorkok dokumentációja	109
6.36.1.1. Music()	109
6.36.2. Tagfüggvények dokumentációja	109
6.36.2.1. getStatus()	109
6.36.2.2. openFromFile()	109
6.36.2.3. play()	109

6.36.2.4. setLoop()	109
6.36.2.5. setVolume()	109
6.36.2.6. stop()	110
6.37. MusicLoadException osztályreferencia	110
6.37.1. Részletes leírás	110
6.37.2. Konstruktork és destruktorok dokumentációja	110
6.37.2.1. MusicLoadException()	110
6.38. MediaPlayer osztályreferencia	111
6.38.1. Részletes leírás	111
6.38.2. Konstruktork és destruktorok dokumentációja	111
6.38.2.1. MediaPlayer()	111
6.38.2.2. ~MediaPlayer()	111
6.38.3. Tagfüggvények dokumentációja	111
6.38.3.1. load_music()	111
6.38.3.2. set_volume()	112
6.38.3.3. toggle_music()	112
6.39. gtest_lite::ostreamRedir osztályreferencia	112
6.39.1. Részletes leírás	112
6.39.2. Konstruktork és destruktorok dokumentációja	113
6.39.2.1. ostreamRedir()	113
6.39.2.2. ~ostreamRedir()	113
6.40. PostProcessor osztályreferencia	113
6.40.1. Részletes leírás	114
6.40.2. Konstruktork és destruktorok dokumentációja	114
6.40.2.1. PostProcessor()	114
6.40.3. Tagfüggvények dokumentációja	114
6.40.3.1. draw()	114
6.40.3.2. setColorOverlay()	114
6.40.3.3. setRenderSize()	115
6.40.3.4. setTextureFor()	115
6.40.3.5. toggle_chromatic_aberration()	115
6.40.3.6. toggle_noise()	116
6.40.3.7. toggle_vignette()	116
6.41. Profession osztályreferencia	116
6.41.1. Részletes leírás	117
6.41.2. Konstruktork és destruktorok dokumentációja	117
6.41.2.1. Profession()	117
6.41.3. Tagfüggvények dokumentációja	117
6.41.3.1. draw()	117
6.41.3.2. load_profession()	118
6.41.3.3. setPosition()	118
6.41.3.4. setTexture()	118

6.41.3.5. to_string()	119
6.42. RandomGenerator osztályreferencia	119
6.42.1. Részletes leírás	119
6.42.2. Konstruktork és destruktorok dokumentációja	120
6.42.2.1. RandomGenerator()	120
6.42.3. Tagfüggvények dokumentációja	120
6.42.3.1. get_instance()	120
6.42.3.2. get_random_int()	120
6.42.3.3. operator=()	121
6.43. ReadSaveFileFail osztályreferencia	121
6.43.1. Részletes leírás	121
6.43.2. Konstruktork és destruktorok dokumentációja	121
6.43.2.1. ReadSaveFileFail()	121
6.44. sf::RectangleShape osztályreferencia	122
6.44.1. Tagfüggvények dokumentációja	122
6.44.1.1. setFillColor()	122
6.44.1.2. setPosition()	122
6.44.1.3. setSize()	122
6.44.2. Adattagok dokumentációja	122
6.44.2.1. position	123
6.45. sf::RenderStates osztályreferencia	123
6.45.1. Konstruktork és destruktorok dokumentációja	123
6.45.1.1. RenderStates()	123
6.45.2. Tagfüggvények dokumentációja	123
6.45.2.1. setBlendMode()	123
6.45.2.2. setTransform()	124
6.45.3. Adattagok dokumentációja	124
6.45.3.1. blendMode	124
6.45.3.2. transform	124
6.46. sf::RenderWindow osztályreferencia	124
6.46.1. Konstruktork és destruktorok dokumentációja	125
6.46.1.1. RenderWindow() [1/3]	125
6.46.1.2. RenderWindow() [2/3]	125
6.46.1.3. RenderWindow() [3/3]	125
6.46.2. Tagfüggvények dokumentációja	125
6.46.2.1. clear() [1/2]	125
6.46.2.2. clear() [2/2]	125
6.46.2.3. close()	125
6.46.2.4. create()	126
6.46.2.5. display()	126
6.46.2.6. draw() [1/3]	126
6.46.2.7. draw() [2/3]	126

6.46.2.8. draw() [3/3]	126
6.46.2.9. isOpen()	126
6.46.2.10. pollEvent()	126
6.46.2.11. setFrameRateLimit()	127
6.47. minerals::ResourceStructure osztályreferencia	127
6.47.1. Részletes leírás	128
6.47.2. Konstruktork és destruktorok dokumentációja	128
6.47.2.1. ResourceStructure()	128
6.47.2.2. ~ResourceStructure()	128
6.47.3. Tagfüggvények dokumentációja	128
6.47.3.1. get_harvested()	128
6.47.3.2. harvest()	128
6.47.4. Adattagok dokumentációja	129
6.47.4.1. harvested	129
6.47.4.2. inner_timer	129
6.48. SaveManager osztályreferencia	129
6.48.1. Részletes leírás	129
6.48.2. Konstruktork és destruktorok dokumentációja	129
6.48.2.1. SaveManager()	129
6.48.3. Tagfüggvények dokumentációja	130
6.48.3.1. deleteFile()	130
6.48.3.2. loadFile()	130
6.48.3.3. saveFile()	130
6.49. Shadowable osztályreferencia	131
6.49.1. Részletes leírás	132
6.49.2. Konstruktork és destruktorok dokumentációja	132
6.49.2.1. ~Shadowable()	132
6.49.3. Tagfüggvények dokumentációja	132
6.49.3.1. drawShadow()	132
6.49.3.2. get_height_offset()	132
6.49.3.3. get_shadow_strength()	133
6.49.3.4. get_skew_offset()	133
6.49.3.5. set_height_offset()	133
6.49.3.6. set_shadow_strength()	133
6.49.3.7. set_skew_offset()	134
6.49.3.8. setShadow()	134
6.49.3.9. setShadowDayNightCycle()	134
6.49.3.10. setShadowPosition()	134
6.49.3.11. setShadowTexture()	135
6.49.4. Adattagok dokumentációja	135
6.49.4.1. height_offset	135
6.50. SimulationException osztályreferencia	135

6.50.1. Részletes leírás	136
6.50.2. Konstruktork és destruktorok dokumentációja	136
6.50.2.1. SimulationException()	136
6.51. creature::Soldier osztályreferencia	136
6.51.1. Részletes leírás	137
6.51.2. Konstruktork és destruktorok dokumentációja	137
6.51.2.1. Soldier()	137
6.51.2.2. ~Soldier()	137
6.51.3. Tagfüggvények dokumentációja	137
6.51.3.1. update_logic()	137
6.52. sf::Sound osztályreferencia	138
6.52.1. Konstruktork és destruktorok dokumentációja	138
6.52.1.1. ~Sound()	138
6.52.2. Tagfüggvények dokumentációja	138
6.52.2.1. play()	138
6.52.2.2. setBuffer()	139
6.52.2.3. stop()	139
6.53. sf::SoundBuffer osztályreferencia	139
6.53.1. Tagfüggvények dokumentációja	139
6.53.1.1. loadFromFile()	139
6.54. SoundPlayer osztályreferencia	139
6.54.1. Részletes leírás	140
6.54.2. Tagfüggvények dokumentációja	140
6.54.2.1. load_sound()	140
6.54.2.2. play_sound()	140
6.54.2.3. stop_sound()	141
6.55. sf::SoundSource osztályreferencia	141
6.55.1. Enumeráció-tagok dokumentációja	141
6.55.1.1. SoundSourceType	141
6.55.2. Konstruktork és destruktorok dokumentációja	142
6.55.2.1. SoundSource()	142
6.55.2.2. ~SoundSource()	142
6.55.3. Adattagok dokumentációja	142
6.55.3.1. type	142
6.56. sf::Sprite osztályreferencia	142
6.56.1. Konstruktork és destruktorok dokumentációja	143
6.56.1.1. Sprite()	143
6.56.1.2. ~Sprite()	143
6.56.2. Tagfüggvények dokumentációja	143
6.56.2.1. draw()	143
6.56.2.2. getGlobalBounds() [1/2]	143
6.56.2.3. getGlobalBounds() [2/2]	143



6.56.2.4. <code>getLocalBounds()</code>	143
6.56.2.5. <code>getPosition()</code>	144
6.56.2.6. <code>getTexture()</code>	144
6.56.2.7. <code>setColor()</code>	144
6.56.2.8. <code>setOrigin()</code>	144
6.56.2.9. <code>setPosition()</code>	144
6.56.2.10. <code>setRotation()</code>	144
6.56.2.11. <code>setScale()</code>	144
6.56.2.12. <code>setTexture()</code>	145
6.56.2.13. <code>setTextureRect()</code>	145
6.57. <code>minerals::Stone</code> osztályreferencia	145
6.57.1. Részletes leírás	146
6.57.2. Konstruktork és destruktorok dokumentációja	146
6.57.2.1. <code>Stone()</code>	146
6.57.3. Tagfüggvények dokumentációja	146
6.57.3.1. <code>get_type()</code>	146
6.57.3.2. <code>harvest()</code>	146
6.57.3.3. <code>update_logic()</code>	146
6.58. <code>creature::Stonemason</code> osztályreferencia	147
6.58.1. Részletes leírás	148
6.58.2. Konstruktork és destruktorok dokumentációja	148
6.58.2.1. <code>Stonemason()</code>	148
6.58.2.2. <code>~Stonemason()</code>	148
6.58.3. Tagfüggvények dokumentációja	148
6.58.3.1. <code>try_mine()</code>	149
6.58.3.2. <code>update_logic()</code>	149
6.58.4. Adattagok dokumentációja	149
6.58.4.1. <code>mining_iron</code>	149
6.59. <code>minerals::Structure</code> osztályreferencia	150
6.59.1. Részletes leírás	151
6.59.2. Konstruktork és destruktorok dokumentációja	151
6.59.2.1. <code>Structure()</code>	151
6.59.2.2. <code>~Structure()</code>	151
6.59.3. Tagfüggvények dokumentációja	151
6.59.3.1. <code>draw()</code>	151
6.59.3.2. <code>draw_logic()</code>	152
6.59.3.3. <code>get_type()</code>	152
6.59.3.4. <code>needs_drawn()</code>	152
6.59.3.5. <code>setPosition()</code>	152
6.59.3.6. <code>setTexture()</code>	153
6.59.3.7. <code>update_logic()</code>	153
6.59.4. Adattagok dokumentációja	153

6.59.4.1. MAX_OBJECT_SIZE	154
6.59.4.2. posix	154
6.59.4.3. posix	154
6.60. StructureException osztályreferencia	154
6.60.1. Részletes leírás	155
6.60.2. Konstruktork és destruktorok dokumentációja	155
6.60.2.1. StructureException()	155
6.61. TerrainContainer< T > osztálysablon-referencia	155
6.61.1. Részletes leírás	156
6.61.2. Konstruktork és destruktorok dokumentációja	156
6.61.2.1. TerrainContainer() [1/2]	156
6.61.2.2. TerrainContainer() [2/2]	157
6.61.2.3. ~TerrainContainer()	157
6.61.3. Tagfüggvények dokumentációja	157
6.61.3.1. clear()	157
6.61.3.2. clear_at()	157
6.61.3.3. draw()	158
6.61.3.4. generate_world()	158
6.61.3.5. get_height()	158
6.61.3.6. get_seed()	159
6.61.3.7. get_width()	159
6.61.3.8. is_on_screen()	159
6.61.3.9. is_valid_coordinate()	159
6.61.3.10. operator[]() [1/2]	160
6.61.3.11. operator[]() [2/2]	160
6.61.3.12. resize()	160
6.61.3.13. set_seed()	161
6.61.3.14. swap_at()	161
6.61.4. Adattagok dokumentációja	161
6.61.4.1. TILE_SIZE	161
6.62. gtest_lite::Test struktúráreferencia	162
6.62.1. Részletes leírás	163
6.62.2. Konstruktork és destruktorok dokumentációja	163
6.62.2.1. ~Test()	163
6.62.3. Tagfüggvények dokumentációja	163
6.62.3.1. astatus()	163
6.62.3.2. begin()	163
6.62.3.3. end()	163
6.62.3.4. expect()	164
6.62.3.5. fail()	164
6.62.3.6. getTest()	164
6.62.4. Adattagok dokumentációja	164

6.62.4.1. ablocks	164
6.62.4.2. failed	164
6.62.4.3. name	165
6.62.4.4. null	165
6.62.4.5. os	165
6.62.4.6. status	165
6.62.4.7. sum	165
6.62.4.8. tmp	165
6.63. sf::Texture osztályreferencia	166
6.63.1. Konstruktork és destruktorok dokumentációja	166
6.63.1.1. Texture()	166
6.63.2. Tagfüggvények dokumentációja	166
6.63.2.1. getSize()	166
6.63.2.2. loadFromFile()	166
6.64. Textureable osztályreferencia	167
6.64.1. Részletes leírás	167
6.64.2. Konstruktork és destruktorok dokumentációja	167
6.64.2.1. ~Textureable()	167
6.64.3. Tagfüggvények dokumentációja	167
6.64.3.1. draw()	167
6.64.3.2. setPosition()	168
6.64.3.3. setTexture()	168
6.65. TextureManager osztályreferencia	169
6.65.1. Részletes leírás	169
6.65.2. Tagfüggvények dokumentációja	169
6.65.2.1. clear()	169
6.65.2.2. getInstance()	169
6.65.2.3. loadTexture()	169
6.66. tiles::Tile osztályreferencia	170
6.66.1. Részletes leírás	170
6.66.2. Tagfüggvények dokumentációja	170
6.66.2.1. draw()	170
6.66.2.2. get_type()	171
6.66.2.3. init()	171
6.66.2.4. setPosition()	171
6.66.2.5. setTexture()	172
6.67. sf::Transform osztályreferencia	172
6.67.1. Konstruktork és destruktorok dokumentációja	172
6.67.1.1. Transform() [1/2]	173
6.67.1.2. Transform() [2/2]	173
6.67.2. Tagfüggvények dokumentációja	173
6.67.2.1. combine()	173

6.67.2.2. transformPoint()	173
6.67.2.3. translate() [1/2]	173
6.67.2.4. translate() [2/2]	174
6.67.3. Adattagok dokumentációja	174
6.67.3.1. matrix	174
6.68. minerals::Tree osztályreferencia	174
6.68.1. Részletes leírás	175
6.68.2. Konstruktork és destruktorok dokumentációja	175
6.68.2.1. Tree()	175
6.68.3. Tagfüggvények dokumentációja	175
6.68.3.1. get_type()	175
6.68.3.2. harvest()	175
6.68.3.3. update_logic()	175
6.69. sf::Vector2f osztályreferencia	176
6.69.1. Konstruktork és destruktorok dokumentációja	176
6.69.1.1. Vector2f() [1/2]	176
6.69.1.2. Vector2f() [2/2]	176
6.69.2. Adattagok dokumentációja	176
6.69.2.1. x	177
6.69.2.2. y	177
6.70. sf::Vector2i osztályreferencia	177
6.70.1. Konstruktork és destruktorok dokumentációja	177
6.70.1.1. Vector2i() [1/2]	177
6.70.1.2. Vector2i() [2/2]	177
6.70.2. Adattagok dokumentációja	178
6.70.2.1. x	178
6.70.2.2. y	178
6.71. sf::VideoMode osztályreferencia	178
6.71.1. Konstruktork és destruktorok dokumentációja	178
6.71.1.1. VideoMode()	179
6.71.2. Tagfüggvények dokumentációja	179
6.71.2.1. getDesktopMode()	179
6.71.2.2. isValid()	179
6.71.3. Adattagok dokumentációja	179
6.71.3.1. bitsPerPixel	179
6.71.3.2. height	179
6.71.3.3. width	179
6.72. creature::Woodcutter osztályreferencia	180
6.72.1. Részletes leírás	180
6.72.2. Konstruktork és destruktorok dokumentációja	180
6.72.2.1. Woodcutter()	180
6.72.2.2. ~Woodcutter()	181

6.72.3. Tagfüggvények dokumentációja	181
6.72.3.1. update_logic()	181
6.73. World osztályreferencia	181
6.73.1. Részletes leírás	183
6.73.2. Konstruktorkok és destruktorkok dokumentációja	183
6.73.2.1. World()	183
6.73.2.2. ~World()	183
6.73.3. Tagfüggvények dokumentációja	183
6.73.3.1. add_resources()	183
6.73.3.2. build_city_center_at()	184
6.73.3.3. clear()	184
6.73.3.4. draw()	184
6.73.3.5. get_border_height()	185
6.73.3.6. get_border_width()	185
6.73.3.7. get_current_city_center()	185
6.73.3.8. get_excluded_entities()	185
6.73.3.9. get_position_nearby_town()	186
6.73.3.10. get_random_house_pos()	186
6.73.3.11. get_random_suitable_position()	186
6.73.3.12. get_structure_type()	187
6.73.3.13. getTileAt()	187
6.73.3.14. is_there_enough_resource()	187
6.73.3.15. populate_world()	188
6.73.3.16. regenerate()	188
6.73.3.17. remove_resources()	188
6.73.3.18. remove_structure_at()	188
6.73.3.19. set_border_height()	189
6.73.3.20. set_border_width()	189
6.73.3.21. spawn_entity()	189
6.73.3.22. spawn_structure()	190
6.73.3.23. spawn_structure_at()	190
6.73.3.24. try_develop_random_role()	191
6.73.3.25. update_world()	191
6.73.3.26. upgrade_house_at()	191
6.73.4. Barát és kapcsolódó függvények dokumentációja	191
6.73.4.1. operator<<	191
6.73.4.2. operator>>	192
<b>7. Fájlok dokumentációja</b>	<b>193</b>
7.1. src/creatures/Goat.cpp fájlreferencia	193
7.2. src/creatures/Goat.d fájlreferencia	193
7.3. src/creatures/Goat.hpp fájlreferencia	193

7.3.1. Részletes leírás . . . . .	194
7.4. src/creatures/HostileInterface.cpp fájlreferencia . . . . .	194
7.5. src/creatures/HostileInterface.d fájlreferencia . . . . .	194
7.6. src/creatures/HostileInterface.hpp fájlreferencia . . . . .	194
7.6.1. Részletes leírás . . . . .	195
7.7. src/creatures/hostiles/Bear.cpp fájlreferencia . . . . .	195
7.8. src/creatures/hostiles/Bear.d fájlreferencia . . . . .	195
7.9. src/creatures/hostiles/Bear.hpp fájlreferencia . . . . .	195
7.9.1. Részletes leírás . . . . .	196
7.10. src/creatures/hostiles/Crocodile.cpp fájlreferencia . . . . .	196
7.11. src/creatures/hostiles/Crocodile.d fájlreferencia . . . . .	196
7.12. src/creatures/hostiles/Crocodile.hpp fájlreferencia . . . . .	196
7.12.1. Részletes leírás . . . . .	197
7.13. src/creatures/hostiles/KillerRobot.cpp fájlreferencia . . . . .	197
7.14. src/creatures/hostiles/KillerRobot.d fájlreferencia . . . . .	197
7.15. src/creatures/hostiles/KillerRobot.hpp fájlreferencia . . . . .	197
7.15.1. Részletes leírás . . . . .	198
7.16. src/creatures/humans/AnglerMiner.cpp fájlreferencia . . . . .	198
7.17. src/creatures/humans/AnglerMiner.d fájlreferencia . . . . .	198
7.18. src/creatures/humans/AnglerMiner.hpp fájlreferencia . . . . .	198
7.18.1. Részletes leírás . . . . .	199
7.19. src/creatures/humans/Builder.cpp fájlreferencia . . . . .	199
7.20. src/creatures/humans/Builder.d fájlreferencia . . . . .	199
7.21. src/creatures/humans/Builder.hpp fájlreferencia . . . . .	199
7.21.1. Részletes leírás . . . . .	200
7.22. src/creatures/humans/Farmer.cpp fájlreferencia . . . . .	200
7.23. src/creatures/humans/Farmer.d fájlreferencia . . . . .	200
7.24. src/creatures/humans/Farmer.hpp fájlreferencia . . . . .	200
7.24.1. Részletes leírás . . . . .	201
7.25. src/creatures/humans/Fisherman.cpp fájlreferencia . . . . .	201
7.26. src/creatures/humans/Fisherman.d fájlreferencia . . . . .	201
7.27. src/creatures/humans/Fisherman.hpp fájlreferencia . . . . .	201
7.27.1. Részletes leírás . . . . .	202
7.28. src/creatures/humans/Human.cpp fájlreferencia . . . . .	202
7.29. src/creatures/humans/Human.d fájlreferencia . . . . .	202
7.30. src/creatures/humans/Human.hpp fájlreferencia . . . . .	202
7.30.1. Részletes leírás . . . . .	203
7.31. src/creatures/humans/King.cpp fájlreferencia . . . . .	203
7.32. src/creatures/humans/King.d fájlreferencia . . . . .	203
7.33. src/creatures/humans/King.hpp fájlreferencia . . . . .	203
7.33.1. Részletes leírás . . . . .	204
7.34. src/creatures/humans/Soldier.cpp fájlreferencia . . . . .	204

7.35. src/creatures/humans/Soldier.d fájlreferencia . . . . .	204
7.36. src/creatures/humans/Soldier.hpp fájlreferencia . . . . .	204
7.36.1. Részletes leírás . . . . .	205
7.37. src/creatures/humans/Stonemason.cpp fájlreferencia . . . . .	205
7.38. src/creatures/humans/Stonemason.d fájlreferencia . . . . .	205
7.39. src/creatures/humans/Stonemason.hpp fájlreferencia . . . . .	205
7.39.1. Részletes leírás . . . . .	206
7.40. src/creatures/humans/Woodcutter.cpp fájlreferencia . . . . .	206
7.41. src/creatures/humans/Woodcutter.d fájlreferencia . . . . .	206
7.42. src/creatures/humans/Woodcutter.hpp fájlreferencia . . . . .	206
7.42.1. Részletes leírás . . . . .	207
7.43. src/creatures/Living.cpp fájlreferencia . . . . .	207
7.44. src/creatures/Living.d fájlreferencia . . . . .	207
7.45. src/creatures/Living.hpp fájlreferencia . . . . .	207
7.45.1. Részletes leírás . . . . .	208
7.46. src/exceptions/FileExceptions.hpp fájlreferencia . . . . .	208
7.46.1. Részletes leírás . . . . .	208
7.47. src/exceptions/MusicLoadException.hpp fájlreferencia . . . . .	209
7.47.1. Részletes leírás . . . . .	209
7.48. src/exceptions/SimulationException.hpp fájlreferencia . . . . .	209
7.48.1. Részletes leírás . . . . .	209
7.49. src/exceptions/WorldExceptions.hpp fájlreferencia . . . . .	210
7.49.1. Részletes leírás . . . . .	210
7.50. src/external/gtest_lite.h fájlreferencia . . . . .	210
7.50.1. Részletes leírás . . . . .	213
7.50.2. Makródefiníciók dokumentációja . . . . .	213
7.50.2.1. ADD_FAILURE . . . . .	213
7.50.2.2. ASSERT_ . . . . .	214
7.50.2.3. ASSERT_EQ . . . . .	214
7.50.2.4. ASSERT_NO_THROW [1/2] . . . . .	214
7.50.2.5. ASSERT_NO_THROW [2/2] . . . . .	214
7.50.2.6. ASSERTTHROW . . . . .	214
7.50.2.7. CREATE_Has_ . . . . .	215
7.50.2.8. CREATE_Has_fn_ . . . . .	215
7.50.2.9. END . . . . .	215
7.50.2.10. ENDM . . . . .	215
7.50.2.11. ENDMsg . . . . .	215
7.50.2.12. EXPECT_ANY_THROW . . . . .	215
7.50.2.13. EXPECT_DOUBLE_EQ . . . . .	215
7.50.2.14. EXPECT_ENVCASEEQ . . . . .	216
7.50.2.15. EXPECT_ENVEQ . . . . .	216
7.50.2.16. EXPECT_EQ . . . . .	216

7.50.2.17.EXPECT_FALSE	216
7.50.2.18.EXPECT_FLOAT_EQ	216
7.50.2.19.EXPECT_GE	216
7.50.2.20.EXPECT_GT	217
7.50.2.21.EXPECT_LE	217
7.50.2.22.EXPECT_LT	217
7.50.2.23.EXPECT_NE	217
7.50.2.24.EXPECT_NO_THROW	217
7.50.2.25.EXPECT_STRCASEEQ	217
7.50.2.26.EXPECT_STRCASENE	217
7.50.2.27.EXPECT_STREQ	218
7.50.2.28.EXPECT_STRNE	218
7.50.2.29.EXPECT_THROW	218
7.50.2.30.EXPECT_THROW_THROW	218
7.50.2.31.EXPECT_TRUE	218
7.50.2.32.EXPECTTHROW	218
7.50.2.33.Nem célszerű közvetlenül használni, vagy módosítani	219
7.50.2.34.FAIL	219
7.50.2.35.GTEND	219
7.50.2.36.GTINIT	219
7.50.2.37.SUCCEED	219
7.50.2.38.TEST	219
7.50.3. Függvények dokumentációja	219
7.50.3.1. hasMember()	219
7.51. src/external/memtrace.cpp fájlreferencia	220
7.52. src/external/memtrace.d fájlreferencia	220
7.53. src/external/memtrace.h fájlreferencia	220
7.54. src/fake_sfml/fake_sfml.cpp fájlreferencia	220
7.55. src/fake_sfml/fake_sfml.d fájlreferencia	220
7.56. src/fake_sfml/fake_sfml.hpp fájlreferencia	220
7.57. src/GameConfig.cpp fájlreferencia	221
7.58. src/GameConfig.d fájlreferencia	221
7.59. src/GameConfig.hpp fájlreferencia	221
7.59.1. Részletes leírás	221
7.60. src/GameManager.cpp fájlreferencia	222
7.61. src/GameManager.d fájlreferencia	222
7.62. src/GameManager.hpp fájlreferencia	222
7.62.1. Részletes leírás	222
7.63. src/main.cpp fájlreferencia	222
7.63.1. Függvények dokumentációja	223
7.63.1.1. main()	223
7.64. src/main.d fájlreferencia	223



7.65. src/MusicPlayer.cpp fájlreferencia . . . . .	223
7.66. src/MusicPlayer.d fájlreferencia . . . . .	223
7.67. src/MusicPlayer.hpp fájlreferencia . . . . .	223
7.67.1. Részletes leírás . . . . .	223
7.68. src/PostProcessor.cpp fájlreferencia . . . . .	223
7.69. src/PostProcessor.d fájlreferencia . . . . .	223
7.70. src/PostProcessor.hpp fájlreferencia . . . . .	223
7.70.1. Részletes leírás . . . . .	224
7.71. src/Profession.cpp fájlreferencia . . . . .	224
7.72. src/Profession.d fájlreferencia . . . . .	224
7.73. src/Profession.hpp fájlreferencia . . . . .	224
7.73.1. Részletes leírás . . . . .	224
7.74. src/Random_Gen.cpp fájlreferencia . . . . .	224
7.75. src/Random_Gen.d fájlreferencia . . . . .	225
7.76. src/Random_Gen.hpp fájlreferencia . . . . .	225
7.76.1. Részletes leírás . . . . .	225
7.77. src/SaveHelpers.cpp fájlreferencia . . . . .	225
7.77.1. Változók dokumentációja . . . . .	225
7.77.1.1. creature_factory . . . . .	226
7.77.1.2. human_factory . . . . .	226
7.77.1.3. resource_factory . . . . .	226
7.78. src/SaveHelpers.d fájlreferencia . . . . .	226
7.79. src/SaveHelpers.hpp fájlreferencia . . . . .	226
7.79.1. Részletes leírás . . . . .	227
7.79.2. Típusdefiníciók dokumentációja . . . . .	228
7.79.2.1. CreatureFactory . . . . .	228
7.79.2.2. HumanFactory . . . . .	228
7.79.2.3. ResourceFactory . . . . .	228
7.79.3. Változók dokumentációja . . . . .	228
7.79.3.1. creature_factory . . . . .	228
7.79.3.2. human_factory . . . . .	228
7.79.3.3. resource_factory . . . . .	228
7.80. src/SaveManager.cpp fájlreferencia . . . . .	228
7.81. src/SaveManager.d fájlreferencia . . . . .	228
7.82. src/SaveManager.hpp fájlreferencia . . . . .	228
7.82.1. Részletes leírás . . . . .	229
7.83. src/Shadowable.cpp fájlreferencia . . . . .	229
7.84. src/Shadowable.d fájlreferencia . . . . .	229
7.85. src/Shadowable.hpp fájlreferencia . . . . .	229
7.85.1. Részletes leírás . . . . .	229
7.86. src/SoundPlayer.cpp fájlreferencia . . . . .	229
7.87. src/SoundPlayer.d fájlreferencia . . . . .	230

7.88. src/SoundPlayer.hpp fájlreferencia . . . . .	230
7.88.1. Részletes leírás . . . . .	230
7.89. src/terrain_tiles/Tile.cpp fájlreferencia . . . . .	230
7.90. src/terrain_tiles/Tile.d fájlreferencia . . . . .	230
7.91. src/terrain_tiles/Tile.hpp fájlreferencia . . . . .	230
7.91.1. Részletes leírás . . . . .	231
7.92. src/TerrainContainer.hpp fájlreferencia . . . . .	231
7.92.1. Részletes leírás . . . . .	231
7.93. src/TerrainContainer.inl fájlreferencia . . . . .	231
7.93.1. Makródefiníciók dokumentációja . . . . .	231
7.93.1.1. DYNAMIC2D_INL . . . . .	232
7.94. src/Textureable.hpp fájlreferencia . . . . .	232
7.94.1. Részletes leírás . . . . .	232
7.95. src/TextureManager.cpp fájlreferencia . . . . .	232
7.96. src/TextureManager.d fájlreferencia . . . . .	232
7.97. src/TextureManager.hpp fájlreferencia . . . . .	232
7.97.1. Részletes leírás . . . . .	232
7.98. src/ui/button.cpp fájlreferencia . . . . .	233
7.99. src/ui/button.d fájlreferencia . . . . .	233
7.100src/ui/button.hpp fájlreferencia . . . . .	233
7.100.1.Részletes leírás . . . . .	233
7.101src/Utils.cpp fájlreferencia . . . . .	233
7.101.1.Függvények dokumentációja . . . . .	234
7.101.1.1.distance_to() . . . . .	234
7.102src/Utils.d fájlreferencia . . . . .	234
7.103src/Utils.hpp fájlreferencia . . . . .	234
7.103.1.Részletes leírás . . . . .	234
7.103.2.Függvények dokumentációja . . . . .	235
7.103.2.1.distance_to() . . . . .	235
7.104src/World.cpp fájlreferencia . . . . .	235
7.104.1.Függvények dokumentációja . . . . .	235
7.104.1.1.operator<<() . . . . .	235
7.104.1.2.operator>>() . . . . .	235
7.105src/World.d fájlreferencia . . . . .	236
7.106src/World.hpp fájlreferencia . . . . .	236
7.106.1.Részletes leírás . . . . .	236
7.107src/world_object/BerryBush.cpp fájlreferencia . . . . .	237
7.108src/world_object/BerryBush.d fájlreferencia . . . . .	237
7.109src/world_object/BerryBush.hpp fájlreferencia . . . . .	237
7.109.1.Részletes leírás . . . . .	237
7.110src/world_object/CityCenter.cpp fájlreferencia . . . . .	237
7.111src/world_object/CityCenter.d fájlreferencia . . . . .	237

7.112src/world_object/CityCenter.hpp fájlreferencia . . . . .	237
7.112.1.Részletes leírás . . . . .	238
7.113src/world_object/House.cpp fájlreferencia . . . . .	238
7.114src/world_object/House.d fájlreferencia . . . . .	238
7.115src/world_object/House.hpp fájlreferencia . . . . .	238
7.115.1.Részletes leírás . . . . .	239
7.116src/world_object/Iron.cpp fájlreferencia . . . . .	239
7.117src/world_object/Iron.d fájlreferencia . . . . .	239
7.118src/world_object/Iron.hpp fájlreferencia . . . . .	239
7.118.1.Részletes leírás . . . . .	239
7.119src/world_object/ResourceStructure.cpp fájlreferencia . . . . .	239
7.120src/world_object/ResourceStructure.d fájlreferencia . . . . .	240
7.121src/world_object/ResourceStructure.hpp fájlreferencia . . . . .	240
7.121.1.Részletes leírás . . . . .	240
7.122src/world_object/Stone.cpp fájlreferencia . . . . .	240
7.123src/world_object/Stone.d fájlreferencia . . . . .	240
7.124src/world_object/Stone.hpp fájlreferencia . . . . .	240
7.124.1.Részletes leírás . . . . .	241
7.125src/world_object/Structure.cpp fájlreferencia . . . . .	241
7.126src/world_object/Structure.d fájlreferencia . . . . .	241
7.127src/world_object/Structure.hpp fájlreferencia . . . . .	241
7.127.1.Részletes leírás . . . . .	242
7.128src/world_object/Tree.cpp fájlreferencia . . . . .	242
7.129src/world_object/Tree.d fájlreferencia . . . . .	242
7.130src/world_object/Tree.hpp fájlreferencia . . . . .	242
7.130.1.Részletes leírás . . . . .	242
<b>Tárgymutató</b>	<b>243</b>



# 1. fejezet

## Névtérmutató

### 1.1. Névtérlista

Az összes névtér listája rövid leírásokkal:

<a href="#">creature</a>	Az összes élőlény és entitás ebben a névtérben van . . . . .	13
<a href="#">gtest_lite</a>	Gtest_lite: a keretrendszer függvényinek és objektumainak névtére . . . . .	15
<a href="#">minerals</a>	Az összes struktúra ebben a névtérben van . . . . .	19
<a href="#">sf</a>	. . . . .	20
<a href="#">tiles</a>	Az összes terepkocka elem ebben a névtérben van . . . . .	22
<a href="#">ui</a>	Az összes UI elem ebben a névtérben van . . . . .	23



## 2. fejezet

# Hierarchikus mutató

### 2.1. Osztályhierarchia

Majdnem (de nem teljesen) betűrendbe szedett leszármazási lista:

_Is_Types< F, T > . . . . .	25
sf::Bound . . . . .	33
sf::Clock . . . . .	42
sf::ClockTime . . . . .	42
sf::Color . . . . .	43
sf::Event . . . . .	49
sf::FloatRect . . . . .	55
GameConfig . . . . .	56
GameManager . . . . .	61
sf::IntRect . . . . .	82
sf::Keyboard . . . . .	87
std::logic_error	
ImportInvalidHumanProfessionException . . . . .	81
InvalidBorderSizeException . . . . .	84
sf::Mouse . . . . .	107
sf::Music . . . . .	108
MusicPlayer . . . . .	111
gtest_lite::ostreamRedir . . . . .	112
PostProcessor . . . . .	113
RandomGenerator . . . . .	119
sf::RectangleShape . . . . .	122
sf::RenderStates . . . . .	123
sf::RenderWindow . . . . .	124
std::runtime_error	
SimulationException . . . . .	135
CityCenterException . . . . .	41
ImportInvalidEntityException . . . . .	79
ImportInvalidHousingLevelException . . . . .	80
ImportInvalidResourceException . . . . .	81
MusicLoadException . . . . .	110
ReadSaveFileFail . . . . .	121
StructureException . . . . .	154
SaveManager . . . . .	129
Shadowable . . . . .	131
creature::Living . . . . .	93

creature::Goat . . . . .	64
creature::HostileInterface . . . . .	66
creature::Bear . . . . .	28
creature::Crocodile . . . . .	46
creature::KillerRobot . . . . .	88
creature::Human . . . . .	73
creature::Builder . . . . .	34
creature::Farmer . . . . .	50
creature::Fisherman . . . . .	52
creature::AnglerMiner . . . . .	26
creature::King . . . . .	91
creature::Soldier . . . . .	136
creature::Stonemason . . . . .	147
creature::AnglerMiner . . . . .	26
creature::Woodcutter . . . . .	180
minerals::Structure . . . . .	150
minerals::CityCenter . . . . .	39
minerals::House . . . . .	70
minerals::ResourceStructure . . . . .	127
minerals::BerryBush . . . . .	31
minerals::Iron . . . . .	84
minerals::Stone . . . . .	145
minerals::Tree . . . . .	174
sf::Sound . . . . .	138
sf::SoundBuffer . . . . .	139
SoundPlayer . . . . .	139
sf::SoundSource . . . . .	141
sf::Sprite . . . . .	142
TerrainContainer< T > . . . . .	155
TerrainContainer< tiles::Tile * > . . . . .	155
gtest_lite::Test . . . . .	162
sf::Texture . . . . .	166
Textureable . . . . .	167
Profession . . . . .	116
creature::Living . . . . .	93
minerals::Structure . . . . .	150
tiles::Tile . . . . .	170
ui::Button . . . . .	35
TextureManager . . . . .	169
sf::Transform . . . . .	172
sf::Vector2f . . . . .	176
sf::Vector2i . . . . .	177
sf::VideoMode . . . . .	178
World . . . . .	181



## 3. fejezet

# Osztálymutató

### 3.1. Osztálylista

Az összes osztály, struktúra, unió és interfész listája rövid leírásokkal:

<a href="#">_Is_Types&lt; F, T &gt;</a>	25
Segédsablon típuskonverzió futás közbeni ellenőrzésére	
<a href="#">creature::AnglerMiner</a>	26
Az "AnglerMiner" szakmájú ember osztály leírása	
<a href="#">creature::Bear</a>	28
A medve osztály leírása	
<a href="#">minerals::BerryBush</a>	31
A bokor osztály leírása. Ételt ad, ha kitermelik	
<a href="#">sf::Bound</a>	33
<a href="#">creature::Builder</a>	34
Az építész szakmájú ember osztály leírása	
<a href="#">ui::Button</a>	35
A gomb osztály leírása. Tárolja a gomb méretét és azt, hogy mit csinál, ha rákattintanak	
<a href="#">minerals::CityCenter</a>	39
A városközpont osztály leírása. E köré épülnek a házak	
<a href="#">CityCenterException</a>	41
Akkor kell dobni, ha a városközpont hibásan működött	
<a href="#">sf::Clock</a>	42
<a href="#">sf::ClockTime</a>	42
<a href="#">sf::Color</a>	43
<a href="#">creature::Crocodile</a>	46
A krokodil osztály leírása	
<a href="#">sf::Event</a>	49
<a href="#">creature::Farmer</a>	50
A farmer szakmájú ember osztály leírása	
<a href="#">creature::Fisherman</a>	52
A halász szakmájú ember osztály leírása	
<a href="#">sf::FloatRect</a>	55
<a href="#">GameConfig</a>	56
A világ szimulációjának leírása	
<a href="#">GameManager</a>	61
A világ szimulálásáért és a kirazolás irányításáért felelős osztály	
<a href="#">creature::Goat</a>	64
A kecske osztály leírása	
<a href="#">creature::HostileInterface</a>	66
A vadállat entitások interface leírása	

<a href="#">minerals::House</a>	
A ház osztály leírása. Szinttől függően idéz embereket	70
<a href="#">creature::Human</a>	
Az alap ember osztály leírása. Minden fajta szakmájú ember innen öröklődik	73
<a href="#">ImportInvalidEntityException</a>	
Akkor kell dobni, ha egy entitás hibásan lett beolvasva	79
<a href="#">ImportInvalidHousingLevelException</a>	
Akkor kell dobni, ha egy ház hibásan lett beolvasva	80
<a href="#">ImportInvalidHumanProfessionException</a>	
Akkor kell dobni, ha egy szakma hibásan lett beolvasva	81
<a href="#">ImportInvalidResourceException</a>	
Akkor kell dobni, ha egy erőforrás hibásan lett beolvasva	81
<a href="#">sf::IntRect</a>	82
<a href="#">InvalidBorderSizeException</a>	
Akkor kell dobni, ha egy világhatárnak nem jó értéket akarnak beállítani	84
<a href="#">minerals::Iron</a>	
A vasérc osztály leírása. Vasat ad, amikor kitermelik	84
<a href="#">sf::Keyboard</a>	87
<a href="#">creature::KillerRobot</a>	
A gyilkos robot osztály leírása	88
<a href="#">creature::King</a>	
A király szakmájú ember osztály leírása	91
<a href="#">creature::Living</a>	
Az élő entitások interface leírása	93
<a href="#">sf::Mouse</a>	107
<a href="#">sf::Music</a>	108
<a href="#">MusicLoadException</a>	
Akkor kell dobni, ha egy zene hibásan lett lejátszva vagy hibás a beolvasott zene	110
<a href="#">MusicPlayer</a>	
A zene játészó osztály leírása	111
<a href="#">gtest_lite::ostreamRedir</a>	112
<a href="#">PostProcessor</a>	
A grafikus szépítő osztály leírása	113
<a href="#">Profession</a>	
A szakma osztály leírása	116
<a href="#">RandomGenerator</a>	
Egy korszerűbb és konfigurálhatóbb véletlen szám generátor osztály	119
<a href="#">ReadSaveFileFail</a>	
Akkor kell dobni, ha egy IO mentés, importálás vagy törlés hibás	121
<a href="#">sf::RectangleShape</a>	122
<a href="#">sf::RenderStates</a>	123
<a href="#">sf::RenderWindow</a>	124
<a href="#">minerals::ResourceStructure</a>	
Az erőforrás struktúra osztály leírása	127
<a href="#">SaveManager</a>	
A fájl menedzseléshez szolgáló osztály leírása	129
<a href="#">Shadowable</a>	
Az árnyékoláshoz szükséges interface	131
<a href="#">SimulationException</a>	
Akkor kell dobni, ha egy szimulációs elem hibásan viselkedik	135
<a href="#">creature::Soldier</a>	
A katona szakmájú ember osztály leírása	136
<a href="#">sf::Sound</a>	138
<a href="#">sf::SoundBuffer</a>	139
<a href="#">SoundPlayer</a>	
A hanglejátszó osztály leírása	139
<a href="#">sf::SoundSource</a>	141
<a href="#">sf::Sprite</a>	142

<a href="#">minerals::Stone</a>	
A kő osztály leírása. követ ad, amikor kitermelik . . . . .	145
<a href="#">creature::Stonemason</a>	
A bányász szakmájú ember osztály leírása . . . . .	147
<a href="#">minerals::Structure</a>	
A struktúra osztály leírása . . . . .	150
<a href="#">StructureException</a>	
Akkor kell dobni, ha egy struktúra hibásan működött . . . . .	154
<a href="#">TerrainContainer&lt; T &gt;</a>	
A világ terepét tároló osztály . . . . .	155
<a href="#">gtest_lite::Test</a> . . . . .	162
<a href="#">sf::Texture</a> . . . . .	166
<a href="#">Textureable</a>	
Egy interface, ami a textúrázáshoz kell . . . . .	167
<a href="#">TextureManager</a>	
A Textúra kezelő osztály . . . . .	169
<a href="#">tiles::Tile</a>	
A terepkocka osztály leírása . . . . .	170
<a href="#">sf::Transform</a> . . . . .	172
<a href="#">minerals::Tree</a>	
A fa osztály leírása. Fát ad, ha kitermelik . . . . .	174
<a href="#">sf::Vector2f</a> . . . . .	176
<a href="#">sf::Vector2i</a> . . . . .	177
<a href="#">sf::VideoMode</a> . . . . .	178
<a href="#">creature::Woodcutter</a>	
A favágó szakmájú ember osztály leírása . . . . .	180
<a href="#">World</a>	
A világ osztály leírása . . . . .	181



## 4. fejezet

# Fájlmutató

### 4.1. Fájllista

Az összes fájl listája rövid leírásokkal:

src/ <a href="#">GameConfig.cpp</a>	221
src/ <a href="#">GameConfig.d</a>	221
src/ <a href="#">GameConfig.hpp</a>	
A Szimuláció konfigurációja itt érhető el	221
src/ <a href="#">GameManager.cpp</a>	222
src/ <a href="#">GameManager.d</a>	222
src/ <a href="#">GameManager.hpp</a>	
A játékmenedzser osztály itt van deklarálva	222
src/ <a href="#">main.cpp</a>	222
src/ <a href="#">main.d</a>	223
src/ <a href="#">MusicPlayer.cpp</a>	223
src/ <a href="#">MusicPlayer.d</a>	223
src/ <a href="#">MusicPlayer.hpp</a>	
A zene lejátszó osztály itt van deklarálva	223
src/ <a href="#">PostProcessor.cpp</a>	223
src/ <a href="#">PostProcessor.d</a>	223
src/ <a href="#">PostProcessor.hpp</a>	
A grafikus szépítő osztály deklarációját tartalmazza	223
src/ <a href="#">Profession.cpp</a>	224
src/ <a href="#">Profession.d</a>	224
src/ <a href="#">Profession.hpp</a>	
Ebben a fájlban van deklarálva a szakma indikátor osztály	224
src/ <a href="#">Random_Gen.cpp</a>	224
src/ <a href="#">Random_Gen.d</a>	225
src/ <a href="#">Random_Gen.hpp</a>	
A véletlen generátor osztályt tároló fájl	225
src/ <a href="#">SaveHelpers.cpp</a>	225
src/ <a href="#">SaveHelpers.d</a>	226
src/ <a href="#">SaveHelpers.hpp</a>	
A mentést segítő factory-k és segédfüggvények	226
src/ <a href="#">SaveManager.cpp</a>	228
src/ <a href="#">SaveManager.d</a>	228
src/ <a href="#">SaveManager.hpp</a>	
A fájl menedzseléshez szolgáló osztály itt van deklarálva	228
src/ <a href="#">Shadowable.cpp</a>	229

src/Shadowable.d	229
src/Shadowable.hpp	
Az árnyék szimulálásához való osztályt tartalmazza	229
src/SoundPlayer.cpp	229
src/SoundPlayer.d	230
src/SoundPlayer.hpp	
Ebben a fájlba vannak a hanglejátszó osztályhoz szükséges deklarációk	230
src/TerrainContainer.hpp	
A Világ terepének a deklarálása ebben a fájlba van	231
src/TerrainContainer.inl	231
src/Textureable.hpp	
Ebbe a fájlba van a textúrázáshoz szükséges osztály	232
src/TextureManager.cpp	232
src/TextureManager.d	232
src/TextureManager.hpp	
Ebbe a fájlba van az az osztály, ami a textúrák betöltéséért, kiosztásáért és tárolásáért felelős	232
src/Utils.cpp	233
src/Utils.d	234
src/Utils.hpp	
Ebben a fájlba vannak a segéd függvények	234
src/World.cpp	235
src/World.d	236
src/World.hpp	
A Világ osztály, ami a fő szimulációs elemek tárolásáért felelős	236
src/creatures/Goat.cpp	193
src/creatures/Goat.d	193
src/creatures/Goat.hpp	
A Kecske osztály itt van deklarálva	193
src/creatures/HostileInterface.cpp	194
src/creatures/HostileInterface.d	194
src/creatures/HostileInterface.hpp	
A vadállat interface itt van deklarálva	194
src/creatures/Living.cpp	207
src/creatures/Living.d	207
src/creatures/Living.hpp	
Az élő interface itt van deklarálva	207
src/creatures/hostiles/Bear.cpp	195
src/creatures/hostiles/Bear.d	195
src/creatures/hostiles/Bear.hpp	
A Medve osztály itt van deklarálva	195
src/creatures/hostiles/Crocodile.cpp	196
src/creatures/hostiles/Crocodile.d	196
src/creatures/hostiles/Crocodile.hpp	
A krokodil osztály itt van deklarálva	196
src/creatures/hostiles/KillerRobot.cpp	197
src/creatures/hostiles/KillerRobot.d	197
src/creatures/hostiles/KillerRobot.hpp	
A Gyilkos Robot osztály itt van deklarálva	197
src/creatures/humans/AnglerMiner.cpp	198
src/creatures/humans/AnglerMiner.d	198
src/creatures/humans/AnglerMiner.hpp	
Az "AnglerMiner" szakmájú ember osztály itt van deklarálva	198
src/creatures/humans/Builder.cpp	199
src/creatures/humans/Builder.d	199
src/creatures/humans/Builder.hpp	
Az építész szakmájú ember osztály itt van deklarálva	199
src/creatures/humans/Farmer.cpp	200
src/creatures/humans/Farmer.d	200

src/creatures/humans/Farmer.hpp	
A farmer szakmájú ember osztály itt van deklarálva	200
src/creatures/humans/Fisherman.cpp	201
src/creatures/humans/Fisherman.d	201
src/creatures/humans/Fisherman.hpp	
A halász szakmájú ember osztály itt van deklarálva	201
src/creatures/humans/Human.cpp	202
src/creatures/humans/Human.d	202
src/creatures/humans/Human.hpp	
Az alap ember osztály itt van deklarálva	202
src/creatures/humans/King.cpp	203
src/creatures/humans/King.d	203
src/creatures/humans/King.hpp	
A király szakmájú ember osztály itt van deklarálva	203
src/creatures/humans/Soldier.cpp	204
src/creatures/humans/Soldier.d	204
src/creatures/humans/Soldier.hpp	
A katona szakmájú ember osztály itt van deklarálva	204
src/creatures/humans/Stonemason.cpp	205
src/creatures/humans/Stonemason.d	205
src/creatures/humans/Stonemason.hpp	
A bányász szakmájú ember osztály itt van deklarálva	205
src/creatures/humans/Woodcutter.cpp	206
src/creatures/humans/Woodcutter.d	206
src/creatures/humans/Woodcutter.hpp	
A favágó szakmájú ember osztály itt van deklarálva	206
src/exceptions/FileExceptions.hpp	
Ebben a fájlban vannak deklarálva azok a hibák, amik az IO-hoz kapcsolódnak	208
src/exceptions/MusicLoadException.hpp	
Ebben a fájlban vannak deklarálva azok a hibák, amik az Zenéhez kapcsolódnak	209
src/exceptions/SimulationException.hpp	
Ebben a fájlban vannak deklarálva azok a hibák, amik az alap Szimulációhoz kapcsolódnak.	
Ezekből a hibákból öröklődik minden olyan hiba, amit a program kezel és dob	209
src/exceptions/WorldExceptions.hpp	
Ebben a fájlban vannak deklarálva azok a hibák, amik az Világhoz kapcsolódnak	210
src/external/gtest_lite.h	210
src/external/memtrace.cpp	220
src/external/memtrace.d	220
src/external/memtrace.h	220
src/fake_sfml/fake_sfml.cpp	220
src/fake_sfml/fake_sfml.d	220
src/fake_sfml/fake_sfml.hpp	220
src/terrain_tiles/Tile.cpp	230
src/terrain_tiles/Tile.d	230
src/terrain_tiles/Tile.hpp	
A Terepkocka osztály itt van deklarálva	230
src/ui/button.cpp	233
src/ui/button.d	233
src/ui/button.hpp	
A gomb osztály itt van deklarálva	233
src/world_object/BerryBush.cpp	237
src/world_object/BerryBush.d	237
src/world_object/BerryBush.hpp	
A bokor osztály itt van deklarálva	237
src/world_object/CityCenter.cpp	237
src/world_object/CityCenter.d	237
src/world_object/CityCenter.hpp	
A városközpont osztály itt van deklarálva	237

src/world_object/ <a href="#">House.cpp</a>	238
src/world_object/ <a href="#">House.d</a>	238
src/world_object/ <a href="#">House.hpp</a>	
A Ház osztály itt van deklarálva	238
src/world_object/ <a href="#">Iron.cpp</a>	239
src/world_object/ <a href="#">Iron.d</a>	239
src/world_object/ <a href="#">Iron.hpp</a>	
A Vasérc osztály itt van deklarálva	239
src/world_object/ <a href="#">ResourceStructure.cpp</a>	239
src/world_object/ <a href="#">ResourceStructure.d</a>	240
src/world_object/ <a href="#">ResourceStructure.hpp</a>	
A kibányászható osztály itt van deklarálva	240
src/world_object/ <a href="#">Stone.cpp</a>	240
src/world_object/ <a href="#">Stone.d</a>	240
src/world_object/ <a href="#">Stone.hpp</a>	
A Kő osztály itt van deklarálva	240
src/world_object/ <a href="#">Structure.cpp</a>	241
src/world_object/ <a href="#">Structure.d</a>	241
src/world_object/ <a href="#">Structure.hpp</a>	
A struktúra osztály itt van deklarálva	241
src/world_object/ <a href="#">Tree.cpp</a>	242
src/world_object/ <a href="#">Tree.d</a>	242
src/world_object/ <a href="#">Tree.hpp</a>	
A fa osztály itt van deklarálva	242



## 5. fejezet

# Névterek dokumentációja

### 5.1. creature névtér-referencia

Az összes élőlény és entitás ebben a névtérben van.

#### Osztályok

- class [Goat](#)  
*A kecske osztály leírása.*
- class [HostileInterface](#)  
*A vadállat entitások interface leírása.*
- class [Bear](#)  
*A medve osztály leírása.*
- class [Crocodile](#)  
*A krokodil osztály leírása.*
- class [KillerRobot](#)  
*A gyilkos robot osztály leírása.*
- class [AnglerMiner](#)  
*Az "AnglerMiner" szakmájú ember osztály leírása.*
- class [Builder](#)  
*Az építész szakmájú ember osztály leírása.*
- class [Farmer](#)  
*A farmer szakmájú ember osztály leírása.*
- class [Fisherman](#)  
*A halász szakmájú ember osztály leírása.*
- class [Human](#)  
*Az alap ember osztály leírása. Minden fajta szakmájú ember innen öröklődik.*
- class [King](#)  
*A király szakmájú ember osztály leírása.*
- class [Soldier](#)  
*A katona szakmájú ember osztály leírása.*
- class [Stonemason](#)  
*A bányász szakmájú ember osztály leírása.*
- class [Woodcutter](#)  
*A favágó szakmájú ember osztály leírása.*
- class [Living](#)  
*Az élő entitások interface leírása.*

## Enumerációk

- enum class `ENTITY_TYPE` : char { `HUMAN` , `ANIMAL` , `ROBOTIC` }
- enum class `ENTITY_GENDER` : char { `MALE` , `FEMALE` }
- enum class `FACING` : bool { `RIGHT` , `LEFT` }
- enum class `LIVINGSTATE` : int {  
    `IDLE` , `RUN` , `WALK` , `DEATH` ,  
    `ATTACKING` , `DOING_ITS_WORK` }

### 5.1.1. Részletes leírás

Az összes élőlény és entitás ebben a névtérben van.

### 5.1.2. Enumerációk dokumentációja

#### 5.1.2.1. ENTITY\_GENDER

```
enum creature::ENTITY_GENDER : char [strong]
```

Enumeráció-értékek

MALE	
FEMALE	

#### 5.1.2.2. ENTITY\_TYPE

```
enum creature::ENTITY_TYPE : char [strong]
```

Enumeráció-értékek

HUMAN	
ANIMAL	
ROBOTIC	

#### 5.1.2.3. FACING

```
enum creature::FACING : bool [strong]
```

## Enumeráció-értékek

RIGHT	
LEFT	

## 5.1.2.4. LIVINGSTATE

```
enum creature::LIVINGSTATE : int [strong]
```

## Enumeráció-értékek

IDLE	
RUN	
WALK	
DEATH	
ATTACKING	
DOING_ITS_WORK	

## 5.2. gtest\_lite névtér-referencia

[gtest\\_lite](#): a keretrendszer függvényinek és objektumainak névtére

## Osztályok

- struct [Test](#)
- class [ostreamRedir](#)

## Függvények

- `template<typename T1, typename T2>`  
`std::ostream & EXPECT_ (T1 exp, T2 act, bool(*pred)(T1, T1), const char *file, int line, const char *expr, const char *lhs="elvart", const char *rhs="aktual")`  
*általános sablon a várt értékhez.*
- `template<typename T1, typename T2>`  
`std::ostream & EXPECT_ (T1 *exp, T2 *act, bool(*pred)(T1 *, T1 *), const char *file, int line, const char *expr, const char *lhs="elvart", const char *rhs="aktual")`  
*pointerre specializált sablon a várt értékhez.*
- `std::ostream & EXPECTSTR (const char *exp, const char *act, bool(*pred)(const char *, const char *), const char *file, int line, const char *expr, const char *lhs="elvart", const char *rhs="aktual")`
- `template<typename T>`  
`bool eq (T a, T b)`
- `bool eqstr (const char *a, const char *b)`
- `bool eqstrcase (const char *a, const char *b)`
- `template<typename T>`  
`bool ne (T a, T b)`

- `bool nestr (const char *a, const char *b)`
- `template<typename T >`  
`bool le (T a, T b)`
- `template<typename T >`  
`bool lt (T a, T b)`
- `template<typename T >`  
`bool ge (T a, T b)`
- `template<typename T >`  
`bool gt (T a, T b)`
- `template<typename T >`  
`bool almostEQ (T a, T b)`

### 5.2.1. Részletes leírás

`gtest_lite`: a keretrendszer függvényinek és objektumainak névtére

## 5.2.2. Függvények dokumentációja

### 5.2.2.1. almostEQ()

```
template<typename T >
bool gtest_lite::almostEQ (
    T a,
    T b )
```

Segédsablon valós számok összehasonlításához Nem bombabiztos, de nekünk most jó lesz Elméleti hátér:  
<http://www.cygnus-software.com/papers/comparingfloats/comparingfloats.htm>

### 5.2.2.2. eq()

```
template<typename T >
bool gtest_lite::eq (
    T a,
    T b )
```

segéd sablonok a relációkhoz. azért nem STL (algorithm), mert csak a függvény lehet, hogy menjen a deduckció

### 5.2.2.3. eqstr()

```
bool gtest_lite::eqstr (
    const char * a,
    const char * b ) [inline]
```

#### 5.2.2.4. eqstrcase()

```
bool gtest_lite::eqstrcase (
    const char * a,
    const char * b ) [inline]
```

#### 5.2.2.5. EXPECT\_() [1/2]

```
template<typename T1 , typename T2 >
std::ostream& gtest_lite::EXPECT_ (
    T1 * exp,
    T2 * act,
    bool(*) (T1 *, T1 *) pred,
    const char * file,
    int line,
    const char * expr,
    const char * lhs = "elvart",
    const char * rhs = "aktual" )
```

pointerre specializált sablon a várt értékhez.

#### 5.2.2.6. EXPECT\_() [2/2]

```
template<typename T1 , typename T2 >
std::ostream& gtest_lite::EXPECT_ (
    T1 exp,
    T2 act,
    bool(*) (T1, T1) pred,
    const char * file,
    int line,
    const char * expr,
    const char * lhs = "elvart",
    const char * rhs = "aktual" )
```

általános sablon a várt értékhez.

#### 5.2.2.7. EXPECTSTR()

```
std::ostream& gtest_lite::EXPECTSTR (
    const char * exp,
    const char * act,
    bool(*) (const char *, const char *) pred,
    const char * file,
    int line,
    const char * expr,
    const char * lhs = "elvart",
    const char * rhs = "aktual" ) [inline]
```

stringek összehasonlításához. azért nem spec. mert a sima EQ-ra másként kell működnie.

**5.2.2.8. ge()**

```
template<typename T >
bool gtest_lite::ge (
    T a,
    T b )
```

**5.2.2.9. gt()**

```
template<typename T >
bool gtest_lite::gt (
    T a,
    T b )
```

**5.2.2.10. le()**

```
template<typename T >
bool gtest_lite::le (
    T a,
    T b )
```

**5.2.2.11. lt()**

```
template<typename T >
bool gtest_lite::lt (
    T a,
    T b )
```

**5.2.2.12. ne()**

```
template<typename T >
bool gtest_lite::ne (
    T a,
    T b )
```

**5.2.2.13. nestr()**

```
bool gtest_lite::nestr (
    const char * a,
    const char * b ) [inline]
```

## 5.3. minerals névtér-referencia

Az összes struktúra ebben a névtérben van.

### Osztályok

- class [BerryBush](#)  
*A bokor osztály leírása. Ételt ad, ha kitermelik.*
- class [CityCenter](#)  
*A városközpont osztály leírása. E köré épülnek a házak.*
- class [House](#)  
*A ház osztály leírása. Szinttől függően idéz embereket.*
- class [Iron](#)  
*A vasérc osztály leírása. Vasat ad, amikor kitermelik.*
- class [ResourceStructure](#)  
*Az erőforrás struktúra osztály leírása.*
- class [Stone](#)  
*A kő osztály leírása. követ ad, amikor kitermelik.*
- class [Structure](#)  
*A struktúra osztály leírása.*
- class [Tree](#)  
*A fa osztály leírása. Fát ad, ha kitermelik.*

### Enumerációk

- enum class [MINERAL\\_TYPE](#) : char {  
    [STONE](#) , [WOOD](#) , [IRON](#) , [FOOD](#) ,  
    [HOUSING](#) , [CITY\\_CENTER](#) }

### Függvények

- std::string [mineral\\_to\\_string](#) ([MINERAL\\_TYPE](#) type)  
*Mentést elősegítő függvények.*

#### 5.3.1. Részletes leírás

Az összes struktúra ebben a névtérben van.

#### 5.3.2. Enumerációk dokumentációja

##### 5.3.2.1. MINERAL\_TYPE

```
enum minerals::MINERAL\_TYPE : char [strong]
```

## Enumeráció-értékek

STONE	
WOOD	
IRON	
FOOD	
HOUSING	
CITY_CENTER	

### 5.3.3. Függvények dokumentációja

#### 5.3.3.1. mineral\_to\_string()

```
std::string minerals::mineral_to_string (
    MINERAL_TYPE type )
```

Mentést elősegítő függvények.

## 5.4. sf névtér-referencia

### Osztályok

- class [Vector2f](#)
- class [Transform](#)
- class [FloatRect](#)
- class [Vector2i](#)
- class [Texture](#)
- class [Bound](#)
- class [Color](#)
- class [IntRect](#)
- class [Sprite](#)
- class [Event](#)
- class [ClockTime](#)
- class [Clock](#)
- class [SoundBuffer](#)
- class [Sound](#)
- class [SoundSource](#)
- class [Music](#)
- class [RectangleShape](#)
- class [Keyboard](#)
- class [RenderStates](#)
- class [VideoMode](#)
- class [RenderWindow](#)
- class [Mouse](#)



## Enumerációk

- enum class `BlendMode` {  
    `None` , `Alpha` , `Additive` , `Multiply` ,  
    `BlendAdd` }

## Függvények

- bool `file_exists_at_path` (const std::string &name)
- std::ostream & `operator<<` (std::ostream &os, const `Color` &color)

## Változók

- bool `hungarian` = false
- constexpr `BlendMode BlendAdd` = `BlendMode::BlendAdd`

### 5.4.1. Enumerációk dokumentációja

#### 5.4.1.1. BlendMode

```
enum sf::BlendMode [strong]
```

##### Enumeráció-értékek

None	
Alpha	
Additive	
Multiply	
BlendAdd	

### 5.4.2. Függvények dokumentációja

#### 5.4.2.1. file\_exists\_at\_path()

```
bool sf::file_exists_at_path (  
    const std::string & name ) [inline]
```

#### 5.4.2.2. operator<<()

```
std::ostream& sf::operator<< (
    std::ostream & os,
    const Color & color )
```

### 5.4.3. Változók dokumentációja

#### 5.4.3.1. BlendAdd

```
constexpr BlendMode sf::BlendAdd = BlendMode::BlendAdd [inline], [constexpr]
```

#### 5.4.3.2. hungarian

```
bool sf::hungarian = false
```

## 5.5. tiles névtér-referencia

Az összes terepkocka elem ebben a névtérben van.

### Osztályok

- class [Tile](#)  
*A terepkocka osztály leírása.*

### Enumerációk

- enum class [TILETYPE](#) : char { [GRASS](#) , [WATER](#) , [MOUNTAIN](#) }

#### 5.5.1. Részletes leírás

Az összes terepkocka elem ebben a névtérben van.

#### 5.5.2. Enumerációk dokumentációja

##### 5.5.2.1. TILETYPE

```
enum tiles::TILETYPE : char [strong]
```

## Enumeráció-értékek

GRASS	
WATER	
MOUNTAIN	

## 5.6. ui névtér-referencia

Az összes UI elem ebben a névtérben van.

### Osztályok

- class [Button](#)

*A gomb osztály leírása. Tárolja a gomb méretét és azt, hogy mit csinál, ha rákattintanak.*

#### 5.6.1. Részletes leírás

Az összes UI elem ebben a névtérben van.



## 6. fejezet

# Osztályok dokumentációja

### 6.1. `_Is_Types< F, T >` struktúrasablon-referencia

Segédsablon típuskonverzió futás közbeni ellenőrzésére.

```
#include <gtest_lite.h>
```

#### Statikus publikus tagfüggvények

- `template<typename D >`  
`static char(& f (D))[1]`
- `template<typename D >`  
`static char(& f (...))[2]`

#### Statikus publikus attribútumok

- `static bool const convertible = sizeof(f<T>(F())) == 1`

#### 6.1.1. Részletes leírás

```
template<typename F, typename T>  
struct _Is_Types< F, T >
```

Segédsablon típuskonverzió futás közbeni ellenőrzésére.

#### 6.1.2. Tagfüggvények dokumentációja

### 6.1.2.1. f() [1/2]

```
template<typename F , typename T >
template<typename D >
static char(& _Is_Types< F, T >::f (
    ... )) [2] [static]
```

### 6.1.2.2. f() [2/2]

```
template<typename F , typename T >
template<typename D >
static char(& _Is_Types< F, T >::f (
    D )) [1] [static]
```

## 6.1.3. Adattagok dokumentációja

### 6.1.3.1. convertible

```
template<typename F , typename T >
bool const _Is_Types< F, T >::convertible = sizeof(f<T>(F())) == 1 [static]
```

Ez a dokumentáció a struktúráról a következő fájl alapján készült:

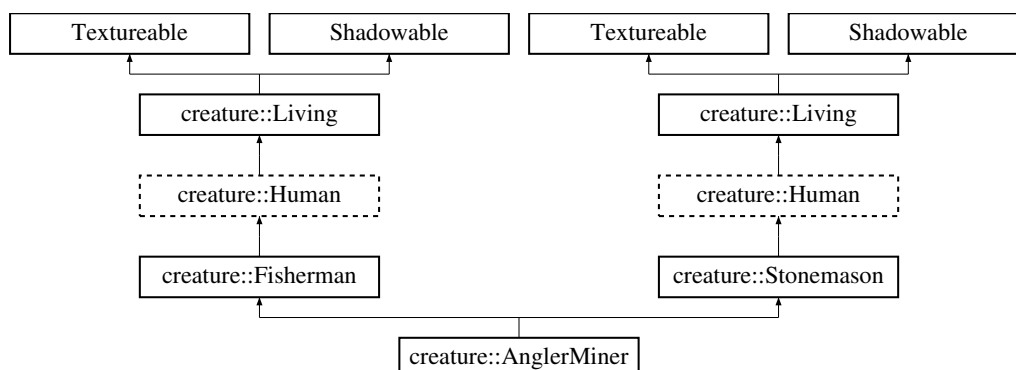
- [src/external/gtest\\_lite.h](#)

## 6.2. creature::AnglerMiner osztályreferencia

Az "AnglerMiner" szakmájú ember osztály leírása.

```
#include <AnglerMiner.hpp>
```

A creature::AnglerMiner osztály származási diagramja:



## Publikus tagfüggvények

- `AnglerMiner` (int x, int y, `ENTITY_GENDER` gender\_modifier)  
*Inicializál egy AnglerMiner-t egy pontos x és y koordinátára és beállítja az attribútumait.*
- void `update_logic` (`World` &world, float deltaTime) override  
*Az entitás frissítési logikája itt van definiálva. Például a halász vizet keres, ha odaért akkor halászik.*
- `~AnglerMiner` ()  
*Az AnglerMiner destruktora.*

## További örökölt tagok

### 6.2.1. Részletes leírás

Az "AnglerMiner" szakmájú ember osztály leírása.

Ez egy speciális szakma, ami tud követ és vasat bányászni és ha akar, még halászni is tud.

### 6.2.2. Konstruktorkok és destruktorkok dokumentációja

#### 6.2.2.1. AnglerMiner()

```
creature::AnglerMiner::AnglerMiner (
    int x,
    int y,
    ENTITY_GENDER gender_modifier )
```

Inicializál egy AnglerMiner-t egy pontos x és y koordinátára és beállítja az attribútumait.

#### Paraméterek

<code>x</code>	Az x koordináta.
<code>y</code>	Az y koordináta.
<code>gender_modifier</code>	Az "AnglerMiner" neve.

#### 6.2.2.2. ~AnglerMiner()

```
creature::AnglerMiner::~~AnglerMiner ( )
```

Az `AnglerMiner` destruktora.

### 6.2.3. Tagfüggvények dokumentációja

### 6.2.3.1. update\_logic()

```
void creature::AnglerMiner::update_logic (
    World & world,
    float deltaTime ) [override], [virtual]
```

Az entitás frissítési logikája itt van definiálva. Például a halász vizet keres, ha odaért akkor halászik.

#### Paraméterek

<i>world</i>	A világ, amibe frissíti magát az entitás.
<i>deltaTime</i>	Az előző frissítés óta eltelt idő.

Megvalósítja a következőket: [creature::Living](#).

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

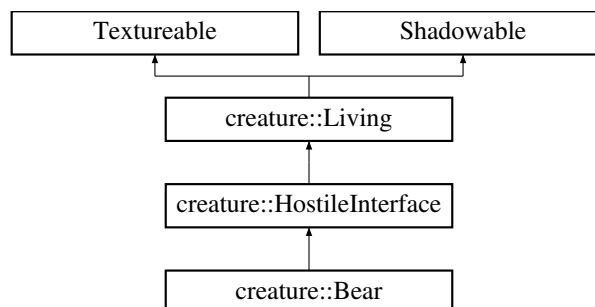
- [src/creatures/humans/AnglerMiner.hpp](#)
- [src/creatures/humans/AnglerMiner.cpp](#)

## 6.3. creature::Bear osztályreferencia

A medve osztály leírása.

```
#include <Bear.hpp>
```

A creature::Bear osztály származási diagramja:



### Publikus tagfüggvények

- [Bear](#) (int x, int y)  
*Idéz egy medvét egy pontos x és y koordinátára és beállítja az attribútumait.*
- [ENTITY\\_TYPE get\\_type](#) () const override  
*Visszaadja az entitás belső szimbólumát. Ez abba segít, hogy a vadállatok például csak embert támadjanak, Az emberek csak vadállatot.*
- void [die](#) () override  
*Mi történjen, ha meghal az entitás.*
- void [update\\_logic](#) (World &world, float deltaTime) override  
*Az entitás frissítési logikája itt van definiálva. Például a halász vizet keres, ha odaért akkor halászik.*
- void [draw\\_logic](#) (sf::RenderWindow &window, float deltaTime, int ofx, int ofy) override  
*Az entitás kirajzolási logikája, például az ember az árnyékát és képét rajzolja ki, de a robot csak a képét.*
- void [select\\_target](#) (World &world) override  
*A logikát írja le, ahogy az entitás a világba kiválasztja magának a célpontot.*
- [~Bear](#) ()  
*A medve destruktora.*



## További örökölt tagok

### 6.3.1. Részletes leírás

A medve osztály leírása.

A medve egy agresszív állat, ami más medvéken kívül mindent támad. Gyorsan fut.

### 6.3.2. Konstruktorok és destruktorok dokumentációja

#### 6.3.2.1. Bear()

```
creature::Bear::Bear (
    int x,
    int y )
```

Idéz egy medvét egy pontos x és y koordinátára és beállítja az attribútumait.

##### Paraméterek

x	Az x koordináta.
y	Az y koordináta.

#### 6.3.2.2. ~Bear()

```
creature::Bear::~~Bear ( )
```

A medve destruktora.

### 6.3.3. Tagfüggvények dokumentációja

#### 6.3.3.1. die()

```
void creature::Bear::die ( ) [override], [virtual]
```

Mi történjen, ha meghal az entitás.

Megvalósítja a következőket: [creature::Living](#).

### 6.3.3.2. draw\_logic()

```
void creature::Bear::draw_logic (
    sf::RenderWindow & window,
    float deltaTime,
    int offx,
    int offy ) [override], [virtual]
```

Az entitás kirajzolás logikája, például az ember az árnyékát és képét rajzolja ki, de a robot csak a képét.

#### Paraméterek

<i>window</i>	Az ablak, ahova rajzolni kell.
<i>deltaTime</i>	Az előző frissítés óta eltelt idő.
<i>offx</i>	A kamera X eltolása.
<i>offy</i>	A kamera Y eltolása.

Megvalósítja a következőket: [creature::Living](#).

### 6.3.3.3. get\_type()

```
ENTITY_TYPE creature::Bear::get_type ( ) const [override], [virtual]
```

Visszaadja az entitás belső szimbólumát. Ez abba segít, hogy a vadállatok például csak embert támadjanak, Az emberek csak vadállatot.

#### Visszatérési érték

A belső szimbólum.

Megvalósítja a következőket: [creature::Living](#).

### 6.3.3.4. select\_target()

```
void creature::Bear::select_target (
    World & world ) [override], [virtual]
```

A logikát írja le, ahogy az entitás a világba kiválasztja magának a célpontot.

#### Paraméterek

<i>world</i>	A világ, amibe a célpontot kell választani.
--------------	---

Megvalósítja a következőket: [creature::HostileInterface](#).

### 6.3.3.5. update\_logic()

```
void creature::Bear::update_logic (
    World & world,
    float deltaTime ) [override], [virtual]
```

Az entitás frissítési logikája itt van definiálva. Például a halász vizet keres, ha odaért akkor halászik.

#### Paraméterek

<i>world</i>	A világ, amibe frissíti magát az entitás.
<i>deltaTime</i>	Az előző frissítés óta eltelt idő.

Megvalósítja a következőket: [creature::Living](#).

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

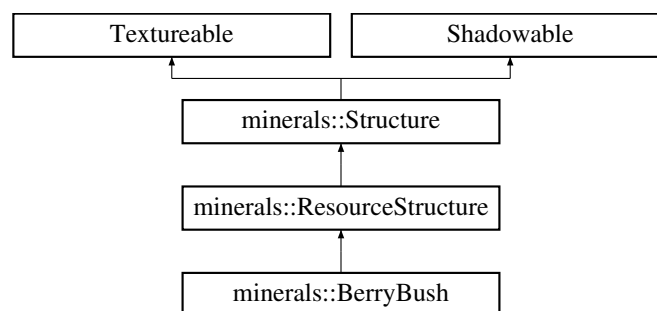
- [src/creatures/hostiles/Bear.hpp](#)
- [src/creatures/hostiles/Bear.cpp](#)

## 6.4. minerals::BerryBush osztályreferencia

A bokr osztály leírása. Ételt ad, ha kitermelik.

```
#include <BerryBush.hpp>
```

A minerals::BerryBush osztály származási diagramja:



### Publikus tagfüggvények

- [BerryBush](#) (int x, int y)  
*Konstruktor ami lerakja az erőforrást egy (x,y) pontra.*
- [MINERAL\\_TYPE get\\_type](#) () const override  
*Szimbólum, ami a fájlba mentéshez kell.*
- void [update\\_logic](#) (float deltaTime) override  
*Frissíti magát az idő függvényében.*
- bool [harvest](#) () override  
*Tisztán virtuális metódus. Ez leírja mi történik, hogy ha kitermelik ezt az erőforrást.*

## További örökölt tagok

### 6.4.1. Részletes leírás

A bokr osztály leírása. Ételt ad, ha kitermelik.

### 6.4.2. Konstruktorkok és destruktorkok dokumentációja

#### 6.4.2.1. BerryBush()

```
minerals::BerryBush::BerryBush (
    int x,
    int y )
```

Konstruktork ami lerakja az erőforrást egy (x,y) pontra.

### 6.4.3. Tagfüggvények dokumentációja

#### 6.4.3.1. get\_type()

```
MINERAL_TYPE minerals::BerryBush::get_type ( ) const [override], [virtual]
```

Szimbólum, ami a fájlba mentéshez kell.

Megvalósítja a következőket: [minerals::Structure](#).

#### 6.4.3.2. harvest()

```
bool minerals::BerryBush::harvest ( ) [override], [virtual]
```

Tisztán virtuális metódus. Ez leírja mi történik, hogy ha kitermelik ezt az erőforrást.

Megvalósítja a következőket: [minerals::ResourceStructure](#).

#### 6.4.3.3. update\_logic()

```
void minerals::BerryBush::update_logic (
    float deltaTime ) [override], [virtual]
```

Frissíti magát az idő függvényében.

## Paraméterek

<i>deltaTime</i>	Az előző frissítés óta eltelt idő.
------------------	------------------------------------

Megvalósítja a következőket: [minerals::Structure](#).

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- [src/world\\_object/BerryBush.hpp](#)
- [src/world\\_object/BerryBush.cpp](#)

## 6.5. sf::Bound osztályreferencia

```
#include <fake_sfml.hpp>
```

### Publikus attribútumok

- `int width`
- `int height`

### 6.5.1. Adattagok dokumentációja

#### 6.5.1.1. height

```
int sf::Bound::height
```

#### 6.5.1.2. width

```
int sf::Bound::width
```

Ez a dokumentáció az osztályról a következő fájl alapján készült:

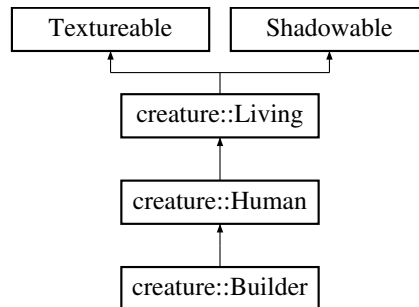
- [src/fake\\_sfml/fake\\_sfml.hpp](#)

## 6.6. creature::Builder osztályreferencia

Az építész szakmájú ember osztály leírása.

```
#include <Builder.hpp>
```

A creature::Builder osztály származási diagramja:



### Publikus tagfüggvények

- **Builder** (int x, int y, ENTITY\_GENDER gender\_modifier)  
*Inicializál egy építészt egy pontos x és y koordinátára és beállítja az attribútumait.*
- void **update\_logic** (World &world, float deltaTime) override  
*Az entitás frissítési logikája itt van definiálva. Például a halász vizet keres, ha odaért akkor halászik.*
- **~Builder** ()  
*Az építész destruktora.*

### További örökölt tagok

#### 6.6.1. Részletes leírás

Az építész szakmájú ember osztály leírása.

Ez a szakmájú ember épületeket fejleszt magasabb szintekre. Ha nincs épület akkor épít még.

#### 6.6.2. Konstruktorok és destruktorok dokumentációja

##### 6.6.2.1. Builder()

```
creature::Builder::Builder (
    int x,
    int y,
    ENTITY_GENDER gender_modifier )
```

Inicializál egy építészt egy pontos x és y koordinátára és beállítja az attribútumait.

## Paraméterek

<i>x</i>	Az x koordináta.
<i>y</i>	Az y koordináta.
<i>gender_modifier</i>	Az építész neme.

## 6.6.2.2. ~Builder()

```
creature::Builder::~~Builder ( )
```

Az építész destruktora.

## 6.6.3. Tagfüggvények dokumentációja

## 6.6.3.1. update\_logic()

```
void creature::Builder::update_logic (
    World & world,
    float deltaTime ) [override], [virtual]
```

Az entitás frissítési logikája itt van definiálva. Például a halász vizet keres, ha odaért akkor halászik.

## Paraméterek

<i>world</i>	A világ, amibe frissíti magát az entitás.
<i>deltaTime</i>	Az előző frissítés óta eltelt idő.

Megvalósítja a következőket: [creature::Living](#).

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

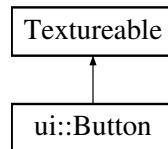
- [src/creatures/humans/Builder.hpp](#)
- [src/creatures/humans/Builder.cpp](#)

## 6.7. ui::Button osztályreferencia

A gomb osztály leírása. Tárolja a gomb méretét és azt, hogy mit csinál, ha rákattintanak.

```
#include <button.hpp>
```

Az ui::Button osztály származási diagramja:



## Publikus tagfüggvények

- **Button** (int px, int py, int width, int height, const std::string &spritepath)  
*A konstruktor ami létrehozza a gombot megadott mérettel és képpel.*
- void **setCallback** (std::function< void()> func)  
*Beállítja, mi történjen, ha a gombra kattintanak.*
- void **try\_hover\_animation** (int mX, int mY)  
*Megnézi, hogy az egér kurzor rajta van-e a gombon.*
- void **onClick** (bool mc)  
*Megnézi, hogy kattintottak-e rá, ha igen akkor végrehajta a függvényt amit neki adtak be.*
- bool **setTexture** (const std::string &filename) override  
*Beállít egy fájl elérési útból egy textúrát. Megvalósítástól függően esetleg 2-t.*
- void **setPosition** (double x, double y) override  
*Beállítja, hogy hova kell rajzolni a textúrázható leszármazottat.*
- void **draw** (sf::RenderWindow &window) override  
*Kirajzolja az objektumot.*

### 6.7.1. Részletes leírás

A gomb osztály leírása. Tárolja a gomb méretét és azt, hogy mit csinál, ha rákattintanak.

### 6.7.2. Konstruktorok és destruktorok dokumentációja

#### 6.7.2.1. Button()

```

ui::Button::Button (
    int px,
    int py,
    int width,
    int height,
    const std::string & spritepath )
  
```

A konstruktor ami létrehozza a gombot megadott mérettel és képpel.

#### Paraméterek

<i>px</i>	A gomb X koordinátája.
<i>py</i>	A gomb Y koordinátája.
<i>width</i>	A gomb szélessége.
<i>height</i>	A gomb magassága.
<i>spritepath</i>	A gomb képének elérési útvonala.



### 6.7.3. Tagfüggvények dokumentációja

#### 6.7.3.1. draw()

```
void ui::Button::draw (
    sf::RenderWindow & window ) [override], [virtual]
```

Kirajzolja az objektumot.

Paraméterek

<i>window</i>	Ahova ki kell rajzolni a textúrázható leszármazottat.
---------------	---

Megvalósítja a következőket: [Textureable](#).

#### 6.7.3.2. onClick()

```
void ui::Button::onClick (
    bool mc )
```

Megnézi, hogy kattintottak-e rá, ha igen akkor végrehajta a függvényt amit neki adtak be.

Paraméterek

<i>mc</i>	Le van-e nyomva az egér gomb.
-----------	-------------------------------

#### 6.7.3.3. setCallback()

```
void ui::Button::setCallback (
    std::function< void()> func )
```

Beállítja, mi történjen, ha a gombra kattintanak.

Paraméterek

<i>func</i>	A függvény, ami le fog futni.
-------------	-------------------------------

#### 6.7.3.4. setPosition()

```
void ui::Button::setPosition (
    double x,
    double y ) [override], [virtual]
```

Beállítja, hogy hova kell rajzolni a textúrázható leszármozottat.

##### Paraméterek

<i>x</i>	Az x koordináta.
<i>y</i>	Az y koordináta.

Megvalósítja a következőket: [Textureable](#).

#### 6.7.3.5. setTexture()

```
bool ui::Button::setTexture (
    const std::string & filename ) [override], [virtual]
```

Beállít egy fájl elérési útból egy textúrát. Megvalósítástól függően esetleg 2-t.

##### Paraméterek

<i>filename</i>	A textúra elérési útja.
-----------------	-------------------------

##### Visszatérési érték

Sikeres volt-e a textúra beállítása.

Megvalósítja a következőket: [Textureable](#).

#### 6.7.3.6. try\_hover\_animation()

```
void ui::Button::try_hover_animation (
    int mX,
    int mY )
```

Megnézi, hogy az egér kurzor rajta van-e a gombon.

##### Paraméterek

<i>mX</i>	A kurzor X koordinátája.
<i>mY</i>	A kurzor Y koordinátája.

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

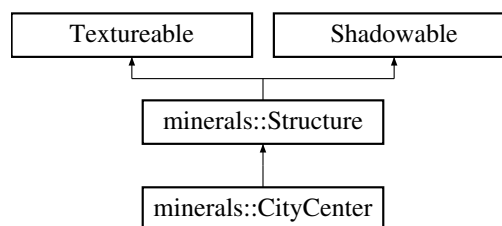
- [src/ui/button.hpp](#)
- [src/ui/button.cpp](#)

## 6.8. minerals::CityCenter osztályreferencia

A városközpont osztály leírása. E köré épülnek a házak.

```
#include <CityCenter.hpp>
```

A minerals::CityCenter osztály származási diagramja:



### Publikus tagfüggvények

- [CityCenter](#) (int x, int y)  
*Konstruktor ami lerakja a házat egy (x,y) pontra.*
- bool [is\\_there\\_room\\_for\\_housing](#) ()  
*Igazat ad vissza, ha lehet még házat építeni köré.*
- void [register\\_new\\_house](#) ()  
*Új házat vesz fel a városhoz.*
- [MINERAL\\_TYPE](#) [get\\_type](#) () const override  
*Szimbólum, ami a fájlba mentéshez kell.*
- void [update\\_logic](#) (float deltaTime) override  
*Frissíti magát az idő függvényében.*
- std::string [get\\_settlement\\_age](#) ()  
*String-ként adja vissza azt, hogy hány másodperces a város.*

### További örökölt tagok

#### 6.8.1. Részletes leírás

A városközpont osztály leírása. E köré épülnek a házak.

#### 6.8.2. Konstruktorok és destruktorok dokumentációja

#### 6.8.2.1. CityCenter()

```
minerals::CityCenter::CityCenter (
    int x,
    int y )
```

Konstruktor ami lerakja a házat egy (x,y) pontra.

### 6.8.3. Tagfüggvények dokumentációja

#### 6.8.3.1. get\_settlement\_age()

```
std::string minerals::CityCenter::get_settlement_age ( )
```

String-ként adja vissza azt, hogy hány másodperces a város.

#### 6.8.3.2. get\_type()

```
MINERAL\_TYPE minerals::CityCenter::get_type ( ) const [override], [virtual]
```

Szimbólum, ami a fájlba mentéshez kell.

Megvalósítja a következőket: [minerals::Structure](#).

#### 6.8.3.3. is\_there\_room\_for\_housing()

```
bool minerals::CityCenter::is_there_room_for_housing ( )
```

Igazat ad vissza, ha lehet még házat építeni köré.

#### 6.8.3.4. register\_new\_house()

```
void minerals::CityCenter::register_new_house ( )
```

Új házat vesz fel a városhoz.

#### 6.8.3.5. update\_logic()

```
void minerals::CityCenter::update_logic (
    float deltaTime ) [override], [virtual]
```

Frissíti magát az idő függvényében.

## Paraméterek

<i>deltaTime</i>	Az előző frissítés óta eltelt idő.
------------------	------------------------------------

Megvalósítja a következőket: [minerals::Structure](#).

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

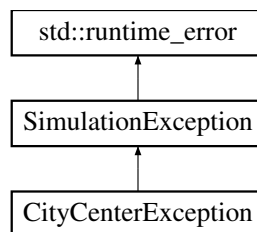
- [src/world\\_object/CityCenter.hpp](#)
- [src/world\\_object/CityCenter.cpp](#)

## 6.9. CityCenterException osztályreferencia

Akkor kell dobni, ha a városközpont hibásan működött.

```
#include <WorldExceptions.hpp>
```

A CityCenterException osztály származási diagramja:



### Publikus tagfüggvények

- [CityCenterException](#) (const std::string &msg)

#### 6.9.1. Részletes leírás

Akkor kell dobni, ha a városközpont hibásan működött.

#### 6.9.2. Konstruktorkok és destruktorkok dokumentációja

##### 6.9.2.1. CityCenterException()

```
CityCenterException::CityCenterException (
    const std::string & msg ) [inline], [explicit]
```

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- [src/exceptions/WorldExceptions.hpp](#)

## 6.10. sf::Clock osztályreferencia

```
#include <fake_sfml.hpp>
```

### Publikus tagfüggvények

- void [restart](#) ()
- [ClockTime](#) & [getElapsedTime](#) ()

### 6.10.1. Tagfüggvények dokumentációja

#### 6.10.1.1. getElapsedTime()

```
ClockTime & sf::Clock::getElapsedTime ( )
```

#### 6.10.1.2. restart()

```
void sf::Clock::restart ( )
```

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- [src/fake\\_sfml/fake\\_sfml.hpp](#)
- [src/fake\\_sfml/fake\\_sfml.cpp](#)

## 6.11. sf::ClockTime osztályreferencia

```
#include <fake_sfml.hpp>
```

### Publikus tagfüggvények

- [ClockTime](#) (std::size\_t atime)
- [ClockTime](#) ()
- float [asSeconds](#) ()
- void [reset](#) ()
- void [increment](#) ()

### 6.11.1. Konstruktorkok és destruktorkok dokumentációja

#### 6.11.1.1. ClockTime() [1/2]

```
sf::ClockTime::ClockTime (
    std::size_t atime )
```

#### 6.11.1.2. ClockTime() [2/2]

```
sf::ClockTime::ClockTime ( )
```

### 6.11.2. Tagfüggvények dokumentációja

#### 6.11.2.1. asSeconds()

```
float sf::ClockTime::asSeconds ( )
```

#### 6.11.2.2. increment()

```
void sf::ClockTime::increment ( )
```

#### 6.11.2.3. reset()

```
void sf::ClockTime::reset ( )
```

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- [src/fake\\_sfml/fake\\_sfml.hpp](#)
- [src/fake\\_sfml/fake\\_sfml.cpp](#)

## 6.12. sf::Color osztályreferencia

```
#include <fake_sfml.hpp>
```

### Publikus tagfüggvények

- [Color](#) ()
- [Color](#) (int \_r, int \_g, int \_b)
- [Color](#) (int \_r, int \_g, int \_b, int \_a)

## Publikus attribútumok

- int `r`
- int `g`
- int `b`
- int `a`

## Statikus publikus attribútumok

- static const `Color Black` = `Color(0, 0, 0)`
- static const `Color White` = `Color(255, 255, 255)`
- static const `Color Red` = `Color(255, 0, 0)`
- static const `Color Green` = `Color(0, 255, 0)`
- static const `Color Blue` = `Color(0, 0, 255)`
- static const `Color Transparent` = `Color(0, 0, 0, 0)`

## 6.12.1. Konstruktorkok és destruktorkok dokumentációja

### 6.12.1.1. `Color()` [1/3]

```
sf::Color::Color ( )
```

### 6.12.1.2. `Color()` [2/3]

```
sf::Color::Color (
    int _r,
    int _g,
    int _b )
```

### 6.12.1.3. `Color()` [3/3]

```
sf::Color::Color (
    int _r,
    int _g,
    int _b,
    int _a )
```

## 6.12.2. Adattagok dokumentációja



**6.12.2.1. a**

```
int sf::Color::a
```

**6.12.2.2. b**

```
int sf::Color::b
```

**6.12.2.3. Black**

```
const Color sf::Color::Black = Color(0, 0, 0) [static]
```

**6.12.2.4. Blue**

```
const Color sf::Color::Blue = Color(0, 0, 255) [static]
```

**6.12.2.5. g**

```
int sf::Color::g
```

**6.12.2.6. Green**

```
const Color sf::Color::Green = Color(0, 255, 0) [static]
```

**6.12.2.7. r**

```
int sf::Color::r
```

**6.12.2.8. Red**

```
const Color sf::Color::Red = Color(255, 0, 0) [static]
```

### 6.12.2.9. Transparent

```
const Color sf::Color::Transparent = Color(0, 0, 0, 0) [static]
```

### 6.12.2.10. White

```
const Color sf::Color::White = Color(255, 255, 255) [static]
```

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

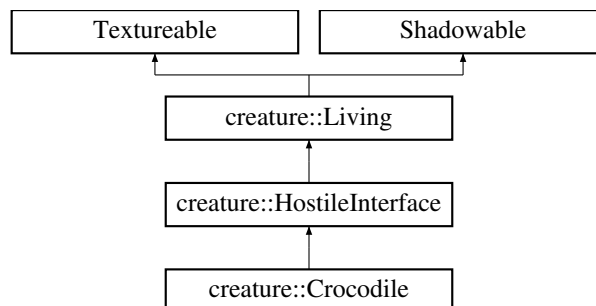
- [src/fake\\_sfml/fake\\_sfml.hpp](#)
- [src/fake\\_sfml/fake\\_sfml.cpp](#)

## 6.13. creature::Crocodile osztályreferencia

A krokodil osztály leírása.

```
#include <Crocodile.hpp>
```

A creature::Crocodile osztály származási diagramja:



### Publikus tagfüggvények

- `Crocodile (int x, int y)`  
*Idéz egy krokodilt egy pontos x és y koordinátára és beállítja az attribútumait.*
- `ENTITY_TYPE get_type ()` const override  
*Visszaadja az entitás belső szimbólumát. Ez abba segít, hogy a vadállatok például csak embert támadjanak, Az emberek csak vadállatot.*
- `void die ()` override  
*Mi történjen, ha meghal az entitás.*
- `void update_logic (World &world, float deltaTime)` override  
*Az entitás frissítési logikája itt van definiálva. Például a halász vizet keres, ha odaért akkor halászik.*
- `void draw_logic (sf::RenderWindow &window, float deltaTime, int ofx, int ofy)` override  
*Az entitás kirajzolás logikája, például az ember az árnyékát és képét rajzolja ki, de a robot csak a képét.*
- `void select_target (World &world)` override  
*A logikát írja le, ahogy az entitás a világba kiválasztja magának a célpontot.*
- `~Crocodile ()`  
*A krokodil destruktora.*

## További örökölt tagok

### 6.13.1. Részletes leírás

A krokodil osztály leírása.

A krokodil egy agresszív állat, ami mindent megeszik a robotokon kívül. Lassan mozog.

### 6.13.2. Konstruktorok és destruktorok dokumentációja

#### 6.13.2.1. Crocodile()

```
creature::Crocodile::Crocodile (
    int x,
    int y )
```

Idéz egy krokodilt egy pontos x és y koordinátára és beállítja az attribútumait.

##### Paraméterek

x	Az x koordináta.
y	Az y koordináta.

#### 6.13.2.2. ~Crocodile()

```
creature::Crocodile::~~Crocodile ( )
```

A krokodil destruktora.

### 6.13.3. Tagfüggvények dokumentációja

#### 6.13.3.1. die()

```
void creature::Crocodile::die ( ) [override], [virtual]
```

Mi történjen, ha meghal az entitás.

Megvalósítja a következőket: [creature::Living](#).

### 6.13.3.2. draw\_logic()

```
void creature::Crocodile::draw_logic (
    sf::RenderWindow & window,
    float deltaTime,
    int offx,
    int offy ) [override], [virtual]
```

Az entitás kirajzolás logikája, például az ember az árnyékát és képét rajzolja ki, de a robot csak a képét.

#### Paraméterek

<i>window</i>	Az ablak, ahova rajzolni kell.
<i>deltaTime</i>	Az előző frissítés óta eltelt idő.
<i>offx</i>	A kamera X eltolása.
<i>offy</i>	A kamera Y eltolása.

Megvalósítja a következőket: [creature::Living](#).

### 6.13.3.3. get\_type()

```
ENTITY_TYPE creature::Crocodile::get_type ( ) const [override], [virtual]
```

Visszaadja az entitás belső szimbólumát. Ez abba segít, hogy a vadállatok például csak embert támadjanak, Az emberek csak vadállatot.

#### Visszatérési érték

A belső szimbólum.

Megvalósítja a következőket: [creature::Living](#).

### 6.13.3.4. select\_target()

```
void creature::Crocodile::select_target (
    World & world ) [override], [virtual]
```

A logikát írja le, ahogy az entitás a világba kiválasztja magának a célpontot.

#### Paraméterek

<i>world</i>	A világ, amibe a célpontot kell választani.
--------------	---

Megvalósítja a következőket: [creature::HostileInterface](#).

### 6.13.3.5. update\_logic()

```
void creature::Crocodile::update_logic (
    World & world,
    float deltaTime ) [override], [virtual]
```

Az entitás frissítési logikája itt van definiálva. Például a halász vizet keres, ha odaért akkor halászik.

#### Paraméterek

<i>world</i>	A világ, amibe frissíti magát az entitás.
<i>deltaTime</i>	Az előző frissítés óta eltelt idő.

Megvalósítja a következőket: [creature::Living](#).

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- [src/creatures/hostiles/Crocodile.hpp](#)
- [src/creatures/hostiles/Crocodile.cpp](#)

## 6.14. sf::Event osztályreferencia

```
#include <fake_sfml.hpp>
```

### Publikus típusok

- enum [EType](#) : char { [Closed](#) , [NoEvent](#) , [Invalid](#) }

### Publikus tagfüggvények

- [Event](#) ()

### Publikus attribútumok

- [EType](#) type

### 6.14.1. Enumeráció-tagok dokumentációja

#### 6.14.1.1. EType

```
enum sf::Event::EType : char
```

## Enumeráció-értékek

Closed	
NoEvent	
Invalid	

## 6.14.2. Konstruktorkok és destruktorkok dokumentációja

### 6.14.2.1. Event()

```
sf::Event::Event ( )
```

## 6.14.3. Adattagok dokumentációja

### 6.14.3.1. type

```
EType sf::Event::type
```

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

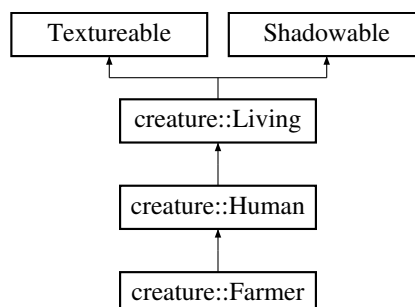
- [src/fake\\_sfml/fake\\_sfml.hpp](#)
- [src/fake\\_sfml/fake\\_sfml.cpp](#)

## 6.15. creature::Farmer osztályreferencia

A farmer szakmájú ember osztály leírása.

```
#include <Farmer.hpp>
```

A creature::Farmer osztály származási diagramja:



## Publikus tagfüggvények

- **Farmer** (int x, int y, ENTITY\_GENDER gender\_modifier)  
*Inicializál egy farmert egy pontos x és y koordinátára és beállítja az attribútumait.*
- void **update\_logic** (World &world, float deltaTime) override  
*Az entitás frissítési logikája itt van definiálva. Például a halászházat keres, ha odaért akkor halászik.*
- **~Farmer** ()  
*A farmer destruktora.*

## További örökölt tagok

### 6.15.1. Részletes leírás

A farmer szakmájú ember osztály leírása.

Ez a szakmájú ember bokrokat keres és kitermeli őket ezzel ételt szerez.

### 6.15.2. Konstruktorok és destruktorkok dokumentációja

#### 6.15.2.1. Farmer()

```
creature::Farmer::Farmer (
    int x,
    int y,
    ENTITY_GENDER gender_modifier )
```

Inicializál egy farmert egy pontos x és y koordinátára és beállítja az attribútumait.

#### Paraméterek

<i>x</i>	Az x koordináta.
<i>y</i>	Az y koordináta.
<i>gender_modifier</i>	A farmer neve.

#### 6.15.2.2. ~Farmer()

```
creature::Farmer::~Farmer ( )
```

A farmer destruktora.

### 6.15.3. Tagfüggvények dokumentációja

### 6.15.3.1. update\_logic()

```
void creature::Farmer::update_logic (
    World & world,
    float deltaTime ) [override], [virtual]
```

Az entitás frissítési logikája itt van definiálva. Például a halász vizet keres, ha odaért akkor halászik.

#### Paraméterek

<i>world</i>	A világ, amibe frissíti magát az entitás.
<i>deltaTime</i>	Az előző frissítés óta eltelt idő.

Megvalósítja a következőket: [creature::Living](#).

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

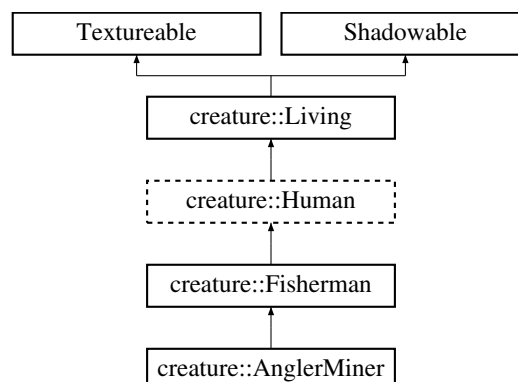
- [src/creatures/humans/Farmer.hpp](#)
- [src/creatures/humans/Farmer.cpp](#)

## 6.16. creature::Fisherman osztályreferencia

A halász szakmájú ember osztály leírása.

```
#include <Fisherman.hpp>
```

A creature::Fisherman osztály származási diagramja:



### Publikus tagfüggvények

- [Fisherman](#) (int x, int y, [ENTITY\\_GENDER](#) gender\_modifier)  
*Inicializál egy halászt egy pontos x és y koordinátára és beállítja az attribútumait.*
- void [update\\_logic](#) ([World](#) &world, float deltaTime) override  
*Az entitás frissítési logikája itt van definiálva. Például a halász vizet keres, ha odaért akkor halászik.*
- [~Fisherman](#) ()  
*A halász destruktora.*



## Védett tagfüggvények

- void `try_fishing` (`World` &world)  
*Megpróbál tavat keresni, ahol halászhat.*

## Védett attribútumok

- bool `fishing`  
*Halászni akar-e az ember jelenleg?*

## További örökölt tagok

### 6.16.1. Részletes leírás

A halász szakmájú ember osztály leírása.

Ez a szakmájú ember víz terepkockát keres és ott halászva ételt szerez.

### 6.16.2. Konstruktorok és destruktorok dokumentációja

#### 6.16.2.1. Fisherman()

```
creature::Fisherman::Fisherman (
    int x,
    int y,
    ENTITY_GENDER gender_modifier )
```

Inicializál egy halászt egy pontos x és y koordinátára és beállítja az attribútumait.

#### Paraméterek

<code>x</code>	Az x koordináta.
<code>y</code>	Az y koordináta.
<code>gender_modifier</code>	A halász neve.

#### 6.16.2.2. ~Fisherman()

```
creature::Fisherman::~~Fisherman ( )
```

A halász destruktora.

### 6.16.3. Tagfüggvények dokumentációja

#### 6.16.3.1. try\_fishing()

```
void creature::Fisherman::try_fishing (
    World & world ) [protected]
```

Megpróbál tavat keresni, ahol halászhat.

##### Paraméterek

<i>world</i>	A világ, amibe tavat kell keresni.
--------------	------------------------------------

#### 6.16.3.2. update\_logic()

```
void creature::Fisherman::update_logic (
    World & world,
    float deltaTime ) [override], [virtual]
```

Az entitás frissítési logikája itt van definiálva. Például a halász vizet keres, ha odaért akkor halászik.

##### Paraméterek

<i>world</i>	A világ, amibe frissíti magát az entitás.
<i>deltaTime</i>	Az előző frissítés óta eltelt idő.

Megvalósítja a következőket: [creature::Living](#).

### 6.16.4. Adattagok dokumentációja

#### 6.16.4.1. fishing

```
bool creature::Fisherman::fishing [protected]
```

Halászni akar-e az ember jelenleg?

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- [src/creatures/humans/Fisherman.hpp](#)
- [src/creatures/humans/Fisherman.cpp](#)

## 6.17. sf::FloatRect osztályreferencia

```
#include <fake_sfml.hpp>
```

### Publikus tagfüggvények

- [FloatRect](#) ()
- [FloatRect](#) (float l, float t, float w, float h)
- bool [contains](#) (float x, float y) const

### Publikus attribútumok

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

### 6.17.1. Konstruktork és destruktork dokumentációja

#### 6.17.1.1. FloatRect() [1/2]

```
sf::FloatRect::FloatRect ( )
```

#### 6.17.1.2. FloatRect() [2/2]

```
sf::FloatRect::FloatRect (
    float l,
    float t,
    float w,
    float h )
```

### 6.17.2. Tagfüggvények dokumentációja

#### 6.17.2.1. contains()

```
bool sf::FloatRect::contains (
    float x,
    float y ) const
```

### 6.17.3. Adattagok dokumentációja

#### 6.17.3.1. height

```
float sf::FloatRect::height
```

#### 6.17.3.2. left

```
float sf::FloatRect::left
```

#### 6.17.3.3. top

```
float sf::FloatRect::top
```

#### 6.17.3.4. width

```
float sf::FloatRect::width
```

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- [src/fake\\_sfml/fake\\_sfml.hpp](#)
- [src/fake\\_sfml/fake\\_sfml.cpp](#)

## 6.18. GameConfig osztályreferencia

A világ szimulációjának leírása.

```
#include <GameConfig.hpp>
```

## Publikus tagfüggvények

- `GameConfig (const GameConfig &)=delete`  
*Nem szükséges a singleton pattern miatt.*
- `GameConfig & operator= (const GameConfig &)=delete`  
*Nem szükséges a singleton pattern miatt.*
- `int get_config_level () const`  
*Visszaadja, hogy mennyire bőbeszédű legyen a hibakezelés / figyelmeztetések.*
- `int get_target_fps () const`  
*Visszaadja az elérni kívánt FPS értékét.*
- `int get_screen_width () const`  
*Visszaadja az ablak szélességét.*
- `int get_screen_height () const`  
*Visszaadja az ablak magasságát.*
- `void set_config_level (int n_flag)`  
*Beállítható, hogy mennyire bőbeszédű legyen a hibakezelés / figyelmeztetések.*
- `int get_world_size () const`  
*Visszaadja a világ konfigurált méretét.*
- `void set_world_size (int newsize)`  
*Beállítja a világ konfigurált méretét.*

## Statikus publikus tagfüggvények

- `static GameConfig & get_instance ()`  
*Visszaad egy referenciát erre az osztály-ra.*

## Publikus attribútumok

- `float berry_growth_time =360.0`  
*Másodpercben mikor nőjön vissza a bogyó a bokrokra.*
- `float max_tree_age =3600.0`  
*Másodpercben mikor pusztuljon el egy a fa.*
- `int max_spawn_tries =99`  
*Meddig próbálkozzon az entitások, objektumok letételével a világ.*
- `int resource_scarcity =4`  
*Milyen bő legyen a világ az erőforrásokban.*
- `int hostiles_count =1`  
*Hány vadállat idéződjön szezononként.*
- `float day_length =400.0`  
*A napok hossza másodpercben.*
- `int max_city_size =16`  
*A maximum ház egy városban.*

### 6.18.1. Részletes leírás

A világ szimulációjának leírása.

Tárolja azokat az értékeket, amiktől függ az, hogy mi mikor és hogyan történik a világba.

## 6.18.2. Konstruktorkok és destruktorkok dokumentációja

### 6.18.2.1. GameConfig()

```
GameConfig::GameConfig (
    const GameConfig & ) [delete]
```

Nem szükséges a singleton pattern miatt.

## 6.18.3. Tagfüggvények dokumentációja

### 6.18.3.1. get\_config\_level()

```
int GameConfig::get_config_level ( ) const
```

Visszaadja, hogy mennyire bőbeszédű legyen a hibakezelés / figyelmeztetések.

### 6.18.3.2. get\_instance()

```
GameConfig & GameConfig::get_instance ( ) [static]
```

Visszaad egy referenciát erre az osztály-ra.

#### Visszatérési érték

A singleton-hoz egy referencia.

### 6.18.3.3. get\_screen\_height()

```
int GameConfig::get_screen_height ( ) const
```

Visszaadja az ablak magasságát.

**6.18.3.4. get\_screen\_width()**

```
int GameConfig::get_screen_width ( ) const
```

Visszaadja az ablak szélességét.

**6.18.3.5. get\_target\_fps()**

```
int GameConfig::get_target_fps ( ) const
```

Visszaadja az elérni kívánt FPS értékét.

**6.18.3.6. get\_world\_size()**

```
int GameConfig::get_world_size ( ) const
```

Visszaadja a világ konfigurált méretét.

**6.18.3.7. operator=()**

```
GameConfig& GameConfig::operator= (
    const GameConfig & ) [delete]
```

Nem szükséges a singleton pattern miatt.

**6.18.3.8. set\_config\_level()**

```
void GameConfig::set_config_level (
    int n_flag )
```

Beállítható, hogy mennyire bőbeszédű legyen a hibakezelés / figyelmeztetések.

**6.18.3.9. set\_world\_size()**

```
void GameConfig::set_world_size (
    int newsize )
```

Beállítja a világ konfigurált méretét.

## 6.18.4. Adattagok dokumentációja

### 6.18.4.1. berry\_growth\_time

```
float GameConfig::berry_growth_time =360.0
```

Másodpercben mikor nőjön vissza a bogyó a bokrokra.

### 6.18.4.2. day\_length

```
float GameConfig::day_length =400.0
```

A napok hossza másodpercben.

### 6.18.4.3. hostiles\_count

```
int GameConfig::hostiles_count =1
```

Hány vadállat idéződjön szezononként.

### 6.18.4.4. max\_city\_size

```
int GameConfig::max_city_size =16
```

A maximum ház egy városban.

### 6.18.4.5. max\_spawn\_tries

```
int GameConfig::max_spawn_tries =99
```

Meddig próbálkozzon az entitások, objektumok letételével a világ.



#### 6.18.4.6. max\_tree\_age

```
float GameConfig::max_tree_age =3600.0
```

Másodpercben mikor pusztuljon el egy a fa.

#### 6.18.4.7. resource\_scarcity

```
int GameConfig::resource_scarcity =4
```

Milyen bő legyen a világ az erőforrásokban.

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- [src/GameConfig.hpp](#)
- [src/GameConfig.cpp](#)

## 6.19. GameManager osztályreferencia

A világ szimulálásáért és a kirazolás irányításáért felelős osztály.

```
#include <GameManager.hpp>
```

### Publikus tagfüggvények

- [GameManager \(\)](#)  
*A konstruktorba létrejön az ablak és az alap változók beállításra kerülnek.*
- void [run \(\)](#)  
*Elindítja a szimulációt és innentől kirajolja a világot, gombokat.*
- void [game\\_loop \(\)](#)  
*A szimuláció loopolását indítja el.*
- void [setup\\_buttons \(\)](#)  
*A gombokat létrehozza, textúrájukat, viselkedésüket betölti.*
- void [update\\_buttons \(\)](#)  
*Frissíti a gombokat, ha 1-re rákattintottak.*
- void [draw\\_buttons \(\)](#)  
*Kirajolja a gombokat.*
- bool [is\\_valid \(\)](#) const  
*Megadja, hogy sikeres lett-e a szimulációs elemek inicializálása.*
- float [get\\_elapsed\\_time \(\)](#) const  
*Megadja az eltelt időt, ami eltelt a szimulációba.*
- void [simulate\\_tick](#) (float e\_time)  
*Szimulál T idő egységnyi időt.*
- [~GameManager \(\)](#)  
*Felszabadítja a világot, gombokat, hangot, textúrákat, render ablakot. Mindent, ami a program tartalmaz.*

### 6.19.1. Részletes leírás

A világ szimulálásáért és a kirazolás irányításáért felelős osztály.

Tárolja a világot, a render ablakot, a kamera adatait, a gombokat és a zene lejátszót. Végül mindent ez az osztály szabadít fel.

### 6.19.2. Konstruktorkok és destruktorkok dokumentációja

#### 6.19.2.1. GameManager()

```
GameManager::GameManager ( )
```

A konstruktorba létrejön az ablak és az alap változók beállításra kerülnek.

#### 6.19.2.2. ~GameManager()

```
GameManager::~~GameManager ( )
```

Felszabadítja a világot, gombokat, hangot, textúrákat, render ablakot. Mindent, ami a program tartalmaz.

### 6.19.3. Tagfüggvények dokumentációja

#### 6.19.3.1. draw\_buttons()

```
void GameManager::draw_buttons ( )
```

Kirajzolja a gombokat.

#### 6.19.3.2. game\_loop()

```
void GameManager::game_loop ( )
```

A szimuláció loopolását indítja el.

#### 6.19.3.3. get\_elapsed\_time()

```
float GameManager::get_elapsed_time ( ) const
```

Megadja az eltelt időt, ami eltelt a szimulációba.

#### 6.19.3.4. is\_valid()

```
bool GameManager::is_valid ( ) const
```

Megadja, hogy sikeres lett-e a szimulációs elemek inicializálása.

#### 6.19.3.5. run()

```
void GameManager::run ( )
```

Elindítja a szimulációt és innentől kirajolja a világot, gombokat.

#### 6.19.3.6. setup\_buttons()

```
void GameManager::setup_buttons ( )
```

A gombokat létrehozza, textúrájukat, viselkedésüket betölti.

#### 6.19.3.7. simulate\_tick()

```
void GameManager::simulate_tick (
    float e_time )
```

Szimulál T idő egységnyi időt.

#### 6.19.3.8. update\_buttons()

```
void GameManager::update_buttons ( )
```

Frissíti a gombokat, ha 1-re rákattintottak.

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

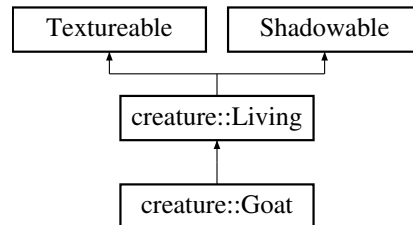
- [src/GameManager.hpp](#)
- [src/GameManager.cpp](#)

## 6.20. creature::Goat osztályreferencia

A kecske osztály leírása.

```
#include <Goat.hpp>
```

A creature::Goat osztály származási diagramja:



### Publikus tagfüggvények

- **Goat** (int x, int y)  
*Idéz egy kecskét egy pontos x és y koordinátára és beállítja az attribútumait.*
- **ENTITY\_TYPE get\_type** () const override  
*Visszaadja az entitás belső szimbólumát. Ez abba segít, hogy a vadállatok például csak embert támadjanak, Az emberek csak vadállatot.*
- void **die** () override  
*Mi történjen, ha meghal az entitás.*
- void **update\_logic** (World &world, float deltaTime) override  
*Az entitás frissítési logikája itt van definiálva. Például a halász vizet keres, ha odaért akkor halászik.*
- void **draw\_logic** (sf::RenderWindow &window, float deltaTime, int offx, int offy) override  
*Az entitás kirajzolás logikája, például az ember az árnyékát és képét rajzolja ki, de a robot csak a képét.*
- virtual **~Goat** ()  
*Virtuális destruktork.*

### További örökölt tagok

#### 6.20.1. Részletes leírás

A kecske osztály leírása.

A kecske egy passzív, nem támadó állat, amit ha az emberek megölnek, ételt ad.

#### 6.20.2. Konstruktorkok és destruktorkok dokumentációja

##### 6.20.2.1. Goat()

```
creature::Goat::Goat (
    int x,
    int y )
```

Idéz egy kecskét egy pontos x és y koordinátára és beállítja az attribútumait.

## Paraméterek

<i>x</i>	Az x koordináta.
<i>y</i>	Az y koordináta.

**6.20.2.2. ~Goat()**

```
creature::Goat::~~Goat ( ) [virtual]
```

Virtuális destruktork.

**6.20.3. Tagfüggvények dokumentációja****6.20.3.1. die()**

```
void creature::Goat::die ( ) [override], [virtual]
```

Mi történjen, ha meghal az entitás.

Megvalósítja a következőket: [creature::Living](#).

**6.20.3.2. draw\_logic()**

```
void creature::Goat::draw_logic (
    sf::RenderWindow & window,
    float deltaTime,
    int offx,
    int offy ) [override], [virtual]
```

Az entitás kirajzolás logikája, például az ember az árnyékát és képét rajzolja ki, de a robot csak a képét.

## Paraméterek

<i>window</i>	Az ablak, ahova rajzolni kell.
<i>deltaTime</i>	Az előző frissítés óta eltelt idő.
<i>offx</i>	A kamera X eltolása.
<i>offy</i>	A kamera Y eltolása.

Megvalósítja a következőket: [creature::Living](#).

### 6.20.3.3. get\_type()

```
ENTITY_TYPE creature::Goat::get_type ( ) const [override], [virtual]
```

Visszaadja az entitás belső szimbólumát. Ez abba segít, hogy a vadállatok például csak embert támadjanak, Az emberek csak vadállatot.

#### Visszatérési érték

A belső szimbólum.

Megvalósítja a következőket: [creature::Living](#).

### 6.20.3.4. update\_logic()

```
void creature::Goat::update_logic (
    World & world,
    float deltaTime ) [override], [virtual]
```

Az entitás frissítési logikája itt van definiálva. Például a halász vizet keres, ha odaért akkor halászik.

#### Paraméterek

<i>world</i>	A világ, amibe frissíti magát az entitás.
<i>deltaTime</i>	Az előző frissítés óta eltelt idő.

Megvalósítja a következőket: [creature::Living](#).

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

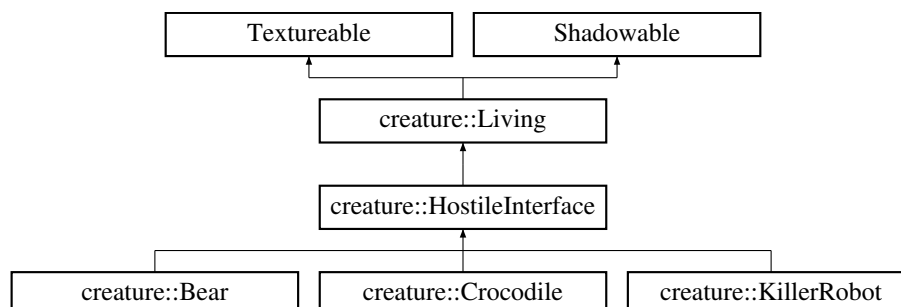
- [src/creatures/Goat.hpp](#)
- [src/creatures/Goat.cpp](#)

## 6.21. creature::HostileInterface osztályreferencia

A vadállat entitások interface leírása.

```
#include <HostileInterface.hpp>
```

A creature::HostileInterface osztály származási diagramja:



## Publikus tagfüggvények

- void `set_hostile_config` (int newdamage, float newattackspeed)  
*Beállítja a vadállat támadási sebességét és sebzését.*
- virtual void `select_target` (World &world)=0  
*A logikát írja le, ahogy az entitás a világba kiválasztja magának a célpontot.*
- virtual `~HostileInterface` ()=default  
*Virtuális destruktork.*
- void `retarget` (Living \*new\_target) override  
*Felidegesíti az entitást a kapott entításra.*
- Living \* `check_aggroed` () const override  
*Visszaadja, hogy kire "ideges az entitás". Ez lehet az, hogy ki ütötte meg vagy hogy kit akar megenni.*

## Védett tagfüggvények

- void `try_attack` ()  
*Megnézi, hogy milyen közel van a célpontja, ha elég közel van, akkor támad.*
- void `hostile_run` (float deltaTime)  
*Egységes futás logika. Addíg fut a célpont felé míg az vagy meghal vagy elég közel lesz.*
- void `hostile_walk` (float deltaTime)  
*Egységes séta logika. Addíg sétál a célpont felé míg az vagy meghal vagy elég közel lesz. Ezt használja a krokodil, ha ebbe a fázisba meghal a célpont, akkor futás módba vált és egyből keres egy új célpontot.*

## Védett attribútumok

- sf::Vector2f `goal`  
*Az célpont entitás pozíciója.*
- int `damage`  
*A vadállat sebzése.*
- float `attack_speed`  
*A vadállat támadási sebessége.*
- Living \* `target`  
*A vadállat célpontja.*

## További örökölt tagok

### 6.21.1. Részletes leírás

A vadállat entitások interface leírása.

Ebbe minden deklarálva van, ami ahhoz kell, hogy egy entitás agresszív legyen. Van célpontjuk, egységes támadási módszereik és sebzésük.

### 6.21.2. Konstruktorkok és destruktorkok dokumentációja

#### 6.21.2.1. ~HostileInterface()

```
virtual creature::HostileInterface::~~HostileInterface ( ) [virtual], [default]
```

Virtuális destruktor.

### 6.21.3. Tagfüggvények dokumentációja

#### 6.21.3.1. check\_aggroed()

```
Living * creature::HostileInterface::check_aggroed ( ) const [override], [virtual]
```

Visszaadja, hogy kire "ideges az entitás". Ez lehet az, hogy ki ütötte meg vagy hogy kit akar megenni.

Visszatérési érték

Az entitás, akire ideges. Nullpointer, ha nincs ilyen entitás.

Újraimplementált ősök: [creature::Living](#).

#### 6.21.3.2. hostile\_run()

```
void creature::HostileInterface::hostile_run (
    float deltaTime ) [protected]
```

Egységes futás logika. Addíg fut a célpont felé míg az vagy meghal vagy elég közel lesz.

#### 6.21.3.3. hostile\_walk()

```
void creature::HostileInterface::hostile_walk (
    float deltaTime ) [protected]
```

Egységes séta logika. Addíg sétál a célpont felé míg az vagy meghal vagy elég közel lesz. Ezt használja a krokodil, ha ebbe a fázisba meghal a célpont, akkor futás módba vált és egyből keres egy új célpontot.

#### 6.21.3.4. retarget()

```
void creature::HostileInterface::retarget (
    Living * new_target ) [override], [virtual]
```

Felidegesíti az entitást a kapott entításra.



## Paraméterek

<i>new_target</i>	Az entitás, akire dühösnek kell lennie.
-------------------	---

Újrimplementált ősök: [creature::Living](#).

**6.21.3.5. select\_target()**

```
virtual void creature::HostileInterface::select_target (
    World & world ) [pure virtual]
```

A logikát írja le, ahogy az entitás a világba kiválasztja magának a célpontot.

## Paraméterek

<i>world</i>	A világ, amibe a célpontot kell választani.
--------------	---

Megvalósítják a következők: [creature::KillerRobot](#), [creature::Crocodile](#) és [creature::Bear](#).

**6.21.3.6. set\_hostile\_config()**

```
void creature::HostileInterface::set_hostile_config (
    int newdamage,
    float newattackspeed )
```

Beállítja a vadállat támadási sebességét és sebzését.

## Paraméterek

<i>newdamage</i>	Az új beállított sebzés.
<i>newattackspeed</i>	Az új beállított sebzési sebesség.

**6.21.3.7. try\_attack()**

```
void creature::HostileInterface::try_attack ( ) [protected]
```

Megnézi, hogy milyen közel van a célpontja, ha elég közel van, akkor támad.

**6.21.4. Adattagok dokumentációja**

#### 6.21.4.1. attack\_speed

```
float creature::HostileInterface::attack_speed [protected]
```

A vadállat támadási sebessége.

#### 6.21.4.2. damage

```
int creature::HostileInterface::damage [protected]
```

A vadállat sebzése.

#### 6.21.4.3. goal

```
sf::Vector2f creature::HostileInterface::goal [protected]
```

Az célpont entitás pozíciója.

#### 6.21.4.4. target

```
Living* creature::HostileInterface::target [protected]
```

A vadállat célpontja.

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

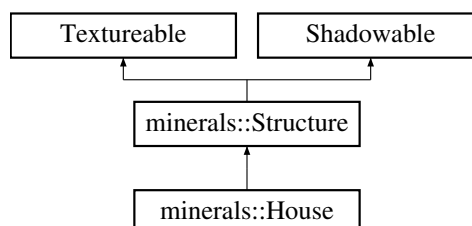
- [src/creatures/HostileInterface.hpp](#)
- [src/creatures/HostileInterface.cpp](#)

## 6.22. minerals::House osztályreferencia

A ház osztály leírása. Szinttől függően idéz embereket.

```
#include <House.hpp>
```

A minerals::House osztály származási diagramja:



## Publikus tagfüggvények

- `House` (int x, int y)  
*Konstruktor ami lerakja a házat egy (x,y) pontra.*
- `MINERAL_TYPE get_type` () const override  
*Szimbólum, ami a fájlba mentéshez kell.*
- void `update_logic` (float deltaTime) override  
*Frissíti magát az idő függvényében.*

## Publikus attribútumok

- int `level`  
*Milyen modern a ház (1-3 -ig).*
- int `stone_req`  
*Mennyi kő kell, hogy a ház elérje a következő szintet.*
- int `wood_req`  
*Mennyi fa kell, hogy a ház elérje a következő szintet.*
- int `iron_req`  
*Mennyi vas kell, hogy a ház elérje a következő szintet.*

## További örökölt tagok

### 6.22.1. Részletes leírás

A ház osztály leírása. Szinttől függően idéz embereket.

### 6.22.2. Konstruktorok és destruktorok dokumentációja

#### 6.22.2.1. `House()`

```
minerals::House::House (  
    int x,  
    int y )
```

Konstruktor ami lerakja a házat egy (x,y) pontra.

### 6.22.3. Tagfüggvények dokumentációja

#### 6.22.3.1. `get_type()`

```
MINERAL_TYPE minerals::House::get_type ( ) const [override], [virtual]
```

Szimbólum, ami a fájlba mentéshez kell.

Megvalósítja a következőket: [minerals::Structure](#).

#### 6.22.3.2. `update_logic()`

```
void minerals::House::update_logic (
    float deltaTime ) [override], [virtual]
```

Frissíti magát az idő függvényében.

##### Paraméterek

<i>deltaTime</i>	Az előző frissítés óta eltelt idő.
------------------	------------------------------------

Megvalósítja a következőket: [minerals::Structure](#).

### 6.22.4. Adattagok dokumentációja

#### 6.22.4.1. `iron_req`

```
int minerals::House::iron_req
```

Mennyi vas kell, hogy a ház elérje a következő szintet.

#### 6.22.4.2. `level`

```
int minerals::House::level
```

Milyen modern a ház (1-3 -ig).

#### 6.22.4.3. `stone_req`

```
int minerals::House::stone_req
```

Mennyi kő kell, hogy a ház elérje a következő szintet.

#### 6.22.4.4. wood\_req

```
int minerals::House::wood_req
```

Mennyi fa kell, hogy a ház elérje a következő szintet.

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

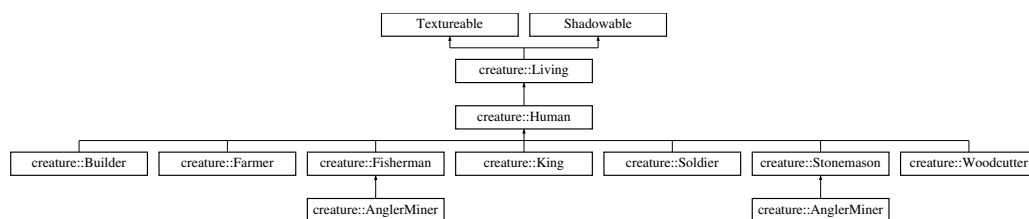
- [src/world\\_object/House.hpp](#)
- [src/world\\_object/House.cpp](#)

## 6.23. creature::Human osztályreferencia

Az alap ember osztály leírása. Minden fajta szakmájú ember innen öröklődik.

```
#include <Human.hpp>
```

A creature::Human osztály származási diagramja:



### Publikus tagfüggvények

- [Human](#) (int x, int y)  
*Az alap konstruktor ami leidézi az embert egy x és y koordinátára.*
- [Human](#) (int x, int y, [ENTITY\\_GENDER](#) const\_gender)  
*Az alap konstruktor ami leidézi az embert egy x és y koordinátára egy megadott nemmel.*
- [ENTITY\\_TYPE](#) [get\\_type](#) () const override  
*Visszaadja az entitás belső szimbólumát. Ez abba segít, hogy a vadállatok például csak embert támadjanak, Az emberek csak vadállatot.*
- void [die](#) () override  
*Mi történjen, ha meghal az entitás.*
- void [update\\_logic](#) ([World](#) &world, float deltaTime) override  
*Az entitás frissítési logikája itt van definiálva. Például a halász vizet keres, ha odaért akkor halászik.*
- void [draw\\_logic](#) ([sf::RenderWindow](#) &window, float deltaTime, int offx, int offy) override  
*Az entitás kirajzolási logikája, például az ember az árnyékát és képét rajzolja ki, de a robot csak a képét.*
- virtual [~Human](#) ()  
*Virtuális destruktork, felszabadítja a szakma ikon pointer-t is.*
- void [initialize](#) (int x, int y)  
*Beállítja az ember tulajdonságait: életpontok, max életkor, nem.*
- void [select\\_texture](#) (int x, int y, int gender\_selector)  
*Beállít egy textúrát ami nagyon különböző lehet emberenként és egyből beállítja, hogy az embert a saját (x,y) koordinátára rajzolják.*
- std::string [get\\_profession\\_string](#) ()  
*Lekérhető az ember szakmájának szöveggé alakított szimbóluma. Ez fontos a fájlba tároláshoz.*

## Publikus attribútumok

- bool `needs_to_be_royal`  
*Kell-e királyá koronázni?*
- bool `needs_promotion`  
*Kell-e neki egy új szakma? Csak akkor igaz, ha már van város.*

## Védett attribútumok

- `Profession` \* `profession`  
*A szakma ikon pointere.*
- `sf::Vector2f` `goal`  
*A cselekvésének a célpontja.*

### 6.23.1. Részletes leírás

Az alap ember osztály leírása. Minden fajta szakmájú ember innen öröklődik.

Tárolja az ember szakma címerét, célkoordinátáját is.

### 6.23.2. Konstruktorkok és destruktorkok dokumentációja

#### 6.23.2.1. `Human()` [1/2]

```
creature::Human::Human (
    int x,
    int y )
```

Az alap konstruktor ami leidézi az embert egy x és y koordinátára.

#### Paraméterek

<code>x</code>	Az x koordináta.
<code>y</code>	Az y koordináta.

#### 6.23.2.2. `Human()` [2/2]

```
creature::Human::Human (
    int x,
    int y,
    ENTITY_GENDER const_gender )
```

Az alap konstruktor ami leidézi az embert egy x és y koordinátára egy megadott nemmel.

## Paraméterek

<i>x</i>	Az x koordináta.
<i>y</i>	Az y koordináta.
<i>const_gender</i>	Az ember neme.

## 6.23.2.3. ~Human()

```
creature::Human::~~Human ( ) [virtual]
```

Virtuális destruktork, felszabadítja a szakma ikon pointert is.

## 6.23.3. Tagfüggvények dokumentációja

## 6.23.3.1. die()

```
void creature::Human::die ( ) [override], [virtual]
```

Mi történjen, ha meghal az entitás.

Megvalósítja a következőket: [creature::Living](#).

## 6.23.3.2. draw\_logic()

```
void creature::Human::draw_logic (
    sf::RenderWindow & window,
    float deltaTime,
    int offx,
    int offy ) [override], [virtual]
```

Az entitás kirajzolás logikája, például az ember az árnyékát és képét rajzolja ki, de a robot csak a képét.

## Paraméterek

<i>window</i>	Az ablak, ahova rajzolni kell.
<i>deltaTime</i>	Az előző frissítés óta eltelt idő.
<i>offx</i>	A kamera X eltolása.
<i>offy</i>	A kamera Y eltolása.

Megvalósítja a következőket: [creature::Living](#).

#### 6.23.3.3. get\_profession\_string()

```
std::string creature::Human::get_profession_string ( )
```

Lekérhető az ember szakmájának szöveggé alakított szimbóluma. Ez fontos a fájlba tároláshoz.

##### Visszatérési érték

Az ember szakmájának szimbóluma.

#### 6.23.3.4. get\_type()

```
ENTITY_TYPE creature::Human::get_type ( ) const [override], [virtual]
```

Visszaadja az entitás belső szimbólumát. Ez abba segít, hogy a vadállatok például csak embert támadjanak, Az emberek csak vadállatot.

##### Visszatérési érték

A belső szimbólum.

Megvalósítja a következőket: [creature::Living](#).

#### 6.23.3.5. initialize()

```
void creature::Human::initialize (
    int x,
    int y )
```

Beállítja az ember tulajdonságait: életpontok, max életkor, nem.

##### Paraméterek

<i>x</i>	Az x koordináta.
<i>y</i>	Az y koordináta.

#### 6.23.3.6. select\_texture()

```
void creature::Human::select_texture (
    int x,
    int y,
    int gender_selector )
```



Beállít egy textúrát ami nagyon különböző lehet emberenéknt és egyből beállítja, hogy az embert a saját (x,y) koordinátára rajzolják.

## Paraméterek

<i>x</i>	Az x koordináta.
<i>y</i>	Az y koordináta.
<i>gender_selector</i>	Egy véletlen szám. Ettől függ a textúra variáció.

**6.23.3.7. update\_logic()**

```
void creature::Human::update_logic (
    World & world,
    float deltaTime ) [override], [virtual]
```

Az entitás frissítési logikája itt van definiálva. Például a halász vizet keres, ha odaért akkor halászik.

## Paraméterek

<i>world</i>	A világ, amibe frissíti magát az entitás.
<i>deltaTime</i>	Az előző frissítés óta eltelt idő.

Megvalósítja a következőket: [creature::Living](#).

Újrimplementáló leszármazottak: [creature::Woodcutter](#), [creature::Stonemason](#), [creature::Soldier](#) és [creature::King](#).

**6.23.4. Adattagok dokumentációja****6.23.4.1. goal**

```
sf::Vector2f creature::Human::goal [protected]
```

A cselekvésének a célpontja.

**6.23.4.2. needs\_promotion**

```
bool creature::Human::needs_promotion
```

Kell-e neki egy új szakma? Csak akkor igaz, ha már van város.

#### 6.23.4.3. needs\_to\_be\_royal

```
bool creature::Human::needs_to_be_royal
```

Kell-e királyá koronázni?

#### 6.23.4.4. profession

```
Profession* creature::Human::profession [protected]
```

A szakma ikon pointerre.

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

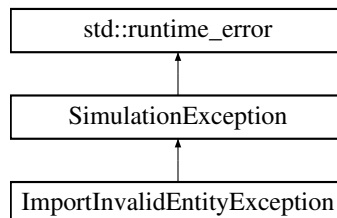
- [src/creatures/humans/Human.hpp](#)
- [src/creatures/humans/Human.cpp](#)

## 6.24. ImportInvalidEntityException osztályreferencia

Akkor kell dobni, ha egy entitás hibásan lett beolvasva.

```
#include <FileExceptions.hpp>
```

Az ImportInvalidEntityException osztály származási diagramja:



### Publikus tagfüggvények

- [ImportInvalidEntityException](#) (const std::string &msg)

#### 6.24.1. Részletes leírás

Akkor kell dobni, ha egy entitás hibásan lett beolvasva.

#### 6.24.2. Konstruktorkok és destruktorkok dokumentációja

#### 6.24.2.1. ImportInvalidEntityException()

```
ImportInvalidEntityException::ImportInvalidEntityException (
    const std::string & msg ) [inline], [explicit]
```

Ez a dokumentáció az osztályról a következő fájl alapján készült:

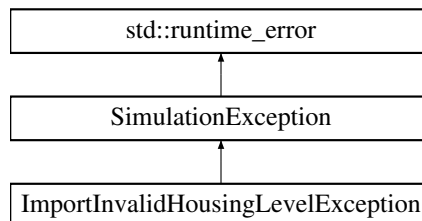
- src/exceptions/[FileExceptions.hpp](#)

### 6.25. ImportInvalidHousingLevelException osztályreferencia

Akkor kell dobni, ha egy ház hibásan lett beolvasva.

```
#include <FileExceptions.hpp>
```

Az ImportInvalidHousingLevelException osztály származási diagramja:



#### Publikus tagfüggvények

- [ImportInvalidHousingLevelException](#) (const std::string &msg)

#### 6.25.1. Részletes leírás

Akkor kell dobni, ha egy ház hibásan lett beolvasva.

#### 6.25.2. Konstruktorkok és destruktorkok dokumentációja

##### 6.25.2.1. ImportInvalidHousingLevelException()

```
ImportInvalidHousingLevelException::ImportInvalidHousingLevelException (
    const std::string & msg ) [inline], [explicit]
```

Ez a dokumentáció az osztályról a következő fájl alapján készült:

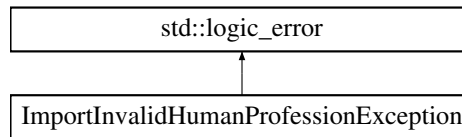
- src/exceptions/[FileExceptions.hpp](#)

## 6.26. ImportInvalidHumanProfessionException osztályreferencia

Akkor kell dobni, ha egy szakma hibásan lett beolvasva.

```
#include <FileExceptions.hpp>
```

Az ImportInvalidHumanProfessionException osztály származási diagramja:



### Publikus tagfüggvények

- [ImportInvalidHumanProfessionException](#) (const std::string &msg)

#### 6.26.1. Részletes leírás

Akkor kell dobni, ha egy szakma hibásan lett beolvasva.

#### 6.26.2. Konstruktorkok és destruktorkok dokumentációja

##### 6.26.2.1. ImportInvalidHumanProfessionException()

```
ImportInvalidHumanProfessionException::ImportInvalidHumanProfessionException (
    const std::string & msg ) [inline], [explicit]
```

Ez a dokumentáció az osztályról a következő fájl alapján készült:

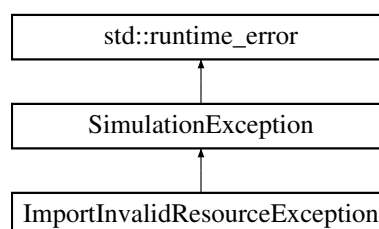
- src/exceptions/[FileExceptions.hpp](#)

## 6.27. ImportInvalidResourceException osztályreferencia

Akkor kell dobni, ha egy erőforrás hibásan lett beolvasva.

```
#include <FileExceptions.hpp>
```

Az ImportInvalidResourceException osztály származási diagramja:



## Publikus tagfüggvények

- [ImportInvalidResourceException](#) (const std::string &msg)

### 6.27.1. Részletes leírás

Akkor kell dobni, ha egy erőforrás hibásan lett beolvasva.

### 6.27.2. Konstruktorok és destruktorok dokumentációja

#### 6.27.2.1. ImportInvalidResourceException()

```
ImportInvalidResourceException::ImportInvalidResourceException (
    const std::string & msg ) [inline], [explicit]
```

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- src/exceptions/[FileExceptions.hpp](#)

## 6.28. sf::IntRect osztályreferencia

```
#include <fake_sfml.hpp>
```

## Publikus tagfüggvények

- [IntRect](#) ()
- [IntRect](#) (int l, int t, int w, int h)

## Publikus attribútumok

- int [left](#)
- int [top](#)
- int [width](#)
- int [height](#)

### 6.28.1. Konstruktorok és destruktorok dokumentációja

**6.28.1.1. IntRect()** [1/2]

```
sf::IntRect::IntRect ( )
```

**6.28.1.2. IntRect()** [2/2]

```
sf::IntRect::IntRect (
    int l,
    int t,
    int w,
    int h )
```

**6.28.2. Adattagok dokumentációja****6.28.2.1. height**

```
int sf::IntRect::height
```

**6.28.2.2. left**

```
int sf::IntRect::left
```

**6.28.2.3. top**

```
int sf::IntRect::top
```

**6.28.2.4. width**

```
int sf::IntRect::width
```

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

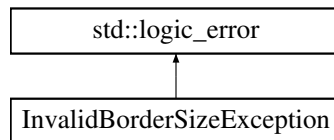
- [src/fake\\_sfml/fake\\_sfml.hpp](#)
- [src/fake\\_sfml/fake\\_sfml.cpp](#)

## 6.29. InvalidBorderSizeException osztályreferencia

Akkor kell dobni, ha egy világhatárnak nem jó értéket akarnak beállítani.

```
#include <WorldExceptions.hpp>
```

Az InvalidBorderSizeException osztály származási diagramja:



### Publikus tagfüggvények

- [InvalidBorderSizeException](#) (const std::string &msg)

#### 6.29.1. Részletes leírás

Akkor kell dobni, ha egy világhatárnak nem jó értéket akarnak beállítani.

#### 6.29.2. Konstruktorkok és destruktorkok dokumentációja

##### 6.29.2.1. InvalidBorderSizeException()

```
InvalidBorderSizeException::InvalidBorderSizeException (
    const std::string & msg ) [inline], [explicit]
```

Ez a dokumentáció az osztályról a következő fájl alapján készült:

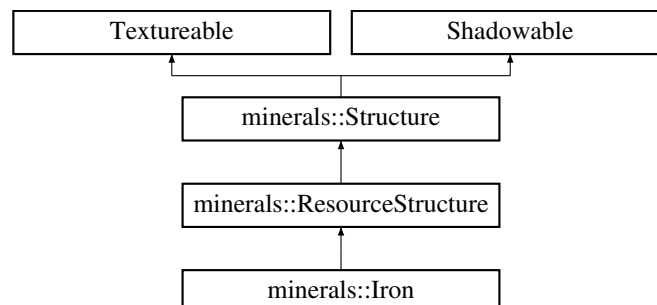
- src/exceptions/[WorldExceptions.hpp](#)

## 6.30. minerals::Iron osztályreferencia

A vasérc osztály leírása. Vasat ad, amikor kitermelik.

```
#include <Iron.hpp>
```

A minerals::Iron osztály származási diagramja:





## Publikus tagfüggvények

- [Iron](#) (int x, int y)  
*Konstruktor ami lerakja a házat egy (x,y) pontra.*
- [MINERAL\\_TYPE get\\_type](#) () const override  
*Szimbólum, ami a fájlba mentéshez kell.*
- void [update\\_logic](#) (float deltaTime) override  
*Frissíti magát az idő függvényében.*
- bool [harvest](#) () override  
*Tisztán virtuális metódus. Ez leírja mi történik, hogy ha kitermelik ezt az erőforrást.*

## További örökölt tagok

### 6.30.1. Részletes leírás

A vasérc osztály leírása. Vasat ad, amikor kitermelik.

### 6.30.2. Konstruktorok és destruktorok dokumentációja

#### 6.30.2.1. Iron()

```
minerals::Iron::Iron (  
    int x,  
    int y )
```

Konstruktor ami lerakja a házat egy (x,y) pontra.

### 6.30.3. Tagfüggvények dokumentációja

#### 6.30.3.1. get\_type()

```
MINERAL\_TYPE minerals::Iron::get_type ( ) const [override], [virtual]
```

Szimbólum, ami a fájlba mentéshez kell.

Megvalósítja a következőket: [minerals::Structure](#).

### 6.30.3.2. harvest()

```
bool minerals::Iron::harvest ( ) [override], [virtual]
```

Tisztán virtuális metódus. Ez leírja mi történik, hogy ha kitermelik ezt az erőforrást.

Megvalósítja a következőket: [minerals::ResourceStructure](#).

### 6.30.3.3. update\_logic()

```
void minerals::Iron::update_logic (
    float deltaTime ) [override], [virtual]
```

Frissíti magát az idő függvényében.

## Paraméterek

<i>deltaTime</i>	Az előző frissítés óta eltelt idő.
------------------	------------------------------------

Megvalósítja a következőket: [minerals::Structure](#).

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- [src/world\\_object/Iron.hpp](#)
- [src/world\\_object/Iron.cpp](#)

## 6.31. sf::Keyboard osztályreferencia

```
#include <fake_sfml.hpp>
```

### Publikus típusok

- enum [Keydowntype](#) : char { [Right](#) , [Left](#) , [Down](#) , [Up](#) }

### Statikus publikus tagfüggvények

- static bool [isKeyPressed](#) ([Keydowntype](#) key)
- static void [simulate\\_key\\_press](#) ([Keydowntype](#) key)
- static void [simulate\\_key\\_release](#) ([Keydowntype](#) key)

### 6.31.1. Enumeráció-tagok dokumentációja

#### 6.31.1.1. Keydowntype

```
enum sf::Keyboard::Keydowntype : char
```

#### Enumeráció-értékek

<a href="#">Right</a>	
<a href="#">Left</a>	
<a href="#">Down</a>	
<a href="#">Up</a>	

### 6.31.2. Tagfüggvények dokumentációja

### 6.31.2.1. isKeyPressed()

```
bool sf::Keyboard::isKeyPressed (
    Keydowntype key ) [static]
```

### 6.31.2.2. simulate\_key\_press()

```
void sf::Keyboard::simulate_key_press (
    Keydowntype key ) [static]
```

### 6.31.2.3. simulate\_key\_release()

```
void sf::Keyboard::simulate_key_release (
    Keydowntype key ) [static]
```

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

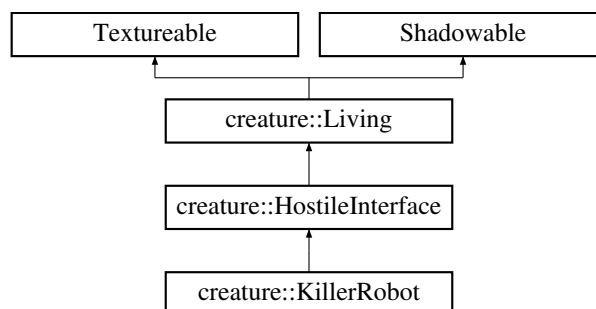
- [src/fake\\_sfml/fake\\_sfml.hpp](#)
- [src/fake\\_sfml/fake\\_sfml.cpp](#)

## 6.32. creature::KillerRobot osztályreferencia

A gyilkos robot osztály leírása.

```
#include <KillerRobot.hpp>
```

A creature::KillerRobot osztály származási diagramja:



## Publikus tagfüggvények

- [KillerRobot](#) (int x, int y)  
*Idéz egy gyilkos robotot egy pontos x és y koordinátára és beállítja az attribútumait.*
- [ENTITY\\_TYPE](#) [get\\_type](#) () const override  
*Visszaadja az entitás belső szimbólumát. Ez abba segít, hogy a vadállatok például csak embert támadjanak, Az emberek csak vadállatot.*
- void [die](#) () override  
*Mi történjen, ha meghal az entitás.*
- void [update\\_logic](#) ([World](#) &world, float deltaTime) override  
*Az entitás frissítési logikája itt van definiálva. Például a halász vizet keres, ha odaért akkor halászik.*
- void [draw\\_logic](#) ([sf::RenderWindow](#) &window, float deltaTime, int ofx, int ofy) override  
*Az entitás kirajzolás logikája, például az ember az árnyékát és képét rajzolja ki, de a robot csak a képét.*
- void [select\\_target](#) ([World](#) &world) override  
*A logikát írja le, ahogy az entitás a világba kiválasztja magának a célpontot.*
- [~KillerRobot](#) ()  
*Virtuális destruktork.*

## További örökölt tagok

### 6.32.1. Részletes leírás

A gyilkos robot osztály leírása.

A gyilkos robot egy ritka ellenség, aminek az az egy célja, hogy kiirtsa az emberiséget, majdnem egy évezredig él (999 évig pontosan), így vagy ő marad, vagy az emberiség.

### 6.32.2. Konstruktorkok és destruktorkok dokumentációja

#### 6.32.2.1. KillerRobot()

```
creature::KillerRobot::KillerRobot (
    int x,
    int y )
```

Idéz egy gyilkos robotot egy pontos x és y koordinátára és beállítja az attribútumait.

#### Paraméterek

x	Az x koordináta.
y	Az y koordináta.

### 6.32.2.2. ~KillerRobot()

```
creature::KillerRobot::~~KillerRobot ( )
```

Virtuális destruktorkor.

## 6.32.3. Tagfüggvények dokumentációja

### 6.32.3.1. die()

```
void creature::KillerRobot::die ( ) [override], [virtual]
```

Mi történjen, ha meghal az entitás.

Megvalósítja a következőket: [creature::Living](#).

### 6.32.3.2. draw\_logic()

```
void creature::KillerRobot::draw_logic (
    sf::RenderWindow & window,
    float deltaTime,
    int offx,
    int offy ) [override], [virtual]
```

Az entitás kirajzolás logikája, például az ember az árnyékát és képét rajzolja ki, de a robot csak a képét.

#### Paraméterek

<i>window</i>	Az ablak, ahova rajzolni kell.
<i>deltaTime</i>	Az előző frissítés óta eltelt idő.
<i>offx</i>	A kamera X eltolása.
<i>offy</i>	A kamera Y eltolása.

Megvalósítja a következőket: [creature::Living](#).

### 6.32.3.3. get\_type()

```
ENTITY_TYPE creature::KillerRobot::get_type ( ) const [override], [virtual]
```

Visszaadja az entitás belső szimbólumát. Ez abba segít, hogy a vadállatok például csak embert támadjanak, Az emberek csak vadállatot.

## Visszatérési érték

A belső szimbólum.

Megvalósítja a következőket: [creature::Living](#).

## 6.32.3.4. select\_target()

```
void creature::KillerRobot::select_target (
    World & world ) [override], [virtual]
```

A logikát írja le, ahogy az entitás a világba kiválasztja magának a célpontot.

## Paraméterek

<i>world</i>	A világ, amibe a célpontot kell választani.
--------------	---

Megvalósítja a következőket: [creature::HostileInterface](#).

## 6.32.3.5. update\_logic()

```
void creature::KillerRobot::update_logic (
    World & world,
    float deltaTime ) [override], [virtual]
```

Az entitás frissítési logikája itt van definiálva. Például a halász vizet keres, ha odaért akkor halászik.

## Paraméterek

<i>world</i>	A világ, amibe frissíti magát az entitás.
<i>deltaTime</i>	Az előző frissítés óta eltelt idő.

Megvalósítja a következőket: [creature::Living](#).

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

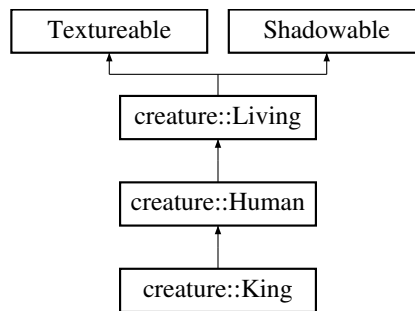
- [src/creatures/hostiles/KillerRobot.hpp](#)
- [src/creatures/hostiles/KillerRobot.cpp](#)

## 6.33. creature::King osztályreferencia

A király szakmájú ember osztály leírása.

```
#include <King.hpp>
```

A creature::King osztály származási diagramja:



## Publikus tagfüggvények

- **King** (int x, int y, ENTITY\_GENDER gender\_modifier)  
*Inicializál egy királyt egy pontos x és y koordinátára és beállítja az attribútumait.*
- void **update\_logic** (World &world, float deltaTime) override  
*Az entitás frissítési logikája itt van definiálva. Például a halász vizet keres, ha odaért akkor halászik.*
- **~King** ()  
*A király destruktora.*

## További örökölt tagok

### 6.33.1. Részletes leírás

A király szakmájú ember osztály leírása.

Ez a szakmájú ember nem sokat csinál. A király szakma csak indikálja, hogy ő alapította a várost. Alapítást után csak örülten bolyong a világba.

### 6.33.2. Konstruktorok és destruktorok dokumentációja

#### 6.33.2.1. King()

```

creature::King::King (
    int x,
    int y,
    ENTITY_GENDER gender_modifier )
  
```

Inicializál egy királyt egy pontos x és y koordinátára és beállítja az attribútumait.

#### Paraméterek

<i>x</i>	Az x koordináta.
<i>y</i>	Az y koordináta.
<i>gender_modifier</i>	A király neve.



### 6.33.2.2. ~King()

```
creature::King::~~King ( )
```

A király destruktora.

## 6.33.3. Tagfüggvények dokumentációja

### 6.33.3.1. update\_logic()

```
void creature::King::update_logic (
    World & world,
    float deltaTime ) [override], [virtual]
```

Az entitás frissítési logikája itt van definiálva. Például a halász vizet keres, ha odaért akkor halászik.

Paraméterek

<i>world</i>	A világ, amibe frissíti magát az entitás.
<i>deltaTime</i>	Az előző frissítés óta eltelt idő.

Újrimplementált ősök: [creature::Human](#).

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

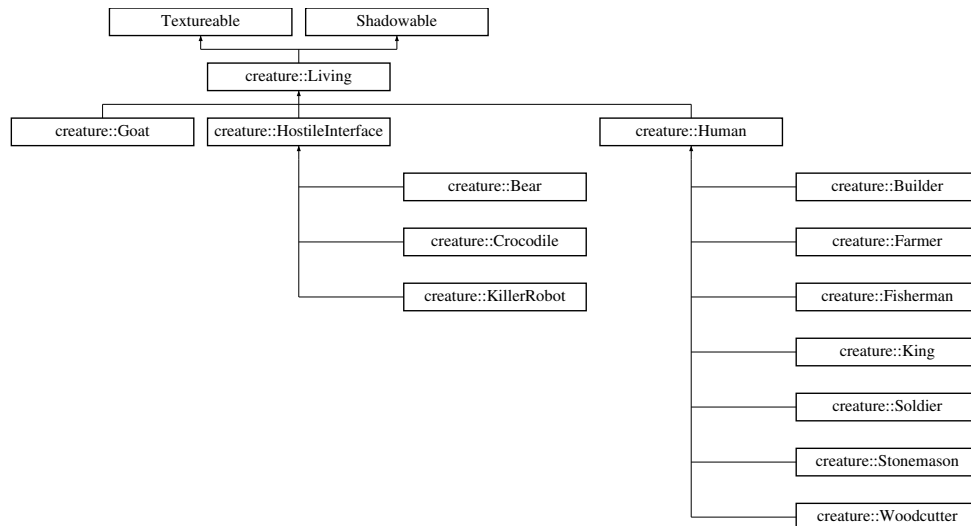
- [src/creatures/humans/King.hpp](#)
- [src/creatures/humans/King.cpp](#)

## 6.34. creature::Living osztályreferencia

Az élő entitások interface leírása.

```
#include <Living.hpp>
```

A creature::Living osztály származási diagramja:



## Publikus tagfüggvények

- `ENTITY_GENDER get_gender ()`  
Visszaadja az entitás nemét.
- `LIVINGSTATE get_state ()`  
Visszaadja az entitás belső állapotát.
- `void damage (Living *dam_by, int amm)`  
Ez a függvény jelzi, hogy megsebezték az entitást és azt, hogy ki sebezte meg.
- `void set_health (int amm)`  
Beállítja az entitás életét ez bizonyos értékre.
- `void init_spritesheet_data (int maxframes, double animspeed)`  
Beállítja az entitásnak azt, hogy hány képkockás animációja legyen és az milyen gyors legyen.
- `void look_left ()`  
Balra nézeti az entitást.
- `void look_right ()`  
Jobbra nézeti az entitást.
- `void set_state (LIVINGSTATE newstate)`  
Beállítja az entitás belső állapotát egy új értékre.
- `virtual ENTITY_TYPE get_type () const =0`  
Visszaadja az entitás belső szimbólumát. Ez abba segít, hogy a vadállatok például csak embert támadjanak, Az emberek csak vadállatot.
- `bool setTexture (const std::string &filename) override`  
Beállít egy fájl elérési útból egy textúrát. Megvalósítástól függően esetleg 2-t.
- `bool setTheShadow (const std::string &filename)`  
Beállítja az entitás árnyék textúráját.
- `void setPosition (double x, double y) override`  
Beállítja, hogy hova kell rajzolni a textúrázható leszármozottat.
- `void update_spritesheet (float deltaTime)`  
Frissíti az entitás animációját az idő függvényében.
- `void draw (sf::RenderWindow &window) override`  
Kirájzolja az objektumot.
- `void set_idle_texture (std::string new_str)`  
Frissíti az entitás "semmit nem csináló" textúráját.
- `void set_attack_texture (std::string new_str)`

- *Frissíti az entitás "támadó" textúráját.*
- void `set_walk_texture` (std::string new\_str)
- *Frissíti az entitás "sétáló" textúráját.*
- void `set_run_texture` (std::string new\_str)
- *Frissíti az entitás "futó" textúráját.*
- void `set_death_texture` (std::string new\_str)
- *Frissíti az entitás "elhalálozó" textúráját.*
- bool `needs_drawn` ()
- *Megnézi, hogy a felhasználó látja-e az entitást.*
- int `get_width` () const
- *Visszaadja az entitás vastagságát.*
- virtual `Living * check_aggroed` () const
- *Visszaadja, hogy kire "ideges az entitás". Ez lehet az, hogy ki ütötte meg vagy hogy kit akar megenni.*
- virtual void `retarget` (`Living *new_target`)
- *Felidegesíti az entitást a kapott entitásra.*
- virtual void `die` ()=0
- *Mi történjen, ha meghal az entitás.*
- virtual void `update_logic` (`World &world`, float deltaTime)=0
- *Az entitás frissítési logikája itt van definiálva. Például a halász vizet keres, ha odaért akkor halászik.*
- virtual void `draw_logic` (`sf::RenderWindow &window`, float deltaTime, int offx, int offy)=0
- *Az entitás kirajzolási logikája, például az ember az árnyékát és képét rajzolja ki, de a robot csak a képét.*
- void `shadow_logic` (`sf::RenderWindow &window`, float elapsed\_time, int offx, int offy)
- *Az entitás árnyékolási logikája, itt állítódik be az árnyék fázisa.*
- void `apply_age` ()
- *Megnézi, hogy hány éves az entitás, ha már meg kell halnia akkor meghal.*
- virtual `~Living` ()
- *Virtuális destruktork.*

## Publikus attribútumok

- std::string `save_name` ="?"
- *Milyen szimbóluma legyen a mentés fájlba.*
- float `death_timer` =0.1f
- *A halál animáció hátralévő idejét méri. Ha ez 0 akkor az entitás felszabadul és megsemmisül.*
- double `posx`
- *Az entitás X pozíciója.*
- double `posy`
- *Az entitás Y pozíciója.*

## Védett attribútumok

- float `max_age`
- *Az entitás maximum életkora. Ha ezt eléri meghal.*
- `Living * damaged_by` =nullptr
- *Arra az entitásra pointer, ami utoljára megsebezte.*
- `ENTITY_GENDER` gender
- *Az entitás neve.*
- `LIVINGSTATE` state
- *Az entitás belső állapota.*

- **FACING facing**

*Jobbra vagy balra néz az entitás.*

- **int health**

*Még mennyi élete maradt az entitásnak. Ha ez  $\leq 0$  akkor meghal az entitás.*

- **float hit\_timer =0.0f**

*Ha megütik az entitást, akkor egy piros szín effektet kap, ez a változó mutatja, hogy még meddig legyen rajta ez az effekt.*

- **float inner\_timer**

*Az entitás születése óta eltelt idő.*

- **float speed**

*Milyen gyorsan sétál az entitás (1 delta idő alatt).*

- **float run\_speed\_modifier**

*Milyen gyorsan fut az entitás (1 delta idő alatt).*

- **const int MAX\_CREATURE\_SIZE =64**

*Mekkora a maximum entitás, amit még a kamera culling nélkül kirajzol, akkor is ha annak a középpontja nincs benne a látótérbe.*

### 6.34.1. Részletes leírás

Az élő entitások interface leírása.

Ebbe minden deklarálva van, amire egy entitásnak szüksége van. Tud fordulni, animált képet rajzolni, futni, mozogni, támadni, meghalni, "csinálni a dolgát". Eltárolja, hogy melyik entitás sebezte meg utoljára.

### 6.34.2. Konstruktorok és destruktorok dokumentációja

#### 6.34.2.1. ~Living()

```
creature::Living::~~Living ( ) [virtual]
```

Virtuális destruktor.

### 6.34.3. Tagfüggvények dokumentációja

#### 6.34.3.1. apply\_age()

```
void creature::Living::apply_age ( )
```

Megnézi, hogy hány éves az entitás, ha már meg kell halnia akkor meghal.

### 6.34.3.2. check\_aggroed()

```
Living * creature::Living::check_aggroed ( ) const [virtual]
```

Visszaadja, hogy kire "ideges az entitás". Ez lehet az, hogy ki ütötte meg vagy hogy kit akar megenni.

#### Visszatérési érték

Az entitás, akire ideges. Nullpointer, ha nincs ilyen entitás.

Újrimplementáló leszármazottak: [creature::HostileInterface](#).

### 6.34.3.3. damage()

```
void creature::Living::damage (
    Living * dam_by,
    int amm )
```

Ez a függvény jelzi, hogy megsebeztek az entitást és azt, hogy ki sebezte meg.

#### Paraméterek

<i>dam_by</i>	Az entitás, aki megsebezte.
<i>amm</i>	Mennyi sebzést kapott. Ezt levonja a metódus az entitás életéből.

### 6.34.3.4. die()

```
virtual void creature::Living::die ( ) [pure virtual]
```

Mi történjen, ha meghal az entitás.

Megvalósítják a következők: [creature::Human](#), [creature::KillerRobot](#), [creature::Crocodile](#), [creature::Bear](#) és [creature::Goat](#).

### 6.34.3.5. draw()

```
void creature::Living::draw (
    sf::RenderWindow & window ) [override], [virtual]
```

Kirajzolja az objektumot.

## Paraméterek

<i>window</i>	Ahova ki kell rajzolni a textúrázható leszármozottat.
---------------	---

Megvalósítja a következőket: [Textureable](#).

**6.34.3.6. draw\_logic()**

```
virtual void creature::Living::draw_logic (
    sf::RenderWindow & window,
    float deltaTime,
    int offx,
    int offy ) [pure virtual]
```

Az entitás kirajzolás logikája, például az ember az árnyékát és képét rajzolja ki, de a robot csak a képét.

## Paraméterek

<i>window</i>	Az ablak, ahova rajzolni kell.
<i>deltaTime</i>	Az előző frissítés óta eltelt idő.
<i>offx</i>	A kamera X eltolása.
<i>offy</i>	A kamera Y eltolása.

Megvalósítják a következők: [creature::Human](#), [creature::KillerRobot](#), [creature::Crocodile](#), [creature::Bear](#) és [creature::Goat](#).

**6.34.3.7. get\_gender()**

```
ENTITY_GENDER creature::Living::get_gender ( )
```

Visszaadja az entitás nemét.

## Visszatérési érték

Az entitás neve.

**6.34.3.8. get\_state()**

```
LIVINGSTATE creature::Living::get_state ( )
```

Visszaadja az entitás belső állapotát.

## Visszatérési érték

Az entitás belső állapota.

#### 6.34.3.9. get\_type()

```
virtual ENTITY_TYPE creature::Living::get_type ( ) const [pure virtual]
```

Visszaadja az entitás belső szimbólumát. Ez abba segít, hogy a vadállatok például csak embert támadjanak, Az emberek csak vadállatot.

##### Visszatérési érték

A belső szimbólum.

Megvalósítják a következők: [creature::Human](#), [creature::KillerRobot](#), [creature::Crocodile](#), [creature::Bear](#) és [creature::Goat](#).

#### 6.34.3.10. get\_width()

```
int creature::Living::get_width ( ) const
```

Visszaadja az entitás vastagságát.

##### Visszatérési érték

Az entitás vastagsága.

#### 6.34.3.11. init\_spritesheet\_data()

```
void creature::Living::init_spritesheet_data (
    int maxframes,
    double animspeed )
```

Beállítja az entitásnak azt, hogy hány képkockás animációja legyen és az milyen gyors legyen.

##### Paraméterek

<i>maxframes</i>	A képkockák száma.
<i>animspeed</i>	Az animáció gyorsasága.

#### 6.34.3.12. look\_left()

```
void creature::Living::look_left ( )
```

Balra nézeti az entitást.

#### 6.34.3.13. look\_right()

```
void creature::Living::look_right ( )
```

Jobbra nézeti az entitást.

#### 6.34.3.14. needs\_drawn()

```
bool creature::Living::needs_drawn ( )
```

Megnézi, hogy a felhasználó látja-e az entitást.

##### Visszatérési érték

Benne van-e a látótérbe.

#### 6.34.3.15. retarget()

```
void creature::Living::retarget (
    Living * new_target ) [virtual]
```

Felidegesíti az entitást a kapott entitásra.

##### Paraméterek

<i>new_target</i>	Az entitás, akire dühösnek kell lennie.
-------------------	---

Újrimplementáló leszármazottak: [creature::HostileInterface](#).

#### 6.34.3.16. set\_attack\_texture()

```
void creature::Living::set_attack_texture (
    std::string new_str )
```

Frissíti az entitás "támadó" textúráját.

##### Paraméterek

<i>new_str</i>	Az új textúra elérési útvonala.
----------------	---------------------------------



**6.34.3.17. set\_death\_texture()**

```
void creature::Living::set_death_texture (
    std::string new_str )
```

Frissíti az entitás "elhalálozó" textúráját.

**Paraméterek**

<i>new_str</i>	Az új textúra elérési útvonala.
----------------	---------------------------------

**6.34.3.18. set\_health()**

```
void creature::Living::set_health (
    int amm )
```

Beállítja az entitás életét ez bizonyos értékre.

**Paraméterek**

<i>amm</i>	Az új életpont szám.
------------	----------------------

**6.34.3.19. set\_idle\_texture()**

```
void creature::Living::set_idle_texture (
    std::string new_str )
```

Frissíti az entitás "semmit nem csináló" textúráját.

**Paraméterek**

<i>new_str</i>	Az új textúra elérési útvonala.
----------------	---------------------------------

**6.34.3.20. set\_run\_texture()**

```
void creature::Living::set_run_texture (
    std::string new_str )
```

Frissíti az entitás "futó" textúráját.

## Paraméterek

<i>new_str</i>	Az új textúra elérési útvonala.
----------------	---------------------------------

**6.34.3.21. set\_state()**

```
void creature::Living::set_state (
    LIVINGSTATE newstate )
```

Beállítja az entitás belső állapotát egy új értékre.

## Paraméterek

<i>newstate</i>	Az új belső állapot.
-----------------	----------------------

**6.34.3.22. set\_walk\_texture()**

```
void creature::Living::set_walk_texture (
    std::string new_str )
```

Frissíti az entitás "sétáló" textúráját.

## Paraméterek

<i>new_str</i>	Az új textúra elérési útvonala.
----------------	---------------------------------

**6.34.3.23. setPosition()**

```
void creature::Living::setPosition (
    double x,
    double y ) [override], [virtual]
```

Beállítja, hogy hova kell rajzolni a textúrázható leszármazottat.

## Paraméterek

<i>x</i>	Az x koordináta.
<i>y</i>	Az y koordináta.

Megvalósítja a következőket: [Textureable](#).

#### 6.34.3.24. setTexture()

```
bool creature::Living::setTexture (
    const std::string & filename ) [override], [virtual]
```

Beállít egy fájl elérési útból egy textúrát. Megvalósítástól függően esetleg 2-t.

##### Paraméterek

<i>filename</i>	A textúra elérési útja.
-----------------	-------------------------

##### Visszatérési érték

Sikeres volt-e a textúra beállítása.

Megvalósítja a következőket: [Textureable](#).

#### 6.34.3.25. setTheShadow()

```
bool creature::Living::setTheShadow (
    const std::string & filename )
```

Beállítja az entitás árnyék textúráját.

##### Paraméterek

<i>filename</i>	Az árnyék textúra elérési útvonala.
-----------------	-------------------------------------

#### 6.34.3.26. shadow\_logic()

```
void creature::Living::shadow_logic (
    sf::RenderWindow & window,
    float elapsed_time,
    int offx,
    int offy )
```

Az entitás árnyékolás logikája, itt állítódik be az árnyék fázisa.

##### Paraméterek

<i>window</i>	Az ablak, ahova rajzolni kell.
<i>elapsed_time</i>	A szimuláció kezdete óta eltelt idő.
<i>offx</i>	A kamera X eltolása.
<i>offy</i>	A kamera Y eltolása.

#### 6.34.3.27. update\_logic()

```
virtual void creature::Living::update_logic (
    World & world,
    float deltaTime ) [pure virtual]
```

Az entitás frissítési logikája itt van definiálva. Például a halász vizet keres, ha odaért akkor halászik.

##### Paraméterek

<i>world</i>	A világ, amibe frissíti magát az entitás.
<i>deltaTime</i>	Az előző frissítés óta eltelt idő.

Megvalósítják a következők: `creature::Woodcutter`, `creature::Stonemason`, `creature::Soldier`, `creature::King`, `creature::Human`, `creature::Fisherman`, `creature::Farmer`, `creature::Builder`, `creature::AnglerMiner`, `creature::KillerRobot`, `creature::Crocodile`, `creature::Bear` és `creature::Goat`.

#### 6.34.3.28. update\_spritesheet()

```
void creature::Living::update_spritesheet (
    float deltaTime )
```

Frissíti az entitás animációját az idő függvényében.

##### Paraméterek

<i>deltaTime</i>	Az előző frissítés óta eltelt idő.
------------------	------------------------------------

### 6.34.4. Adattagok dokumentációja

#### 6.34.4.1. damaged\_by

```
Living* creature::Living::damaged_by =nullptr [protected]
```

Arra az entításra pointer, ami utoljára megsebezte.

#### 6.34.4.2. death\_timer

```
float creature::Living::death_timer =0.1f
```

A halál animáció hátralévő idejét méri. Ha ez 0 akkor az entitás felszabadul és megsemmisül.

#### 6.34.4.3. facing

```
FACING creature::Living::facing [protected]
```

Jobbra vagy balra néz az entitás.

#### 6.34.4.4. gender

```
ENTITY_GENDER creature::Living::gender [protected]
```

Az entitás neve.

#### 6.34.4.5. health

```
int creature::Living::health [protected]
```

Még mennyi élete maradt az entitásnak. Ha ez  $\leq 0$  akkor meghal az entitás.

#### 6.34.4.6. hit\_timer

```
float creature::Living::hit_timer =0.0f [protected]
```

Ha megütik az entitást, akkor egy piros szín effektet kap, ez a változó mutatja, hogy még meddig legyen rajta ez az effekt.

#### 6.34.4.7. inner\_timer

```
float creature::Living::inner_timer [protected]
```

Az entitás születése óta eltelt idő.

#### 6.34.4.8. max\_age

```
float creature::Living::max_age [protected]
```

Az entitás maximum életkora. Ha ezt eléri meghal.

#### 6.34.4.9. MAX\_CREATURE\_SIZE

```
const int creature::Living::MAX_CREATURE_SIZE =64 [protected]
```

Mekkora a maximum entitás, amit még a kamera culling nélkül kirajzol, akkor is ha annak a középpontja nincs benne a látótérbe.

#### 6.34.4.10. posX

```
double creature::Living::posx
```

Az entitás X pozíciója.

#### 6.34.4.11. posY

```
double creature::Living::posy
```

Az entitás Y pozíciója.

#### 6.34.4.12. run\_speed\_modifier

```
float creature::Living::run_speed_modifier [protected]
```

Milyen gyorsan fut az entitás (1 delta idő alatt).

#### 6.34.4.13. save\_name

```
std::string creature::Living::save_name = "?"
```

Milyen szimbóluma legyen a mentés fájlba.

#### 6.34.4.14. speed

```
float creature::Living::speed [protected]
```

Milyen gyorsan sétál az entitás (1 delta idő alatt).

#### 6.34.4.15. state

```
LIVINGSTATE creature::Living::state [protected]
```

Az entitás belső állapota.

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- src/creatures/Living.hpp
- src/creatures/Living.cpp

## 6.35. sf::Mouse osztályreferencia

```
#include <fake_sfml.hpp>
```

### Publikus típusok

- enum [MouseDowntype](#) : char { [Right](#) , [Left](#) }

### Statikus publikus tagfüggvények

- static bool [isButtonPressed](#) ([MouseDowntype](#) key)
- static [Vector2i](#) [getPosition](#) ([RenderWindow](#) &window)
- static void [simulate\\_key\\_press](#) ([MouseDowntype](#) key)
- static void [simulate\\_key\\_release](#) ([MouseDowntype](#) key)

### 6.35.1. Enumeráció-tagok dokumentációja

#### 6.35.1.1. MouseDowntype

```
enum sf::Mouse::MouseDowntype : char
```

##### Enumeráció-értékek

Right	
Left	

## 6.35.2. Tagfüggvények dokumentációja

### 6.35.2.1. getPosition()

```
Vector2i sf::Mouse::getPosition (
    RenderWindow & window ) [static]
```

### 6.35.2.2. isButtonPressed()

```
bool sf::Mouse::isButtonPressed (
    MouseButtonType key ) [static]
```

### 6.35.2.3. simulate\_key\_press()

```
void sf::Mouse::simulate_key_press (
    MouseButtonType key ) [static]
```

### 6.35.2.4. simulate\_key\_release()

```
void sf::Mouse::simulate_key_release (
    MouseButtonType key ) [static]
```

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- [src/fake\\_sfml/fake\\_sfml.hpp](#)
- [src/fake\\_sfml/fake\\_sfml.cpp](#)

## 6.36. sf::Music osztályreferencia

```
#include <fake_sfml.hpp>
```

### Publikus tagfüggvények

- [Music](#) ()
- void [setLoop](#) (bool newval)
- void [setVolume](#) (double new\_db)
- [SoundSource::SoundSourceType](#) [getStatus](#) ()
- void [play](#) ()
- void [stop](#) ()
- bool [openFromFile](#) (const std::string &filepath)



## 6.36.1. Konstruktorkok és destruktorkok dokumentációja

### 6.36.1.1. Music()

```
sf::Music::Music ( )
```

## 6.36.2. Tagfüggvények dokumentációja

### 6.36.2.1. getStatus()

```
SoundSource::SoundSourceType sf::Music::getStatus ( )
```

### 6.36.2.2. openFromFile()

```
bool sf::Music::openFromFile (
    const std::string & filepath )
```

### 6.36.2.3. play()

```
void sf::Music::play ( )
```

### 6.36.2.4. setLoop()

```
void sf::Music::setLoop (
    bool newval )
```

### 6.36.2.5. setVolume()

```
void sf::Music::setVolume (
    double new_db )
```

### 6.36.2.6. stop()

```
void sf::Music::stop ( )
```

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

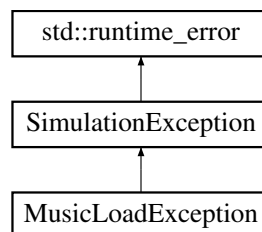
- [src/fake\\_sfml/fake\\_sfml.hpp](#)
- [src/fake\\_sfml/fake\\_sfml.cpp](#)

## 6.37. MusicLoadException osztályreferencia

Akkor kell dobni, ha egy zene hibásan lett lejátszva vagy hibás a beolvasott zene.

```
#include <MusicLoadException.hpp>
```

A MusicLoadException osztály származási diagramja:



### Publikus tagfüggvények

- [MusicLoadException](#) (const std::string &msg)

### 6.37.1. Részletes leírás

Akkor kell dobni, ha egy zene hibásan lett lejátszva vagy hibás a beolvasott zene.

### 6.37.2. Konstruktorkok és destruktorkok dokumentációja

#### 6.37.2.1. MusicLoadException()

```
MusicLoadException::MusicLoadException (
    const std::string & msg ) [inline], [explicit]
```

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- [src/exceptions/MusicLoadException.hpp](#)

## 6.38. MusicPlayer osztályreferencia

A zene játszó osztály leírása.

```
#include <MusicPlayer.hpp>
```

### Publikus tagfüggvények

- `MusicPlayer ()`  
*Alap konstruktor, beállítja a toggled és load\_music értéket hamisra.*
- void `load_music (const std::string &filename)`  
*Betölti az elérési útvonal végén lévő fájlból a zenét.*
- void `toggle_music ()`  
*Ki-be kapcsolja a zenét.*
- void `set_volume (float vol)`  
*Beállítja a hangerőt X decibelre.*
- `~MusicPlayer ()`  
*Destruktor, ami megállítja a zenét. Ez kiküszöböli a sound blasting-et, ami e-nélkül lenne.*

### 6.38.1. Részletes leírás

A zene játszó osztály leírása.

Képes zenét betölteni, ki-be kapcsolni és lejátszani megadott hangerőn.

### 6.38.2. Konstruktorok és destruktorok dokumentációja

#### 6.38.2.1. MusicPlayer()

```
MusicPlayer::MusicPlayer ( )
```

Alap konstruktor, beállítja a toggled és load\_music értéket hamisra.

#### 6.38.2.2. ~MusicPlayer()

```
MusicPlayer::~~MusicPlayer ( )
```

Destruktor, ami megállítja a zenét. Ez kiküszöböli a sound blasting-et, ami e-nélkül lenne.

### 6.38.3. Tagfüggvények dokumentációja

#### 6.38.3.1. load\_music()

```
void MusicPlayer::load_music (
    const std::string & filename )
```

Betölti az elérési útvonal végén lévő fájlból a zenét.

## Paraméterek

<i>filename</i>	A fájl elérési útvonala.
-----------------	--------------------------

**6.38.3.2. set\_volume()**

```
void MusicPlayer::set_volume (
    float vol )
```

Beállítja a hangerőt X decibelre.

## Paraméterek

<i>vol</i>	Mekkora decibel.
------------	------------------

**6.38.3.3. toggle\_music()**

```
void MusicPlayer::toggle_music ( )
```

Ki-be kapcsolja a zenét.

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- [src/MusicPlayer.hpp](#)
- [src/MusicPlayer.cpp](#)

**6.39. gtest\_lite::ostreamRedir osztályreferencia**

```
#include <gtest_lite.h>
```

**Publikus tagfüggvények**

- [ostreamRedir](#) (std::ostream &src, std::ostream &dst)
- [~ostreamRedir](#) ()

**6.39.1. Részletes leírás**

Segédsablon ostream átirányításához A destruktorkor visszaállít

## 6.39.2. Konstruktorok és destruktorok dokumentációja

### 6.39.2.1. ostreamRedir()

```
gtest_lite::ostreamRedir::ostreamRedir (
    std::ostream & src,
    std::ostream & dst ) [inline]
```

### 6.39.2.2. ~ostreamRedir()

```
gtest_lite::ostreamRedir::~ostreamRedir ( ) [inline]
```

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- [src/external/gtest\\_lite.h](#)

## 6.40. PostProcessor osztályreferencia

A grafikus szépítő osztály leírása.

```
#include <PostProcessor.hpp>
```

### Publikus tagfüggvények

- [PostProcessor](#) ()  
*A konstruktor, a használt textúrák betöltése itt történik.*
- void [toggle\\_vignette](#) (bool newval)  
*Ki-be kapcsolja a vignettát.*
- void [toggle\\_noise](#) (bool newval)  
*Ki-be kapcsolja a zajt.*
- void [toggle\\_chromatic\\_aberration](#) (bool newval)  
*Ki-be kapcsolja a Chromatic aberration-t.*
- bool [setTextureFor](#) (sf::Sprite &what, const std::string &filename)  
*Beállít egy képnek egy új textúrát.*
- void [setRenderSize](#) (double x, double y)  
*Beállítja azt a négyzetet (0,0) (x,y)-ig, ahol a szépítő osztály dolgozni fog.*
- void [draw](#) (sf::RenderWindow &window)  
*Kirajzolódik az osztály.*
- void [setColorOverlay](#) (int r, int g, int b, int a)  
*Beállítja az új szín réteget.*

### 6.40.1. Részletes leírás

A grafikus szépítő osztály leírása.

Különböző szépítések beállíthatóak: Zaj, Szín, Chromatic aberration, Vignette.

### 6.40.2. Konstruktork és destruktorok dokumentációja

#### 6.40.2.1. PostProcessor()

```
PostProcessor::PostProcessor ( )
```

A konstruktor, a használt textúrák betöltése itt történik.

### 6.40.3. Tagfüggvények dokumentációja

#### 6.40.3.1. draw()

```
void PostProcessor::draw (
    sf::RenderWindow & window )
```

Kirajzolódik az osztály.

##### Paraméterek

<i>window</i>	Az ablak, amire rajzolódik.
---------------	-----------------------------

#### 6.40.3.2. setColorOverlay()

```
void PostProcessor::setColorOverlay (
    int r,
    int g,
    int b,
    int a )
```

Beállítja az új szín réteget.

##### Paraméterek

<i>r</i>	Piros komponens.
----------	------------------

## Paraméterek

<i>g</i>	Zöld komponens.
<i>b</i>	Kék komponens.
<i>a</i>	Alfa komponens.

**6.40.3.3. setRenderSize()**

```
void PostProcessor::setRenderSize (
    double x,
    double y )
```

Beállítja azt a négyzetet (0,0) (x,y)-ig, ahol a szépítő osztály dolgozni fog.

## Paraméterek

<i>x</i>	A szélesség.
<i>y</i>	A magasság.

**6.40.3.4. setTextureFor()**

```
bool PostProcessor::setTextureFor (
    sf::Sprite & what,
    const std::string & filename )
```

Beállít egy képnek egy új textúrát.

## Paraméterek

<i>what</i>	Azt a képet, amit be kell állítani.
<i>filename</i>	Az új textúra elérési útvonala.

## Visszatérési érték

Sikerült-e.

**6.40.3.5. toggle\_chromatic\_aberration()**

```
void PostProcessor::toggle_chromatic_aberration (
    bool newval )
```

Ki-be kapcsolja a Chromatic aberration-t.

## Paraméterek

<i>newval</i>	Ha igaz, akkor ezen túl ki lesz rajzolva, különben nem lesz.
---------------	--

**6.40.3.6. toggle\_noise()**

```
void PostProcessor::toggle_noise (
    bool newval )
```

Ki-be kapcsolja a zajt.

## Paraméterek

<i>newval</i>	Ha igaz, akkor ezen túl ki lesz rajzolva, különben nem lesz.
---------------	--

**6.40.3.7. toggle\_vignette()**

```
void PostProcessor::toggle_vignette (
    bool newval )
```

Ki-be kapcsolja a vignettát.

## Paraméterek

<i>newval</i>	Ha igaz, akkor ezen túl ki lesz rajzolva, különben nem lesz.
---------------	--

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

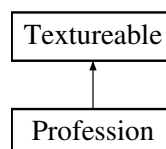
- [src/PostProcessor.hpp](#)
- [src/PostProcessor.cpp](#)

**6.41. Profession osztályreferencia**

A szakma osztály leírása.

```
#include <Profession.hpp>
```

A Profession osztály származási diagramja:





## Publikus tagfüggvények

- `Profession` (`const std::string &intype`)  
*A konstruktor, ami egy szimbólum alapján betölti az ikon képet.*
- `bool setTexture` (`const std::string &filename`) `override`  
*Beállít egy fájl elérési útból egy textúrát. Megvalósítástól függően esetleg 2-t.*
- `void setPosition` (`double x, double y`) `override`  
*Beállítja, hogy hova kell rajzolni a textúrázható leszármazottat.*
- `void draw` (`sf::RenderWindow &window`) `override`  
*Kirajzolja az objektumot.*
- `void load_profession` (`const std::string &new_profession`)  
*Egy szimbólum alapján betölti az ikon képet.*
- `std::string to_string` ()  
*Egy getter a szakma szimbólumához.*

### 6.41.1. Részletes leírás

A szakma osztály leírása.

Tárolja a szakma ikonját és szimbólumát is.

### 6.41.2. Konstruktorok és destruktorok dokumentációja

#### 6.41.2.1. Profession()

```
Profession::Profession (
    const std::string & intype )
```

A konstruktor, ami egy szimbólum alapján betölti az ikon képet.

Paraméterek

<code>intype</code>	A szakma szimbólum.
---------------------	---------------------

### 6.41.3. Tagfüggvények dokumentációja

#### 6.41.3.1. draw()

```
void Profession::draw (
    sf::RenderWindow & window ) [override], [virtual]
```

Kirajzolja az objektumot.

## Paraméterek

<i>window</i>	Ahova ki kell rajzolni a textúrázható leszármazottat.
---------------	---

Megvalósítja a következőket: [Textureable](#).

**6.41.3.2. load\_profession()**

```
void Profession::load_profession (
    const std::string & new_profession )
```

Egy szimbólum alapján betölti az ikon képet.

## Paraméterek

<i>intype</i>	A szakma szimbólum.
---------------	---------------------

**6.41.3.3. setPosition()**

```
void Profession::setPosition (
    double x,
    double y ) [override], [virtual]
```

Beállítja, hogy hova kell rajzolni a textúrázható leszármazottat.

## Paraméterek

<i>x</i>	Az x koordináta.
<i>y</i>	Az y koordináta.

Megvalósítja a következőket: [Textureable](#).

**6.41.3.4. setTexture()**

```
bool Profession::setTexture (
    const std::string & filename ) [override], [virtual]
```

Beállít egy fájl elérési útból egy textúrát. Megvalósítástól függően esetleg 2-t.

## Paraméterek

<i>filename</i>	A textúra elérési útja.
-----------------	-------------------------

**Visszatérési érték**

Sikeres volt-e a textúra beállítása.

Megvalósítja a következőket: [Textureable](#).

**6.41.3.5. to\_string()**

```
std::string Profession::to_string ( )
```

Egy getter a szakma szimbólumához.

**Visszatérési érték**

A szakma szimbóluma szöveggént.

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- [src/Profession.hpp](#)
- [src/Profession.cpp](#)

## 6.42. RandomGenerator osztályreferencia

Egy korszerűbb és konfigurálhatóbb véletlen szám generátor osztály.

```
#include <Random_Gen.hpp>
```

**Publikus tagfüggvények**

- [RandomGenerator](#) (const [RandomGenerator](#) &)=delete  
*A singleton pattern miatt törölve.*
- [RandomGenerator](#) & operator= (const [RandomGenerator](#) &)=delete  
*A singleton pattern miatt törölve.*
- int [get\\_random\\_int](#) (int max)  
*0 és a max-1 számok között visszaad egy véletlen számot.*

**Statikus publikus tagfüggvények**

- static [RandomGenerator](#) & [get\\_instance](#) ()  
*Visszaad egy referenciát erre az osztály-ra.*

**6.42.1. Részletes leírás**

Egy korszerűbb és konfigurálhatóbb véletlen szám generátor osztály.

Singleton pattern-t használ.

## 6.42.2. Konstruktorkok és destruktorkok dokumentációja

### 6.42.2.1. RandomGenerator()

```
RandomGenerator::RandomGenerator (
    const RandomGenerator & ) [delete]
```

A singleton pattern miatt törölve.

## 6.42.3. Tagfüggvények dokumentációja

### 6.42.3.1. get\_instance()

```
RandomGenerator & RandomGenerator::get_instance ( ) [static]
```

Visszaad egy referenciát erre az osztály-ra.

#### Visszatérési érték

A singleton-hoz egy referencia.

### 6.42.3.2. get\_random\_int()

```
int RandomGenerator::get_random_int (
    int max )
```

0 és a max-1 számok között visszaad egy véletlen számot.

#### Paraméterek

<i>max</i>	A maximum érték, aminél már csak kisebb számokat ad vissza.
------------	---

#### Visszatérési érték

A határ mérete.

### 6.42.3.3. operator=()

```
RandomGenerator& RandomGenerator::operator= (
    const RandomGenerator & ) [delete]
```

A singleton pattern miatt törölve.

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

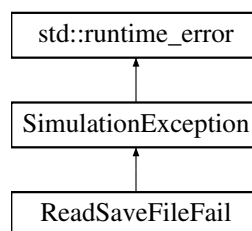
- [src/Random\\_Gen.hpp](#)
- [src/Random\\_Gen.cpp](#)

## 6.43. ReadSaveFileFail osztályreferencia

Akkor kell dobni, ha egy IO mentés, importálás vagy törlés hibás.

```
#include <FileExceptions.hpp>
```

A ReadSaveFileFail osztály származási diagramja:



### Publikus tagfüggvények

- [ReadSaveFileFail](#) (const std::string &msg)

### 6.43.1. Részletes leírás

Akkor kell dobni, ha egy IO mentés, importálás vagy törlés hibás.

### 6.43.2. Konstruktorkok és destruktorkok dokumentációja

#### 6.43.2.1. ReadSaveFileFail()

```
ReadSaveFileFail::ReadSaveFileFail (
    const std::string & msg ) [inline], [explicit]
```

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- [src/exceptions/FileExceptions.hpp](#)

## 6.44. sf::RectangleShape osztályreferencia

```
#include <fake_sfml.hpp>
```

### Publikus tagfüggvények

- void [setFillColor](#) ([Color](#) new\_color)
- void [setSize](#) ([Vector2f](#) newsize)
- void [setPosition](#) ([Vector2f](#) newsize)

### Publikus attribútumok

- [Vector2f](#) position

### 6.44.1. Tagfüggvények dokumentációja

#### 6.44.1.1. setFillColor()

```
void sf::RectangleShape::setFillColor (
    Color new_color )
```

#### 6.44.1.2. setPosition()

```
void sf::RectangleShape::setPosition (
    Vector2f newsize )
```

#### 6.44.1.3. setSize()

```
void sf::RectangleShape::setSize (
    Vector2f newsize )
```

### 6.44.2. Adattagok dokumentációja

#### 6.44.2.1. position

```
Vector2f sf::RectangleShape::position
```

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- [src/fake\\_sfml/fake\\_sfml.hpp](#)
- [src/fake\\_sfml/fake\\_sfml.cpp](#)

## 6.45. sf::RenderStates osztályreferencia

```
#include <fake_sfml.hpp>
```

### Publikus tagfüggvények

- [RenderStates](#) ()
- void [setBlendMode](#) ([BlendMode](#) mode)
- void [setTransform](#) (const float newTransform[4][4])

### Publikus attribútumok

- [Transform](#) transform
- [BlendMode](#) blendMode

### 6.45.1. Konstruktorkok és destruktorkok dokumentációja

#### 6.45.1.1. RenderStates()

```
sf::RenderStates::RenderStates ( )
```

### 6.45.2. Tagfüggvények dokumentációja

#### 6.45.2.1. setBlendMode()

```
void sf::RenderStates::setBlendMode (
    BlendMode mode )
```

### 6.45.2.2. setTransform()

```
void sf::RenderStates::setTransform (
    const float newTransform[4][4] )
```

## 6.45.3. Adattagok dokumentációja

### 6.45.3.1. blendMode

[BlendMode](#) sf::RenderStates::blendMode

### 6.45.3.2. transform

[Transform](#) sf::RenderStates::transform

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- [src/fake\\_sfml/fake\\_sfml.hpp](#)
- [src/fake\\_sfml/fake\\_sfml.cpp](#)

## 6.46. sf::RenderWindow osztályreferencia

```
#include <fake_sfml.hpp>
```

### Publikus tagfüggvények

- [RenderWindow](#) ()
- [RenderWindow](#) (const std::string &title\_, std::size\_t w, std::size\_t h)
- [RenderWindow](#) ([VideoMode](#) vmode, const std::string &title\_)
- void [create](#) (const std::string &title\_, std::size\_t w, std::size\_t h)
- bool [isOpen](#) () const
- bool [pollEvent](#) ([Event](#) &event)
- void [close](#) ()
- void [setFramerateLimit](#) (std::size\_t limit)
- void [clear](#) ()
- void [draw](#) (const [Sprite](#) &sprite)
- void [draw](#) (const [Sprite](#) &sprite, [RenderStates](#) states)
- void [draw](#) (const [RectangleShape](#) &shape)
- void [display](#) ()
- void [clear](#) ([Color](#) clr)



## 6.46.1. Konstruktorkok és destruktorkok dokumentációja

### 6.46.1.1. RenderWindow() [1/3]

```
sf::RenderWindow::RenderWindow ( )
```

### 6.46.1.2. RenderWindow() [2/3]

```
sf::RenderWindow::RenderWindow (
    const std::string & title_,
    std::size_t w,
    std::size_t h )
```

### 6.46.1.3. RenderWindow() [3/3]

```
sf::RenderWindow::RenderWindow (
    VideoMode vmode,
    const std::string & title_ )
```

## 6.46.2. Tagfüggvények dokumentációja

### 6.46.2.1. clear() [1/2]

```
void sf::RenderWindow::clear ( )
```

### 6.46.2.2. clear() [2/2]

```
void sf::RenderWindow::clear (
    Color clr )
```

### 6.46.2.3. close()

```
void sf::RenderWindow::close ( )
```

**6.46.2.4. create()**

```
void sf::RenderWindow::create (
    const std::string & title_,
    std::size_t w,
    std::size_t h )
```

**6.46.2.5. display()**

```
void sf::RenderWindow::display ( )
```

**6.46.2.6. draw()** [1/3]

```
void sf::RenderWindow::draw (
    const RectangleShape & shape )
```

**6.46.2.7. draw()** [2/3]

```
void sf::RenderWindow::draw (
    const Sprite & sprite )
```

**6.46.2.8. draw()** [3/3]

```
void sf::RenderWindow::draw (
    const Sprite & sprite,
    RenderStates states )
```

**6.46.2.9. isOpen()**

```
bool sf::RenderWindow::isOpen ( ) const
```

**6.46.2.10. pollEvent()**

```
bool sf::RenderWindow::pollEvent (
    Event & event )
```

## 6.46.2.11. setFramerateLimit()

```
void sf::RenderWindow::setFramerateLimit (
    std::size_t limit )
```

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

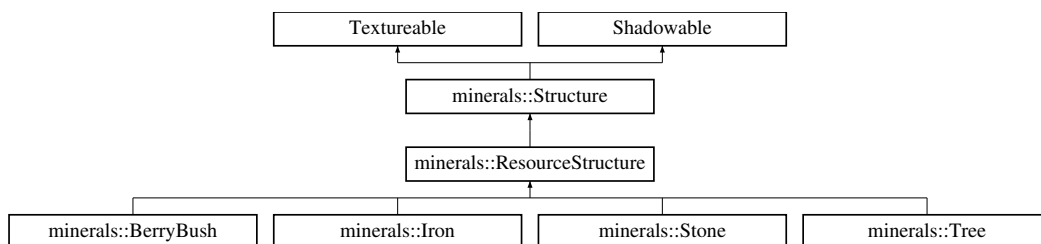
- [src/fake\\_sfml/fake\\_sfml.hpp](#)
- [src/fake\\_sfml/fake\\_sfml.cpp](#)

## 6.47. minerals::ResourceStructure osztályreferencia

Az erőforrás struktúra osztály leírása.

```
#include <ResourceStructure.hpp>
```

A minerals::ResourceStructure osztály származási diagramja:



## Publikus tagfüggvények

- bool [get\\_harvested](#) () const  
*Megadja, hogy kitermelt-e az erőforrás.*
- [ResourceStructure](#) (int x, int y)  
*Konstruktor ami lerakja az erőforrást egy (x,y) pontra.*
- virtual bool [harvest](#) ()=0  
*Tisztán virtuális metódus. Ez leírja mi történik, hogy ha kitermelik ezt az erőforrást.*
- virtual [~ResourceStructure](#) ()=default  
*Virtuális destruktork.*

## Védett attribútumok

- float [inner\\_timer](#)  
*Az idéződés óta eltelt idő.*
- bool [harvested](#)  
*Ki van-e termelve az erőforrás.*

## További örökölt tagok

### 6.47.1. Részletes leírás

Az erőforrás struktúra osztály leírása.

Ez az interface rendelkezik azokról az adatokról, hogy kibányászható-e még ez az objektum és ahhoz szükséges metódusokkal.

### 6.47.2. Konstruktorkok és destruktorkok dokumentációja

#### 6.47.2.1. ResourceStructure()

```
minerals::ResourceStructure::ResourceStructure (
    int x,
    int y )
```

Konstruktork ami lerakja az erőforrást egy (x,y) pontra.

#### 6.47.2.2. ~ResourceStructure()

```
virtual minerals::ResourceStructure::~~ResourceStructure ( ) [virtual], [default]
```

Virtuális destruktork.

### 6.47.3. Tagfüggvények dokumentációja

#### 6.47.3.1. get\_harvested()

```
bool minerals::ResourceStructure::get_harvested ( ) const
```

Megadja, hogy kitermelt-e az erőforrás.

#### 6.47.3.2. harvest()

```
virtual bool minerals::ResourceStructure::harvest ( ) [pure virtual]
```

Tisztán virtuális metódus. Ez leírja mi történik, hogy ha kitermelik ezt az erőforrást.

Megvalósítják a következők: [minerals::Tree](#), [minerals::Stone](#), [minerals::Iron](#) és [minerals::BerryBush](#).

### 6.47.4. Adattagok dokumentációja

#### 6.47.4.1. harvested

```
bool minerals::ResourceStructure::harvested [protected]
```

Ki van-e termelve az erőforrás.

#### 6.47.4.2. inner\_timer

```
float minerals::ResourceStructure::inner_timer [protected]
```

Az idéződés óta eltelt idő.

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- src/world\_object/[ResourceStructure.hpp](#)
- src/world\_object/[ResourceStructure.cpp](#)

## 6.48. SaveManager osztályreferencia

A fájl menedzseléshez szolgáló osztály leírása.

```
#include <SaveManager.hpp>
```

### Publikus tagfüggvények

- [SaveManager](#) (const std::string &file)  
*A konstruktor, ahol beállítható, hogy mi a neve és elérési útvonala a mentés fájljának.*
- void [saveFile](#) ([World](#) &world)  
*Elment egy világot a fájlba.*
- void [loadFile](#) ([World](#) &world)  
*Elment egy fájlt a világba.*
- void [deleteFile](#) ()  
*Kitörli a jelenlegi mentés fájl tartalmát.*

#### 6.48.1. Részletes leírás

A fájl menedzseléshez szolgáló osztály leírása.

Képes betölteni mentést, eltárolni és törölni is.

#### 6.48.2. Konstruktorok és destruktorok dokumentációja

##### 6.48.2.1. SaveManager()

```
SaveManager::SaveManager (  
    const std::string & file )
```

A konstruktor, ahol beállítható, hogy mi a neve és elérési útvonala a mentés fájljának.

**Paraméterek**

<i>file</i>	Az elérési útvonal.
-------------	---------------------

### 6.48.3. Tagfüggvények dokumentációja

#### 6.48.3.1. deleteFile()

```
void SaveManager::deleteFile ( )
```

Kitörli a jelenlegi mentés fájl tartalmát.

#### 6.48.3.2. loadFile()

```
void SaveManager::loadFile (
    World & world )
```

Elment egy fájlt a világba.

**Paraméterek**

<i>world</i>	Referencia a világra.
--------------	-----------------------

#### 6.48.3.3. saveFile()

```
void SaveManager::saveFile (
    World & world )
```

Elment egy világot a fájlba.

**Paraméterek**

<i>world</i>	Referencia a világra.
--------------	-----------------------

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

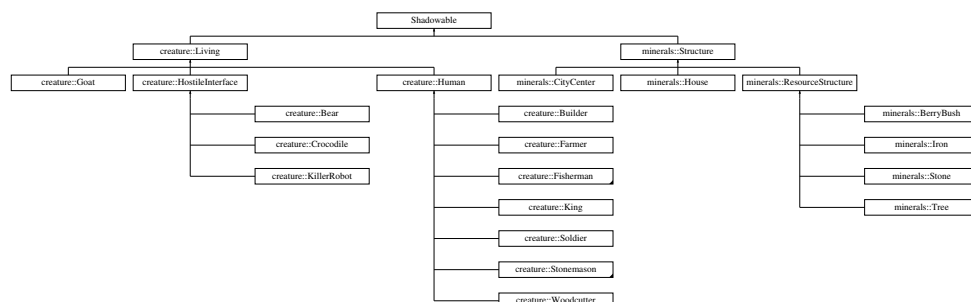
- [src/SaveManager.hpp](#)
- [src/SaveManager.cpp](#)

## 6.49. Shadowable osztályreferencia

Az árnyékoláshoz szükséges interface.

```
#include <Shadowable.hpp>
```

A Shadowable osztály származási diagramja:



## Publikus tagfüggvények

- double `get_height_offset` () const  
*Egy getter a magasságpont eltolásának megszerzésére.*
- int `get_shadow_strength` () const  
*Egy getter az árnyék erősségre.*
- float `get_skew_offset` () const  
*Egy getter az elnyújtás mértékére.*
- void `set_height_offset` (double new\_val)  
*Egy setter a magasságpont eltolásához.*
- void `set_shadow_strength` (int new\_val)  
*Egy setter az árnyék erősséghez.*
- void `set_skew_offset` (float new\_val)  
*Egy setter a elnyújtás mértékéhez.*
- virtual `~Shadowable` ()=default  
*Virtuális destruktork.*
- bool `setShadowTexture` (const std::string &filename)  
*Az árnyék kinézetét állítja be.*
- void `setShadow` (float ySize, float xSkew)  
*Beállítja az árnyék nyújtását és eltolását.*
- void `setShadowDayNightCycle` (float delta\_time)  
*Beállítja az árnyék nyújtását a napszaktól függően.*
- void `setShadowPosition` (double x, double y)  
*Beállítja az árnyék helyét.*
- void `drawShadow` (sf::RenderWindow &>window)  
*Kirajzolja az árnyékot.*

## Védett attribútumok

- double `height_offset` = 0.0
- Milyen messze kezdődjön az árnyék az objektum alsó pontjától.*

### 6.49.1. Részletes leírás

Az árnyékoláshoz szükséges interface.

Tárolja az árnyék textúráját, valamiért felelős annak mozgatásáért és rendes kirajzolásáért.

### 6.49.2. Konstruktorkok és destruktorkok dokumentációja

#### 6.49.2.1. ~Shadowable()

```
virtual Shadowable::~~Shadowable ( ) [virtual], [default]
```

Virtuális destruktork.

### 6.49.3. Tagfüggvények dokumentációja

#### 6.49.3.1. drawShadow()

```
void Shadowable::drawShadow (
    sf::RenderWindow & window )
```

Kirajzolja az árnyékot.

##### Paraméterek

<i>window</i>	Az ablak, ahova ki kell rajzolni.
---------------	-----------------------------------

#### 6.49.3.2. get\_height\_offset()

```
double Shadowable::get_height_offset ( ) const
```

Egy getter a magasságpont eltolásának megszerzésére.

##### Visszatérési érték

A magasságpont eltolásának értéke.



#### 6.49.3.3. get\_shadow\_strength()

```
int Shadowable::get_shadow_strength ( ) const
```

Egy getter az árnyék erősségre.

##### Visszatérési érték

Az árnyék erőssége.

#### 6.49.3.4. get\_skew\_offset()

```
float Shadowable::get_skew_offset ( ) const
```

Egy getter az elnyújtás mértékére.

##### Visszatérési érték

Az elnyújtás mértéke.

#### 6.49.3.5. set\_height\_offset()

```
void Shadowable::set_height_offset (
    double new_val )
```

Egy setter a magasságpont eltolásához.

##### Paraméterek

<i>new_val</i>	Az új érték, amire be kell állítani.
----------------	--------------------------------------

#### 6.49.3.6. set\_shadow\_strength()

```
void Shadowable::set_shadow_strength (
    int new_val )
```

Egy setter az árnyék erősséghez.

##### Paraméterek

<i>new_val</i>	Az új érték, amire be kell állítani.
----------------	--------------------------------------

#### 6.49.3.7. set\_skew\_offset()

```
void Shadowable::set_skew_offset (
    float new_val )
```

Egy setter a elnyújtás mértékéhez.

##### Paraméterek

<i>new_val</i>	Az új érték, amire be kell állítani.
----------------	--------------------------------------

#### 6.49.3.8. setShadow()

```
void Shadowable::setShadow (
    float ySize,
    float xSkew )
```

Beállítja az árnyék nyújtását és eltolását.

##### Paraméterek

<i>ySize</i>	Az Y tengelyen való nyújtás.
<i>xSkew</i>	Az X elnyújtás.

#### 6.49.3.9. setShadowDayNightCycle()

```
void Shadowable::setShadowDayNightCycle (
    float delta_time )
```

Beállítja az árnyék nyújtását a napszaktól függően.

##### Paraméterek

<i>delta_time</i>	Az előző frissítés óta eltelt idő.
-------------------	------------------------------------

#### 6.49.3.10. setShadowPosition()

```
void Shadowable::setShadowPosition (
    double x,
    double y )
```

Beállítja az árnyék helyét.

#### Paraméterek

<i>x</i>	Az x koordináta.
<i>y</i>	Az y koordináta.

#### 6.49.3.11. setShadowTexture()

```
bool Shadowable::setShadowTexture (
    const std::string & filename )
```

Az árnyék kinézetét állítja be.

#### Paraméterek

<i>filename</i>	A textúra elérési útvonala
-----------------	----------------------------

#### Visszatérési érték

Igaz, ha sikeres a textúra beállítás, különben hamis.

### 6.49.4. Adattagok dokumentációja

#### 6.49.4.1. height\_offset

```
double Shadowable::height_offset =0.0 [protected]
```

Milyen messze kezdődjön az árnyék az objektum alsó pontjától.

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

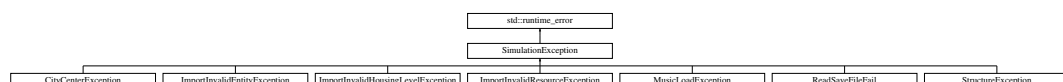
- [src/Shadowable.hpp](#)
- [src/Shadowable.cpp](#)

## 6.50. SimulationException osztályreferencia

Akkor kell dobni, ha egy szimulációs elem hibásan viselkedik.

```
#include <SimulationException.hpp>
```

A SimulationException osztály származási diagramja:



## Publikus tagfüggvények

- [SimulationException](#) (const std::string &msg)

### 6.50.1. Részletes leírás

Akkor kell dobni, ha egy szimulációs elem hibásan viselkedik.

### 6.50.2. Konstruktorkok és destruktorkok dokumentációja

#### 6.50.2.1. SimulationException()

```
SimulationException::SimulationException (
    const std::string & msg ) [inline], [explicit]
```

Ez a dokumentáció az osztályról a következő fájl alapján készült:

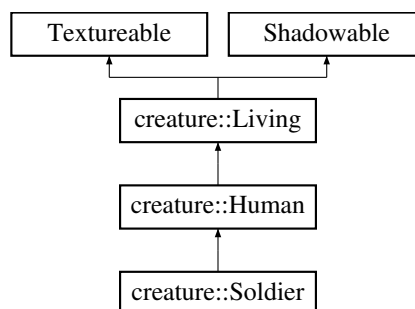
- src/exceptions/[SimulationException.hpp](#)

## 6.51. creature::Soldier osztályreferencia

A katona szakmájú ember osztály leírása.

```
#include <Soldier.hpp>
```

A creature::Soldier osztály származási diagramja:



## Publikus tagfüggvények

- [Soldier](#) (int x, int y, [ENTITY\\_GENDER](#) gender\_modifier)  
*Inicializál egy katonát egy pontos x és y koordinátára és beállítja az attribútumait.*
- void [update\\_logic](#) ([World](#) &world, float deltaTime) override  
*Az entitás frissítési logikája itt van definiálva. Például a halász vizet keres, ha odaért akkor halászik.*
- [~Soldier](#) ()  
*A katona destruktora.*

## További örökölt tagok

### 6.51.1. Részletes leírás

A katona szakmájú ember osztály leírása.

Ez a szakmájú ember vadászik állatokat és megvédi a népet az ellenséges entitásoktól.

### 6.51.2. Konstruktorok és destruktorok dokumentációja

#### 6.51.2.1. Soldier()

```
creature::Soldier::Soldier (
    int x,
    int y,
    ENTITY_GENDER gender_modifier )
```

Inicializál egy katonát egy pontos x és y koordinátára és beállítja az attribútumait.

##### Paraméterek

<i>x</i>	Az x koordináta.
<i>y</i>	Az y koordináta.
<i>gender_modifier</i>	A katona neme.

#### 6.51.2.2. ~Soldier()

```
creature::Soldier::~~Soldier ( )
```

A katona destruktora.

### 6.51.3. Tagfüggvények dokumentációja

#### 6.51.3.1. update\_logic()

```
void creature::Soldier::update_logic (
    World & world,
    float deltaTime ) [override], [virtual]
```

Az entitás frissítési logikája itt van definiálva. Például a halász vizet keres, ha odaért akkor halászik.

## Paraméterek

<i>world</i>	A világ, amibe frissíti magát az entitás.
<i>deltaTime</i>	Az előző frissítés óta eltelt idő.

Újrimplementált ősök: [creature::Human](#).

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- [src/creatures/humans/Soldier.hpp](#)
- [src/creatures/humans/Soldier.cpp](#)

## 6.52. sf::Sound osztályreferencia

```
#include <fake_sfml.hpp>
```

### Publikus tagfüggvények

- void [setBuffer](#) ([SoundBuffer](#) &buf)
- void [play](#) ()
- void [stop](#) ()
- [~Sound](#) ()

### 6.52.1. Konstruktorok és destruktorok dokumentációja

#### 6.52.1.1. ~Sound()

```
sf::Sound::~~Sound ( )
```

### 6.52.2. Tagfüggvények dokumentációja

#### 6.52.2.1. play()

```
void sf::Sound::play ( )
```

#### 6.52.2.2. setBuffer()

```
void sf::Sound::setBuffer (
    SoundBuffer & buf )
```

#### 6.52.2.3. stop()

```
void sf::Sound::stop ( )
```

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- [src/fake\\_sfml/fake\\_sfml.hpp](#)
- [src/fake\\_sfml/fake\\_sfml.cpp](#)

### 6.53. sf::SoundBuffer osztályreferencia

```
#include <fake_sfml.hpp>
```

#### Publikus tagfüggvények

- bool [loadFromFile](#) (const std::string &filepath)

#### 6.53.1. Tagfüggvények dokumentációja

##### 6.53.1.1. loadFromFile()

```
bool sf::SoundBuffer::loadFromFile (
    const std::string & filepath )
```

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- [src/fake\\_sfml/fake\\_sfml.hpp](#)
- [src/fake\\_sfml/fake\\_sfml.cpp](#)

### 6.54. SoundPlayer osztályreferencia

A hanglejátszó osztály leírása.

```
#include <SoundPlayer.hpp>
```

## Publikus tagfüggvények

- void `load_sound` (const std::string &filename)  
*Betölt egy hangot az elérési útvonalról.*
- void `play_sound` (const std::string &filename)  
*Lejátszik egy hangot az elérési útvonalról. Ha még nem volt ez betöltve akkor először betölti.*
- void `stop_sound` ()  
*Megállítja az éppen lejátszott hangot.*

### 6.54.1. Részletes leírás

A hanglejátszó osztály leírása.

Képes hangokat betölteni, elindítani, lejátszani és megállítani.

### 6.54.2. Tagfüggvények dokumentációja

#### 6.54.2.1. load\_sound()

```
void SoundPlayer::load_sound (  
    const std::string & filename )
```

Betölt egy hangot az elérési útvonalról.

##### Paraméterek

<i>filename</i>	Az elérési útvonal.
-----------------	---------------------

#### 6.54.2.2. play\_sound()

```
void SoundPlayer::play_sound (  
    const std::string & filename )
```

Lejátszik egy hangot az elérési útvonalról. Ha még nem volt ez betöltve akkor először betölti.

##### Paraméterek

<i>filename</i>	Az elérési útvonal.
-----------------	---------------------



### 6.54.2.3. stop\_sound()

```
void SoundPlayer::stop_sound ( )
```

Megállítja az éppen lejátszott hangot.

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- src/[SoundPlayer.hpp](#)
- src/[SoundPlayer.cpp](#)

## 6.55. sf::SoundSource osztályreferencia

```
#include <fake_sfml.hpp>
```

### Publikus típusok

- enum [SoundSourceType](#) : char { [Playing](#) , [Stopped](#) , [Paused](#) }

### Publikus tagfüggvények

- [SoundSource](#) ()
- virtual [~SoundSource](#) ()=default

### Publikus attribútumok

- [SoundSourceType](#) type

### 6.55.1. Enumeráció-tagok dokumentációja

#### 6.55.1.1. SoundSourceType

```
enum sf::SoundSource::SoundSourceType : char
```

Enumeráció-értékek

Playing	
Stopped	
Paused	

## 6.55.2. Konstruktorok és destruktorok dokumentációja

### 6.55.2.1. SoundSource()

```
sf::SoundSource::SoundSource ( )
```

### 6.55.2.2. ~SoundSource()

```
virtual sf::SoundSource::~~SoundSource ( ) [virtual], [default]
```

## 6.55.3. Adattagok dokumentációja

### 6.55.3.1. type

```
SoundSourceType sf::SoundSource::type
```

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- [src/fake\\_sfml/fake\\_sfml.hpp](#)
- [src/fake\\_sfml/fake\\_sfml.cpp](#)

## 6.56. sf::Sprite osztályreferencia

```
#include <fake_sfml.hpp>
```

### Publikus tagfüggvények

- [Sprite](#) ()
- void [setTexture](#) (const [Texture](#) &tex)
- void [setTextureRect](#) (const [IntRect](#) &tex)
- [Texture](#) \* [getTexture](#) () const
- void [setPosition](#) (float x, float y)
- void [setOrigin](#) (float \_x, float \_y)
- void [setRotation](#) (float deg)
- [Vector2f](#) [getPosition](#) ()
- void [setScale](#) (float sx, float sy)
- [FloatRect](#) [getLocalBounds](#) ()
- [FloatRect](#) [getGlobalBounds](#) ()
- [FloatRect](#) [getGlobalBounds](#) () const
- void [draw](#) () const
- void [setColor](#) ([Color](#) \_clr)
- [~Sprite](#) ()

## 6.56.1. Konstruktorkok és destruktorkok dokumentációja

### 6.56.1.1. Sprite()

```
sf::Sprite::Sprite ( )
```

### 6.56.1.2. ~Sprite()

```
sf::Sprite::~~Sprite ( )
```

## 6.56.2. Tagfüggvények dokumentációja

### 6.56.2.1. draw()

```
void sf::Sprite::draw ( ) const
```

### 6.56.2.2. getGlobalBounds() [1/2]

```
FloatRect sf::Sprite::getGlobalBounds ( )
```

### 6.56.2.3. getGlobalBounds() [2/2]

```
FloatRect sf::Sprite::getGlobalBounds ( ) const
```

### 6.56.2.4. getLocalBounds()

```
FloatRect sf::Sprite::getLocalBounds ( )
```

**6.56.2.5. getPosition()**

```
Vector2f sf::Sprite::getPosition ( )
```

**6.56.2.6. getTexture()**

```
Texture * sf::Sprite::getTexture ( ) const
```

**6.56.2.7. setColor()**

```
void sf::Sprite::setColor (
    Color _clr )
```

**6.56.2.8. setOrigin()**

```
void sf::Sprite::setOrigin (
    float _x,
    float _y )
```

**6.56.2.9. setPosition()**

```
void sf::Sprite::setPosition (
    float x,
    float y )
```

**6.56.2.10. setRotation()**

```
void sf::Sprite::setRotation (
    float deg )
```

**6.56.2.11. setScale()**

```
void sf::Sprite::setScale (
    float sx,
    float sy )
```

## 6.56.2.12. setTexture()

```
void sf::Sprite::setTexture (
    const Texture & tex )
```

## 6.56.2.13. setTextureRect()

```
void sf::Sprite::setTextureRect (
    const IntRect & tex )
```

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

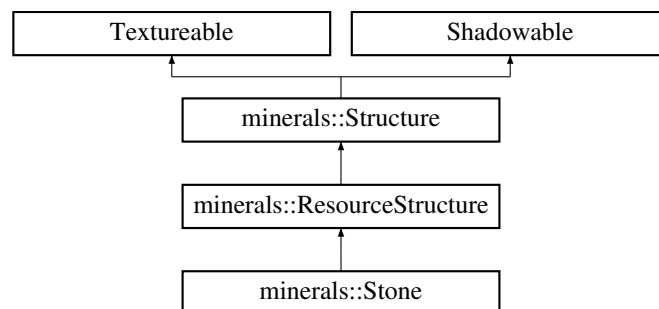
- [src/fake\\_sfml/fake\\_sfml.hpp](#)
- [src/fake\\_sfml/fake\\_sfml.cpp](#)

## 6.57. minerals::Stone osztályreferencia

A kő osztály leírása. követ ad, amikor kitermelik.

```
#include <Stone.hpp>
```

A minerals::Stone osztály származási diagramja:



## Publikus tagfüggvények

- [Stone](#) (int x, int y)  
*Konstruktor ami lerakja a házat egy (x,y) pontra.*
- [MINERAL\\_TYPE get\\_type](#) () const override  
*Szimbólum, ami a fájlba mentéshez kell.*
- void [update\\_logic](#) (float deltaTime) override  
*Frissíti magát az idő függvényében.*
- bool [harvest](#) () override  
*Tisztán virtuális metódus. Ez leírja mi történik, hogy ha kitermelik ezt az erőforrást.*

## További örökölt tagok

### 6.57.1. Részletes leírás

A kő osztály leírása. követ ad, amikor kitermelik.

### 6.57.2. Konstruktorkok és destruktorkok dokumentációja

#### 6.57.2.1. Stone()

```
minerals::Stone::Stone (
    int x,
    int y )
```

Konstruktork ami lerakja a házat egy (x,y) pontra.

### 6.57.3. Tagfüggvények dokumentációja

#### 6.57.3.1. get\_type()

```
MINERAL_TYPE minerals::Stone::get_type ( ) const [override], [virtual]
```

Szimbólum, ami a fájlba mentéshez kell.

Megvalósítja a következőket: [minerals::Structure](#).

#### 6.57.3.2. harvest()

```
bool minerals::Stone::harvest ( ) [override], [virtual]
```

Tisztán virtuális metódus. Ez leírja mi történik, hogy ha kitermelik ezt az erőforrást.

Megvalósítja a következőket: [minerals::ResourceStructure](#).

#### 6.57.3.3. update\_logic()

```
void minerals::Stone::update_logic (
    float deltaTime ) [override], [virtual]
```

Frissíti magát az idő függvényében.

## Paraméterek

<code>deltaTime</code>	Az előző frissítés óta eltelt idő.
------------------------	------------------------------------

Megvalósítja a következőket: [minerals::Structure](#).

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

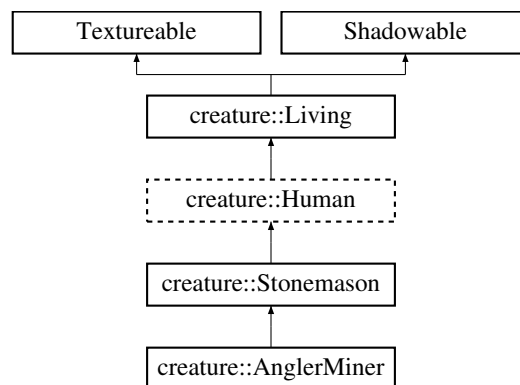
- [src/world\\_object/Stone.hpp](#)
- [src/world\\_object/Stone.cpp](#)

## 6.58. creature::Stonemason osztályreferencia

A bányász szakmájú ember osztály leírása.

```
#include <Stonemason.hpp>
```

A creature::Stonemason osztály származási diagramja:



### Publikus tagfüggvények

- [Stonemason](#) (int x, int y, [ENTITY\\_GENDER](#) gender\_modifier)  
*Inicializál egy bányászt egy pontos x és y koordinátára és beállítja az attribútumait.*
- void [update\\_logic](#) ([World](#) &world, float deltaTime) override  
*Az entitás frissítési logikája itt van definiálva. Például a halász vizet keres, ha odaért akkor halászik.*
- [~Stonemason](#) ()  
*A bányász destruktora.*

### Védett tagfüggvények

- void [try\\_mine](#) ([World](#) &world)  
*Megpróbál magának egy kőoszlopot vagy vasércet keresni, amit aztán ki fog bányászni.*

## Védett attribútumok

- bool `mining_iron`

*Vasat bányászik-e? Ha hamis, akkor követ bányászik.*

## További örökölt tagok

### 6.58.1. Részletes leírás

A bányász szakmájú ember osztály leírása.

Ez a szakmájú ember követ vagy vasat keres és kitermeli őket, így követ és vasat szerez.

### 6.58.2. Konstruktork és destruktork dokumentációja

#### 6.58.2.1. `Stonemason()`

```
creature::Stonemason::Stonemason (
    int x,
    int y,
    ENTITY_GENDER gender_modifier )
```

Inicializál egy bányászt egy pontos x és y koordinátára és beállítja az attribútumait.

#### Paraméterek

<code>x</code>	Az x koordináta.
<code>y</code>	Az y koordináta.
<code>gender_modifier</code>	A bányász neme.

#### 6.58.2.2. `~Stonemason()`

```
creature::Stonemason::~~Stonemason ( )
```

A bányász destruktora.

### 6.58.3. Tagfüggvények dokumentációja



### 6.58.3.1. try\_mine()

```
void creature::Stonemason::try_mine (
    World & world ) [protected]
```

Megpróbál magának egy kőoszlopot vagy vasércet keresni, amit aztán ki fog bányászni.

#### Paraméterek

<i>world</i>	A világ, amibe keresni kell az érceket.
--------------	---

### 6.58.3.2. update\_logic()

```
void creature::Stonemason::update_logic (
    World & world,
    float deltaTime ) [override], [virtual]
```

Az entitás frissítési logikája itt van definiálva. Például a halász vizet keres, ha odaért akkor halászik.

#### Paraméterek

<i>world</i>	A világ, amibe frissíti magát az entitás.
<i>deltaTime</i>	Az előző frissítés óta eltelt idő.

Újraimplementált ősök: [creature::Human](#).

## 6.58.4. Adattagok dokumentációja

### 6.58.4.1. mining\_iron

```
bool creature::Stonemason::mining_iron [protected]
```

Vasat bányászik-e? Ha hamis, akkor követ bányászik.

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

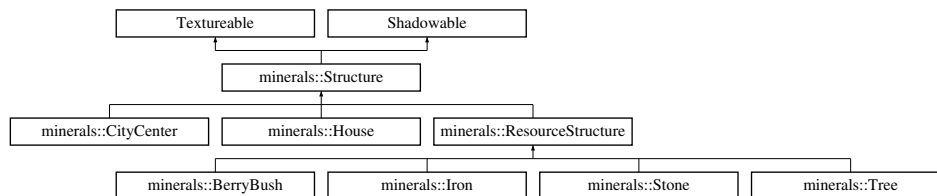
- [src/creatures/humans/Stonemason.hpp](#)
- [src/creatures/humans/Stonemason.cpp](#)

## 6.59. minerals::Structure osztályreferencia

A struktúra osztály leírása.

```
#include <Structure.hpp>
```

A minerals::Structure osztály származási diagramja:



### Publikus tagfüggvények

- **Structure** (int x, int y)  
*Létrehozza magát az x és y megadott pontban.*
- bool **setTexture** (const std::string &filename) override  
*Beállít egy fájl elérési útból egy textúrát. Megvalósítástól függően esetleg 2-t.*
- void **setPosition** (double x, double y) override  
*Beállítja, hogy hova kell rajzolni a textúrázható leszármazottat.*
- void **draw** (sf::RenderWindow &window) override  
*Kirajzolja az objektumot.*
- bool **needs\_drawn** ()  
*Igazat ad vissza, ha látható és ezért ki kell rajzolni.*
- virtual **MINERAL\_TYPE get\_type** () const =0  
*Szimbólum, ami a fájlba mentéshez kell.*
- virtual void **update\_logic** (float deltaTime)=0  
*Frissíti magát az idő függvényében.*
- void **draw\_logic** (sf::RenderWindow &window, float elapsed\_time, int offx, int offy)  
*Kirajzolja a struktúrát attól függően, hogy ki kell-e.*
- virtual **~Structure** ()=default  
*Alap virtuális destruktork.*

### Publikus attribútumok

- int **posx**  
*Az X koordináta, amin elhelyezkedik.*
- int **posy**  
*Az Y koordináta, amin elhelyezkedik.*

### Védett attribútumok

- const int **MAX\_OBJECT\_SIZE** =64  
*Egy határ, minnél nagyobb, annál nagyobb környezetbe lesznek kirajzolva a nézőpontból.*

### 6.59.1. Részletes leírás

A struktúra osztály leírása.

A Textúrázható és Árnyékolható interface-ből öröklődik. Alaposztály amiből később jönnek a házak, erőforrások.

### 6.59.2. Konstruktorok és destruktorok dokumentációja

#### 6.59.2.1. Structure()

```
minerals::Structure::Structure (
    int x,
    int y )
```

Létrehozza magát az x és y megadott pontban.

##### Paraméterek

<i>x</i>	Az x koordináta.
<i>y</i>	Az y koordináta.

#### 6.59.2.2. ~Structure()

```
virtual minerals::Structure::~~Structure ( ) [virtual], [default]
```

Alap virtuális destruktor.

### 6.59.3. Tagfüggvények dokumentációja

#### 6.59.3.1. draw()

```
void minerals::Structure::draw (
    sf::RenderWindow & window ) [override], [virtual]
```

Kirajzolja az objektumot.

##### Paraméterek

<i>window</i>	Ahova ki kell rajzolni a textúrázható leszármazottat.
---------------	---

Megvalósítja a következőket: [Textureable](#).

#### 6.59.3.2. draw\_logic()

```
void minerals::Structure::draw_logic (
    sf::RenderWindow & window,
    float elapsed_time,
    int offx,
    int offy )
```

Kirajzolja a struktúrát attól függően, hogy ki kell-e.

##### Paraméterek

<i>window</i>	Az ablak, ahova rajzolni kell.
<i>elapsed_time</i>	A világ megléte óta eltelt idő.
<i>offx</i>	A kamera X eltolása.
<i>offy</i>	A kamera Y eltolása.

#### 6.59.3.3. get\_type()

```
virtual MINERAL_TYPE minerals::Structure::get_type ( ) const [pure virtual]
```

Szimbólum, ami a fájlba mentéshez kell.

Megvalósítják a következők: [minerals::Tree](#), [minerals::Stone](#), [minerals::Iron](#), [minerals::House](#), [minerals::CityCenter](#) és [minerals::BerryBush](#).

#### 6.59.3.4. needs\_drawn()

```
bool minerals::Structure::needs_drawn ( )
```

Igazat ad vissza, ha látható és ezért ki kell rajzolni.

#### 6.59.3.5. setPosition()

```
void minerals::Structure::setPosition (
    double x,
    double y ) [override], [virtual]
```

Beállítja, hogy hova kell rajzolni a textúrázható leszármozottat.

## Paraméterek

<i>x</i>	Az x koordináta.
<i>y</i>	Az y koordináta.

Megvalósítja a következőket: [Textureable](#).

**6.59.3.6. setTexture()**

```
bool minerals::Structure::setTexture (
    const std::string & filename ) [override], [virtual]
```

Beállít egy fájl elérési útból egy textúrát. Megvalósítástól függően esetleg 2-t.

## Paraméterek

<i>filename</i>	A textúra elérési útja.
-----------------	-------------------------

## Visszatérési érték

Sikeres volt-e a textúra beállítása.

Megvalósítja a következőket: [Textureable](#).

**6.59.3.7. update\_logic()**

```
virtual void minerals::Structure::update_logic (
    float deltaTime ) [pure virtual]
```

Frissíti magát az idő függvényében.

## Paraméterek

<i>deltaTime</i>	Az előző frissítés óta eltelt idő.
------------------	------------------------------------

Megvalósítják a következők: [minerals::Tree](#), [minerals::Stone](#), [minerals::Iron](#), [minerals::House](#), [minerals::CityCenter](#) és [minerals::BerryBush](#).

**6.59.4. Adattagok dokumentációja**

#### 6.59.4.1. MAX\_OBJECT\_SIZE

```
const int minerals::Structure::MAX_OBJECT_SIZE = 64 [protected]
```

Egy határ, minnél nagyobb, annál nagyobb környezetbe lesznek kirajzolva a nézőpontból.

#### 6.59.4.2. posx

```
int minerals::Structure::posx
```

Az X koordináta, amin elhelyezkedik.

#### 6.59.4.3. posy

```
int minerals::Structure::posy
```

Az Y koordináta, amin elhelyezkedik.

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

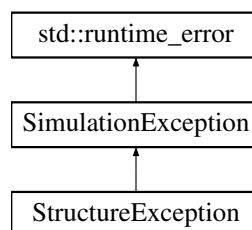
- [src/world\\_object/Structure.hpp](#)
- [src/world\\_object/Structure.cpp](#)

## 6.60. StructureException osztályreferencia

Akkor kell dobni, ha egy struktúra hibásan működött.

```
#include <WorldExceptions.hpp>
```

A StructureException osztály származási diagramja:



### Publikus tagfüggvények

- [StructureException](#) (const std::string &msg)

### 6.60.1. Részletes leírás

Akkor kell dobni, ha egy struktúra hibásan működött.

### 6.60.2. Konstruktorkok és destruktorkok dokumentációja

#### 6.60.2.1. StructureException()

```
StructureException::StructureException (
    const std::string & msg ) [inline], [explicit]
```

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- src/exceptions/[WorldExceptions.hpp](#)

## 6.61. TerrainContainer< T > osztálysablon-referencia

A világ terepét tároló osztály.

```
#include <TerrainContainer.hpp>
```

### Publikus tagfüggvények

- int [get\\_width](#) () const  
*Egy getter a tömb szélességére.*
- int [get\\_height](#) () const  
*Egy getter a tömb magasságára.*
- [TerrainContainer](#) ()  
*Alapkonstruktor. Nem történik memória foglalás.*
- [TerrainContainer](#) (int awidth, int aheight, T def\_value)  
*Konstruktor, ami már foglal memóriát és generál terepet.*
- void [swap\\_at](#) (int x1, int y1, int x2, int y2)  
*Kicseréli a (x1,y1) kockát az (x2,y2) helyen lévő kockával.*
- bool [is\\_valid\\_coordinate](#) (int x, int y)  
*Visszaadja, hogy az (x,y) koordinátán lévő terepkocka helyesen van definiálva-e.*
- bool [is\\_on\\_screen](#) (int x, int y)  
*Visszaadja, hogy az (x,y) koordinátán lévő terepkocka rajta van-e a látható síkon.*
- T \*& [operator\[\]](#) (std::size\_t row)  
*operator[] hogy elérhetőek legyenek a belső elemek.*
- T \*const & [operator\[\]](#) (std::size\_t row) const  
*konstans operator[] hogy elérhetőek legyenek a belső elemek.*
- void [draw](#) (sf::RenderWindow &window, int offx, int offy)  
*Kirajzolja az (x,y) koordinátán lévő terepkockát.*
- void [generate\\_world](#) ()

- *Generál egy új terepet.*
- void `clear_at` (int x, int y)  
*Felszabadítja az adott sorban és oszlopban elhelyezkedő terepkockát.*
- void `clear` ()  
*Felszabadítja a tárolt terepkockákat.*
- void `set_seed` (int new\_value)  
*Setter, beállítja a seedet az új megadott értékre.*
- int `get_seed` () const  
*Getter, visszaadja a seedet.*
- template<typename Defvalue >  
void `resize` (int awidth, int aheight)  
*Újraméretezi a terepet az új méreteir és feltölti a megadott típusú adattal.*
- ~`TerrainContainer` ()  
*A destruktork, ami kitörli az összes tárolt terepkockát a `clear()` meghívásával.*

## Publikus attribútumok

- const int `TILE_SIZE` =32  
*Egy terep maximális textúra mérete. Ennél nagyobb terepkockák talán nem rajzolódnak ki, mert a kamera úgy érzékeli, hogy már nincs benne a látótérben.*

### 6.61.1. Részletes leírás

```
template<typename T>
class TerrainContainer< T >
```

A világ terepét tároló osztály.

Sablon paraméterek

<code>T</code>	A generikus elem, amiket eltárol ez a konténer.
----------------	---

Egy dinamikus 2 dimenziós n\*m-es tömb. Rendelkezik a szükséges getterekkel. Ez az osztály felelős a világ terepének a véletlen generálásáért is.

### 6.61.2. Konstruktork és destruktork dokumentációja

#### 6.61.2.1. TerrainContainer() [1/2]

```
template<typename T >
TerrainContainer< T >::TerrainContainer ( ) [default]
```

Alapkonstruktor. Nem történik memória foglalás.



### 6.61.2.2. TerrainContainer() [2/2]

```
template<typename T >
TerrainContainer< T >::TerrainContainer (
    int awidth,
    int aheight,
    T def_value )
```

Konstruktor, ami már foglal memóriát és generál terepet.

#### Paraméterek

<i>awidth</i>	Az új szélesség.
<i>ahheight</i>	Az új magasság.
<i>def_value</i>	Alap érték, amivel feltöltődik a tömb.

### 6.61.2.3. ~TerrainContainer()

```
template<typename T >
TerrainContainer< T >::~~TerrainContainer
```

A destruktor, ami kitörli az összes tárolt terepkockát a [clear\(\)](#) meghívásával.

## 6.61.3. Tagfüggvények dokumentációja

### 6.61.3.1. clear()

```
template<typename T >
void TerrainContainer< T >::clear
```

Felszabadítja a tárolt terepkockákat.

### 6.61.3.2. clear\_at()

```
template<typename T >
void TerrainContainer< T >::clear_at (
    int x,
    int y )
```

Felszabadítja az adott sorban és oszlopban elhelyezkedő terepkockát.

## Paraméterek

<i>x</i>	Oszlop index.
<i>y</i>	Sor index.

**6.61.3.3. draw()**

```
template<typename T >
void TerrainContainer< T >::draw (
    sf::RenderWindow & window,
    int offx,
    int offy )
```

Kirajzolja az (x,y) koordinátán lévő terepkockát.

## Paraméterek

<i>window</i>	Ahova ki kell rajzolni a terepkockát.
<i>x</i>	Oszlop index.
<i>y</i>	Sor index.

**6.61.3.4. generate\_world()**

```
template<typename T >
void TerrainContainer< T >::generate_world
```

Generál egy új terepet.

**6.61.3.5. get\_height()**

```
template<typename T >
int TerrainContainer< T >::get_height
```

Egy getter a tömb magasságára.

## Visszatérési érték

A tömb magassága.

#### 6.61.3.6. get\_seed()

```
template<typename T >
int TerrainContainer< T >::get_seed
```

Getter, visszaadja a seedet.

##### Visszatérési érték

A seed.

#### 6.61.3.7. get\_width()

```
template<typename T >
int TerrainContainer< T >::get_width
```

Egy getter a tömb szélességére.

##### Visszatérési érték

A tömb szélessége.

#### 6.61.3.8. is\_on\_screen()

```
template<typename T >
bool TerrainContainer< T >::is_on_screen (
    int x,
    int y )
```

Visszaadja, hogy az (x,y) koordinátán lévő terepkocka rajta van-e a látható síkon.

##### Paraméterek

x	Oszlop index.
y	Sor index.

##### Visszatérési érték

Igaz, rajta van, különben hamis.

#### 6.61.3.9. is\_valid\_coordinate()

```
template<typename T >
bool TerrainContainer< T >::is_valid_coordinate (
```

```
int x,
int y )
```

Visszaadja, hogy az (x,y) koordinátán lévő terepkocka helyesen van definiálva-e.

#### Paraméterek

<i>x</i>	Oszlop index.
<i>y</i>	Sor index.

#### Visszatérési érték

Igaz, ha rendesen van definiálva és már használható, különben hamis.

#### 6.61.3.10. operator[]() [1/2]

```
template<typename T >
T *& TerrainContainer< T >::operator[] (
    std::size_t row )
```

operator[] hogy elérhetőek legyenek a belső elemek.

#### 6.61.3.11. operator[]() [2/2]

```
template<typename T >
T *const & TerrainContainer< T >::operator[] (
    std::size_t row ) const
```

konstans operator[] hogy elérhetőek legyenek a belső elemek.

#### 6.61.3.12. resize()

```
template<typename T >
template<typename Defvalue >
void TerrainContainer< T >::resize (
    int awidth,
    int aheight )
```

Újraméretezi a terepet az új méreteir és feltölti a megadott típusú adattal.

#### Sablon paraméterek

<i>Defvalue</i>	Az érték amivel az új terep fel lesz töltve.
-----------------	--

## Paraméterek

<i>awidth</i>	Az új érték.
<i>aheight</i>	Az új érték.

**6.61.3.13. set\_seed()**

```
template<typename T >
void TerrainContainer< T >::set_seed (
    int new_value )
```

Setter, beállítja a seedet az új megadott értékre.

## Paraméterek

<i>new_value</i>	Az új érték.
------------------	--------------

**6.61.3.14. swap\_at()**

```
template<typename T >
void TerrainContainer< T >::swap_at (
    int x1,
    int y1,
    int x2,
    int y2 )
```

Kicseréli a (x1,y1) kockát az (x2,y2) helyen lévő kockával.

## Paraméterek

<i>x1</i>	Az 1. kocka x koordinátája.
<i>y1</i>	Az 1. kocka y koordinátája.
<i>x2</i>	A 2. kocka x koordinátája.
<i>y2</i>	A 2. kocka y koordinátája.

**6.61.4. Adattagok dokumentációja****6.61.4.1. TILE\_SIZE**

```
template<typename T >
const int TerrainContainer< T >::TILE_SIZE =32
```

Egy terep maximális textúra mérete. Ennél nagyobb terepkockák talán nem rajzolódnak ki, mert a kamera úgy érzékeli, hogy már nincs benne a látótérben.

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- src/[TerrainContainer.hpp](#)
- src/[TerrainContainer.inl](#)

## 6.62. gtest\_lite::Test struktúrareferencia

```
#include <gtest_lite.h>
```

### Publikus tagfüggvények

- void [begin](#) (const char \*n)  
*Teszt kezdete.*
- std::ostream & [end](#) (bool memchk=false)  
*Teszt vége.*
- bool [fail](#) ()
- bool [astatus](#) ()
- std::ostream & [expect](#) (bool st, const char \*file, int line, const char \*expr, bool pr=false)  
*Eredményt adminisztráló tagfüggvény True a jó eset.*
- [~Test](#) ()  
*Destruktor.*

### Statikus publikus tagfüggvények

- static [Test](#) & [getTest](#) ()

### Publikus attribútumok

- int [sum](#)  
*tesztek számlálója*
- int [failed](#)  
*hibás tesztek*
- int [ablocks](#)  
*allokált blokkok száma*
- bool [status](#)  
*éppen futó teszt státusza.*
- bool [tmp](#)  
*temp a kivételkezeléshez;*
- std::string [name](#)  
*éppen futó teszt neve.*
- std::fstream [null](#)  
*nyelő, ha nem kell kiírni semmit*
- std::ostream & [os](#)  
*ide írunk*

### 6.62.1. Részletes leírás

Tesztek állapotát tároló osztály. Egyetlen egy statikus példány keletkezik, aminek a destruktora a futás végén hívódik meg.

### 6.62.2. Konstruktorkok és destruktorkok dokumentációja

#### 6.62.2.1. ~Test()

```
gtest_lite::Test::~~Test ( ) [inline]
```

Destruktor.

### 6.62.3. Tagfüggvények dokumentációja

#### 6.62.3.1. astatus()

```
bool gtest_lite::Test::astatus ( ) [inline]
```

#### 6.62.3.2. begin()

```
void gtest_lite::Test::begin (
    const char * n ) [inline]
```

Teszt kezdete.

#### 6.62.3.3. end()

```
std::ostream& gtest_lite::Test::end (
    bool memchk = false ) [inline]
```

Teszt vége.

#### 6.62.3.4. expect()

```
std::ostream& gtest_lite::Test::expect (
    bool st,
    const char * file,
    int line,
    const char * expr,
    bool pr = false ) [inline]
```

Eredményt adminisztráló tagfüggvény True a jó eset.

#### 6.62.3.5. fail()

```
bool gtest_lite::Test::fail ( ) [inline]
```

#### 6.62.3.6. getTest()

```
static Test& gtest_lite::Test::getTest ( ) [inline], [static]
```

< egyedüli (singleton) példány

### 6.62.4. Adattagok dokumentációja

#### 6.62.4.1. ablocks

```
int gtest_lite::Test::ablocks
```

allokált blokkok száma

#### 6.62.4.2. failed

```
int gtest_lite::Test::failed
```

hibás tesztek



#### 6.62.4.3. name

```
std::string gtest_lite::Test::name
```

éppen futó teszt neve.

#### 6.62.4.4. null

```
std::fstream gtest_lite::Test::null
```

nyelő, ha nem kell kiírni semmit

#### 6.62.4.5. os

```
std::ostream& gtest_lite::Test::os
```

ide írunk

#### 6.62.4.6. status

```
bool gtest_lite::Test::status
```

éppen futó teszt státusza.

#### 6.62.4.7. sum

```
int gtest_lite::Test::sum
```

tesztek számlálója

#### 6.62.4.8. tmp

```
bool gtest_lite::Test::tmp
```

temp a kivételkezeléshez;

Ez a dokumentáció a struktúráról a következő fájl alapján készült:

- src/external/[gtest\\_lite.h](#)

## 6.63. sf::Texture osztályreferencia

```
#include <fake_sfml.hpp>
```

### Publikus tagfüggvények

- [Texture](#) ()
- bool [loadFromFile](#) (const std::string &filepath)
- [Vector2i](#) [getSize](#) ()

### 6.63.1. Konstruktorkok és destruktorkok dokumentációja

#### 6.63.1.1. Texture()

```
sf::Texture::Texture ( )
```

### 6.63.2. Tagfüggvények dokumentációja

#### 6.63.2.1. getSize()

```
Vector2i sf::Texture::getSize ( )
```

#### 6.63.2.2. loadFromFile()

```
bool sf::Texture::loadFromFile (
    const std::string & filepath )
```

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

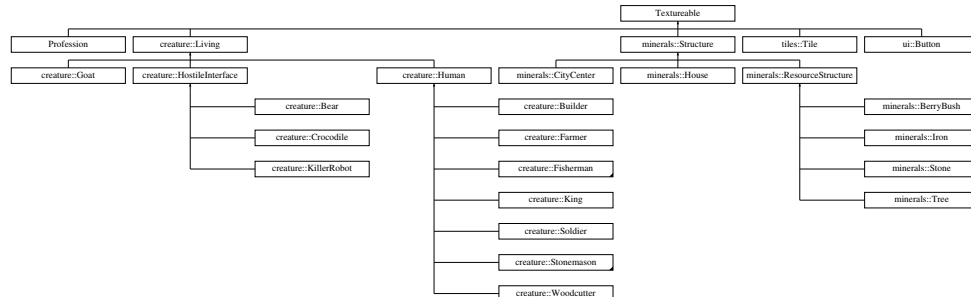
- [src/fake\\_sfml/fake\\_sfml.hpp](#)
- [src/fake\\_sfml/fake\\_sfml.cpp](#)

## 6.64. Textureable osztályreferencia

Egy interface, ami a textúrázáshoz kell.

```
#include <Textureable.hpp>
```

A Textureable osztály származási diagramja:



### Publikus tagfüggvények

- virtual `~Textureable()`=default  
*Alap virtuális destruktor.*
- virtual bool `setTexture(const std::string &filename)=0`  
*Beállít egy fájl elérési útból egy textúrát. Megvalósítástól függően esetleg 2-t.*
- virtual void `setPosition(double x, double y)=0`  
*Beállítja, hogy hova kell rajzolni a textúrázható leszármazottat.*
- virtual void `draw(sf::RenderWindow &window)=0`  
*Kirajzolja az objektumot.*

#### 6.64.1. Részletes leírás

Egy interface, ami a textúrázáshoz kell.

Jelzi, hogy a kirajzoláshoz és a világban való megjelenítéséhez milyen metódusokat kell elkészíteni.

#### 6.64.2. Konstruktork és destruktorok dokumentációja

##### 6.64.2.1. ~Textureable()

```
virtual Textureable::~~Textureable ( ) [virtual], [default]
```

Alap virtuális destruktor.

#### 6.64.3. Tagfüggvények dokumentációja

##### 6.64.3.1. draw()

```
virtual void Textureable::draw (
    sf::RenderWindow & window ) [pure virtual]
```

Kirajzolja az objektumot.

## Paraméterek

<i>window</i>	Ahova ki kell rajzolni a textúrázható leszármozottat.
---------------	---

Megvalósítják a következők: [minerals::Structure](#), [ui::Button](#), [tiles::Tile](#), [Profession](#) és [creature::Living](#).

**6.64.3.2. setPosition()**

```
virtual void Textureable::setPosition (
    double x,
    double y ) [pure virtual]
```

Beállítja, hogy hova kell rajzolni a textúrázható leszármozottat.

## Paraméterek

<i>x</i>	Az x koordináta.
<i>y</i>	Az y koordináta.

Megvalósítják a következők: [minerals::Structure](#), [ui::Button](#), [tiles::Tile](#), [Profession](#) és [creature::Living](#).

**6.64.3.3. setTexture()**

```
virtual bool Textureable::setTexture (
    const std::string & filename ) [pure virtual]
```

Beállít egy fájl elérési útból egy textúrát. Megvalósítástól függően esetleg 2-t.

## Paraméterek

<i>filename</i>	A textúra elérési útja.
-----------------	-------------------------

## Visszatérési érték

Sikeres volt-e a textúra beállítása.

Megvalósítják a következők: [minerals::Structure](#), [ui::Button](#), [tiles::Tile](#), [Profession](#) és [creature::Living](#).

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- [src/Textureable.hpp](#)

## 6.65. TextureManager osztályreferencia

A Textúra kezelő osztály.

```
#include <TextureManager.hpp>
```

### Publikus tagfüggvények

- `sf::Texture * loadTexture` (const std::string &filename)  
*Betölti a kért textúrát, ha kell. Ha nem, akkor csak visszaadja a már régen betöltött textúrát.*
- void `clear` ()  
*Kitörli az összes betöltött textúrát.*

### Statikus publikus tagfüggvények

- static `TextureManager & getInstance` ()  
*Odaadja a referenciát a singleton-ra.*

#### 6.65.1. Részletes leírás

A Textúra kezelő osztály.

A textúrák betöltéséért, tárolásáért és kiosztásáért felelős osztály. Rendelkezik egy tisztítás metódussal is.

#### 6.65.2. Tagfüggvények dokumentációja

##### 6.65.2.1. clear()

```
void TextureManager::clear ( )
```

Kitörli az összes betöltött textúrát.

##### 6.65.2.2. getInstance()

```
TextureManager & TextureManager::getInstance ( ) [static]
```

Odaadja a referenciát a singleton-ra.

Visszatérési érték

A referencia a textúramezelőre.

##### 6.65.2.3. loadTexture()

```
sf::Texture * TextureManager::loadTexture (
    const std::string & filename )
```

Betölti a kért textúrát, ha kell. Ha nem, akkor csak visszaadja a már régen betöltött textúrát.

## Paraméterek

<code>filename</code>	A textúra elérési útja.
-----------------------	-------------------------

## Visszatérési érték

Referencia a kért textúrára.

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

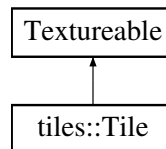
- [src/TextureManager.hpp](#)
- [src/TextureManager.cpp](#)

## 6.66. tiles::Tile osztályreferencia

A terepkocka osztály leírása.

```
#include <Tile.hpp>
```

A tiles::Tile osztály származási diagramja:



### Publikus tagfüggvények

- void [init](#) ([TILETYPE](#) newtype)  
*Inicializálja a terepkocka kinézetét a biotípusa alapján.*
- [TILETYPE get\\_type](#) () const  
*Egy getter ami visszaadja a terepkocka biotípusát.*
- bool [setTexture](#) (const std::string &filename) override  
*Beállít egy fájl elérési útból egy textúrát. Megvalósítástól függően esetleg 2-t.*
- void [setPosition](#) (double x, double y) override  
*Beállítja, hogy hova kell rajzolni a textúrázható leszármazottat.*
- void [draw](#) ([sf::RenderWindow](#) &window) override  
*Kirajzolja az objektumot.*

### 6.66.1. Részletes leírás

A terepkocka osztály leírása.

Tárolja a terepkocka kinézetét és azt, hogy milyen biom típusú.

### 6.66.2. Tagfüggvények dokumentációja

#### 6.66.2.1. draw()

```
void tiles::Tile::draw (
    sf::RenderWindow & window ) [override], [virtual]
```

Kirajzolja az objektumot.

## Paraméterek

<i>window</i>	Ahova ki kell rajzolni a textúrázható leszármozottat.
---------------	---

Megvalósítja a következőket: [Textureable](#).

**6.66.2.2. get\_type()**

```
TILETYPE tiles::Tile::get_type ( ) const
```

Egy getter ami visszaadja a terepkocka biotípusát.

## Visszatérési érték

A biotípusa.

**6.66.2.3. init()**

```
void tiles::Tile::init (
    TILETYPE newtype )
```

Inicializálja a terepkocka kinézetét a biotípusa alapján.

## Paraméterek

<i>newtype</i>	A biotípusa.
----------------	--------------

**6.66.2.4. setPosition()**

```
void tiles::Tile::setPosition (
    double x,
    double y ) [override], [virtual]
```

Beállítja, hogy hova kell rajzolni a textúrázható leszármozottat.

## Paraméterek

<i>x</i>	Az x koordináta.
<i>y</i>	Az y koordináta.

Megvalósítja a következőket: [Textureable](#).

### 6.66.2.5. setTexture()

```
bool tiles::Tile::setTexture (
    const std::string & filename ) [override], [virtual]
```

Beállít egy fájl elérési útból egy textúrát. Megvalósítástól függően esetleg 2-t.

#### Paraméterek

<i>filename</i>	A textúra elérési útja.
-----------------	-------------------------

#### Visszatérési érték

Sikeres volt-e a textúra beállítása.

Megvalósítja a következőket: [Textureable](#).

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- [src/terrain\\_tiles/Tile.hpp](#)
- [src/terrain\\_tiles/Tile.cpp](#)

## 6.67. sf::Transform osztályreferencia

```
#include <fake_sfml.hpp>
```

### Publikus tagfüggvények

- [Transform](#) ()
- [Transform](#) (float a00, float a01, float a02, float a10, float a11, float a12, float a20, float a21, float a22)
- [Transform combine](#) (const [Transform](#) &other)
- void [transformPoint](#) (float x, float y) const
- void [translate](#) (float tx, float ty)
- void [translate](#) ([Vector2f](#) Vy)

### Publikus attribútumok

- float [matrix](#) [9]

### 6.67.1. Konstruktorok és destruktorok dokumentációja



#### 6.67.1.1. Transform() [1/2]

```
sf::Transform::Transform ( )
```

#### 6.67.1.2. Transform() [2/2]

```
sf::Transform::Transform (
    float a00,
    float a01,
    float a02,
    float a10,
    float a11,
    float a12,
    float a20,
    float a21,
    float a22 )
```

### 6.67.2. Tagfüggvények dokumentációja

#### 6.67.2.1. combine()

```
Transform sf::Transform::combine (
    const Transform & other )
```

#### 6.67.2.2. transformPoint()

```
void sf::Transform::transformPoint (
    float x,
    float y ) const
```

#### 6.67.2.3. translate() [1/2]

```
void sf::Transform::translate (
    float tx,
    float ty )
```

#### 6.67.2.4. translate() [2/2]

```
void sf::Transform::translate (
    Vector2f Vy )
```

### 6.67.3. Adattagok dokumentációja

#### 6.67.3.1. matrix

```
float sf::Transform::matrix[9]
```

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

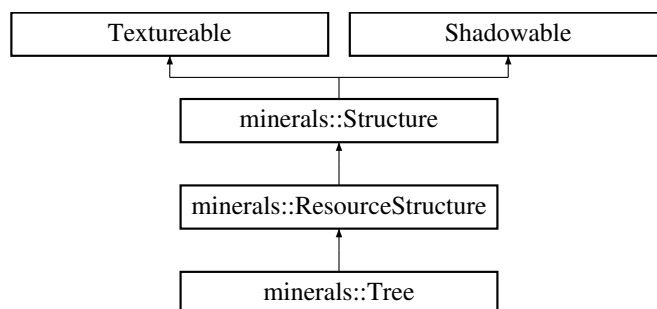
- [src/fake\\_sfml/fake\\_sfml.hpp](#)
- [src/fake\\_sfml/fake\\_sfml.cpp](#)

## 6.68. minerals::Tree osztályreferencia

A fa osztály leírása. Fát ad, ha kitermelik.

```
#include <Tree.hpp>
```

A minerals::Tree osztály származási diagramja:



### Publikus tagfüggvények

- [Tree](#) (int x, int y)  
*Konstruktor ami lerakja az erőforrást egy (x,y) pontra.*
- [MINERAL\\_TYPE get\\_type](#) () const override  
*Szimbólum, ami a fájlba mentéshez kell.*
- void [update\\_logic](#) (float deltaTime) override  
*Frissíti magát az idő függvényében.*
- bool [harvest](#) () override  
*Tisztán virtuális metódus. Ez leírja mi történik, hogy ha kitermelik ezt az erőforrást.*

## További örökölt tagok

### 6.68.1. Részletes leírás

A fa osztály leírása. Fát ad, ha kitermelik.

### 6.68.2. Konstruktorkok és destruktorkok dokumentációja

#### 6.68.2.1. Tree()

```
minerals::Tree::Tree (
    int x,
    int y )
```

Konstruktork ami lerakja az erőforrást egy (x,y) pontra.

### 6.68.3. Tagfüggvények dokumentációja

#### 6.68.3.1. get\_type()

```
MINERAL_TYPE minerals::Tree::get_type ( ) const [override], [virtual]
```

Szimbólum, ami a fájlba mentéshez kell.

Megvalósítja a következőket: [minerals::Structure](#).

#### 6.68.3.2. harvest()

```
bool minerals::Tree::harvest ( ) [override], [virtual]
```

Tisztán virtuális metódus. Ez leírja mi történik, hogy ha kitermelik ezt az erőforrást.

Megvalósítja a következőket: [minerals::ResourceStructure](#).

#### 6.68.3.3. update\_logic()

```
void minerals::Tree::update_logic (
    float deltaTime ) [override], [virtual]
```

Frissíti magát az idő függvényében.

## Paraméterek

<i>deltaTime</i>	Az előző frissítés óta eltelt idő.
------------------	------------------------------------

Megvalósítja a következőket: [minerals::Structure](#).

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- `src/world_object/Tree.hpp`
- `src/world_object/Tree.cpp`

## 6.69. sf::Vector2f osztályreferencia

```
#include <fake_sfml.hpp>
```

### Publikus tagfüggvények

- [Vector2f](#) ()
- [Vector2f](#) (float x1, float y1)

### Publikus attribútumok

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

### 6.69.1. Konstruktorok és destruktorok dokumentációja

#### 6.69.1.1. Vector2f() [1/2]

```
sf::Vector2f::Vector2f ( )
```

#### 6.69.1.2. Vector2f() [2/2]

```
sf::Vector2f::Vector2f (
    float x1,
    float y1 )
```

### 6.69.2. Adattagok dokumentációja

#### 6.69.2.1. x

```
float sf::Vector2f::x
```

#### 6.69.2.2. y

```
float sf::Vector2f::y
```

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- [src/fake\\_sfml/fake\\_sfml.hpp](#)
- [src/fake\\_sfml/fake\\_sfml.cpp](#)

## 6.70. sf::Vector2i osztályreferencia

```
#include <fake_sfml.hpp>
```

### Publikus tagfüggvények

- [Vector2i](#) ()
- [Vector2i](#) (int x1, int y1)

### Publikus attribútumok

- int [x](#)
- int [y](#)

### 6.70.1. Konstruktorkok és destruktorkok dokumentációja

#### 6.70.1.1. Vector2i() [1/2]

```
sf::Vector2i::Vector2i ( )
```

#### 6.70.1.2. Vector2i() [2/2]

```
sf::Vector2i::Vector2i (
    int x1,
    int y1 )
```

## 6.70.2. Adattagok dokumentációja

### 6.70.2.1. x

```
int sf::Vector2i::x
```

### 6.70.2.2. y

```
int sf::Vector2i::y
```

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- [src/fake\\_sfml/fake\\_sfml.hpp](#)
- [src/fake\\_sfml/fake\\_sfml.cpp](#)

## 6.71. sf::VideoMode osztályreferencia

```
#include <fake_sfml.hpp>
```

### Publikus tagfüggvények

- [VideoMode](#) (std::size\_t w=800, std::size\_t h=600)
- bool [isValid](#) () const

### Statikus publikus tagfüggvények

- static [VideoMode](#) [getDesktopMode](#) ()

### Publikus attribútumok

- std::size\_t [width](#)
- std::size\_t [height](#)
- std::size\_t [bitsPerPixel](#)

### 6.71.1. Konstruktorkok és destruktorkok dokumentációja

### 6.71.1.1. VideoMode()

```
sf::VideoMode::VideoMode (
    std::size_t w = 800,
    std::size_t h = 600 )
```

## 6.71.2. Tagfüggvények dokumentációja

### 6.71.2.1. getDesktopMode()

```
VideoMode sf::VideoMode::getDesktopMode ( ) [static]
```

< (HD)

### 6.71.2.2. isValid()

```
bool sf::VideoMode::isValid ( ) const
```

## 6.71.3. Adattagok dokumentációja

### 6.71.3.1. bitsPerPixel

```
std::size_t sf::VideoMode::bitsPerPixel
```

### 6.71.3.2. height

```
std::size_t sf::VideoMode::height
```

### 6.71.3.3. width

```
std::size_t sf::VideoMode::width
```

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

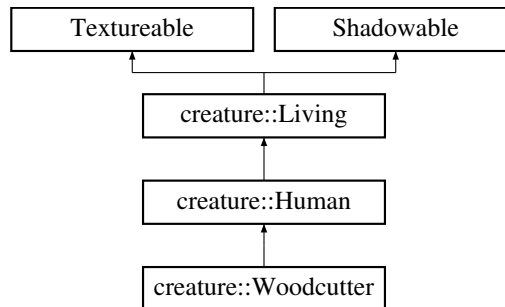
- [src/fake\\_sfml/fake\\_sfml.hpp](#)
- [src/fake\\_sfml/fake\\_sfml.cpp](#)

## 6.72. creature::Woodcutter osztályreferencia

A favágó szakmájú ember osztály leírása.

```
#include <Woodcutter.hpp>
```

A creature::Woodcutter osztály származási diagramja:



### Publikus tagfüggvények

- **Woodcutter** (int x, int y, ENTITY\_GENDER gender\_modifier)  
*Inicializál egy favágót egy pontos x és y koordinátára és beállítja az attribútumait.*
- void **update\_logic** (World &world, float deltaTime) override  
*Az entitás frissítési logikája itt van definiálva. Például a halász vizet keres, ha odaért akkor halászik.*
- **~Woodcutter** ()  
*A favágó destruktora.*

### További örökölt tagok

#### 6.72.1. Részletes leírás

A favágó szakmájú ember osztály leírása.

Ez a szakmájú ember fákat keres és kivágja őket, így fát szerez.

#### 6.72.2. Konstruktorok és destruktorok dokumentációja

##### 6.72.2.1. Woodcutter()

```
creature::Woodcutter::Woodcutter (
    int x,
    int y,
    ENTITY_GENDER gender_modifier )
```

Inicializál egy favágót egy pontos x és y koordinátára és beállítja az attribútumait.



## Paraméterek

<i>x</i>	Az x koordináta.
<i>y</i>	Az y koordináta.
<i>gender_modifier</i>	A favágó neve.

## 6.72.2.2. ~Woodcutter()

```
creature::Woodcutter::~~Woodcutter ( )
```

A favágó destruktora.

## 6.72.3. Tagfüggvények dokumentációja

## 6.72.3.1. update\_logic()

```
void creature::Woodcutter::update_logic (
    World & world,
    float deltaTime ) [override], [virtual]
```

Az entitás frissítési logikája itt van definiálva. Például a halász vizet keres, ha odaért akkor halászik.

## Paraméterek

<i>world</i>	A világ, amibe frissíti magát az entitás.
<i>deltaTime</i>	Az előző frissítés óta eltelt idő.

Újrimplementált ősök: [creature::Human](#).

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- [src/creatures/humans/Woodcutter.hpp](#)
- [src/creatures/humans/Woodcutter.cpp](#)

## 6.73. World osztályreferencia

A világ osztály leírása.

```
#include <World.hpp>
```

## Publikus tagfüggvények

- `int get_border_width () const`  
*Egy getter a horizontális kamera határ nagyságához.*
- `int get_border_height () const`  
*Egy getter a vertikális kamera határ nagyságához.*
- `void set_border_width (int newwidth)`  
*Egy setter a horizontális kamera határ nagyságához.*
- `void set_border_height (int newheight)`  
*Egy setter a vertikális kamera határ nagyságához.*
- `void clear ()`  
*Segédfüggvény a világ törléséhez. Felszabadítja az entitásokat, erőforrásokat.*
- `minerals::CityCenter * get_current_city_center ()`  
*Egy függvény, amivel kívülről el lehet érni a jelenlegi városközpontot.*
- `World ()`  
*A világ konstruktora. A konstruktor generál egy alap világot.*
- `~World ()`  
*A világ destruktora. Itt felszabadul minden, ami a világban "van". Emberek, állatok, város, erőforrások.*
- `void draw (sf::RenderWindow &window, float delta_time, int offx, int offy)`  
*Kirajzol mindent, ami a világba van.*
- `void update_world (float delta_time)`  
*Frissíti a világban lévő entitásokat, napszakot, árnyékolást.*
- `void regenerate ()`  
*Újra épít egy világot az előző világ helyére. Az előző világ tartalmát üríti.*
- `tiles::Tile & getTileAt (int x, int y) const`  
*Visszaadja a világban az [y][x]-edik terepkockát.*
- `void add_resources (std::string what, int amount)`  
*Az emberek által gyűjtött erőforrásokhoz hozzáad egy típusból valamennyit.*
- `void remove_resources (std::string what, int amount)`  
*Az emberek által gyűjtött erőforrásokból kiszed egy típusból valamennyit.*
- `bool is_there_enough_resource (std::string from_what, int needed_amount)`  
*Megnézi, hogy az emberek már szedtek-e elég erőforrást valamiből.*
- `void populate_world ()`  
*Idéz entításokat és fákat, bokrokat a világba. Ha pehhesek az emberek akkor egy gyilkos robot is megjelenik :(.*
- `template<typename T >`  
`void spawn_structure (bool mountain_exclusive)`  
*Idéz egy struktúrát. Opcionálisan bekapcsolható, hogy csak a hegyekben idéződjön.*
- `template<typename T >`  
`void spawn_structure_at (int x, int y)`  
*Idéz egy struktúrát egy pontos x és y koordinátára.*
- `template<typename T >`  
`void spawn_entity (tiles::TILETYPE goal_habitat, const std::string &savefile_identifier)`  
*Idéz egy entitást, egy bizonyos típusú biomba.*
- `sf::Vector2f get_random_suitable_position (tiles::TILETYPE suitable_tile)`  
*Keres egy optimális helyet terepkocka típus szerint.*
- `void build_city_center_at (int x, int y)`  
*Épít egy városközpontot egy x és y koordinátára. Ha már létezik városközpont, hiba keletkezik.*
- `void try_develop_random_role (creature::Human *&human_ptr)`  
*Kiválaszt egy új szakmát egy embernek. Ha ez az ember építette a várost, király lesz belőle.*
- `minerals::Structure * get_structure_type (minerals::MINERAL_TYPE atype)`  
*Keres egy olyan erőforrás struktúrát, ami bizonyos erőforrást tartalmaz.*

- void `remove_structure_at` (int x, int y)  
*Lerombol egy struktúrát egy bizonyos x és y koordinátán. Ezt az emberek hívják meg bányászatkor, favágáskor. Ha házra vagy városközpontra hívódik meg akkor hiba keletkezik.*
- `sf::Vector2f get_position_nearby_town` ()  
*Keres egy nem foglalt lakóterületnek való helyet a városközpontához közel.*
- `sf::Vector2f get_random_house_pos` ()  
*Keres egy olyan (x,y) koordinátát, amin van ház.*
- void `upgrade_house_at` (int x, int y)  
*Megpróbál megfrissíteni egy házat az (x,y) koordinátán. Ezt az építész ember hívja meg.*
- `creature::Living * get_excluded_entities` (`creature::ENTITY_TYPE` excluded\_type)  
*Visszaad egy olyan entitás típust, ami nem a specifikált típus. Ezt a ragadozó állatok hívják meg, hogy ne egymást vadásszák. Az ember vadász sem öl embereket.*

## Barátok

- `std::ostream & operator<<` (`std::ostream &os`, `const World &w`)  
*A világ adatait segíti kimenteni egy folyamba.*
- `std::ifstream & operator>>` (`std::ifstream &in`, `World &w`)  
*Egy folyamból tölti fel a világot új adatokkal.*

## 6.73.1. Részletes leírás

A világ osztály leírása.

Tárolja az erőforrásokat, embereket, entitásokat, a terepet. Rendelkezik a szimulációhoz tartozó metódusokkal

## 6.73.2. Konstruktorkok és destruktorkok dokumentációja

### 6.73.2.1. World()

```
World::World ( )
```

A világ konstruktora. A konstruktor generál egy alap világot.

### 6.73.2.2. ~World()

```
World::~~World ( )
```

A világ destruktora. Itt felszabadul minden, ami a világban "van". Emberek, állatok, város, erőforrások.

## 6.73.3. Tagfüggvények dokumentációja

### 6.73.3.1. add\_resources()

```
void World::add_resources (
    std::string what,
    int amount )
```

Az emberek által gyűjtött erőforrásokhoz hozzáad egy típusból valamennyit.

## Paraméterek

<i>what</i>	Mit adjon hozzá.
<i>amount</i>	Mennyit adjon hozzá.

**6.73.3.2. build\_city\_center\_at()**

```
void World::build_city_center_at (
    int x,
    int y )
```

Épít egy városközpontot egy x és y koordinátára. Ha már létezik városközpont, hiba keletkezik.

## Paraméterek

<i>x</i>	Az x koordináta.
<i>y</i>	Az y koordináta.

**6.73.3.3. clear()**

```
void World::clear ( )
```

Segédfüggvény a világ törléséhez. Felszabadítja az entitásokat, erőforrásokat.

**6.73.3.4. draw()**

```
void World::draw (
    sf::RenderWindow & window,
    float delta_time,
    int offx,
    int offy )
```

Kirajzol mindent, ami a világba van.

## Paraméterek

<i>window</i>	A játékalak.
<i>delta_time</i>	Az előző frissítés óta eltelt idő.
<i>offx</i>	A kamera elmozdulása horizontálisan a felső bal csücsökhöz képest.
<i>offy</i>	A kamera elmozdulása vertikálisan a felső bal csücsökhöz képest.

#### 6.73.3.5. get\_border\_height()

```
int World::get_border_height ( ) const
```

Egy getter a vertikális kamera határ nagyságához.

##### Visszatérési érték

A határ mérete.

#### 6.73.3.6. get\_border\_width()

```
int World::get_border_width ( ) const
```

Egy getter a horizontális kamera határ nagyságához.

##### Visszatérési érték

A határ mérete.

#### 6.73.3.7. get\_current\_city\_center()

```
minerals::CityCenter * World::get_current_city_center ( )
```

Egy függvény, amivel kívülről el lehet érni a jelenlegi városközpontot.

##### Visszatérési érték

A városközpont.

#### 6.73.3.8. get\_excluded\_entities()

```
creature::Living * World::get_excluded_entities (
    creature::ENTITY_TYPE excluded_type )
```

Visszaad egy olyan entitás típust, ami nem a specifikált típus. Ezt a ragadozó állatok hívják meg, hogy ne egymást vadásszák. Az ember vadász sem öl embereket.

##### Paraméterek

<i>excluded_type</i>	A típus amit nem akar megkapni.
----------------------	---------------------------------

**Visszatérési érték**

Egy entitás pointer.

**6.73.3.9. get\_position\_nearby\_town()**

```
sf::Vector2f World::get_position_nearby_town ( )
```

Keres egy nem foglalt lakóterületnek való helyet a városközponthoz közel.

**Visszatérési érték**

Egy (x,y) koordináta, ahova lehet házat építeni. Ha nincs ilyen akkor (-1,-1).

**6.73.3.10. get\_random\_house\_pos()**

```
sf::Vector2f World::get_random_house_pos ( )
```

Keres egy olyan (x,y) koordinátát, amin van ház.

**Visszatérési érték**

Egy (x,y) koordinátpár ahol van ház. (-1,-1) ha nincs ilyen.

**6.73.3.11. get\_random\_suitable\_position()**

```
sf::Vector2f World::get_random_suitable_position (
    tiles::TILETYPE suitable_tile )
```

Keres egy optimális helyet terepkocka típus szerint.

**Paraméterek**

<i>suitable_tile</i>	Az optimális terepkocka típusa.
----------------------	---------------------------------

**Visszatérési érték**

A szabad koordináta vektora. Ha nincs jó hely akkor a (-1,-1) vektor.

**6.73.3.12. get\_structure\_type()**

```
minerals::Structure * World::get_structure_type (
    minerals::MINERAL_TYPE atype )
```

Keres egy olyan erőforrás struktúrát, ami bizonyos erőforrást tartalmaz.

**Paraméterek**

<i>atype</i>	Az erőforrás típusa, ami keresett.
--------------	------------------------------------

**Visszatérési érték**

Egy struktúra pointer.

**6.73.3.13. getTileAt()**

```
tiles::Tile & World::getTileAt (
    int x,
    int y ) const
```

Visszaadja a világban az [y][x]-edik terepkockát.

**Paraméterek**

<i>x</i>	Oszlop index.
<i>y</i>	Sor index.

**Visszatérési érték**

Az x.Oszlop y.Sor-i terepkocka.

**6.73.3.14. is\_there\_enough\_resource()**

```
bool World::is_there_enough_resource (
    std::string from_what,
    int needed_amount )
```

Megnézi, hogy az emberek már szedtek-e elég erőforrást valamiből.

**Paraméterek**

<i>Miből</i>	kell.
<i>Mennyi</i>	kell, hogy legyen.

**Visszatérési érték**

Ha van elég, akkor igaz, ha nincs akkor hamis.

**6.73.3.15. populate\_world()**

```
void World::populate_world ( )
```

Idéz entitásokat és fákat, bokrokat a világba. Ha pehhesek az emberek akkor egy gyilkos robot is megjelenik :(.

**6.73.3.16. regenerate()**

```
void World::regenerate ( )
```

Újra épít egy világot az előző világ helyére. Az előző világ tartalmát üríti.

**6.73.3.17. remove\_resources()**

```
void World::remove_resources (
    std::string what,
    int amount )
```

Az emberek által gyűjtött erőforrásokból kised egy típusból valamennyit.

**Paraméterek**

<i>Mit</i>	vesz el.
------------	----------

**Visszatérési érték**

Mennyit vegyen el.

**6.73.3.18. remove\_structure\_at()**

```
void World::remove_structure_at (
    int x,
    int y )
```

Lerombol egy struktúrát egy bizonyos x és y koordinátán. Ezt az emberek hívják meg bányászatkor, favágáskor. Ha házra vagy városközpontre hívódik meg akkor hiba keletkezik.



## Paraméterek

<i>x</i>	Az x koordináta.
<i>y</i>	Az y koordináta.

**6.73.3.19. set\_border\_height()**

```
void World::set_border_height (
    int newheight )
```

Egy setter a vertikális kamera határ nagyságához.

## Paraméterek

<i>newheight</i>	Az új határ méret.
------------------	--------------------

**6.73.3.20. set\_border\_width()**

```
void World::set_border_width (
    int newwidth )
```

Egy setter a horizontális kamera határ nagyságához.

## Paraméterek

<i>newwidth</i>	Az új határ méret.
-----------------	--------------------

**6.73.3.21. spawn\_entity()**

```
template<typename T >
void World::spawn_entity (
    tiles::TILETYPE goal_habitat,
    const std::string & savefile_identifier )
```

Idéz egy entitást, egy bizonyos típusú biomba.

## Sablon paraméterek

<i>T</i>	Az entitás típusa.
----------	--------------------

## Paraméterek

<i>goal_habitat</i>	A terepkocka, ami a cél.
<i>savefile_identifier</i>	Ez egy azonosító, így fog a mentés fájlba megjelenni.

**6.73.3.22. spawn\_structure()**

```
template<typename T >
void World::spawn_structure (
    bool mountain_exclusive )
```

Idéz egy struktúrát. Opcionálisan bekapcsolható, hogy csak a hegyekben idéződjön.

## Sablon paraméterek

<i>T</i>	A struktúra.
----------	--------------

## Paraméterek

<i>mountain_exclusive</i>	Ha igaz, akkor csak a hegyekben idéződik a struktúra.
---------------------------	---

**6.73.3.23. spawn\_structure\_at()**

```
template<typename T >
void World::spawn_structure_at (
    int x,
    int y )
```

Idéz egy struktúrát egy pontos x és y koordinátára.

## Sablon paraméterek

<i>T</i>	A struktúra.
----------	--------------

## Paraméterek

<i>x</i>	Az x koordináta.
<i>y</i>	Az y koordináta.

#### 6.73.3.24. try\_develop\_random\_role()

```
void World::try_develop_random_role (
    creature::Human * & human_ptr )
```

Kiválaszt egy új szakmát egy embernek. Ha ez az ember építette a várost, király lesz belőle.

##### Paraméterek

<i>human_ptr</i>	Az ember pointer referenciája, akinek új szakmát kell adni.
------------------	---

#### 6.73.3.25. update\_world()

```
void World::update_world (
    float delta_time )
```

Frissíti a világban lévő entitásokat, napszakot, árnyékolást.

##### Paraméterek

<i>delta_time</i>	Az előző frissítés óta eltelt idő.
-------------------	------------------------------------

#### 6.73.3.26. upgrade\_house\_at()

```
void World::upgrade_house_at (
    int x,
    int y )
```

Megpróbál megfrissíteni egy házat az (x,y) koordinátán. Ezt az építész ember hívja meg.

##### Paraméterek

<i>x</i>	Az x koordináta.
<i>y</i>	Az y koordináta.

### 6.73.4. Barát és kapcsolódó függvények dokumentációja

#### 6.73.4.1. operator<<

```
std::ostream& operator<< (
    std::ostream & os,
    const World & w ) [friend]
```

A világ adatait segíti kimenteni egy folyamba.

#### 6.73.4.2. `operator>>`

```
std::ifstream& operator>> (  
    std::ifstream & in,  
    World & w ) [friend]
```

Egy folyamból tölti fel a világot új adatokkal.

Ez a dokumentáció az osztályról a következő fájlok alapján készült:

- [src/World.hpp](#)
- [src/World.cpp](#)

## 7. fejezet

# Fájlok dokumentációja

### 7.1. src/creatures/Goat.cpp fájlreferencia

```
#include "Goat.hpp"  
#include "../World.hpp"
```

#### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

### 7.2. src/creatures/Goat.d fájlreferencia

### 7.3. src/creatures/Goat.hpp fájlreferencia

A Kecske osztály itt van deklarálva.

```
#include "Living.hpp"  
#include "../Random_Gen.hpp"  
#include "../Utils.hpp"
```

#### Osztályok

- class [creature::Goat](#)

*A kecske osztály leírása.*

#### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

### 7.3.1. Részletes leírás

A Kecske osztály itt van deklarálva.

Szerző

Funk Gábor

Dátum

2025-04-21

## 7.4. src/creatures/HostileInterface.cpp fájlreferencia

```
#include "HostileInterface.hpp"
#include "../World.hpp"
```

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

## 7.5. src/creatures/HostileInterface.d fájlreferencia

## 7.6. src/creatures/HostileInterface.hpp fájlreferencia

A vadállat interface itt van deklarálva.

```
#include "Living.hpp"
```

### Osztályok

- class [creature::HostileInterface](#)

*A vadállat entitások interface leírása.*

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

### 7.6.1. Részletes leírás

A vadállat interface itt van deklarálva.

#### Szerző

Funk Gábor

#### Dátum

2025-04-21

## 7.7. src/creatures/hostiles/Bear.cpp fájlreferencia

```
#include "Bear.hpp"
#include "../../World.hpp"
```

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

## 7.8. src/creatures/hostiles/Bear.d fájlreferencia

## 7.9. src/creatures/hostiles/Bear.hpp fájlreferencia

A Medve osztály itt van deklarálva.

```
#include "../HostileInterface.hpp"
#include "../../Random_Gen.hpp"
#include "../../Utils.hpp"
```

### Osztályok

- class [creature::Bear](#)

*A medve osztály leírása.*

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

### 7.9.1. Részletes leírás

A Medve osztály itt van deklarálva.

Szerző

Funk Gábor

Dátum

2025-04-21

## 7.10. src/creatures/hostiles/Crocodile.cpp fájlreferencia

```
#include "Crocodile.hpp"  
#include "../World.hpp"
```

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

## 7.11. src/creatures/hostiles/Crocodile.d fájlreferencia

## 7.12. src/creatures/hostiles/Crocodile.hpp fájlreferencia

A krokodil osztály itt van deklarálva.

```
#include "../HostileInterface.hpp"  
#include "../Random_Gen.hpp"  
#include "../Utils.hpp"
```

### Osztályok

- class [creature::Crocodile](#)

*A krokodil osztály leírása.*

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*



### 7.12.1. Részletes leírás

A krokodil osztály itt van deklarálva.

Szerző

Funk Gábor

Dátum

2025-04-21

## 7.13. src/creatures/hostiles/KillerRobot.cpp fájlreferencia

```
#include "KillerRobot.hpp"
#include "../World.hpp"
```

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

## 7.14. src/creatures/hostiles/KillerRobot.d fájlreferencia

## 7.15. src/creatures/hostiles/KillerRobot.hpp fájlreferencia

A Gyilkos Robot osztály itt van deklarálva.

```
#include "../HostileInterface.hpp"
#include "../Random_Gen.hpp"
#include "../Utils.hpp"
```

### Osztályok

- class [creature::KillerRobot](#)

*A gyilkos robot osztály leírása.*

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

### 7.15.1. Részletes leírás

A Gyilkos Robot osztály itt van deklarálva.

Szerző

Funk Gábor

Dátum

2025-04-21

## 7.16. src/creatures/humans/AnglerMiner.cpp fájlreferencia

```
#include "AnglerMiner.hpp"  
#include "../World.hpp"
```

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

## 7.17. src/creatures/humans/AnglerMiner.d fájlreferencia

## 7.18. src/creatures/humans/AnglerMiner.hpp fájlreferencia

Az "AnglerMiner" szakmájú ember osztály itt van deklarálva.

```
#include "Fisherman.hpp"  
#include "Stonemason.hpp"
```

### Osztályok

- class [creature::AnglerMiner](#)

*Az "AnglerMiner" szakmájú ember osztály leírása.*

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

### 7.18.1. Részletes leírás

Az "AnglerMiner" szakmájú ember osztály itt van deklarálva.

Szerző

Funk Gábor

Dátum

2025-04-21

## 7.19. src/creatures/humans/Builder.cpp fájlreferencia

```
#include "Builder.hpp"  
#include "../World.hpp"
```

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

## 7.20. src/creatures/humans/Builder.d fájlreferencia

## 7.21. src/creatures/humans/Builder.hpp fájlreferencia

Az építész szakmájú ember osztály itt van deklarálva.

```
#include "Human.hpp"
```

### Osztályok

- class [creature::Builder](#)

*Az építész szakmájú ember osztály leírása.*

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

### 7.21.1. Részletes leírás

Az építész szakmájú ember osztály itt van deklarálva.

Szerző

Funk Gábor

Dátum

2025-04-21

## 7.22. src/creatures/humans/Farmer.cpp fájlreferencia

```
#include "Farmer.hpp"
#include "../World.hpp"
```

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

## 7.23. src/creatures/humans/Farmer.d fájlreferencia

## 7.24. src/creatures/humans/Farmer.hpp fájlreferencia

A farmer szakmájú ember osztály itt van deklarálva.

```
#include "Human.hpp"
```

### Osztályok

- class [creature::Farmer](#)

*A farmer szakmájú ember osztály leírása.*

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

### 7.24.1. Részletes leírás

A farmer szakmájú ember osztály itt van deklarálva.

Szerző

Funk Gábor

Dátum

2025-04-21

## 7.25. src/creatures/humans/Fisherman.cpp fájlreferencia

```
#include "Fisherman.hpp"  
#include "../World.hpp"
```

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

## 7.26. src/creatures/humans/Fisherman.d fájlreferencia

## 7.27. src/creatures/humans/Fisherman.hpp fájlreferencia

A halász szakmájú ember osztály itt van deklarálva.

```
#include "Human.hpp"
```

### Osztályok

- class [creature::Fisherman](#)

*A halász szakmájú ember osztály leírása.*

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

### 7.27.1. Részletes leírás

A halász szakmájú ember osztály itt van deklarálva.

Szerző

Funk Gábor

Dátum

2025-04-21

## 7.28. src/creatures/humans/Human.cpp fájlreferencia

```
#include "Human.hpp"  
#include "../World.hpp"
```

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

## 7.29. src/creatures/humans/Human.d fájlreferencia

## 7.30. src/creatures/humans/Human.hpp fájlreferencia

Az alap ember osztály itt van deklarálva.

```
#include "../Living.hpp"  
#include "../Random_Gen.hpp"  
#include "../Profession.hpp"  
#include "../world_object/CityCenter.hpp"  
#include "../Utils.hpp"
```

### Osztályok

- class [creature::Human](#)

*Az alap ember osztály leírása. Minden fajta szakmájú ember innen öröklődik.*

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

### 7.30.1. Részletes leírás

Az alap ember osztály itt van deklarálva.

Szerző

Funk Gábor

Dátum

2025-04-21

## 7.31. src/creatures/humans/King.cpp fájlreferencia

```
#include "King.hpp"
#include "../World.hpp"
```

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

## 7.32. src/creatures/humans/King.d fájlreferencia

## 7.33. src/creatures/humans/King.hpp fájlreferencia

A király szakmájú ember osztály itt van deklarálva.

```
#include "Human.hpp"
```

### Osztályok

- class [creature::King](#)

*A király szakmájú ember osztály leírása.*

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

### 7.33.1. Részletes leírás

A király szakmájú ember osztály itt van deklarálva.

Szerző

Funk Gábor

Dátum

2025-04-21

## 7.34. src/creatures/humans/Soldier.cpp fájlreferencia

```
#include "Soldier.hpp"  
#include "../World.hpp"
```

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

## 7.35. src/creatures/humans/Soldier.d fájlreferencia

## 7.36. src/creatures/humans/Soldier.hpp fájlreferencia

A katona szakmájú ember osztály itt van deklarálva.

```
#include "Human.hpp"
```

### Osztályok

- class [creature::Soldier](#)

*A katona szakmájú ember osztály leírása.*

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*



### 7.36.1. Részletes leírás

A katona szakmájú ember osztály itt van deklarálva.

Szerző

Funk Gábor

Dátum

2025-04-21

## 7.37. src/creatures/humans/Stonemason.cpp fájlreferencia

```
#include "Stonemason.hpp"  
#include "../World.hpp"
```

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

## 7.38. src/creatures/humans/Stonemason.d fájlreferencia

## 7.39. src/creatures/humans/Stonemason.hpp fájlreferencia

A bányász szakmájú ember osztály itt van deklarálva.

```
#include "Human.hpp"
```

### Osztályok

- class [creature::Stonemason](#)

*A bányász szakmájú ember osztály leírása.*

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

### 7.39.1. Részletes leírás

A bányász szakmájú ember osztály itt van deklarálva.

Szerző

Funk Gábor

Dátum

2025-04-21

## 7.40. src/creatures/humans/Woodcutter.cpp fájlreferencia

```
#include "Woodcutter.hpp"  
#include "../World.hpp"
```

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

## 7.41. src/creatures/humans/Woodcutter.d fájlreferencia

## 7.42. src/creatures/humans/Woodcutter.hpp fájlreferencia

A favágó szakmájú ember osztály itt van deklarálva.

```
#include "Human.hpp"
```

### Osztályok

- class [creature::Woodcutter](#)

*A favágó szakmájú ember osztály leírása.*

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

### 7.42.1. Részletes leírás

A favágó szakmájú ember osztály itt van deklarálva.

Szerző

Funk Gábor

Dátum

2025-04-21

## 7.43. src/creatures/Living.cpp fájlreferencia

```
#include "Living.hpp"
```

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

## 7.44. src/creatures/Living.d fájlreferencia

## 7.45. src/creatures/Living.hpp fájlreferencia

Az élő interface itt van deklarálva.

```
#include "../Textureable.hpp"  
#include "../TextureManager.hpp"  
#include "../GameConfig.hpp"  
#include "../Shadowable.hpp"  
#include <string>  
#include <iostream>
```

### Osztályok

- class [creature::Living](#)

*Az élő entitások interface leírása.*

### Névterek

- [creature](#)

*Az összes élőlény és entitás ebben a névtérben van.*

## Enumerációk

- enum class `creature::ENTITY_TYPE` : char { `creature::HUMAN` , `creature::ANIMAL` , `creature::ROBOTIC` }
- enum class `creature::ENTITY_GENDER` : char { `creature::MALE` , `creature::FEMALE` }
- enum class `creature::FACING` : bool { `creature::RIGHT` , `creature::LEFT` }
- enum class `creature::LIVINGSTATE` : int {  
    `creature::IDLE` , `creature::RUN` , `creature::WALK` , `creature::DEATH` ,  
    `creature::ATTACKING` , `creature::DOING_ITS_WORK` }

### 7.45.1. Részletes leírás

Az élő interface itt van deklarálva.

Szerző

Funk Gábor

Dátum

2025-04-21

## 7.46. src/exceptions/FileExceptions.hpp fájlreferencia

Ebben a fájlban vannak deklarálva azok a hibák, amik az IO-hoz kapcsolódnak.

```
#include "SimulationException.hpp"
```

## Osztályok

- class `ImportInvalidEntityException`  
    *Akkor kell dobni, ha egy entitás hibásan lett beolvasva.*
- class `ImportInvalidHumanProfessionException`  
    *Akkor kell dobni, ha egy szakma hibásan lett beolvasva.*
- class `ImportInvalidHousingLevelException`  
    *Akkor kell dobni, ha egy ház hibásan lett beolvasva.*
- class `ImportInvalidResourceException`  
    *Akkor kell dobni, ha egy erőforrás hibásan lett beolvasva.*
- class `ReadSaveFileFail`  
    *Akkor kell dobni, ha egy IO mentés, importálás vagy törlés hibás.*

### 7.46.1. Részletes leírás

Ebben a fájlban vannak deklarálva azok a hibák, amik az IO-hoz kapcsolódnak.

Szerző

Funk Gábor

Dátum

2025-04-21

## 7.47. src/exceptions/MusicLoadException.hpp fájlreferencia

Ebben a fájlban vannak deklarálva azok a hibák, amik az Zenéhez kapcsolódnak.

```
#include "SimulationException.hpp"
```

### Osztályok

- class [MusicLoadException](#)

*Akkor kell dobni, ha egy zene hibásan lett lejátszva vagy hibás a beolvasott zene.*

#### 7.47.1. Részletes leírás

Ebben a fájlban vannak deklarálva azok a hibák, amik az Zenéhez kapcsolódnak.

##### Szerző

Funk Gábor

##### Dátum

2025-04-21

## 7.48. src/exceptions/SimulationException.hpp fájlreferencia

Ebben a fájlban vannak deklarálva azok a hibák, amik az alap Szimulációhoz kapcsolódnak. Ezekből a hibákból öröklődik minden olyan hiba, amit a program kezel és dob.

```
#include <stdexcept>
#include <string>
```

### Osztályok

- class [SimulationException](#)

*Akkor kell dobni, ha egy szimulációs elem hibásan viselkedik.*

#### 7.48.1. Részletes leírás

Ebben a fájlban vannak deklarálva azok a hibák, amik az alap Szimulációhoz kapcsolódnak. Ezekből a hibákból öröklődik minden olyan hiba, amit a program kezel és dob.

##### Szerző

Funk Gábor

##### Dátum

2025-04-21

## 7.49. src/exceptions/WorldExceptions.hpp fájlreferencia

Ebben a fájlban vannak deklarálva azok a hibák, amik az Világhoz kapcsolódnak.

```
#include "SimulationException.hpp"
```

### Osztályok

- class [CityCenterException](#)  
*Akkor kell dobni, ha a városközpont hibásan működött.*
- class [StructureException](#)  
*Akkor kell dobni, ha egy struktúra hibásan működött.*
- class [InvalidBorderSizeException](#)  
*Akkor kell dobni, ha egy világhatárnak nem jó értéket akarnak beállítani.*

### 7.49.1. Részletes leírás

Ebben a fájlban vannak deklarálva azok a hibák, amik az Világhoz kapcsolódnak.

#### Szerző

Funk Gábor

#### Dátum

2025-04-21

## 7.50. src/external/gtest\_lite.h fájlreferencia

```
#include <iostream>
#include <cassert>
#include <cmath>
#include <cstring>
#include <limits>
#include <cstdlib>
#include <string>
#include <fstream>
```

### Osztályok

- struct [\\_Is\\_Types< F, T >](#)  
*Segédsablon típuskonverzió futás közbeni ellenőrzésére.*
- struct [gtest\\_lite::Test](#)
- class [gtest\\_lite::ostreamRedir](#)

## Névterek

- [gtest\\_lite](#)

*gtest\_lite*: a keretrendszer függvényinek és objektumainak névtére

## Makródefiníciók

- `#define TEST(C, N) do { gtest_lite::test.begin(#C"."#N);`
- `#define END gtest_lite::test.end(); } while (false);`  
*Tesztelés vége.*
- `#define ENDM gtest_lite::test.end(true); } while (false);`
- `#define ENDMsg(t) gtest_lite::test.end(true) << t << std::endl; } while (false);`
- `#define SUCCEED() gtest_lite::test.expect(true, __FILE__, __LINE__, "SUCCEED()", true)`  
*Sikeres teszt makrója.*
- `#define FAIL() gtest_lite::test.expect(false, __FILE__, __LINE__, "FAIL()", true)`  
*Sikertelen teszt fatális hiba makrója.*
- `#define ADD_FAILURE() gtest_lite::test.expect(false, __FILE__, __LINE__, "ADD_FAILURE()", true)`  
*Sikertelen teszt makrója.*
- `#define EXPECT_EQ(expected, actual) gtest_lite::EXPECT_(expected, actual, gtest_lite::eq, __FILE__, __LINE__, "EXPECT_EQ(" #expected ", " #actual ")")`  
*Azonosságot elváró makró*
- `#define EXPECT_NE(expected, actual) gtest_lite::EXPECT_(expected, actual, gtest_lite::ne, __FILE__, __LINE__, "EXPECT_NE(" #expected ", " #actual ")", "etalon")`  
*Eltérést elváró makró*
- `#define EXPECT_LE(expected, actual) gtest_lite::EXPECT_(expected, actual, gtest_lite::le, __FILE__, __LINE__, "EXPECT_LE(" #expected ", " #actual ")", "etalon")`  
*Kisebb, vagy egyenlő relációt elváró makró*
- `#define EXPECT_LT(expected, actual) gtest_lite::EXPECT_(expected, actual, gtest_lite::lt, __FILE__, __LINE__, "EXPECT_LT(" #expected ", " #actual ")", "etalon")`  
*Kisebb, mint relációt elváró makró*
- `#define EXPECT_GE(expected, actual) gtest_lite::EXPECT_(expected, actual, gtest_lite::ge, __FILE__, __LINE__, "EXPECT_GE(" #expected ", " #actual ")", "etalon")`  
*Nagyobb, vagy egyenlő relációt elváró makró*
- `#define EXPECT_GT(expected, actual) gtest_lite::EXPECT_(expected, actual, gtest_lite::gt, __FILE__, __LINE__, "EXPECT_GT(" #expected ", " #actual ")", "etalon")`  
*Nagyobb, mint relációt elváró makró*
- `#define EXPECT_TRUE(actual) gtest_lite::EXPECT_(true, actual, gtest_lite::eq, __FILE__, __LINE__, "EXPECT_TRUE(" #actual ")")`  
*Igaz értéket elváró makró*
- `#define EXPECT_FALSE(actual) gtest_lite::EXPECT_(false, actual, gtest_lite::eq, __FILE__, __LINE__, "EXPECT_FALSE(" #actual ")")`  
*Hamis értéket elváró makró*
- `#define EXPECT_FLOAT_EQ(expected, actual) gtest_lite::EXPECT_(expected, actual, gtest_lite::almostEQ, __FILE__, __LINE__, "EXPECT_FLOAT_EQ(" #expected ", " #actual ")")`  
*Valós számok azonosságát elváró makró*
- `#define EXPECT_DOUBLE_EQ(expected, actual) gtest_lite::EXPECT_(expected, actual, gtest_lite::almostEQ, __FILE__, __LINE__, "EXPECT_DOUBLE_EQ(" #expected ", " #actual ")")`  
*Valós számok azonosságát elváró makró*
- `#define EXPECT_STREQ(expected, actual) gtest_lite::EXPECTSTR_(expected, actual, gtest_lite::eqstr, __FILE__, __LINE__, "EXPECT_STREQ(" #expected ", " #actual ")")`  
*C stringek (const char \*) azonosságát tesztelő makró*

- `#define EXPECT_STRNE(expected, actual) gtest_lite::EXPECTSTR(expected, actual, gtest_lite::nestr, __FILE__, __LINE__, "EXPECT_STRNE(" #expected ", " #actual ")", "etalon")`  
*C stringek (const char \*) eltéréset tesztelő makró*
- `#define EXPECT_STRCASEEQ(expected, actual) gtest_lite::EXPECTSTR(expected, actual, gtest_lite::eqstrcase, __FILE__, __LINE__, "EXPECT_STRCASEEQ(" #expected ", " #actual ")", "etalon")`  
*C stringek (const char \*) azonosságát tesztelő makró (kisbetű/nagybetű azonos)*
- `#define EXPECT_STRCASENE(expected, actual) gtest_lite::EXPECTSTR(expected, actual, gtest_lite::nestrcase, __FILE__, __LINE__, "EXPECT_STRCASENE(" #expected ", " #actual ")", "etalon")`  
*C stringek (const char \*) eltéréset tesztelő makró (kisbetű/nagybetű azonos)*
- `#define EXPECT_THROW(statement, exception_type)`  
*Kivételt várunk.*
- `#define EXPECT_ANY_THROW(statement)`  
*Kivételt várunk.*
- `#define EXPECT_NO_THROW(statement)`  
*Nem várunk kivételt.*
- `#define ASSERT_NO_THROW(statement)`  
*Nem várunk kivételt.*
- `#define EXPECT_THROW_THROW(statement, exception_type)`  
*Kivételt várunk és továbbdobjuk – ilyen nincs a gtest-ben.*
- `#define EXPECT_ENVEQ(expected, actual) gtest_lite::EXPECTSTR(std::getenv(expected), actual, gtest_lite::eqstr, __FILE__, __LINE__, "EXPECT_ENVEQ(" #expected ", " #actual ")", "etalon")`  
*Környezeti változóhoz hasonlít – ilyen nincs a gtest-ben.*
- `#define EXPECT_ENVCASEEQ(expected, actual) gtest_lite::EXPECTSTR(std::getenv(expected), actual, gtest_lite::eqstrcase, __FILE__, __LINE__, "EXPECT_ENVCASEEQ(" #expected ", " #actual ")", "etalon")`  
*Környezeti változóhoz hasonlít – ilyen nincs a gtest-ben (kisbetű/nagybetű azonos)*
- `#define ASSERT_EQ(expected, actual) gtest_lite::ASSERT_(expected, actual, gtest_lite::eq, "ASSERT_EQ")`  
*Azonosságot elváró makró*
- `#define ASSERT_NO_THROW(statement)`  
*Nem várunk kivételt.*
- `#define CREATE_Has_(X)`
- `#define CREATE_Has_fn_(X, S)`
- `#define EXPECT_THROW(statement, exp, act)`  
*EXPECT\_THROW: kivételkezelés.*
- `#define ASSERT_THROW(statement, exp, act)`
- `#define ASSERT_(expected, actual, fn, op)`
- `#define GTINIT(IS)`
- `#define GTEND(os)`

## Függvények

- `void hasMember (...)`
- `template<typename T1, typename T2>  
std::ostream & gtest_lite::EXPECT_(T1 exp, T2 act, bool(*pred)(T1, T1), const char *file, int line, const char *expr, const char *lhs="elvart", const char *rhs="aktual")`  
*általános sablon a várt értékhez.*
- `template<typename T1, typename T2>  
std::ostream & gtest_lite::EXPECT_(T1 *exp, T2 *act, bool(*pred)(T1 *, T1 *), const char *file, int line, const char *expr, const char *lhs="elvart", const char *rhs="aktual")`  
*pointerre specializált sablon a várt értékhez.*
- `std::ostream & gtest_lite::EXPECTSTR(const char *exp, const char *act, bool(*pred)(const char *, const char *), const char *file, int line, const char *expr, const char *lhs="elvart", const char *rhs="aktual")`



- `template<typename T>`  
`bool gtest_lite::eq (T a, T b)`
- `bool gtest_lite::eqstr (const char *a, const char *b)`
- `bool gtest_lite::eqstrcase (const char *a, const char *b)`
- `template<typename T>`  
`bool gtest_lite::ne (T a, T b)`
- `bool gtest_lite::nestr (const char *a, const char *b)`
- `template<typename T>`  
`bool gtest_lite::le (T a, T b)`
- `template<typename T>`  
`bool gtest_lite::lt (T a, T b)`
- `template<typename T>`  
`bool gtest_lite::ge (T a, T b)`
- `template<typename T>`  
`bool gtest_lite::gt (T a, T b)`
- `template<typename T>`  
`bool gtest_lite::almostEQ (T a, T b)`

### 7.50.1. Részletes leírás

(v4/2022)

Google gtest keretrendszerhez hasonló rendszer. Sz.l. 2015., 2016., 2017. (*Has\_X*) Sz.l. 2018 (*template*), *ENDM*, *ENDMsg*, *nullptr\_t* Sz.l. 2019 *singleton* Sz.l. 2021 *ASSERT...*, *STRCASE...* Sz.l. 2021 *EXPECT\_REGEX*, *CREATE\_Has\_fn*, *cmp w. NULL*, *EXPECT\_param* fix V.B., Sz.l. 2022 *almostEQ* fix, Sz.l. 2022. *EXPECT\_THROW* fix

A tesztelés legalapvetőbb funkcióit támogató függvények és makrók. Nem szálbiztos megvalósítás.

Szabadon felhasználható, bővíthető.

Használati példa: Teszteljük az  $f(x)=2*x$  függvényt: `int f(int x) { return 2*x; }`

```
int main() { TEST(TeszEsetNeve, TesztNeve) EXPECT_EQ(0, f(0)); EXPECT_EQ(4, f(2)) << "A függvény hibás
eredményt adott" << std::endl; ... END ... // Fatális hiba esetén a tesztelés nem fut tovább. Ezek az AS-
SERT... makrók. // Nem lehet a kiírásukhoz további üzenetet fűzni. PL: TEST(TeszEsetNeve, TesztNeve)
ASSERT_NO_THROW(f(0)); // itt nem lehet << "duma" EXPECT_EQ(4, f(2)) << "A függvény hibás eredményt
adott" << std::endl; ... END ...
```

A működés részleteinek megértése szorgalmi feladat.

### 7.50.2. Makródefiníciók dokumentációja

#### 7.50.2.1. ADD\_FAILURE

```
#define ADD_FAILURE( ) gtest_lite::test.expect(false, __FILE__, __LINE__, "ADD_FAILURE()",
true)
```

Sikertelen teszt makrója.

### 7.50.2.2. ASSERT\_

```
#define ASSERT_(
    expected,
    actual,
    fn,
    op )
```

Érték:

```
EXPECT_(expected, actual, fn, __FILE__, __LINE__, #op "(" #expected ", " #actual ")" ); \
if (!gtest_lite::test.status) { gtest_lite::test.end(); break; }
```

### 7.50.2.3. ASSERT\_EQ

```
#define ASSERT_EQ(
    expected,
    actual ) gtest_lite::ASSERT_(expected, actual, gtest_lite::eq, "ASER_EQ")
```

Azonosságot elváró makró

ASSERT típusú ellenőrzések. CSak 1-2 van megvalósítva. Nem ostream& -val térnek vissza !!! Kivételt várunk

---

### 7.50.2.4. ASSERT\_NO\_THROW [1/2]

```
#define ASSERT_NO_THROW(
    statement )
```

Érték:

```
try { gtest_lite::test.tmp = true; statement; } \
catch (...) { gtest_lite::test.tmp = false; }\
ASSERTTHROW(statement, "nem dob kivetelt.", "kivetelt dobott.")
```

Nem várunk kivételt.

### 7.50.2.5. ASSERT\_NO\_THROW [2/2]

```
#define ASSERT_NO_THROW(
    statement )
```

Érték:

```
try { gtest_lite::test.tmp = true; statement; } \
catch (...) { gtest_lite::test.tmp = false; }\
ASSERTTHROW(statement, "nem dob kivetelt.", "kivetelt dobott.")
```

Nem várunk kivételt.

### 7.50.2.6. ASSERTTHROW

```
#define ASSERTTHROW(
    statement,
    exp,
    act )
```

Érték:

```
gtest_lite::test.expect(gtest_lite::test.tmp, __FILE__, __LINE__, #statement) \
« "*** Az utasítás " « (act) \
« "\n** Azt vartuk, hogy " « (exp) « std::endl; if (!gtest_lite::test.status) { gtest_lite::test.end(); \
break; }
```

**7.50.2.7. CREATE\_Has\_**

```
#define CREATE_Has_(
    X )
```

**Érték:**

```
template<typename T> struct _Has_##X { \
    struct Fallback { int X; }; \
    struct Derived : T, Fallback {}; \
    template<typename C, C> struct ChT; \
    template<typename D> static char (&f(ChT<int Fallback::*, &D::X>*)) [1]; \
    template<typename D> static char (&f(...)) [2]; \
    static bool const member = sizeof(f<Derived>(0)) == 2; \
};
```

Segédmakró egy adattag, vagy tagfüggvény létezésének tesztelésére futási időben Ötlet: <https://cpptalk.wordpress.com/2009/09/12/substitution-failure-is-not-an-error-2>

Használat: `CREATE_Has_(size) ... if (_Has_size<std::string>::member)...`

**7.50.2.8. CREATE\_Has\_fn\_**

```
#define CREATE_Has_fn_(
    X,
    S )
```

**Érték:**

```
template<typename R, typename T> struct _Has_fn_##X##_##S { \
    template<typename C, R (C::*f)() S> struct ChT; \
    template<typename D> static char (&f(ChT<D, &D::X>*)) [1]; \
    template<typename D> static char (&f(...)) [2]; \
    static bool const fn = sizeof(f<T>(0)) == 1; \
};
```

**7.50.2.9. END**

```
#define END gtest_lite::test.end(); } while (false);
```

Tesztet vége.

**7.50.2.10. ENDM**

```
#define ENDM gtest_lite::test.end(true); } while (false);
```

Tesztet vége allokált blokkok számának összehasonlításával Ez az ellenőrzés nem bomba biztos.

**7.50.2.11. ENDMsg**

```
#define ENDMsg(
    t ) gtest_lite::test.end(true) << t << std::endl; } while (false);
```

Tesztet vége allokált blokkok számának összehasonlításával Ez az ellenőrzés nem bomba biztos. Ha hiba van kiírja az üzenetet.

**7.50.2.12. EXPECT\_ANY\_THROW**

```
#define EXPECT_ANY_THROW(
    statement )
```

**Érték:**

```
try { gtest_lite::test.tmp = false; statement; } \
catch (...) { gtest_lite::test.tmp = true; } \
EXPECT_THROW(statement, "kivetelt dob.", "nem dobott kivetelt.")
```

Kivételt várunk.

**7.50.2.13. EXPECT\_DOUBLE\_EQ**

```
#define EXPECT_DOUBLE_EQ(
    expected,
```

```

        actual ) gtest_lite::EXPECT_(expected, actual, gtest_lite::almostEQ, __FILE__, ↵
__LINE__, "EXPECT_DOUBLE_EQ(" #expected ", " #actual ")") )

```

Valós számok azonosságát elváró makró

#### 7.50.2.14. EXPECT\_ENVCASEEQ

```

#define EXPECT_ENVCASEEQ(
    expected,
    actual ) gtest_lite::EXPECTSTR(std::getenv(expected), actual, gtest_lite::eqstrcase,
__FILE__, __LINE__, "EXPECT_ENVCASEEQ(" #expected ", " #actual ")") )

```

Környezeti változóhoz hasonlít – ilyen nincs a gtest-ben (kisbetű/nagybetű azonos)

#### 7.50.2.15. EXPECT\_ENVEQ

```

#define EXPECT_ENVEQ(
    expected,
    actual ) gtest_lite::EXPECTSTR(std::getenv(expected), actual, gtest_lite::eqstr,
__FILE__, __LINE__, "EXPECT_ENVEQ(" #expected ", " #actual ")") )

```

Környezeti változóhoz hasonlít – ilyen nincs a gtest-ben.

#### 7.50.2.16. EXPECT\_EQ

```

#define EXPECT_EQ(
    expected,
    actual ) gtest_lite::EXPECT_(expected, actual, gtest_lite::eq, __FILE__, __LINE↵
__, "EXPECT_EQ(" #expected ", " #actual ")") )

```

Azonosságot elváró makró

#### 7.50.2.17. EXPECT\_FALSE

```

#define EXPECT_FALSE(
    actual ) gtest_lite::EXPECT_(false, actual, gtest_lite::eq, __FILE__, __LINE__,
"EXPECT_FALSE(" #actual ")") )

```

Hamis értéket elváró makró

#### 7.50.2.18. EXPECT\_FLOAT\_EQ

```

#define EXPECT_FLOAT_EQ(
    expected,
    actual ) gtest_lite::EXPECT_(expected, actual, gtest_lite::almostEQ, __FILE__, ↵
__LINE__, "EXPECT_FLOAT_EQ(" #expected ", " #actual ")") )

```

Valós számok azonosságát elváró makró

#### 7.50.2.19. EXPECT\_GE

```

#define EXPECT_GE(
    expected,
    actual ) gtest_lite::EXPECT_(expected, actual, gtest_lite::ge, __FILE__, __LINE↵
__, "EXPECT_GE(" #expected ", " #actual ")", "etalon") )

```

Nagyobb, vagy egyenlő relációt elváró makró

**7.50.2.20. EXPECT\_GT**

```
#define EXPECT_GT(
    expected,
    actual ) gtest_lite::EXPECT_(expected, actual, gtest_lite::gt, __FILE__, __LINE__↵
    __, "EXPECT_GT(" #expected ", " #actual ")", "etalon" )
```

Nagyobb, mint relációt elváró makró

**7.50.2.21. EXPECT\_LE**

```
#define EXPECT_LE(
    expected,
    actual ) gtest_lite::EXPECT_(expected, actual, gtest_lite::le, __FILE__, __LINE__↵
    __, "EXPECT_LE(" #expected ", " #actual ")", "etalon" )
```

Kisebb, vagy egyenlő relációt elváró makró

**7.50.2.22. EXPECT\_LT**

```
#define EXPECT_LT(
    expected,
    actual ) gtest_lite::EXPECT_(expected, actual, gtest_lite::lt, __FILE__, __LINE__↵
    __, "EXPECT_LT(" #expected ", " #actual ")", "etalon" )
```

Kisebb, mint relációt elváró makró

**7.50.2.23. EXPECT\_NE**

```
#define EXPECT_NE(
    expected,
    actual ) gtest_lite::EXPECT_(expected, actual, gtest_lite::ne, __FILE__, __LINE__↵
    __, "EXPECT_NE(" #expected ", " #actual ")", "etalon" )
```

Eltérést elváró makró

**7.50.2.24. EXPECT\_NO\_THROW**

```
#define EXPECT_NO_THROW(
    statement )
```

Érték:

```
try { gtest_lite::test.tmp = true; statement; } \
catch (...) { gtest_lite::test.tmp = false; }\
EXPECTTHROW(statement, "nem dob kivételt.", "kivételt dobott.")
```

Nem várunk kivételt.

**7.50.2.25. EXPECT\_STRCASEEQ**

```
#define EXPECT_STRCASEEQ(
    expected,
    actual ) gtest_lite::EXPECTSTR(expected, actual, gtest_lite::eqstrcase, __FILE__↵
    __, __LINE__, "EXPECT_STRCASEEQ(" #expected ", " #actual ")" )
```

C stringek (const char \*) azonosságát tesztelő makró (kisbetű/nagybetű azonos)

**7.50.2.26. EXPECT\_STRCASENE**

```
#define EXPECT_STRCASENE(
    expected,
```

```

        actual ) gtest_lite::EXPECTSTR(expected, actual, gtest_lite::nestr, __FILE__, ↵
_, __LINE__, "EXPECT_STRCASENE(" #expected ", " #actual ")", "etalon" )

```

C stringek (const char \*) eltéréset tesztelő makró (kisbetű/nagybetű azonos)

#### 7.50.2.27. EXPECT\_STREQ

```

#define EXPECT_STREQ(
    expected,
    actual ) gtest_lite::EXPECTSTR(expected, actual, gtest_lite::eqstr, __FILE__, ↵
__LINE__, "EXPECT_STREQ(" #expected ", " #actual " )" )

```

C stringek (const char \*) azonosságát tesztelő makró

#### 7.50.2.28. EXPECT\_STRNE

```

#define EXPECT_STRNE(
    expected,
    actual ) gtest_lite::EXPECTSTR(expected, actual, gtest_lite::nestr, __FILE__, ↵
__LINE__, "EXPECT_STRNE(" #expected ", " #actual ")", "etalon" )

```

C stringek (const char \*) eltéréset tesztelő makró

#### 7.50.2.29. EXPECT\_THROW

```

#define EXPECT_THROW(
    statement,
    exception_type )

```

Érték:

```

try { gtest_lite::test.tmp = false; statement; } \
catch (exception_type &e) { gtest_lite::test.tmp = true; } \
catch (...) { } \
EXPECT_THROW(statement, "kivetelt dob.", "nem dobott '" #exception_type "' kivetelt.")

```

Kivetelt várunk.

#### 7.50.2.30. EXPECT\_THROW\_THROW

```

#define EXPECT_THROW_THROW(
    statement,
    exception_type )

```

Érték:

```

try { gtest_lite::test.tmp = false; statement; } \
catch (exception_type &e) { gtest_lite::test.tmp = true; throw; } \
EXPECT_THROW(statement, "kivetelt dob.", "nem dobott '" #exception_type "' kivetelt.")

```

Kivetelt várunk és továbbdobjuk – ilyen nincs a gtest-ben.

#### 7.50.2.31. EXPECT\_TRUE

```

#define EXPECT_TRUE(
    actual ) gtest_lite::EXPECT_(true, actual, gtest_lite::eq, __FILE__, __LINE__ ↵
, "EXPECT_TRUE(" #actual " )" )

```

Igaz értéket elváró makró

#### 7.50.2.32. EXPECTTHROW

```

#define EXPECTTHROW(
    statement,
    exp,
    act )

```

**Érték:**

```
gtest_lite::test.expect(gtest_lite::test.tmp, __FILE__, __LINE__, #statement) \
« "*** Az utasítás " « (act) \
« "\n*** Azt vártuk, hogy " « (exp) « std::endl
```

**EXPECTTHROW; kivételkezelés.**


---

Belső megvalósításhoz tartozó makrók, és osztályok.

**7.50.2.33. Nem célszerű közvetlenül használni, vagy módosítani****7.50.2.34. FAIL**

```
#define FAIL( ) gtest_lite::test.expect(false, __FILE__, __LINE__, "FAIL()", true)
```

Sikertelen teszt fatális hiba makrója.

**7.50.2.35. GTEND**

```
#define GTEND(
    os )
```

**7.50.2.36. GTINIT**

```
#define GTINIT(
    IS )
```

**7.50.2.37. SUCCEED**

```
#define SUCCEED( ) gtest_lite::test.expect(true, __FILE__, __LINE__, "SUCCEED()", true)
```

Sikeres teszt makrója.

**7.50.2.38. TEST**

```
#define TEST(
    C,
    N ) do { gtest_lite::test.begin(#C"."#N);
```

Teszt kezdete. A makró paraméterezése hasonlít a gtest paraméterezéséhez. Így az itt elkészített tesztek könnyen átemelhetők a gtest keretrendszerbe.

**Paraméterek**

<i>C</i>	- teszt eset neve (csak a gtest kompatibilitás miatt van külön neve az eseteknek)
<i>N</i>	- teszt neve

**7.50.3. Függvények dokumentációja****7.50.3.1. hasMember()**

```
void hasMember (
    ... ) [inline]
```

Segédfüggvény egy publikus adattag, vagy tagfüggvény létezésének tesztelésére fordítási időben

## 7.51. src/external/memtrace.cpp fájlreferencia

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <ctype.h>
```

## 7.52. src/external/memtrace.d fájlreferencia

## 7.53. src/external/memtrace.h fájlreferencia

## 7.54. src/fake\_sfml/fake\_sfml.cpp fájlreferencia

```
#include "fake_sfml.hpp"
```

### Névterek

- [sf](#)

### Függvények

- bool [sf::file\\_exists\\_at\\_path](#) (const std::string &name)
- std::ostream & [sf::operator<<](#) (std::ostream &os, const Color &color)

### Változók

- bool [sf::hungarian](#) = false

## 7.55. src/fake\_sfml/fake\_sfml.d fájlreferencia

## 7.56. src/fake\_sfml/fake\_sfml.hpp fájlreferencia

```
#include <iostream>
#include <string>
#include <fstream>
#include <algorithm>
```

### Osztályok

- class [sf::Vector2f](#)
- class [sf::Transform](#)
- class [sf::FloatRect](#)
- class [sf::Vector2i](#)
- class [sf::Texture](#)
- class [sf::Bound](#)
- class [sf::Color](#)
- class [sf::IntRect](#)
- class [sf::Sprite](#)
- class [sf::Event](#)
- class [sf::ClockTime](#)
- class [sf::Clock](#)



- class [sf::SoundBuffer](#)
- class [sf::Sound](#)
- class [sf::SoundSource](#)
- class [sf::Music](#)
- class [sf::RectangleShape](#)
- class [sf::Keyboard](#)
- class [sf::RenderStates](#)
- class [sf::VideoMode](#)
- class [sf::RenderWindow](#)
- class [sf::Mouse](#)

## Névterek

- [sf](#)

## Enumerációk

- enum class [sf::BlendMode](#) {  
    [sf::None](#) , [sf::Alpha](#) , [sf::Additive](#) , [sf::Multiply](#) ,  
    [sf::BlendAdd](#) }

## Függvények

- bool [sf::file\\_exists\\_at\\_path](#) (const std::string &name)

## Változók

- constexpr BlendMode [sf::BlendAdd](#) = BlendMode::BlendAdd

## 7.57. src/GameConfig.cpp fájlreferencia

```
#include "GameConfig.hpp"
```

## 7.58. src/GameConfig.d fájlreferencia

## 7.59. src/GameConfig.hpp fájlreferencia

A Szimuláció konfigurációja itt érhető el.

```
#include <mutex>
#include <iostream>
```

## Osztályok

- class [GameConfig](#)  
    *A világ szimulációjának leírása.*

### 7.59.1. Részletes leírás

A Szimuláció konfigurációja itt érhető el.

#### Szerző

Funk Gábor

Dátum

2025-04-21

## 7.60. src/GameManager.cpp fájlreferencia

```
#include "GameManager.hpp"
```

## 7.61. src/GameManager.d fájlreferencia

## 7.62. src/GameManager.hpp fájlreferencia

A játékmenedzser osztály itt van deklarálva.

```
#include "Utils.hpp"
#include "../fake_sfml/fake_sfml.hpp"
#include <iostream>
#include <vector>
#include "GameConfig.hpp"
#include "ui/button.hpp"
#include "World.hpp"
#include "PostProcessor.hpp"
#include "MusicPlayer.hpp"
#include "SaveManager.hpp"
#include "SoundPlayer.hpp"
```

### Osztályok

- class [GameManager](#)  
*A világ szimulálásáért és a kirazolás irányításáért felelős osztály.*

### 7.62.1. Részletes leírás

A játékmenedzser osztály itt van deklarálva.

Szerző

Funk Gábor

Dátum

2025-04-21

## 7.63. src/main.cpp fájlreferencia

```
#include "GameManager.hpp"
#include "GameConfig.hpp"
#include "World.hpp"
#include "TextureManager.hpp"
#include "Random_Gen.hpp"
#include "external/gtest_lite.h"
```

### Függvények

- int [main](#) ()

### 7.63.1. Függvények dokumentációja

#### 7.63.1.1. main()

```
int main ( )
```

### 7.64. src/main.d fájlreferencia

### 7.65. src/MusicPlayer.cpp fájlreferencia

```
#include "MusicPlayer.hpp"
```

### 7.66. src/MusicPlayer.d fájlreferencia

### 7.67. src/MusicPlayer.hpp fájlreferencia

A zene lejátszó osztály itt van deklarálva.

```
#include "Utils.hpp"
#include "../fake_sfml/fake_sfml.hpp"
#include <unordered_map>
#include <string>
#include <memory>
#include "../exceptions/MusicLoadException.hpp"
```

#### Osztályok

- class [MusicPlayer](#)

*A zene játsszó osztály leírása.*

#### 7.67.1. Részletes leírás

A zene lejátszó osztály itt van deklarálva.

##### Szerző

Funk Gábor

##### Dátum

2025-04-21

### 7.68. src/PostProcessor.cpp fájlreferencia

```
#include "PostProcessor.hpp"
```

### 7.69. src/PostProcessor.d fájlreferencia

### 7.70. src/PostProcessor.hpp fájlreferencia

A grafikus szépítő osztály deklarációját tartalmazza.

```
#include "Utils.hpp"
#include "../fake_sfml/fake_sfml.hpp"
#include <string>
#include "TextureManager.hpp"
```

## Osztályok

- class [PostProcessor](#)

*A grafikus szépítő osztály leírása.*

### 7.70.1. Részletes leírás

A grafikus szépítő osztály deklarációját tartalmazza.

#### Szerző

Funk Gábor

#### Dátum

2025-04-21

## 7.71. src/Profession.cpp fájlreferencia

```
#include "Profession.hpp"
```

## 7.72. src/Profession.d fájlreferencia

## 7.73. src/Profession.hpp fájlreferencia

Ebben a fájlban van deklarálva a szakma indikátor osztály.

```
#include "Textureable.hpp"
#include "TextureManager.hpp"
```

## Osztályok

- class [Profession](#)

*A szakma osztály leírása.*

### 7.73.1. Részletes leírás

Ebben a fájlban van deklarálva a szakma indikátor osztály.

#### Szerző

Funk Gábor

#### Dátum

2025-04-21

## 7.74. src/Random\_Gen.cpp fájlreferencia

```
#include "Random_Gen.hpp"
```

## 7.75. src/Random\_Gen.d fájlreferencia

## 7.76. src/Random\_Gen.hpp fájlreferencia

A véletlen generátor osztályt tároló fájl.

```
#include <iostream>
#include <chrono>
#include <random>
#include <mutex>
```

### Osztályok

- class [RandomGenerator](#)

*Egy korszerűbb és konfigurálhatóbb véletlen szám generátor osztály.*

#### 7.76.1. Részletes leírás

A véletlen generátor osztályt tároló fájl.

#### Szerző

Funk Gábor

#### Dátum

2025-04-21

## 7.77. src/SaveHelpers.cpp fájlreferencia

```
#include "SaveHelpers.hpp"
```

### Névterek

- [minerals](#)

*Az összes struktúra ebben a névtérben van.*

### Függvények

- std::string [minerals::mineral\\_to\\_string](#) (MINERAL\_TYPE type)

*Mentést elősegítő függvények.*

### Változók

- std::unordered\_map< std::string, [CreatureFactory](#) > [creature\\_factory](#)  
*Globális változók deklarálása.*
- std::unordered\_map< std::string, [HumanFactory](#) > [human\\_factory](#)
- std::unordered\_map< std::string, [ResourceFactory](#) > [resource\\_factory](#)

#### 7.77.1. Változók dokumentációja

### 7.77.1.1. creature\_factory

```
std::unordered_map<std::string, CreatureFactory> creature_factory
```

**Kezdő érték:**

```
= {
    {"Crocodile", [](int x, int y) {
        auto* temp = new creature::Crocodile(x, y);
        temp->save_name = "Crocodile";
        return temp;
    }},
    {"Robot", [](int x, int y) {
        auto* temp = new creature::KillerRobot(x, y);
        temp->save_name = "Robot";
        return temp;
    }},
    {"Goat", [](int x, int y) {
        auto* temp = new creature::Goat(x, y);
        temp->save_name = "Goat";
        return temp;
    }},
    {"Bear", [](int x, int y) {
        auto* temp = new creature::Bear(x, y);
        temp->save_name = "Bear";
        return temp;
    }}
}
```

Globális változók deklarálása.

### 7.77.1.2. human\_factory

```
std::unordered_map<std::string, HumanFactory> human_factory
```

**Kezdő érték:**

```
= {
    {"empty",          [](int x, int y, creature::ENTITY_GENDER g) { return new creature::Human(x, y, g); }},
    {"?",              [](int x, int y, creature::ENTITY_GENDER g) { return new creature::Human(x, y, g); }},
    {"axe",             [](int x, int y, creature::ENTITY_GENDER g) { return new creature::Woodcutter(x, y, g); }},
    {"bricks",          [](int x, int y, creature::ENTITY_GENDER g) { return new creature::Builder(x, y, g); }},
    {"crown",           [](int x, int y, creature::ENTITY_GENDER g) { return new creature::King(x, y, g); }},
    {"fishing_rod",     [](int x, int y, creature::ENTITY_GENDER g) { return new creature::Fisherman(x, y, g); }},
    {"pickaxe",         [](int x, int y, creature::ENTITY_GENDER g) { return new creature::Stonemason(x, y, g); }},
    {"saw",             [](int x, int y, creature::ENTITY_GENDER g) { return new creature::AnglerMiner(x, y, g); }},
    {"scythe",          [](int x, int y, creature::ENTITY_GENDER g) { return new creature::Farmer(x, y, g); }},
    {"sword",           [](int x, int y, creature::ENTITY_GENDER g) { return new creature::Soldier(x, y, g); }}
}
```

### 7.77.1.3. resource\_factory

```
std::unordered_map<std::string, ResourceFactory> resource_factory
```

**Kezdő érték:**

```
= {
    {"FOOD",           [](int x, int y) { return new minerals::BerryBush(x, y); }},
    {"IRON",           [](int x, int y) { return new minerals::Iron(x, y); }},
    {"WOOD",           [](int x, int y) { return new minerals::Tree(x, y); }},
    {"STONE",          [](int x, int y) { return new minerals::Stone(x, y); }}
}
```

## 7.78. src/SaveHelpers.d fájlreferencia

## 7.79. src/SaveHelpers.hpp fájlreferencia

A mentést segítő factory-k és segédfüggvények.

```
#include <string>
#include <unordered_map>
#include <functional>
#include "creatures/Living.hpp"
#include "creatures/humans/Human.hpp"
```

```
#include "creatures/humans/Woodcutter.hpp"
#include "creatures/humans/Farmer.hpp"
#include "creatures/humans/Stonemason.hpp"
#include "creatures/humans/Fisherman.hpp"
#include "creatures/humans/Builder.hpp"
#include "creatures/humans/King.hpp"
#include "creatures/humans/AnglerMiner.hpp"
#include "creatures/humans/Soldier.hpp"
#include "creatures/Goat.hpp"
#include "creatures/hostiles/Crocodile.hpp"
#include "creatures/hostiles/Bear.hpp"
#include "creatures/hostiles/KillerRobot.hpp"
#include "world_object/Structure.hpp"
#include "world_object/BerryBush.hpp"
#include "world_object/Stone.hpp"
#include "world_object/Tree.hpp"
#include "world_object/Iron.hpp"
#include "world_object/CityCenter.hpp"
#include "world_object/House.hpp"
```

## Névterek

- [minerals](#)

*Az összes struktúra ebben a névtérben van.*

## Típusdefiníciók

- using [CreatureFactory](#) = std::function< [creature::Living](#) \*(int, int)>  
*Factory-k.*
- using [HumanFactory](#) = std::function< [creature::Human](#) \*(int, int, [creature::ENTITY\\_GENDER](#))>
- using [ResourceFactory](#) = std::function< [minerals::Structure](#) \*(int, int)>

## Függvények

- std::string [minerals::mineral\\_to\\_string](#) (MINERAL\_TYPE type)

*Mentést elősegítő függvények.*

## Változók

- std::unordered\_map< std::string, [CreatureFactory](#) > [creature\\_factory](#)  
*Globális változók deklarálása.*
- std::unordered\_map< std::string, [HumanFactory](#) > [human\\_factory](#)
- std::unordered\_map< std::string, [ResourceFactory](#) > [resource\\_factory](#)

### 7.79.1. Részletes leírás

A mentést segítő factory-k és segédfüggvények.

#### Szerző

Funk Gábor

#### Dátum

2025-04-21

## 7.79.2. Típusdefiníciók dokumentációja

### 7.79.2.1. CreatureFactory

using CreatureFactory = std::function<creature::Living\*(int, int)>  
Factory-k.

### 7.79.2.2. HumanFactory

using HumanFactory = std::function<creature::Human\*(int, int, creature::ENTITY\_GENDER)>

### 7.79.2.3. ResourceFactory

using ResourceFactory = std::function<minerals::Structure\*(int, int)>

## 7.79.3. Változók dokumentációja

### 7.79.3.1. creature\_factory

std::unordered\_map<std::string, CreatureFactory> creature\_factory [extern]  
Globális változók deklarálása.

### 7.79.3.2. human\_factory

std::unordered\_map<std::string, HumanFactory> human\_factory [extern]

### 7.79.3.3. resource\_factory

std::unordered\_map<std::string, ResourceFactory> resource\_factory [extern]

## 7.80. src/SaveManager.cpp fájlreferencia

```
#include "SaveManager.hpp"
```

## 7.81. src/SaveManager.d fájlreferencia

## 7.82. src/SaveManager.hpp fájlreferencia

A fájl menedzseléshez szolgáló osztály itt van deklarálva.

```
#include <iostream>
#include <string>
#include <fstream>
#include <cstdio>
#include "World.hpp"
#include "SaveHelpers.hpp"
#include "GameConfig.hpp"
#include "../exceptions/FileExceptions.hpp"
```



## Osztályok

- class [SaveManager](#)

*A fájl menedzseléshez szolgáló osztály leírása.*

### 7.82.1. Részletes leírás

A fájl menedzseléshez szolgáló osztály itt van deklarálva.

#### Szerző

Funk Gábor

#### Dátum

2025-04-21

## 7.83. src/Shadowable.cpp fájlreferencia

```
#include "Shadowable.hpp"
```

## 7.84. src/Shadowable.d fájlreferencia

## 7.85. src/Shadowable.hpp fájlreferencia

Az árnyék szimulálásához való osztályt tartalmazza.

```
#include "TextureManager.hpp"  
#include "GameConfig.hpp"  
#include "Utils.hpp"  
#include "../fake_sfml/fake_sfml.hpp"  
#include <string>  
#include <cmath>
```

## Osztályok

- class [Shadowable](#)

*Az árnyékoláshoz szükséges interface.*

### 7.85.1. Részletes leírás

Az árnyék szimulálásához való osztályt tartalmazza.

#### Szerző

Funk Gábor

#### Dátum

2025-04-21

## 7.86. src/SoundPlayer.cpp fájlreferencia

```
#include "SoundPlayer.hpp"
```

## 7.87. src/SoundPlayer.d fájlreferencia

## 7.88. src/SoundPlayer.hpp fájlreferencia

Ebben a fájlba vannak a hanglejátszó osztályhoz szükséges deklarációk.

```
#include "Utils.hpp"
#include "../fake_sfml/fake_sfml.hpp"
#include <unordered_map>
#include <string>
#include <memory>
#include <iostream>
```

### Osztályok

- class [SoundPlayer](#)  
*A hanglejátszó osztály leírása.*

#### 7.88.1. Részletes leírás

Ebben a fájlba vannak a hanglejátszó osztályhoz szükséges deklarációk.

#### Szerző

Funk Gábor

#### Dátum

2025-04-21

## 7.89. src/terrain\_tiles/Tile.cpp fájlreferencia

```
#include "Tile.hpp"
```

### Névterek

- [tiles](#)  
*Az összes terepkocka elem ebben a névtérben van.*

## 7.90. src/terrain\_tiles/Tile.d fájlreferencia

## 7.91. src/terrain\_tiles/Tile.hpp fájlreferencia

A Terepkocka osztály itt van deklarálv.

```
#include "../Textureable.hpp"
#include "../TextureManager.hpp"
```

### Osztályok

- class [tiles::Tile](#)  
*A terepkocka osztály leírása.*

### Névterek

- [tiles](#)  
*Az összes terepkocka elem ebben a névtérben van.*

## Enumerációk

- enum class `tiles::TILETYPE` : char { `tiles::GRASS` , `tiles::WATER` , `tiles::MOUNTAIN` }

### 7.91.1. Részletes leírás

A Terepkocka osztály itt van deklarálva.

#### Szerző

Funk Gábor

#### Dátum

2025-04-21

## 7.92. src/TerrainContainer.hpp fájlreferencia

A Világ terepének a deklarálása ebben a fájlba van.

```
#include "Utils.hpp"
#include "../fake_sfml/fake_sfml.hpp"
#include "GameManager.hpp"
#include "GameConfig.hpp"
#include "Random_Gen.hpp"
#include "terrain_tiles/Tile.hpp"
#include "TerrainContainer.inl"
```

## Osztályok

- class `TerrainContainer< T >`

*A világ terepét tároló osztály.*

### 7.92.1. Részletes leírás

A Világ terepének a deklarálása ebben a fájlba van.

#### Szerző

Funk Gábor

#### Dátum

2025-04-21

## 7.93. src/TerrainContainer.inl fájlreferencia

```
#include "TerrainContainer.hpp"
```

## Makródefiníciók

- #define `DYNAMIC2D_INL`

### 7.93.1. Makródefiníciók dokumentációja

#### 7.93.1.1. DYNAMIC2D\_INL

```
#define DYNAMIC2D_INL
```

### 7.94. src/Textureable.hpp fájlreferencia

Ebbe a fájlba van a textúrázáshoz szükséges osztály.

```
#include "Utils.hpp"
#include "../fake_sfml/fake_sfml.hpp"
#include <string>
```

#### Osztályok

- class [Textureable](#)

*Egy interface, ami a textúrázáshoz kell.*

#### 7.94.1. Részletes leírás

Ebbe a fájlba van a textúrázáshoz szükséges osztály.

##### Szerző

Funk Gábor

##### Dátum

2025-04-21

### 7.95. src/TextureManager.cpp fájlreferencia

```
#include "TextureManager.hpp"
```

### 7.96. src/TextureManager.d fájlreferencia

### 7.97. src/TextureManager.hpp fájlreferencia

Ebbe a fájlba van az az osztály, ami a textúrák betöltéséért, kiosztásáért és tárolásáért felelős.

```
#include "Utils.hpp"
#include "../fake_sfml/fake_sfml.hpp"
#include <unordered_map>
#include <string>
#include <memory>
#include <iostream>
```

#### Osztályok

- class [TextureManager](#)

*A Textúra kezelő osztály.*

#### 7.97.1. Részletes leírás

Ebbe a fájlba van az az osztály, ami a textúrák betöltéséért, kiosztásáért és tárolásáért felelős.

A [TextureManager](#) osztály singleton pattern-t használ.

**Szerző**

Funk Gábor

**Dátum**

2025-04-21

**7.98. src/ui/button.cpp fájlreferencia**

```
#include "button.hpp"
```

**Névterek**

- [ui](#)

*Az összes UI elem ebben a névtérben van.*

**7.99. src/ui/button.d fájlreferencia****7.100. src/ui/button.hpp fájlreferencia**

A gomb osztály itt van deklarálva.

```
#include "../Textureable.hpp"  
#include "../TextureManager.hpp"  
#include <iostream>  
#include <functional>  
#include <string>
```

**Osztályok**

- class [ui::Button](#)

*A gomb osztály leírása. Tárolja a gomb méretét és azt, hogy mit csinál, ha rákattintanak.*

**Névterek**

- [ui](#)

*Az összes UI elem ebben a névtérben van.*

**7.100.1. Részletes leírás**

A gomb osztály itt van deklarálva.

**Szerző**

Funk Gábor

**Dátum**

2025-04-21

**7.101. src/Uutils.cpp fájlreferencia**

```
#include "Uutils.hpp"
```

## Függvények

- double [distance\\_to](#) (double x1, double y1, double x2, double y2)

Visszaadja a távolságot az (x1,y1) (x2,y2) pont között.

### 7.101.1. Függvények dokumentációja

#### 7.101.1.1. distance\_to()

```
double distance_to (  
    double x1,  
    double y1,  
    double x2,  
    double y2 )
```

Visszaadja a távolságot az (x1,y1) (x2,y2) pont között.

#### Paraméterek

x1	Az 1. pont x koordinátája.
y1	Az 1. pont y koordinátája.
x2	Az 2. pont x koordinátája.
y2	Az 2. pont y koordinátája.

#### Visszatérési érték

A távolság a 2 pont között.

#### Figyelmeztetés

Ez a függvény jelenleg nincs használatban.

## 7.102. src/Utils.d fájlreferencia

## 7.103. src/Utils.hpp fájlreferencia

Ebben a fájlba vannak a segéd függvények.

```
#include <cmath>
```

## Függvények

- double [distance\\_to](#) (double x1, double y1, double x2, double y2)

Visszaadja a távolságot az (x1,y1) (x2,y2) pont között.

### 7.103.1. Részletes leírás

Ebben a fájlba vannak a segéd függvények.

#### Szerző

Funk Gábor

#### Dátum

2025-04-21

## 7.103.2. Függvények dokumentációja

### 7.103.2.1. distance\_to()

```
double distance_to (
    double x1,
    double y1,
    double x2,
    double y2 )
```

Visszaadja a távolságot az (x1,y1) (x2,y2) pont között.

#### Paraméterek

<i>x1</i>	Az 1. pont x koordinátája.
<i>y1</i>	Az 1. pont y koordinátája.
<i>x2</i>	Az 2. pont x koordinátája.
<i>y2</i>	Az 2. pont y koordinátája.

#### Visszatérési érték

A távolság a 2 pont között.

#### Figyelmeztetés

Ez a függvény jelenleg nincs használatban.

## 7.104. src/World.cpp fájlreferencia

```
#include "World.hpp"
```

### Függvények

- std::ostream & [operator<<](#) (std::ostream &os, const [World](#) &w)
- std::ifstream & [operator>>](#) (std::ifstream &in, [World](#) &w)

### 7.104.1. Függvények dokumentációja

#### 7.104.1.1. operator<<()

```
std::ostream& operator<< (
    std::ostream & os,
    const World & w )
```

#### 7.104.1.2. operator>>()

```
std::ifstream& operator>> (
    std::ifstream & in,
    World & w )
```

## 7.105. src/World.d fájlreferencia

## 7.106. src/World.hpp fájlreferencia

A Világ osztály, ami a fő szimulációs elemek tárolásáért felelős.

```
#include "terrain_tiles/Tile.hpp"
#include <fstream>
#include <string>
#include <sstream>
#include "TerrainContainer.hpp"
#include "creatures/Living.hpp"
#include "creatures/humans/Human.hpp"
#include "creatures/humans/Woodcutter.hpp"
#include "creatures/humans/Farmer.hpp"
#include "creatures/humans/Stonemason.hpp"
#include "creatures/humans/Fisherman.hpp"
#include "creatures/humans/Builder.hpp"
#include "creatures/humans/King.hpp"
#include "creatures/humans/AnglerMiner.hpp"
#include "creatures/humans/Soldier.hpp"
#include "creatures/Goat.hpp"
#include "creatures/hostiles/Crocodile.hpp"
#include "creatures/hostiles/Bear.hpp"
#include "creatures/hostiles/KillerRobot.hpp"
#include <unordered_map>
#include <queue>
#include "world_object/Structure.hpp"
#include "world_object/BerryBush.hpp"
#include "world_object/Stone.hpp"
#include "world_object/Tree.hpp"
#include "world_object/Iron.hpp"
#include "world_object/CityCenter.hpp"
#include "world_object/House.hpp"
#include <vector>
#include "SoundPlayer.hpp"
#include "SaveHelpers.hpp"
#include "../exceptions/FileExceptions.hpp"
#include "../exceptions/WorldExceptions.hpp"
```

### Osztályok

- class [World](#)

*A világ osztály leírása.*

#### 7.106.1. Részletes leírás

A Világ osztály, ami a fő szimulációs elemek tárolásáért felelős.

Ez az osztály felelős a szimulációs elemekért, felszabadítja őket, ha kell.

#### Szerző

Funk Gábor

#### Dátum

2025-04-21



## 7.107. src/world\_object/BerryBush.cpp fájlreferencia

```
#include "BerryBush.hpp"
```

### Névterek

- [minerals](#)

*Az összes struktúra ebben a névtérben van.*

## 7.108. src/world\_object/BerryBush.d fájlreferencia

## 7.109. src/world\_object/BerryBush.hpp fájlreferencia

A bokor osztály itt van deklarálva.

```
#include "ResourceStructure.hpp"
```

### Osztályok

- class [minerals::BerryBush](#)

*A bokor osztály leírása. Ételt ad, ha kitermelik.*

### Névterek

- [minerals](#)

*Az összes struktúra ebben a névtérben van.*

### 7.109.1. Részletes leírás

A bokor osztály itt van deklarálva.

#### Szerző

Funk Gábor

#### Dátum

2025-04-21

## 7.110. src/world\_object/CityCenter.cpp fájlreferencia

```
#include "CityCenter.hpp"
```

### Névterek

- [minerals](#)

*Az összes struktúra ebben a névtérben van.*

## 7.111. src/world\_object/CityCenter.d fájlreferencia

## 7.112. src/world\_object/CityCenter.hpp fájlreferencia

A városközpont osztály itt van deklarálva.

```
#include "Structure.hpp"  
#include <string>
```

## Osztályok

- class [minerals::CityCenter](#)

*A városközpont osztály leírása. E köré épülnek a házak.*

## Névterek

- [minerals](#)

*Az összes struktúra ebben a névtérben van.*

### 7.112.1. Részletes leírás

A városközpont osztály itt van deklarálnva.

#### Szerző

Funk Gábor

#### Dátum

2025-04-21

## 7.113. src/world\_object/House.cpp fájlreferencia

```
#include "House.hpp"
```

## Névterek

- [minerals](#)

*Az összes struktúra ebben a névtérben van.*

## 7.114. src/world\_object/House.d fájlreferencia

## 7.115. src/world\_object/House.hpp fájlreferencia

A Ház osztály itt van deklarálnva.

```
#include "Structure.hpp"  
#include <string>  
#include "../Random_Gen.hpp"
```

## Osztályok

- class [minerals::House](#)

*A ház osztály leírása. Szinttől függően idéz embereket.*

## Névterek

- [minerals](#)

*Az összes struktúra ebben a névtérben van.*

### 7.115.1. Részletes leírás

A Ház osztály itt van deklarálva.

Szerző

Funk Gábor

Dátum

2025-04-21

## 7.116. src/world\_object/Iron.cpp fájlreferencia

```
#include "Iron.hpp"
```

### Névterek

- [minerals](#)

*Az összes struktúra ebben a névtérben van.*

## 7.117. src/world\_object/Iron.d fájlreferencia

## 7.118. src/world\_object/Iron.hpp fájlreferencia

A Vasérc osztály itt van deklarálva.

```
#include "ResourceStructure.hpp"
```

```
#include "../Random_Gen.hpp"
```

### Osztályok

- class [minerals::Iron](#)

*A vasérc osztály leírása. Vasat ad, amikor kitermelik.*

### Névterek

- [minerals](#)

*Az összes struktúra ebben a névtérben van.*

### 7.118.1. Részletes leírás

A Vasérc osztály itt van deklarálva.

Szerző

Funk Gábor

Dátum

2025-04-21

## 7.119. src/world\_object/ResourceStructure.cpp fájlreferencia

```
#include "ResourceStructure.hpp"
```

## Névterek

- [minerals](#)

*Az összes struktúra ebben a névtérben van.*

## 7.120. src/world\_object/ResourceStructure.d fájlreferencia

## 7.121. src/world\_object/ResourceStructure.hpp fájlreferencia

A kibányászható osztály itt van deklarálva.

```
#include "Structure.hpp"
#include <string>
```

## Osztályok

- class [minerals::ResourceStructure](#)

*Az erőforrás struktúra osztály leírása.*

## Névterek

- [minerals](#)

*Az összes struktúra ebben a névtérben van.*

### 7.121.1. Részletes leírás

A kibányászható osztály itt van deklarálva.

#### Szerző

Funk Gábor

#### Dátum

2025-04-21

## 7.122. src/world\_object/Stone.cpp fájlreferencia

```
#include "Stone.hpp"
```

## Névterek

- [minerals](#)

*Az összes struktúra ebben a névtérben van.*

## 7.123. src/world\_object/Stone.d fájlreferencia

## 7.124. src/world\_object/Stone.hpp fájlreferencia

A Kő osztály itt van deklarálva.

```
#include "ResourceStructure.hpp"
#include "../Random_Gen.hpp"
```

## Osztályok

- class [minerals::Stone](#)

*A kő osztály leírása. követ ad, amikor kitermelik.*

## Névterek

- [minerals](#)

*Az összes struktúra ebben a névtérben van.*

### 7.124.1. Részletes leírás

A Kő osztály itt van deklarálva.

#### Szerző

Funk Gábor

#### Dátum

2025-04-21

## 7.125. src/world\_object/Structure.cpp fájlreferencia

```
#include "Structure.hpp"
```

## Névterek

- [minerals](#)

*Az összes struktúra ebben a névtérben van.*

## 7.126. src/world\_object/Structure.d fájlreferencia

## 7.127. src/world\_object/Structure.hpp fájlreferencia

A struktúra osztály itt van deklarálva.

```
#include "../Textureable.hpp"  
#include "../TextureManager.hpp"  
#include "../GameConfig.hpp"  
#include "../Shadowable.hpp"  
#include <string>  
#include <iostream>
```

## Osztályok

- class [minerals::Structure](#)

*A struktúra osztály leírása.*

## Névterek

- [minerals](#)

*Az összes struktúra ebben a névtérben van.*

## Enumerációk

- enum class `minerals::MINERAL_TYPE` : char {  
    `minerals::STONE` , `minerals::WOOD` , `minerals::IRON` , `minerals::FOOD` ,  
    `minerals::HOUSING` , `minerals::CITY_CENTER` }

### 7.127.1. Részletes leírás

A struktúra osztály itt van deklarálva.

#### Szerző

Funk Gábor

#### Dátum

2025-04-21

## 7.128. src/world\_object/Tree.cpp fájlreferencia

```
#include "Tree.hpp"
```

## Névterek

- `minerals`

*Az összes struktúra ebben a névtérben van.*

## 7.129. src/world\_object/Tree.d fájlreferencia

## 7.130. src/world\_object/Tree.hpp fájlreferencia

A fa osztály itt van deklarálva.

```
#include "ResourceStructure.hpp"  
#include "../Random_Gen.hpp"
```

## Osztályok

- class `minerals::Tree`

*A fa osztály leírása. Fát ad, ha kitermelik.*

## Névterek

- `minerals`

*Az összes struktúra ebben a névtérben van.*

### 7.130.1. Részletes leírás

A fa osztály itt van deklarálva.

#### Szerző

Funk Gábor

#### Dátum

2025-04-21

# Tárgymutató

`_Is_Types< F, T >`, [25](#)  
    `convertable`, [26](#)  
    `f`, [25](#), [26](#)  
`~AnglerMiner`  
    `creature::AnglerMiner`, [27](#)  
`~Bear`  
    `creature::Bear`, [29](#)  
`~Builder`  
    `creature::Builder`, [35](#)  
`~Crocodile`  
    `creature::Crocodile`, [47](#)  
`~Farmer`  
    `creature::Farmer`, [51](#)  
`~Fisherman`  
    `creature::Fisherman`, [53](#)  
`~GameManager`  
    `GameManager`, [62](#)  
`~Goat`  
    `creature::Goat`, [65](#)  
`~HostileInterface`  
    `creature::HostileInterface`, [67](#)  
`~Human`  
    `creature::Human`, [75](#)  
`~KillerRobot`  
    `creature::KillerRobot`, [89](#)  
`~King`  
    `creature::King`, [93](#)  
`~Living`  
    `creature::Living`, [96](#)  
`~MusicPlayer`  
    `MusicPlayer`, [111](#)  
`~ResourceStructure`  
    `minerals::ResourceStructure`, [128](#)  
`~Shadowable`  
    `Shadowable`, [132](#)  
`~Soldier`  
    `creature::Soldier`, [137](#)  
`~Sound`  
    `sf::Sound`, [138](#)  
`~SoundSource`  
    `sf::SoundSource`, [142](#)  
`~Sprite`  
    `sf::Sprite`, [143](#)  
`~Stonemason`  
    `creature::Stonemason`, [148](#)  
`~Structure`  
    `minerals::Structure`, [151](#)  
`~TerrainContainer`  
    `TerrainContainer< T >`, [157](#)  
  
`~Test`  
    `gtest_lite::Test`, [163](#)  
`~Textureable`  
    `Textureable`, [167](#)  
`~Woodcutter`  
    `creature::Woodcutter`, [181](#)  
`~World`  
    `World`, [183](#)  
`~ostreamRedir`  
    `gtest_lite::ostreamRedir`, [113](#)  
  
`a`  
    `sf::Color`, [44](#)  
`ablocks`  
    `gtest_lite::Test`, [164](#)  
`ADD_FAILURE`  
    `gtest_lite.h`, [213](#)  
`add_resources`  
    `World`, [183](#)  
`Additive`  
    `sf`, [21](#)  
`almostEQ`  
    `gtest_lite`, [16](#)  
`Alpha`  
    `sf`, [21](#)  
`AnglerMiner`  
    `creature::AnglerMiner`, [27](#)  
`ANIMAL`  
    `creature`, [14](#)  
`apply_age`  
    `creature::Living`, [96](#)  
`asSeconds`  
    `sf::ClockTime`, [43](#)  
`ASSERT_`  
    `gtest_lite.h`, [213](#)  
`ASSERT_EQ`  
    `gtest_lite.h`, [214](#)  
`ASSERT_NO_THROW`  
    `gtest_lite.h`, [214](#)  
`ASSERTTHROW`  
    `gtest_lite.h`, [214](#)  
`astatus`  
    `gtest_lite::Test`, [163](#)  
`attack_speed`  
    `creature::HostileInterface`, [69](#)  
`ATTACKING`  
    `creature`, [15](#)  
  
`b`

- sf::Color, 45
- Bear
  - creature::Bear, 29
- begin
  - gtest\_lite::Test, 163
- berry\_growth\_time
  - GameConfig, 60
- BerryBush
  - minerals::BerryBush, 32
- bitsPerPixel
  - sf::VideoMode, 179
- Black
  - sf::Color, 45
- BlendAdd
  - sf, 21, 22
- BlendMode
  - sf, 21
- blendMode
  - sf::RenderStates, 124
- Blue
  - sf::Color, 45
- build\_city\_center\_at
  - World, 184
- Builder
  - creature::Builder, 34
- Button
  - ui::Button, 36
- check\_aggroed
  - creature::HostileInterface, 68
  - creature::Living, 96
- CITY\_CENTER
  - minerals, 20
- CityCenter
  - minerals::CityCenter, 39
- CityCenterException, 41
  - CityCenterException, 41
- clear
  - sf::RenderWindow, 125
  - TerrainContainer< T >, 157
  - TextureManager, 169
  - World, 184
- clear\_at
  - TerrainContainer< T >, 157
- ClockTime
  - sf::ClockTime, 42, 43
- close
  - sf::RenderWindow, 125
- Closed
  - sf::Event, 50
- Color
  - sf::Color, 44
- combine
  - sf::Transform, 173
- contains
  - sf::FloatRect, 55
- convertable
  - \_Is\_Types< F, T >, 26
- create
  - sf::RenderWindow, 125
- CREATE\_Has\_
  - gtest\_lite.h, 214
- CREATE\_Has\_fn\_
  - gtest\_lite.h, 215
- creature, 13
  - ANIMAL, 14
  - ATTACKING, 15
  - DEATH, 15
  - DOING\_ITS\_WORK, 15
  - ENTITY\_GENDER, 14
  - ENTITY\_TYPE, 14
  - FACING, 14
  - FEMALE, 14
  - HUMAN, 14
  - IDLE, 15
  - LEFT, 15
  - LIVINGSTATE, 15
  - MALE, 14
  - RIGHT, 15
  - ROBOTIC, 14
  - RUN, 15
  - WALK, 15
- creature::AnglerMiner, 26
  - ~AnglerMiner, 27
  - AnglerMiner, 27
  - update\_logic, 27
- creature::Bear, 28
  - ~Bear, 29
  - Bear, 29
  - die, 29
  - draw\_logic, 29
  - get\_type, 30
  - select\_target, 30
  - update\_logic, 30
- creature::Builder, 34
  - ~Builder, 35
  - Builder, 34
  - update\_logic, 35
- creature::Crocodile, 46
  - ~Crocodile, 47
  - Crocodile, 47
  - die, 47
  - draw\_logic, 47
  - get\_type, 48
  - select\_target, 48
  - update\_logic, 48
- creature::Farmer, 50
  - ~Farmer, 51
  - Farmer, 51
  - update\_logic, 51
- creature::Fisherman, 52
  - ~Fisherman, 53
  - Fisherman, 53
  - fishing, 54
  - try\_fishing, 54
  - update\_logic, 54
- creature::Goat, 64



- ~Goat, 65
- die, 65
- draw\_logic, 65
- get\_type, 65
- Goat, 64
- update\_logic, 66
- creature::HostileInterface, 66
  - ~HostileInterface, 67
  - attack\_speed, 69
  - check\_aggroed, 68
  - damage, 70
  - goal, 70
  - hostile\_run, 68
  - hostile\_walk, 68
  - retarget, 68
  - select\_target, 69
  - set\_hostile\_config, 69
  - target, 70
  - try\_attack, 69
- creature::Human, 73
  - ~Human, 75
  - die, 75
  - draw\_logic, 75
  - get\_profession\_string, 75
  - get\_type, 76
  - goal, 78
  - Human, 74
  - initialize, 76
  - needs\_promotion, 78
  - needs\_to\_be\_royal, 78
  - profession, 79
  - select\_texture, 76
  - update\_logic, 78
- creature::KillerRobot, 88
  - ~KillerRobot, 89
  - die, 90
  - draw\_logic, 90
  - get\_type, 90
  - KillerRobot, 89
  - select\_target, 91
  - update\_logic, 91
- creature::King, 91
  - ~King, 93
  - King, 92
  - update\_logic, 93
- creature::Living, 93
  - ~Living, 96
  - apply\_age, 96
  - check\_aggroed, 96
  - damage, 97
  - damaged\_by, 104
  - death\_timer, 104
  - die, 97
  - draw, 97
  - draw\_logic, 98
  - facing, 105
  - gender, 105
  - get\_gender, 98
  - get\_state, 98
  - get\_type, 98
  - get\_width, 99
  - health, 105
  - hit\_timer, 105
  - init\_spritesheet\_data, 99
  - inner\_timer, 105
  - look\_left, 99
  - look\_right, 99
  - max\_age, 105
  - MAX\_CREATURE\_SIZE, 106
  - needs\_drawn, 100
  - posx, 106
  - posy, 106
  - retarget, 100
  - run\_speed\_modifier, 106
  - save\_name, 106
  - set\_attack\_texture, 100
  - set\_death\_texture, 100
  - set\_health, 101
  - set\_idle\_texture, 101
  - set\_run\_texture, 101
  - set\_state, 102
  - set\_walk\_texture, 102
  - setPosition, 102
  - setTexture, 102
  - setTheShadow, 103
  - shadow\_logic, 103
  - speed, 106
  - state, 107
  - update\_logic, 104
  - update\_spritesheet, 104
- creature::Soldier, 136
  - ~Soldier, 137
  - Soldier, 137
  - update\_logic, 137
- creature::Stonemason, 147
  - ~Stonemason, 148
  - mining\_iron, 149
  - Stonemason, 148
  - try\_mine, 148
  - update\_logic, 149
- creature::Woodcutter, 180
  - ~Woodcutter, 181
  - update\_logic, 181
  - Woodcutter, 180
- creature\_factory
  - SaveHelpers.cpp, 225
  - SaveHelpers.hpp, 228
- CreatureFactory
  - SaveHelpers.hpp, 228
- Crocodile
  - creature::Crocodile, 47
- damage
  - creature::HostileInterface, 70
  - creature::Living, 97
- damaged\_by
  - creature::Living, 104

day\_length  
     GameConfig, 60  
 DEATH  
     creature, 15  
 death\_timer  
     creature::Living, 104  
 deleteFile  
     SaveManager, 130  
 die  
     creature::Bear, 29  
     creature::Crocodile, 47  
     creature::Goat, 65  
     creature::Human, 75  
     creature::KillerRobot, 90  
     creature::Living, 97  
 display  
     sf::RenderWindow, 126  
 distance\_to  
     Utils.cpp, 234  
     Utils.hpp, 235  
 DOING\_ITS\_WORK  
     creature, 15  
 Down  
     sf::Keyboard, 87  
 draw  
     creature::Living, 97  
     minerals::Structure, 151  
     PostProcessor, 114  
     Profession, 117  
     sf::RenderWindow, 126  
     sf::Sprite, 143  
     TerrainContainer< T >, 158  
     Textureable, 167  
     tiles::Tile, 170  
     ui::Button, 37  
     World, 184  
 draw\_buttons  
     GameManager, 62  
 draw\_logic  
     creature::Bear, 29  
     creature::Crocodile, 47  
     creature::Goat, 65  
     creature::Human, 75  
     creature::KillerRobot, 90  
     creature::Living, 98  
     minerals::Structure, 152  
 drawShadow  
     Shadowable, 132  
 DYNAMIC2D\_INL  
     TerrainContainer.inl, 231  
 END  
     gtest\_lite.h, 215  
 end  
     gtest\_lite::Test, 163  
 ENDM  
     gtest\_lite.h, 215  
 ENDMsg  
     gtest\_lite.h, 215

ENTITY\_GENDER  
     creature, 14  
 ENTITY\_TYPE  
     creature, 14  
 eq  
     gtest\_lite, 16  
 eqstr  
     gtest\_lite, 16  
 eqstrcase  
     gtest\_lite, 16  
 EType  
     sf::Event, 49  
 Event  
     sf::Event, 50  
 expect  
     gtest\_lite::Test, 163  
 EXPECT\_  
     gtest\_lite, 17  
 EXPECT\_ANY\_THROW  
     gtest\_lite.h, 215  
 EXPECT\_DOUBLE\_EQ  
     gtest\_lite.h, 215  
 EXPECT\_ENVCASEEQ  
     gtest\_lite.h, 216  
 EXPECT\_ENVEQ  
     gtest\_lite.h, 216  
 EXPECT\_EQ  
     gtest\_lite.h, 216  
 EXPECT\_FALSE  
     gtest\_lite.h, 216  
 EXPECT\_FLOAT\_EQ  
     gtest\_lite.h, 216  
 EXPECT\_GE  
     gtest\_lite.h, 216  
 EXPECT\_GT  
     gtest\_lite.h, 216  
 EXPECT\_LE  
     gtest\_lite.h, 217  
 EXPECT\_LT  
     gtest\_lite.h, 217  
 EXPECT\_NE  
     gtest\_lite.h, 217  
 EXPECT\_NO\_THROW  
     gtest\_lite.h, 217  
 EXPECT\_STRCASEEQ  
     gtest\_lite.h, 217  
 EXPECT\_STRCASENE  
     gtest\_lite.h, 217  
 EXPECT\_STREQ  
     gtest\_lite.h, 218  
 EXPECT\_STRNE  
     gtest\_lite.h, 218  
 EXPECT\_THROW  
     gtest\_lite.h, 218  
 EXPECT\_THROW\_THROW  
     gtest\_lite.h, 218  
 EXPECT\_TRUE  
     gtest\_lite.h, 218

EXPECTSTR  
     gtest\_lite, 17  
 EXPECTTHROW  
     gtest\_lite.h, 218  
  
 f  
     \_Is\_Types< F, T >, 25, 26  
 FACING  
     creature, 14  
 facing  
     creature::Living, 105  
 FAIL  
     gtest\_lite.h, 219  
 fail  
     gtest\_lite::Test, 164  
 failed  
     gtest\_lite::Test, 164  
 Farmer  
     creature::Farmer, 51  
 FEMALE  
     creature, 14  
 file\_exists\_at\_path  
     sf, 21  
 Fisherman  
     creature::Fisherman, 53  
 fishing  
     creature::Fisherman, 54  
 FloatRect  
     sf::FloatRect, 55  
 FOOD  
     minerals, 20  
  
 g  
     sf::Color, 45  
 game\_loop  
     GameManager, 62  
 GameConfig, 56  
     berry\_growth\_time, 60  
     day\_length, 60  
     GameConfig, 58  
     get\_config\_level, 58  
     get\_instance, 58  
     get\_screen\_height, 58  
     get\_screen\_width, 58  
     get\_target\_fps, 59  
     get\_world\_size, 59  
     hostiles\_count, 60  
     max\_city\_size, 60  
     max\_spawn\_tries, 60  
     max\_tree\_age, 60  
     operator=, 59  
     resource\_scarcity, 61  
     set\_config\_level, 59  
     set\_world\_size, 59  
 GameManager, 61  
     ~GameManager, 62  
     draw\_buttons, 62  
     game\_loop, 62  
     GameManager, 62  
     get\_elapsed\_time, 62  
     is\_valid, 63  
     run, 63  
     setup\_buttons, 63  
     simulate\_tick, 63  
     update\_buttons, 63  
 ge  
     gtest\_lite, 17  
 gender  
     creature::Living, 105  
 generate\_world  
     TerrainContainer< T >, 158  
 get\_border\_height  
     World, 184  
 get\_border\_width  
     World, 185  
 get\_config\_level  
     GameConfig, 58  
 get\_current\_city\_center  
     World, 185  
 get\_elapsed\_time  
     GameManager, 62  
 get\_excluded\_entities  
     World, 185  
 get\_gender  
     creature::Living, 98  
 get\_harvested  
     minerals::ResourceStructure, 128  
 get\_height  
     TerrainContainer< T >, 158  
 get\_height\_offset  
     Shadowable, 132  
 get\_instance  
     GameConfig, 58  
     RandomGenerator, 120  
 get\_position\_nearby\_town  
     World, 186  
 get\_profession\_string  
     creature::Human, 75  
 get\_random\_house\_pos  
     World, 186  
 get\_random\_int  
     RandomGenerator, 120  
 get\_random\_suitable\_position  
     World, 186  
 get\_screen\_height  
     GameConfig, 58  
 get\_screen\_width  
     GameConfig, 58  
 get\_seed  
     TerrainContainer< T >, 158  
 get\_settlement\_age  
     minerals::CityCenter, 40  
 get\_shadow\_strength  
     Shadowable, 132  
 get\_skew\_offset  
     Shadowable, 133  
 get\_state

- creature::Living, 98
- get\_structure\_type
  - World, 186
- get\_target\_fps
  - GameConfig, 59
- get\_type
  - creature::Bear, 30
  - creature::Crocodile, 48
  - creature::Goat, 65
  - creature::Human, 76
  - creature::KillerRobot, 90
  - creature::Living, 98
  - minerals::BerryBush, 32
  - minerals::CityCenter, 40
  - minerals::House, 71
  - minerals::Iron, 85
  - minerals::Stone, 146
  - minerals::Structure, 152
  - minerals::Tree, 175
  - tiles::Tile, 171
- get\_width
  - creature::Living, 99
  - TerrainContainer< T >, 159
- get\_world\_size
  - GameConfig, 59
- getDesktopMode
  - sf::VideoMode, 179
- getElapsedTime
  - sf::Clock, 42
- getGlobalBounds
  - sf::Sprite, 143
- getInstance
  - TextureManager, 169
- getLocalBounds
  - sf::Sprite, 143
- getPosition
  - sf::Mouse, 108
  - sf::Sprite, 143
- getSize
  - sf::Texture, 166
- getStatus
  - sf::Music, 109
- getTest
  - gtest\_lite::Test, 164
- getTexture
  - sf::Sprite, 144
- getTileAt
  - World, 187
- goal
  - creature::HostileInterface, 70
  - creature::Human, 78
- Goat
  - creature::Goat, 64
- GRASS
  - tiles, 23
- Green
  - sf::Color, 45
- gt

- gtest\_lite, 18
- GTEND
  - gtest\_lite.h, 219
- gtest\_lite, 15
  - almostEQ, 16
  - eq, 16
  - eqstr, 16
  - eqstrcase, 16
  - EXPECT\_, 17
  - EXPECTSTR, 17
  - ge, 17
  - gt, 18
  - le, 18
  - lt, 18
  - ne, 18
  - nestr, 18
- gtest\_lite.h
  - ADD\_FAILURE, 213
  - ASSERT\_, 213
  - ASSERT\_EQ, 214
  - ASSERT\_NO\_THROW, 214
  - ASSERTTHROW, 214
  - CREATE\_Has\_, 214
  - CREATE\_Has\_fn\_, 215
  - END, 215
  - ENDM, 215
  - ENDMsg, 215
  - EXPECT\_ANY\_THROW, 215
  - EXPECT\_DOUBLE\_EQ, 215
  - EXPECT\_ENVCASEEQ, 216
  - EXPECT\_ENVEQ, 216
  - EXPECT\_EQ, 216
  - EXPECT\_FALSE, 216
  - EXPECT\_FLOAT\_EQ, 216
  - EXPECT\_GE, 216
  - EXPECT\_GT, 216
  - EXPECT\_LE, 217
  - EXPECT\_LT, 217
  - EXPECT\_NE, 217
  - EXPECT\_NO\_THROW, 217
  - EXPECT\_STRCASEEQ, 217
  - EXPECT\_STRCASENE, 217
  - EXPECT\_STREQ, 218
  - EXPECT\_STRNE, 218
  - EXPECT\_THROW, 218
  - EXPECT\_THROW\_THROW, 218
  - EXPECT\_TRUE, 218
  - EXPECTTHROW, 218
  - FAIL, 219
  - GTEND, 219
  - GTINIT, 219
  - hasMember, 219
  - SUCCEED, 219
  - TEST, 219
- gtest\_lite::ostreamRedir, 112
  - ~ostreamRedir, 113
  - ostreamRedir, 113
- gtest\_lite::Test, 162

- ~Test, [163](#)
- ablocks, [164](#)
- astatus, [163](#)
- begin, [163](#)
- end, [163](#)
- expect, [163](#)
- fail, [164](#)
- failed, [164](#)
- getTest, [164](#)
- name, [164](#)
- null, [165](#)
- os, [165](#)
- status, [165](#)
- sum, [165](#)
- tmp, [165](#)
- GTINIT
  - gtest\_lite.h, [219](#)
- harvest
  - minerals::BerryBush, [32](#)
  - minerals::Iron, [85](#)
  - minerals::ResourceStructure, [128](#)
  - minerals::Stone, [146](#)
  - minerals::Tree, [175](#)
- harvested
  - minerals::ResourceStructure, [129](#)
- hasMember
  - gtest\_lite.h, [219](#)
- health
  - creature::Living, [105](#)
- height
  - sf::Bound, [33](#)
  - sf::FloatRect, [56](#)
  - sf::IntRect, [83](#)
  - sf::VideoMode, [179](#)
- height\_offset
  - Shadowable, [135](#)
- hit\_timer
  - creature::Living, [105](#)
- hostile\_run
  - creature::HostileInterface, [68](#)
- hostile\_walk
  - creature::HostileInterface, [68](#)
- hostiles\_count
  - GameConfig, [60](#)
- House
  - minerals::House, [71](#)
- HOUSING
  - minerals, [20](#)
- HUMAN
  - creature, [14](#)
- Human
  - creature::Human, [74](#)
- human\_factory
  - SaveHelpers.cpp, [226](#)
  - SaveHelpers.hpp, [228](#)
- HumanFactory
  - SaveHelpers.hpp, [228](#)
- hungarian
  - sf, [22](#)
- IDLE
  - creature, [15](#)
- ImportInvalidEntityException, [79](#)
  - ImportInvalidEntityException, [79](#)
- ImportInvalidHousingLevelException, [80](#)
  - ImportInvalidHousingLevelException, [80](#)
- ImportInvalidHumanProfessionException, [81](#)
  - ImportInvalidHumanProfessionException, [81](#)
- ImportInvalidResourceException, [81](#)
  - ImportInvalidResourceException, [82](#)
- increment
  - sf::ClockTime, [43](#)
- init
  - tiles::Tile, [171](#)
- init\_spritesheet\_data
  - creature::Living, [99](#)
- initialize
  - creature::Human, [76](#)
- inner\_timer
  - creature::Living, [105](#)
  - minerals::ResourceStructure, [129](#)
- IntRect
  - sf::IntRect, [82](#), [83](#)
- Invalid
  - sf::Event, [50](#)
- InvalidBorderSizeException, [84](#)
  - InvalidBorderSizeException, [84](#)
- IRON
  - minerals, [20](#)
- Iron
  - minerals::Iron, [85](#)
- iron\_req
  - minerals::House, [72](#)
- is\_on\_screen
  - TerrainContainer< T >, [159](#)
- is\_there\_enough\_resource
  - World, [187](#)
- is\_there\_room\_for\_housing
  - minerals::CityCenter, [40](#)
- is\_valid
  - GameManager, [63](#)
- is\_valid\_coordinate
  - TerrainContainer< T >, [159](#)
- isButtonPressed
  - sf::Mouse, [108](#)
- isKeyPressed
  - sf::Keyboard, [87](#)
- isOpen
  - sf::RenderWindow, [126](#)
- isValid
  - sf::VideoMode, [179](#)
- Keydownntype
  - sf::Keyboard, [87](#)
- KillerRobot
  - creature::KillerRobot, [89](#)
- King

- creature::King, [92](#)
- le
  - gtest\_lite, [18](#)
- LEFT
  - creature, [15](#)
- Left
  - sf::Keyboard, [87](#)
  - sf::Mouse, [107](#)
- left
  - sf::FloatRect, [56](#)
  - sf::IntRect, [83](#)
- level
  - minerals::House, [72](#)
- LIVINGSTATE
  - creature, [15](#)
- load\_music
  - MusicPlayer, [111](#)
- load\_profession
  - Profession, [118](#)
- load\_sound
  - SoundPlayer, [140](#)
- loadFile
  - SaveManager, [130](#)
- loadFromFile
  - sf::SoundBuffer, [139](#)
  - sf::Texture, [166](#)
- loadTexture
  - TextureManager, [169](#)
- look\_left
  - creature::Living, [99](#)
- look\_right
  - creature::Living, [99](#)
- It
  - gtest\_lite, [18](#)
- main
  - main.cpp, [223](#)
- main.cpp
  - main, [223](#)
- MALE
  - creature, [14](#)
- matrix
  - sf::Transform, [174](#)
- max\_age
  - creature::Living, [105](#)
- max\_city\_size
  - GameConfig, [60](#)
- MAX\_CREATURE\_SIZE
  - creature::Living, [106](#)
- MAX\_OBJECT\_SIZE
  - minerals::Structure, [153](#)
- max\_spawn\_tries
  - GameConfig, [60](#)
- max\_tree\_age
  - GameConfig, [60](#)
- mineral\_to\_string
  - minerals, [20](#)
- MINERAL\_TYPE
  - minerals, [19](#)
- minerals, [19](#)
  - CITY\_CENTER, [20](#)
  - FOOD, [20](#)
  - HOUSING, [20](#)
  - IRON, [20](#)
  - mineral\_to\_string, [20](#)
  - MINERAL\_TYPE, [19](#)
  - STONE, [20](#)
  - WOOD, [20](#)
- minerals::BerryBush, [31](#)
  - BerryBush, [32](#)
  - get\_type, [32](#)
  - harvest, [32](#)
  - update\_logic, [32](#)
- minerals::CityCenter, [39](#)
  - CityCenter, [39](#)
  - get\_settlement\_age, [40](#)
  - get\_type, [40](#)
  - is\_there\_room\_for\_housing, [40](#)
  - register\_new\_house, [40](#)
  - update\_logic, [40](#)
- minerals::House, [70](#)
  - get\_type, [71](#)
  - House, [71](#)
  - iron\_req, [72](#)
  - level, [72](#)
  - stone\_req, [72](#)
  - update\_logic, [72](#)
  - wood\_req, [72](#)
- minerals::Iron, [84](#)
  - get\_type, [85](#)
  - harvest, [85](#)
  - Iron, [85](#)
  - update\_logic, [86](#)
- minerals::ResourceStructure, [127](#)
  - ~ResourceStructure, [128](#)
  - get\_harvested, [128](#)
  - harvest, [128](#)
  - harvested, [129](#)
  - inner\_timer, [129](#)
  - ResourceStructure, [128](#)
- minerals::Stone, [145](#)
  - get\_type, [146](#)
  - harvest, [146](#)
  - Stone, [146](#)
  - update\_logic, [146](#)
- minerals::Structure, [150](#)
  - ~Structure, [151](#)
  - draw, [151](#)
  - draw\_logic, [152](#)
  - get\_type, [152](#)
  - MAX\_OBJECT\_SIZE, [153](#)
  - needs\_drawn, [152](#)
  - posx, [154](#)
  - posy, [154](#)
  - setPosition, [152](#)
  - setTexture, [153](#)

- Structure, [151](#)
- update\_logic, [153](#)
- minerals::Tree, [174](#)
  - get\_type, [175](#)
  - harvest, [175](#)
  - Tree, [175](#)
  - update\_logic, [175](#)
- mining\_iron
  - creature::Stonemason, [149](#)
- MOUNTAIN
  - tiles, [23](#)
- Mousedowntype
  - sf::Mouse, [107](#)
- Multiply
  - sf, [21](#)
- Music
  - sf::Music, [109](#)
- MusicLoadException, [110](#)
  - MusicLoadException, [110](#)
- MusicPlayer, [111](#)
  - ~MusicPlayer, [111](#)
  - load\_music, [111](#)
  - MusicPlayer, [111](#)
  - set\_volume, [112](#)
  - toggle\_music, [112](#)
- name
  - gtest\_lite::Test, [164](#)
- ne
  - gtest\_lite, [18](#)
- needs\_drawn
  - creature::Living, [100](#)
  - minerals::Structure, [152](#)
- needs\_promotion
  - creature::Human, [78](#)
- needs\_to\_be\_royal
  - creature::Human, [78](#)
- nestr
  - gtest\_lite, [18](#)
- NoEvent
  - sf::Event, [50](#)
- None
  - sf, [21](#)
- null
  - gtest\_lite::Test, [165](#)
- onClick
  - ui::Button, [37](#)
- openFromFile
  - sf::Music, [109](#)
- operator<<
  - sf, [21](#)
  - World, [191](#)
  - World.cpp, [235](#)
- operator>>
  - World, [192](#)
  - World.cpp, [235](#)
- operator=
  - GameConfig, [59](#)
- RandomGenerator, [120](#)
- operator[]
  - TerrainContainer< T >, [160](#)
- os
  - gtest\_lite::Test, [165](#)
- ostreamRedir
  - gtest\_lite::ostreamRedir, [113](#)
- Paused
  - sf::SoundSource, [141](#)
- play
  - sf::Music, [109](#)
  - sf::Sound, [138](#)
- play\_sound
  - SoundPlayer, [140](#)
- Playing
  - sf::SoundSource, [141](#)
- pollEvent
  - sf::RenderWindow, [126](#)
- populate\_world
  - World, [188](#)
- position
  - sf::RectangleShape, [122](#)
- PostProcessor, [113](#)
  - draw, [114](#)
  - PostProcessor, [114](#)
  - setColorOverlay, [114](#)
  - setRenderSize, [115](#)
  - setTextureFor, [115](#)
  - toggle\_chromatic\_aberration, [115](#)
  - toggle\_noise, [116](#)
  - toggle\_vignette, [116](#)
- posx
  - creature::Living, [106](#)
  - minerals::Structure, [154](#)
- posy
  - creature::Living, [106](#)
  - minerals::Structure, [154](#)
- Profession, [116](#)
  - draw, [117](#)
  - load\_profession, [118](#)
  - Profession, [117](#)
  - setPosition, [118](#)
  - setTexture, [118](#)
  - to\_string, [119](#)
- profession
  - creature::Human, [79](#)
- r
  - sf::Color, [45](#)
- RandomGenerator, [119](#)
  - get\_instance, [120](#)
  - get\_random\_int, [120](#)
  - operator=, [120](#)
  - RandomGenerator, [120](#)
- ReadSaveFileFail, [121](#)
  - ReadSaveFileFail, [121](#)
- Red
  - sf::Color, [45](#)

- regenerate
  - World, 188
- register\_new\_house
  - minerals::CityCenter, 40
- remove\_resources
  - World, 188
- remove\_structure\_at
  - World, 188
- RenderStates
  - sf::RenderStates, 123
- RenderWindow
  - sf::RenderWindow, 125
- reset
  - sf::ClockTime, 43
- resize
  - TerrainContainer< T >, 160
- resource\_factory
  - SaveHelpers.cpp, 226
  - SaveHelpers.hpp, 228
- resource\_scarcity
  - GameConfig, 61
- ResourceFactory
  - SaveHelpers.hpp, 228
- ResourceStructure
  - minerals::ResourceStructure, 128
- restart
  - sf::Clock, 42
- retarget
  - creature::HostileInterface, 68
  - creature::Living, 100
- RIGHT
  - creature, 15
- Right
  - sf::Keyboard, 87
  - sf::Mouse, 107
- ROBOTIC
  - creature, 14
- RUN
  - creature, 15
- run
  - GameManager, 63
- run\_speed\_modifier
  - creature::Living, 106
- save\_name
  - creature::Living, 106
- saveFile
  - SaveManager, 130
- SaveHelpers.cpp
  - creature\_factory, 225
  - human\_factory, 226
  - resource\_factory, 226
- SaveHelpers.hpp
  - creature\_factory, 228
  - CreatureFactory, 228
  - human\_factory, 228
  - HumanFactory, 228
  - resource\_factory, 228
  - ResourceFactory, 228
- SaveManager, 129
  - deleteFile, 130
  - loadFile, 130
  - saveFile, 130
  - SaveManager, 129
- select\_target
  - creature::Bear, 30
  - creature::Crocodile, 48
  - creature::HostileInterface, 69
  - creature::KillerRobot, 91
- select\_texture
  - creature::Human, 76
- set\_attack\_texture
  - creature::Living, 100
- set\_border\_height
  - World, 189
- set\_border\_width
  - World, 189
- set\_config\_level
  - GameConfig, 59
- set\_death\_texture
  - creature::Living, 100
- set\_health
  - creature::Living, 101
- set\_height\_offset
  - Shadowable, 133
- set\_hostile\_config
  - creature::HostileInterface, 69
- set\_idle\_texture
  - creature::Living, 101
- set\_run\_texture
  - creature::Living, 101
- set\_seed
  - TerrainContainer< T >, 161
- set\_shadow\_strength
  - Shadowable, 133
- set\_skew\_offset
  - Shadowable, 134
- set\_state
  - creature::Living, 102
- set\_volume
  - MusicPlayer, 112
- set\_walk\_texture
  - creature::Living, 102
- set\_world\_size
  - GameConfig, 59
- setBlendMode
  - sf::RenderStates, 123
- setBuffer
  - sf::Sound, 138
- setCallback
  - ui::Button, 37
- setColor
  - sf::Sprite, 144
- setColorOverlay
  - PostProcessor, 114
- setFillColor
  - sf::RectangleShape, 122



- setFramerateLimit
  - sf::RenderWindow, 126
- setLoop
  - sf::Music, 109
- setOrigin
  - sf::Sprite, 144
- setPosition
  - creature::Living, 102
  - minerals::Structure, 152
  - Profession, 118
  - sf::RectangleShape, 122
  - sf::Sprite, 144
  - Textureable, 168
  - tiles::Tile, 171
  - ui::Button, 37
- setRenderSize
  - PostProcessor, 115
- setRotation
  - sf::Sprite, 144
- setScale
  - sf::Sprite, 144
- setShadow
  - Shadowable, 134
- setShadowDayNightCycle
  - Shadowable, 134
- setShadowPosition
  - Shadowable, 134
- setShadowTexture
  - Shadowable, 135
- setSize
  - sf::RectangleShape, 122
- setTexture
  - creature::Living, 102
  - minerals::Structure, 153
  - Profession, 118
  - sf::Sprite, 144
  - Textureable, 168
  - tiles::Tile, 172
  - ui::Button, 38
- setTextureFor
  - PostProcessor, 115
- setTextureRect
  - sf::Sprite, 145
- setTheShadow
  - creature::Living, 103
- setTransform
  - sf::RenderStates, 123
- setup\_buttons
  - GameManager, 63
- setVolume
  - sf::Music, 109
- sf, 20
  - Additive, 21
  - Alpha, 21
  - BlendAdd, 21, 22
  - BlendMode, 21
  - file\_exists\_at\_path, 21
  - hungarian, 22
  - Multiply, 21
  - None, 21
  - operator<<, 21
- sf::Bound, 33
  - height, 33
  - width, 33
- sf::Clock, 42
  - getElapsedTime, 42
  - restart, 42
- sf::ClockTime, 42
  - asSeconds, 43
  - ClockTime, 42, 43
  - increment, 43
  - reset, 43
- sf::Color, 43
  - a, 44
  - b, 45
  - Black, 45
  - Blue, 45
  - Color, 44
  - g, 45
  - Green, 45
  - r, 45
  - Red, 45
  - Transparent, 45
  - White, 46
- sf::Event, 49
  - Closed, 50
  - EType, 49
  - Event, 50
  - Invalid, 50
  - NoEvent, 50
  - type, 50
- sf::FloatRect, 55
  - contains, 55
  - FloatRect, 55
  - height, 56
  - left, 56
  - top, 56
  - width, 56
- sf::IntRect, 82
  - height, 83
  - IntRect, 82, 83
  - left, 83
  - top, 83
  - width, 83
- sf::Keyboard, 87
  - Down, 87
  - isKeyPressed, 87
  - Keydownntype, 87
  - Left, 87
  - Right, 87
  - simulate\_key\_press, 88
  - simulate\_key\_release, 88
  - Up, 87
- sf::Mouse, 107
  - getPosition, 108
  - isButtonPressed, 108

- Left, 107
- MousedownType, 107
- Right, 107
- simulate\_key\_press, 108
- simulate\_key\_release, 108
- sf::Music, 108
  - getStatus, 109
  - Music, 109
  - openFromFile, 109
  - play, 109
  - setLoop, 109
  - setVolume, 109
  - stop, 109
- sf::RectangleShape, 122
  - position, 122
  - setFillColor, 122
  - setPosition, 122
  - setSize, 122
- sf::RenderStates, 123
  - blendMode, 124
  - RenderStates, 123
  - setBlendMode, 123
  - setTransform, 123
  - transform, 124
- sf::RenderWindow, 124
  - clear, 125
  - close, 125
  - create, 125
  - display, 126
  - draw, 126
  - isOpen, 126
  - pollEvent, 126
  - RenderWindow, 125
  - setFramerateLimit, 126
- sf::Sound, 138
  - ~Sound, 138
  - play, 138
  - setBuffer, 138
  - stop, 139
- sf::SoundBuffer, 139
  - loadFromFile, 139
- sf::SoundSource, 141
  - ~SoundSource, 142
  - Paused, 141
  - Playing, 141
  - SoundSource, 142
  - SoundSourceType, 141
  - Stopped, 141
  - type, 142
- sf::Sprite, 142
  - ~Sprite, 143
  - draw, 143
  - getGlobalBounds, 143
  - getLocalBounds, 143
  - getPosition, 143
  - getTexture, 144
  - setColor, 144
  - setOrigin, 144
  - setPosition, 144
  - setRotation, 144
  - setScale, 144
  - setTexture, 144
  - setTextureRect, 145
  - Sprite, 143
- sf::Texture, 166
  - getSize, 166
  - loadFromFile, 166
  - Texture, 166
- sf::Transform, 172
  - combine, 173
  - matrix, 174
  - Transform, 172, 173
  - transformPoint, 173
  - translate, 173
- sf::Vector2f, 176
  - Vector2f, 176
  - x, 176
  - y, 177
- sf::Vector2i, 177
  - Vector2i, 177
  - x, 178
  - y, 178
- sf::VideoMode, 178
  - bitsPerPixel, 179
  - getDesktopMode, 179
  - height, 179
  - isValid, 179
  - VideoMode, 178
  - width, 179
- shadow\_logic
  - creature::Living, 103
- Shadowable, 131
  - ~Shadowable, 132
  - drawShadow, 132
  - get\_height\_offset, 132
  - get\_shadow\_strength, 132
  - get\_skew\_offset, 133
  - height\_offset, 135
  - set\_height\_offset, 133
  - set\_shadow\_strength, 133
  - set\_skew\_offset, 134
  - setShadow, 134
  - setShadowDayNightCycle, 134
  - setShadowPosition, 134
  - setShadowTexture, 135
- simulate\_key\_press
  - sf::Keyboard, 88
  - sf::Mouse, 108
- simulate\_key\_release
  - sf::Keyboard, 88
  - sf::Mouse, 108
- simulate\_tick
  - GameManager, 63
- SimulationException, 135
  - SimulationException, 136
- Soldier

creature::Soldier, [137](#)  
 SoundPlayer, [139](#)  
     load\_sound, [140](#)  
     play\_sound, [140](#)  
     stop\_sound, [140](#)  
 SoundSource  
     sf::SoundSource, [142](#)  
 SoundSourceType  
     sf::SoundSource, [141](#)  
 spawn\_entity  
     World, [189](#)  
 spawn\_structure  
     World, [190](#)  
 spawn\_structure\_at  
     World, [190](#)  
 speed  
     creature::Living, [106](#)  
 Sprite  
     sf::Sprite, [143](#)  
 src/creatures/Goat.cpp, [193](#)  
 src/creatures/Goat.d, [193](#)  
 src/creatures/Goat.hpp, [193](#)  
 src/creatures/HostileInterface.cpp, [194](#)  
 src/creatures/HostileInterface.d, [194](#)  
 src/creatures/HostileInterface.hpp, [194](#)  
 src/creatures/hostiles/Bear.cpp, [195](#)  
 src/creatures/hostiles/Bear.d, [195](#)  
 src/creatures/hostiles/Bear.hpp, [195](#)  
 src/creatures/hostiles/Crocodile.cpp, [196](#)  
 src/creatures/hostiles/Crocodile.d, [196](#)  
 src/creatures/hostiles/Crocodile.hpp, [196](#)  
 src/creatures/hostiles/KillerRobot.cpp, [197](#)  
 src/creatures/hostiles/KillerRobot.d, [197](#)  
 src/creatures/hostiles/KillerRobot.hpp, [197](#)  
 src/creatures/humans/AnglerMiner.cpp, [198](#)  
 src/creatures/humans/AnglerMiner.d, [198](#)  
 src/creatures/humans/AnglerMiner.hpp, [198](#)  
 src/creatures/humans/Builder.cpp, [199](#)  
 src/creatures/humans/Builder.d, [199](#)  
 src/creatures/humans/Builder.hpp, [199](#)  
 src/creatures/humans/Farmer.cpp, [200](#)  
 src/creatures/humans/Farmer.d, [200](#)  
 src/creatures/humans/Farmer.hpp, [200](#)  
 src/creatures/humans/Fisherman.cpp, [201](#)  
 src/creatures/humans/Fisherman.d, [201](#)  
 src/creatures/humans/Fisherman.hpp, [201](#)  
 src/creatures/humans/Human.cpp, [202](#)  
 src/creatures/humans/Human.d, [202](#)  
 src/creatures/humans/Human.hpp, [202](#)  
 src/creatures/humans/King.cpp, [203](#)  
 src/creatures/humans/King.d, [203](#)  
 src/creatures/humans/King.hpp, [203](#)  
 src/creatures/humans/Soldier.cpp, [204](#)  
 src/creatures/humans/Soldier.d, [204](#)  
 src/creatures/humans/Soldier.hpp, [204](#)  
 src/creatures/humans/Stonemason.cpp, [205](#)  
 src/creatures/humans/Stonemason.d, [205](#)  
 src/creatures/humans/Stonemason.hpp, [205](#)  
 src/creatures/humans/Woodcutter.cpp, [206](#)  
 src/creatures/humans/Woodcutter.d, [206](#)  
 src/creatures/humans/Woodcutter.hpp, [206](#)  
 src/creatures/Living.cpp, [207](#)  
 src/creatures/Living.d, [207](#)  
 src/creatures/Living.hpp, [207](#)  
 src/exceptions/FileExceptions.hpp, [208](#)  
 src/exceptions/MusicLoadException.hpp, [209](#)  
 src/exceptions/SimulationException.hpp, [209](#)  
 src/exceptions/WorldExceptions.hpp, [210](#)  
 src/external/gtest\_lite.h, [210](#)  
 src/external/memtrace.cpp, [220](#)  
 src/external/memtrace.d, [220](#)  
 src/external/memtrace.h, [220](#)  
 src/fake\_sfml/fake\_sfml.cpp, [220](#)  
 src/fake\_sfml/fake\_sfml.d, [220](#)  
 src/fake\_sfml/fake\_sfml.hpp, [220](#)  
 src/GameConfig.cpp, [221](#)  
 src/GameConfig.d, [221](#)  
 src/GameConfig.hpp, [221](#)  
 src/GameManager.cpp, [222](#)  
 src/GameManager.d, [222](#)  
 src/GameManager.hpp, [222](#)  
 src/main.cpp, [222](#)  
 src/main.d, [223](#)  
 src/MusicPlayer.cpp, [223](#)  
 src/MusicPlayer.d, [223](#)  
 src/MusicPlayer.hpp, [223](#)  
 src/PostProcessor.cpp, [223](#)  
 src/PostProcessor.d, [223](#)  
 src/PostProcessor.hpp, [223](#)  
 src/Profession.cpp, [224](#)  
 src/Profession.d, [224](#)  
 src/Profession.hpp, [224](#)  
 src/Random\_Gen.cpp, [224](#)  
 src/Random\_Gen.d, [225](#)  
 src/Random\_Gen.hpp, [225](#)  
 src/SaveHelpers.cpp, [225](#)  
 src/SaveHelpers.d, [226](#)  
 src/SaveHelpers.hpp, [226](#)  
 src/SaveManager.cpp, [228](#)  
 src/SaveManager.d, [228](#)  
 src/SaveManager.hpp, [228](#)  
 src/Shadowable.cpp, [229](#)  
 src/Shadowable.d, [229](#)  
 src/Shadowable.hpp, [229](#)  
 src/SoundPlayer.cpp, [229](#)  
 src/SoundPlayer.d, [230](#)  
 src/SoundPlayer.hpp, [230](#)  
 src/terrain\_tiles/Tile.cpp, [230](#)  
 src/terrain\_tiles/Tile.d, [230](#)  
 src/terrain\_tiles/Tile.hpp, [230](#)  
 src/TerrainContainer.hpp, [231](#)  
 src/TerrainContainer.inl, [231](#)  
 src/Textureable.hpp, [232](#)  
 src/TextureManager.cpp, [232](#)  
 src/TextureManager.d, [232](#)  
 src/TextureManager.hpp, [232](#)

- src/ui/button.cpp, 233
- src/ui/button.d, 233
- src/ui/button.hpp, 233
- src/Utils.cpp, 233
- src/Utils.d, 234
- src/Utils.hpp, 234
- src/World.cpp, 235
- src/World.d, 236
- src/World.hpp, 236
- src/world\_object/BerryBush.cpp, 237
- src/world\_object/BerryBush.d, 237
- src/world\_object/BerryBush.hpp, 237
- src/world\_object/CityCenter.cpp, 237
- src/world\_object/CityCenter.d, 237
- src/world\_object/CityCenter.hpp, 237
- src/world\_object/House.cpp, 238
- src/world\_object/House.d, 238
- src/world\_object/House.hpp, 238
- src/world\_object/Iron.cpp, 239
- src/world\_object/Iron.d, 239
- src/world\_object/Iron.hpp, 239
- src/world\_object/ResourceStructure.cpp, 239
- src/world\_object/ResourceStructure.d, 240
- src/world\_object/ResourceStructure.hpp, 240
- src/world\_object/Stone.cpp, 240
- src/world\_object/Stone.d, 240
- src/world\_object/Stone.hpp, 240
- src/world\_object/Structure.cpp, 241
- src/world\_object/Structure.d, 241
- src/world\_object/Structure.hpp, 241
- src/world\_object/Tree.cpp, 242
- src/world\_object/Tree.d, 242
- src/world\_object/Tree.hpp, 242
- state
  - creature::Living, 107
- status
  - gtest\_lite::Test, 165
- STONE
  - minerals, 20
- Stone
  - minerals::Stone, 146
- stone\_req
  - minerals::House, 72
- Stonemason
  - creature::Stonemason, 148
- stop
  - sf::Music, 109
  - sf::Sound, 139
- stop\_sound
  - SoundPlayer, 140
- Stopped
  - sf::SoundSource, 141
- Structure
  - minerals::Structure, 151
- StructureException, 154
  - StructureException, 155
- SUCCEED
  - gtest\_lite.h, 219
- sum
  - gtest\_lite::Test, 165
- swap\_at
  - TerrainContainer< T >, 161
- target
  - creature::HostileInterface, 70
- TerrainContainer
  - TerrainContainer< T >, 156
- TerrainContainer< T >, 155
  - ~TerrainContainer, 157
  - clear, 157
  - clear\_at, 157
  - draw, 158
  - generate\_world, 158
  - get\_height, 158
  - get\_seed, 158
  - get\_width, 159
  - is\_on\_screen, 159
  - is\_valid\_coordinate, 159
  - operator[], 160
  - resize, 160
  - set\_seed, 161
  - swap\_at, 161
  - TerrainContainer, 156
  - TILE\_SIZE, 161
- TerrainContainer.inl
  - DYNAMIC2D\_INL, 231
- TEST
  - gtest\_lite.h, 219
- Texture
  - sf::Texture, 166
- Textureable, 167
  - ~Textureable, 167
  - draw, 167
  - setPosition, 168
  - setTexture, 168
- TextureManager, 169
  - clear, 169
  - getInstance, 169
  - loadTexture, 169
- TILE\_SIZE
  - TerrainContainer< T >, 161
- tiles, 22
  - GRASS, 23
  - MOUNTAIN, 23
  - TILETYPE, 22
  - WATER, 23
- tiles::Tile, 170
  - draw, 170
  - get\_type, 171
  - init, 171
  - setPosition, 171
  - setTexture, 172
- TILETYPE
  - tiles, 22
- tmp
  - gtest\_lite::Test, 165
- to\_string

- Profession, [119](#)
- toggle\_chromatic\_aberration
  - PostProcessor, [115](#)
- toggle\_music
  - MusicPlayer, [112](#)
- toggle\_noise
  - PostProcessor, [116](#)
- toggle\_vignette
  - PostProcessor, [116](#)
- top
  - sf::FloatRect, [56](#)
  - sf::IntRect, [83](#)
- Transform
  - sf::Transform, [172](#), [173](#)
- transform
  - sf::RenderStates, [124](#)
- transformPoint
  - sf::Transform, [173](#)
- translate
  - sf::Transform, [173](#)
- Transparent
  - sf::Color, [45](#)
- Tree
  - minerals::Tree, [175](#)
- try\_attack
  - creature::HostileInterface, [69](#)
- try\_develop\_random\_role
  - World, [190](#)
- try\_fishing
  - creature::Fisherman, [54](#)
- try\_hover\_animation
  - ui::Button, [38](#)
- try\_mine
  - creature::Stonemason, [148](#)
- type
  - sf::Event, [50](#)
  - sf::SoundSource, [142](#)
- ui, [23](#)
- ui::Button, [35](#)
  - Button, [36](#)
  - draw, [37](#)
  - onClick, [37](#)
  - setCallback, [37](#)
  - setPosition, [37](#)
  - setTexture, [38](#)
  - try\_hover\_animation, [38](#)
- Up
  - sf::Keyboard, [87](#)
- update\_buttons
  - GameManager, [63](#)
- update\_logic
  - creature::AnglerMiner, [27](#)
  - creature::Bear, [30](#)
  - creature::Builder, [35](#)
  - creature::Crocodile, [48](#)
  - creature::Farmer, [51](#)
  - creature::Fisherman, [54](#)
  - creature::Goat, [66](#)
  - creature::Human, [78](#)
  - creature::KillerRobot, [91](#)
  - creature::King, [93](#)
  - creature::Living, [104](#)
  - creature::Soldier, [137](#)
  - creature::Stonemason, [149](#)
  - creature::Woodcutter, [181](#)
  - minerals::BerryBush, [32](#)
  - minerals::CityCenter, [40](#)
  - minerals::House, [72](#)
  - minerals::Iron, [86](#)
  - minerals::Stone, [146](#)
  - minerals::Structure, [153](#)
  - minerals::Tree, [175](#)
- update\_spritesheet
  - creature::Living, [104](#)
- update\_world
  - World, [191](#)
- upgrade\_house\_at
  - World, [191](#)
- Utils.cpp
  - distance\_to, [234](#)
- Utils.hpp
  - distance\_to, [235](#)
- Vector2f
  - sf::Vector2f, [176](#)
- Vector2i
  - sf::Vector2i, [177](#)
- VideoMode
  - sf::VideoMode, [178](#)
- WALK
  - creature, [15](#)
- WATER
  - tiles, [23](#)
- White
  - sf::Color, [46](#)
- width
  - sf::Bound, [33](#)
  - sf::FloatRect, [56](#)
  - sf::IntRect, [83](#)
  - sf::VideoMode, [179](#)
- WOOD
  - minerals, [20](#)
- wood\_req
  - minerals::House, [72](#)
- Woodcutter
  - creature::Woodcutter, [180](#)
- World, [181](#)
  - ~World, [183](#)
  - add\_resources, [183](#)
  - build\_city\_center\_at, [184](#)
  - clear, [184](#)
  - draw, [184](#)
  - get\_border\_height, [184](#)
  - get\_border\_width, [185](#)
  - get\_current\_city\_center, [185](#)
  - get\_excluded\_entities, [185](#)

- get\_position\_nearby\_town, [186](#)
- get\_random\_house\_pos, [186](#)
- get\_random\_suitable\_position, [186](#)
- get\_structure\_type, [186](#)
- getTileAt, [187](#)
- is\_there\_enough\_resource, [187](#)
- operator<<, [191](#)
- operator>>, [192](#)
- populate\_world, [188](#)
- regenerate, [188](#)
- remove\_resources, [188](#)
- remove\_structure\_at, [188](#)
- set\_border\_height, [189](#)
- set\_border\_width, [189](#)
- spawn\_entity, [189](#)
- spawn\_structure, [190](#)
- spawn\_structure\_at, [190](#)
- try\_develop\_random\_role, [190](#)
- update\_world, [191](#)
- upgrade\_house\_at, [191](#)
- World, [183](#)

#### World.cpp

- operator<<, [235](#)
- operator>>, [235](#)

#### x

- sf::Vector2f, [176](#)
- sf::Vector2i, [178](#)

#### y

- sf::Vector2f, [177](#)
- sf::Vector2i, [178](#)