

PingForce TD

Készítette Doxygen 1.9.4

1. Névtérmutató	1
1.1. Névtérlista	1
2. Hierarchikus mutató	3
2.1. Osztályhierarchia	3
3. Adatszerkezet-mutató	5
3.1. Adatszerkezetek	5
4. Fájlmutató	7
4.1. Fájllista	7
5. Névterek dokumentációja	9
5.1. pftd névtér-referencia	9
5.1.1. Típusdefiníciók dokumentációja	10
5.1.1.1. EndPoint	10
5.2. pftd::gr névtér-referencia	10
5.3. pftd::utils névtér-referencia	10
5.3.1. Típusdefiníciók dokumentációja	10
5.3.1.1. Vec2f	10
5.3.1.2. Vec2i	11
5.4. pftd::utils::parser névtér-referencia	11
6. Adatszerkezetek dokumentációja	13
6.1. pftd::App osztályreferencia	13
6.1.1. Részletes leírás	14
6.1.2. Konstruktorkok és destruktorkok dokumentációja	14
6.1.2.1. App() [1/3]	14
6.1.2.2. App() [2/3]	14
6.1.2.3. ~App()	14
6.1.2.4. App() [3/3]	14
6.1.3. Tagfüggvények dokumentációja	15
6.1.3.1. addScene()	15
6.1.3.2. changeScene()	15
6.1.3.3. create()	15
6.1.3.4. destroy()	16
6.1.3.5. getInstance()	16
6.1.3.6. getWindowHeight()	16
6.1.3.7. getWindowWidth()	17
6.1.3.8. isRunning()	17
6.1.3.9. operator=()	17
6.1.3.10. run()	17
6.1.4. Adatmezők dokumentációja	17
6.1.4.1. m_activeSceneID	17

6.1.4.2.	m_instance	18
6.1.4.3.	m_renderer	18
6.1.4.4.	m_running	18
6.1.4.5.	m_scenes	18
6.2.	pftd::gr::Button osztályreferencia	18
6.2.1.	Részletes leírás	19
6.2.2.	Konstruktorok és destruktorok dokumentációja	19
6.2.2.1.	Button()	19
6.2.2.2.	~Button()	20
6.2.3.	Tagfüggvények dokumentációja	20
6.2.3.1.	draw()	20
6.2.3.2.	getLabel()	20
6.2.3.3.	handleClick()	20
6.2.3.4.	setBackground()	21
6.2.3.5.	setSound()	21
6.2.4.	Adatmezők dokumentációja	21
6.2.4.1.	label	21
6.2.4.2.	m_background	21
6.2.4.3.	m_clickSound	22
6.2.4.4.	m_rect	22
6.3.	pftd::Clickable osztályreferencia	22
6.3.1.	Részletes leírás	23
6.3.2.	Konstruktorok és destruktorok dokumentációja	23
6.3.2.1.	Clickable() [1/2]	23
6.3.2.2.	Clickable() [2/2]	23
6.3.3.	Tagfüggvények dokumentációja	23
6.3.3.1.	handleClick()	23
6.3.3.2.	setCallback()	23
6.3.4.	Adatmezők dokumentációja	25
6.3.4.1.	isActive	25
6.3.4.2.	m_callback	25
6.4.	pftd::Object::Compare struktúrareferencia	25
6.4.1.	Részletes leírás	25
6.4.2.	Tagfüggvények dokumentációja	25
6.4.2.1.	operator() [1/2]	26
6.4.2.2.	operator() [2/2]	26
6.5.	pftd::utils::Container< T, C > osztálysablon-referencia	26
6.5.1.	Részletes leírás	26
6.5.2.	Konstruktorok és destruktorok dokumentációja	27
6.5.2.1.	Container()	27
6.5.2.2.	~Container()	27
6.5.3.	Tagfüggvények dokumentációja	27

6.5.3.1.	append()	27
6.5.3.2.	getContainer()	27
6.5.3.3.	size()	28
6.5.4.	Adatmezők dokumentációja	28
6.5.4.1.	m_container	28
6.6.	pftd::Cub osztályreferencia	28
6.6.1.	Részletes leírás	29
6.6.2.	Konstruktorok és destruktorok dokumentációja	29
6.6.2.1.	Cub()	29
6.6.3.	Tagfüggvények dokumentációja	29
6.6.3.1.	clone()	29
6.7.	pftd::Entity osztályreferencia	30
6.7.1.	Részletes leírás	31
6.7.2.	Konstruktorok és destruktorok dokumentációja	31
6.7.2.1.	Entity() [1/4]	31
6.7.2.2.	Entity() [2/4]	31
6.7.2.3.	Entity() [3/4]	31
6.7.2.4.	Entity() [4/4]	32
6.7.2.5.	~Entity()	32
6.7.3.	Tagfüggvények dokumentációja	32
6.7.3.1.	advanceAnimationFrame()	32
6.7.3.2.	clone()	32
6.7.3.3.	draw()	32
6.7.3.4.	getPosition()	33
6.7.3.5.	getSprite() [1/2]	33
6.7.3.6.	getSprite() [2/2]	33
6.7.3.7.	setPosition()	33
6.7.3.8.	update()	34
6.7.4.	Adatmezők dokumentációja	34
6.7.4.1.	CELL_N	34
6.7.4.2.	cellSize	34
6.7.4.3.	currentCell	34
6.7.4.4.	currentSprite	34
6.7.4.5.	frameDurationSec	35
6.7.4.6.	isAnimated	35
6.7.4.7.	spriteSheet	35
6.7.4.8.	totalElapsedSec	35
6.8.	pftd::utils::parser::SaveFileParser::EntityInfo struktúrareferencia	35
6.8.1.	Részletes leírás	36
6.8.2.	Adatmezők dokumentációja	36
6.8.2.1.		36
6.8.2.2.	entityType	36

6.8.2.3.	hp	36
6.8.2.4.	position	36
6.8.2.5.	sealID	36
6.8.2.6.	towerID	36
6.9.	pftd::FollowPath struktúrareferencia	37
6.9.1.	Részletes leírás	37
6.9.2.	Konstruktorok és destruktorok dokumentációja	37
6.9.2.1.	FollowPath()	37
6.10.	pftd::FZC osztályreferencia	37
6.10.1.	Részletes leírás	38
6.10.2.	Konstruktorok és destruktorok dokumentációja	38
6.10.2.1.	FZC()	38
6.10.3.	Tagfüggvények dokumentációja	38
6.10.3.1.	clone()	38
6.11.	pftd::GameScene osztályreferencia	39
6.11.1.	Részletes leírás	40
6.11.2.	Konstruktorok és destruktorok dokumentációja	40
6.11.2.1.	GameScene()	40
6.11.2.2.	~GameScene()	40
6.11.3.	Tagfüggvények dokumentációja	40
6.11.3.1.	onEvent()	40
6.11.3.2.	startGame()	40
6.11.3.3.	toggleActive()	41
6.11.3.4.	update()	41
6.11.3.5.	updateScore()	41
6.11.3.6.	updateWealth()	41
6.11.4.	Adatmezők dokumentációja	42
6.11.4.1.	m_gameoverSound	42
6.11.4.2.	m_hornSound	42
6.11.4.3.	m_inventory	42
6.11.4.4.	m_level	42
6.11.4.5.	m_moneyCounter	42
6.11.4.6.	m_saveButt	42
6.11.4.7.	m_scoreCounter	43
6.12.	pftd::IcicleStabber osztályreferencia	43
6.12.1.	Részletes leírás	43
6.12.2.	Konstruktorok és destruktorok dokumentációja	43
6.12.2.1.	IcicleStabber() [1/2]	44
6.12.2.2.	IcicleStabber() [2/2]	44
6.12.2.3.	~IcicleStabber()	44
6.12.3.	Tagfüggvények dokumentációja	44
6.12.3.1.	clone()	44

6.12.3.2. update()	44
6.13. pftd::GameScene::Inventory struktúráreferencia	45
6.13.1. Részletes leírás	45
6.13.2. Konstruktork és destruktorok dokumentációja	45
6.13.2.1. Inventory()	45
6.13.2.2. ~Inventory()	46
6.13.3. Tagfüggvények dokumentációja	46
6.13.3.1. draw()	46
6.13.4. Adatmezők dokumentációja	46
6.13.4.1. background	46
6.14. pftd::GameScene::InventoryItem struktúráreferencia	47
6.14.1. Részletes leírás	47
6.14.2. Konstruktork és destruktorok dokumentációja	47
6.14.2.1. InventoryItem()	47
6.14.2.2. ~InventoryItem()	48
6.14.3. Tagfüggvények dokumentációja	48
6.14.3.1. draw()	48
6.14.4. Adatmezők dokumentációja	48
6.14.4.1. frame	48
6.14.4.2. icon	48
6.14.4.3. priceLabel	48
6.14.4.4. towerToSpawn	49
6.15. pftd::gr::Label osztályreferencia	49
6.15.1. Részletes leírás	49
6.15.2. Konstruktork és destruktorok dokumentációja	50
6.15.2.1. Label() [1/2]	50
6.15.2.2. Label() [2/2]	50
6.15.3. Tagfüggvények dokumentációja	50
6.15.3.1. draw()	50
6.15.3.2. getText() [1/2]	51
6.15.3.3. getText() [2/2]	51
6.15.3.4. setLabel()	51
6.15.3.5. setOutline()	51
6.15.4. Adatmezők dokumentációja	52
6.15.4.1. m_text	52
6.16. pftd::Level osztályreferencia	52
6.16.1. Részletes leírás	53
6.16.2. Konstruktork és destruktorok dokumentációja	53
6.16.2.1. Level()	53
6.16.2.2. ~Level()	54
6.16.3. Tagfüggvények dokumentációja	54
6.16.3.1. deselectTower()	54

6.16.3.2. draw()	54
6.16.3.3. isGameOver()	54
6.16.3.4. loseHP()	55
6.16.3.5. placeTower()	55
6.16.3.6. selectTower()	55
6.16.3.7. spawnSeal()	55
6.16.3.8. update()	56
6.16.4. Adatmezők dokumentációja	56
6.16.4.1. config	56
6.16.4.2. followPath	56
6.16.4.3. m_accuTimeSpawnSec	56
6.16.4.4. nest	56
6.16.4.5. projectiles	56
6.16.4.6. seals	57
6.16.4.7. selectedTower	57
6.16.4.8. stats	57
6.16.4.9. towers	57
6.17. pftd::utils::parser::LevelConfigParser osztályreferencia	57
6.17.1. Részletes leírás	58
6.17.2. Konstruktork és destruktorok dokumentációja	58
6.17.2.1. LevelConfigParser()	58
6.17.2.2. ~LevelConfigParser()	58
6.17.3. Tagfüggvények dokumentációja	58
6.17.3.1. _getAttribute()	58
6.17.3.2. getAttribute()	59
6.17.3.3. parse()	59
6.17.4. Adatmezők dokumentációja	59
6.17.4.1. m_attribs	59
6.17.4.2. m_lastAttribute	59
6.18. pftd::MenuScene osztályreferencia	59
6.18.1. Részletes leírás	60
6.18.2. Konstruktork és destruktorok dokumentációja	60
6.18.2.1. MenuScene()	60
6.18.2.2. ~MenuScene()	60
6.18.3. Tagfüggvények dokumentációja	60
6.18.3.1. onEvent()	60
6.18.3.2. update()	61
6.18.4. Adatmezők dokumentációja	61
6.18.4.1. m_buttons	61
6.19. pftd::Level::Nest struktúráreferencia	61
6.19.1. Részletes leírás	62
6.19.2. Konstruktork és destruktorok dokumentációja	62

6.19.2.1. Nest() [1/2]	62
6.19.2.2. Nest() [2/2]	62
6.19.2.3. ~Nest()	62
6.19.3. Adatmezők dokumentációja	63
6.19.3.1. radiusPixel	63
6.20. pftd::Object osztályreferencia	63
6.20.1. Részletes leírás	64
6.20.2. Konstruktork és destruktorok dokumentációja	64
6.20.2.1. Object() [1/4]	64
6.20.2.2. Object() [2/4]	64
6.20.2.3. Object() [3/4]	64
6.20.2.4. Object() [4/4]	64
6.20.2.5. ~Object()	64
6.20.3. Tagfüggvények dokumentációja	65
6.20.3.1. draw()	65
6.20.4. Adatmezők dokumentációja	65
6.20.4.1. position	65
6.20.4.2. size	65
6.20.4.3. zIndex	65
6.21. pftd::utils::parser::Parser osztályreferencia	66
6.21.1. Konstruktork és destruktorok dokumentációja	66
6.21.1.1. Parser()	66
6.21.1.2. ~Parser()	67
6.21.2. Tagfüggvények dokumentációja	67
6.21.2.1. _skipLine()	67
6.21.2.2. _skipWhitespace()	67
6.21.2.3. get()	67
6.21.2.4. isLabelValid()	67
6.21.2.5. parse()	68
6.21.2.6. peekAhead()	68
6.21.2.7. reset()	68
6.21.2.8. skip() [1/2]	68
6.21.2.9. skip() [2/2]	69
6.21.3. Adatmezők dokumentációja	69
6.21.3.1. commentDenoter	69
6.21.3.2. sourceStream	69
6.21.3.3. validLabel	69
6.22. pftd::Projectile osztályreferencia	70
6.22.1. Konstruktork és destruktorok dokumentációja	70
6.22.1.1. ~Projectile()	70
6.22.1.2. Projectile()	71
6.22.2. Tagfüggvények dokumentációja	71

6.22.2.1. update()	71
6.22.3. Adatmezők dokumentációja	71
6.22.3.1. angularVelocityRadPerSec	71
6.22.3.2. direction	72
6.22.3.3. linearSpeed	72
6.23. pftd::utils::Random osztályreferencia	72
6.23.1. Részletes leírás	72
6.23.2. Konstruktork és destruktorok dokumentációja	72
6.23.2.1. Random()	73
6.23.3. Tagfüggvények dokumentációja	73
6.23.3.1. generate()	73
6.23.4. Adatmezők dokumentációja	73
6.23.4.1. randomEngine	73
6.24. pftd::RegularSeal osztályreferencia	74
6.24.1. Részletes leírás	74
6.24.2. Konstruktork és destruktorok dokumentációja	74
6.24.2.1. RegularSeal()	74
6.24.3. Tagfüggvények dokumentációja	75
6.24.3.1. clone()	75
6.25. pftd::Renderer osztályreferencia	75
6.25.1. Részletes leírás	76
6.25.2. Konstruktork és destruktorok dokumentációja	76
6.25.2.1. Renderer() [1/3]	76
6.25.2.2. Renderer() [2/3]	76
6.25.2.3. Renderer() [3/3]	76
6.25.2.4. ~Renderer()	76
6.25.3. Tagfüggvények dokumentációja	77
6.25.3.1. clear()	77
6.25.3.2. display()	77
6.25.3.3. getWindow()	77
6.25.3.4. pushQueue()	77
6.25.3.5. render()	78
6.25.4. Barát és kapcsolódó függvények dokumentációja	78
6.25.4.1. App	78
6.25.5. Adatmezők dokumentációja	78
6.25.5.1. m_height	78
6.25.5.2. m_queue	78
6.25.5.3. m_width	78
6.25.5.4. m_window	78
6.26. pftd::ResourceManager osztályreferencia	79
6.26.1. Részletes leírás	79
6.26.2. Konstruktork és destruktorok dokumentációja	80

6.26.2.1. ResourceManager() [1/3]	80
6.26.2.2. ResourceManager() [2/3]	80
6.26.2.3. ~ResourceManager()	80
6.26.2.4. ResourceManager() [3/3]	80
6.26.3. Tagfüggvények dokumentációja	80
6.26.3.1. create()	80
6.26.3.2. destroy()	81
6.26.3.3. getDefaultFont()	81
6.26.3.4. getInstance()	81
6.26.3.5. getSound()	81
6.26.3.6. getTexture()	82
6.26.3.7. loadDefaultFont()	82
6.26.4. Adatmezők dokumentációja	82
6.26.4.1. m_defaultFont	82
6.26.4.2. m_instance	83
6.26.4.3. m_sounds	83
6.26.4.4. m_textures	83
6.27. pftd::utils::parser::SaveFileParser osztályreferencia	83
6.27.1. Részletes leírás	84
6.27.2. Enumeráció-tagok dokumentációja	84
6.27.2.1. EntityType	84
6.27.3. Konstruktorkok és destruktorkok dokumentációja	84
6.27.3.1. SaveFileParser()	84
6.27.3.2. ~SaveFileParser()	85
6.27.4. Tagfüggvények dokumentációja	85
6.27.4.1. getEntities()	85
6.27.4.2. getStats()	85
6.27.4.3. parse()	85
6.27.5. Adatmezők dokumentációja	85
6.27.5.1. m_entities	86
6.27.5.2. m_readStats	86
6.28. pftd::Scene osztályreferencia	86
6.28.1. Részletes leírás	87
6.28.2. Típusdefiníció-tagok dokumentációja	87
6.28.2.1. ObjPtrVec	87
6.28.3. Konstruktorkok és destruktorkok dokumentációja	87
6.28.3.1. Scene() [1/3]	87
6.28.3.2. Scene() [2/3]	87
6.28.3.3. Scene() [3/3]	87
6.28.3.4. ~Scene()	88
6.28.4. Tagfüggvények dokumentációja	88
6.28.4.1. getObjects() [1/2]	88

6.28.4.2. getObjects() [2/2]	88
6.28.4.3. onEvent()	88
6.28.4.4. setMusic()	89
6.28.4.5. toggleActive()	89
6.28.4.6. update()	89
6.28.5. Adatmezők dokumentációja	90
6.28.5.1. backgroundMusic	90
6.28.5.2. isActive	90
6.28.5.3. objects	90
6.29. pftd::SceneError struktúrareferencia	90
6.29.1. Részletes leírás	90
6.30. pftd::Seal osztályreferencia	91
6.30.1. Részletes leírás	92
6.30.2. Konstruktorok és destruktorok dokumentációja	92
6.30.2.1. ~Seal()	92
6.30.2.2. Seal()	92
6.30.3. Tagfüggvények dokumentációja	93
6.30.3.1. advanceAnimationFrame()	93
6.30.3.2. clone()	93
6.30.3.3. damage()	93
6.30.3.4. hasCompletedPath()	93
6.30.3.5. hasReachedNest()	94
6.30.3.6. lerpPath()	94
6.30.3.7. update()	94
6.30.4. Adatmezők dokumentációja	94
6.30.4.1. followPath	94
6.30.4.2. hp	95
6.30.4.3. isCurrentlyStealing	95
6.30.4.4. lerpParam	95
6.30.4.5. nextPoint	95
6.30.4.6. reachedNest	95
6.30.4.7. returned	95
6.30.4.8. speed	95
6.30.4.9. value	96
6.31. pftd::utils::Serializable osztályreferencia	96
6.31.1. Részletes leírás	96
6.31.2. Tagfüggvények dokumentációja	96
6.31.2.1. deserialize()	96
6.31.2.2. serialize()	97
6.32. pftd::Snowball osztályreferencia	97
6.32.1. Részletes leírás	98
6.32.2. Konstruktorok és destruktorok dokumentációja	98

6.32.2.1. Snowball()	98
6.32.3. Tagfüggvények dokumentációja	98
6.32.3.1. clone()	98
6.33. pftd::Snowballer osztályreferencia	99
6.33.1. Részletes leírás	99
6.33.2. Konstruktorkok és destruktorkok dokumentációja	99
6.33.2.1. Snowballer() [1/2]	99
6.33.2.2. Snowballer() [2/2]	100
6.33.2.3. ~Snowballer()	100
6.33.3. Tagfüggvények dokumentációja	100
6.33.3.1. attack()	100
6.33.3.2. clone()	100
6.33.3.3. update()	100
6.34. pftd::gr::Sprite osztályreferencia	101
6.34.1. Részletes leírás	102
6.34.2. Konstruktorkok és destruktorkok dokumentációja	102
6.34.2.1. Sprite() [1/5]	102
6.34.2.2. Sprite() [2/5]	102
6.34.2.3. Sprite() [3/5]	102
6.34.2.4. Sprite() [4/5]	102
6.34.2.5. Sprite() [5/5]	103
6.34.2.6. ~Sprite()	103
6.34.3. Tagfüggvények dokumentációja	103
6.34.3.1. draw()	103
6.34.3.2. flipY()	103
6.34.3.3. modColor()	103
6.34.3.4. setSpriteRect()	104
6.34.4. Adatmezők dokumentációja	104
6.34.4.1. m_sprite	104
6.34.4.2. m_texture	104
6.35. pftd::Level::Stats struktúrareferencia	104
6.35.1. Részletes leírás	105
6.35.2. Konstruktorkok és destruktorkok dokumentációja	105
6.35.2.1. Stats() [1/2]	105
6.35.2.2. Stats() [2/2]	105
6.35.3. Tagfüggvények dokumentációja	105
6.35.3.1. deserialize()	105
6.35.3.2. serialize()	106
6.35.4. Adatmezők dokumentációja	106
6.35.4.1. hp	106
6.35.4.2. MAX_HP	106
6.35.4.3. money	106

6.35.4.4. score	107
6.36. pftd::utils::parser::SaveFileParser::StatsInfo struktúráreferencia	107
6.36.1. Részletes leírás	107
6.36.2. Adatmezők dokumentációja	107
6.36.2.1. hp	107
6.36.2.2. score	107
6.36.2.3. wealth	108
6.37. pftd::Tower osztályreferencia	108
6.37.1. Részletes leírás	109
6.37.2. Típusdefiníció-tagok dokumentációja	109
6.37.2.1. ProjSpawnFunc	109
6.37.3. Konstruktorkok és destruktorkok dokumentációja	109
6.37.3.1. Tower() [1/3]	110
6.37.3.2. Tower() [2/3]	110
6.37.3.3. Tower() [3/3]	110
6.37.3.4. ~Tower()	110
6.37.4. Tagfüggvények dokumentációja	110
6.37.4.1. advanceAnimationFrame()	110
6.37.4.2. attack()	111
6.37.4.3. clone()	111
6.37.4.4. lookForTarget()	111
6.37.4.5. setProjSpawnCb()	111
6.37.4.6. update()	112
6.37.5. Adatmezők dokumentációja	112
6.37.5.1. attackDamage	112
6.37.5.2. attackRangePixel	112
6.37.5.3. attackSpeedSec	112
6.37.5.4. attackTimerSec	113
6.37.5.5. instantAttack	113
6.37.5.6. price	113
6.37.5.7. radiusPixel	113
6.37.5.8. spawnProjectile	113
6.37.5.9. target	113
6.38. pftd::utils::Vec2< T > struktúrasablon-referencia	113
6.38.1. Részletes leírás	114
6.38.2. Konstruktorkok és destruktorkok dokumentációja	114
6.38.2.1. Vec2() [1/3]	114
6.38.2.2. Vec2() [2/3]	115
6.38.2.3. Vec2() [3/3]	115
6.38.3. Tagfüggvények dokumentációja	115
6.38.3.1. distance()	115
6.38.3.2. normalize()	115

6.38.3.3. operator!=()	116
6.38.3.4. operator*()	116
6.38.3.5. operator+()	116
6.38.3.6. operator-()	116
6.38.3.7. operator/()	116
6.38.3.8. operator=()	116
6.38.3.9. operator==()	117
6.38.4. Barát és kapcsolódó függvények dokumentációja	117
6.38.4.1. operator<<	117
6.38.5. Adatmezők dokumentációja	117
6.38.5.1. x	117
6.38.5.2. y	117
6.39. pftd::ZombieSeal osztályreferencia	118
6.39.1. Részletes leírás	118
6.39.2. Konstruktorkok és destruktorkok dokumentációja	118
6.39.2.1. ZombieSeal()	118
6.39.3. Tagfüggvények dokumentációja	119
6.39.3.1. clone()	119
7. Fájlok dokumentációja	121
7.1. include/all.hpp fájlreferencia	121
7.2. all.hpp	121
7.3. include/app.hpp fájlreferencia	122
7.4. app.hpp	122
7.5. include/game/level.hpp fájlreferencia	123
7.6. level.hpp	124
7.7. include/objects/clickable.hpp fájlreferencia	125
7.8. clickable.hpp	125
7.9. include/objects/entities/entity_base.hpp fájlreferencia	126
7.10. entity_base.hpp	126
7.11. include/objects/entities/projectiles/projectile_base.hpp fájlreferencia	127
7.12. projectile_base.hpp	127
7.13. include/objects/entities/projectiles/snowball.hpp fájlreferencia	127
7.14. snowball.hpp	128
7.15. include/objects/entities/seals/cub.hpp fájlreferencia	128
7.16. cub.hpp	128
7.17. include/objects/entities/seals/followpath.hpp fájlreferencia	129
7.18. followpath.hpp	129
7.19. include/objects/entities/seals/fortified_zombie_cub.hpp fájlreferencia	129
7.20. fortified_zombie_cub.hpp	130
7.21. include/objects/entities/seals/regular.hpp fájlreferencia	130
7.22. regular.hpp	130

7.23. include/objects/entities/seals/seal_base.hpp fájlreferencia	130
7.24. seal_base.hpp	131
7.25. include/objects/entities/seals/zombie.hpp fájlreferencia	131
7.26. zombie.hpp	132
7.27. include/objects/entities/towers/iclestabber.hpp fájlreferencia	132
7.28. iclestabber.hpp	132
7.29. include/objects/entities/towers/snowballer.hpp fájlreferencia	133
7.30. snowballer.hpp	133
7.31. include/objects/entities/towers/tower_base.hpp fájlreferencia	133
7.32. tower_base.hpp	134
7.33. include/objects/gui/button.hpp fájlreferencia	134
7.34. button.hpp	135
7.35. include/objects/gui/image.hpp fájlreferencia	135
7.36. image.hpp	136
7.37. include/objects/gui/label.hpp fájlreferencia	136
7.38. label.hpp	137
7.39. include/objects/object_base.hpp fájlreferencia	137
7.40. object_base.hpp	138
7.41. include/resources.hpp fájlreferencia	138
7.42. resources.hpp	139
7.43. include/scene.hpp fájlreferencia	139
7.44. scene.hpp	140
7.45. include/scenes/game.hpp fájlreferencia	140
7.46. game.hpp	141
7.47. include/scenes/menu.hpp fájlreferencia	142
7.48. menu.hpp	142
7.49. include/utls/hetero_collection.hpp fájlreferencia	142
7.50. hetero_collection.hpp	143
7.51. include/utls/logger.hpp fájlreferencia	143
7.51.1. Makródefiníciók dokumentációja	143
7.51.1.1. print	143
7.51.1.2. where	144
7.52. logger.hpp	144
7.53. include/utls/parsers.hpp fájlreferencia	144
7.54. parsers.hpp	145
7.55. include/utls/random_gen.hpp fájlreferencia	146
7.56. random_gen.hpp	147
7.57. include/utls/serializable.hpp fájlreferencia	147
7.58. serializable.hpp	147
7.59. include/utls/substitute_types.hpp fájlreferencia	148
7.60. substitute_types.hpp	148

Tárgymutató

151

1. fejezet

Névtérmutató

1.1. Névtérlista

Az összes névtér listája rövid leírásokkal:

<code>pftd</code>	9
<code>pftd::gr</code>	10
<code>pftd::utils</code>	10
<code>pftd::utils::parser</code>	11

2. fejezet

Hierarchikus mutató

2.1. Osztályhierarchia

Majdnem (de nem teljesen) betűrendbe szedett leszármazási lista:

pftd::App	13
pftd::Object::Compare	25
pftd::utils::Container< T, C >	26
pftd::utils::Container< EndPoint >	26
pftd::FollowPath	37
pftd::utils::Container< InventoryItem >	26
pftd::GameScene::Inventory	45
sf::Drawable	
pftd::Object	63
pftd::Clickable	22
pftd::GameScene::InventoryItem	47
pftd::gr::Button	18
pftd::Entity	30
pftd::Level::Nest	61
pftd::Projectile	70
pftd::Snowball	97
pftd::Seal	91
pftd::Cub	28
pftd::FZC	37
pftd::RegularSeal	74
pftd::ZombieSeal	118
pftd::Tower	108
pftd::IcicleStabber	43
pftd::Snowballer	99
pftd::GameScene::Inventory	45
pftd::Level	52
pftd::gr::Label	49
pftd::gr::Sprite	101
pftd::utils::parser::SaveFileParser::EntityInfo	35
pftd::utils::parser::Parser	66
pftd::utils::parser::LevelConfigParser	57
pftd::utils::parser::SaveFileParser	83
pftd::utils::Random	72
pftd::Renderer	75

pftd::ResourceManager	79
std::runtime_error	
pftd::SceneError	90
pftd::Scene	86
pftd::GameScene	39
pftd::MenuScene	59
pftd::utils::Serializable	96
pftd::Entity	30
pftd::Level::Stats	104
pftd::utils::parser::SaveFileParser::StatsInfo	107
pftd::utils::Vec2< T >	113
pftd::utils::Vec2< float >	113

3. fejezet

Adatszerkezet-mutató

3.1. Adatszerkezetek

Az összes adatszerkezet listája rövid leírásokkal:

pftd::App	13
pftd::gr::Button	18
pftd::Clickable	22
pftd::Object::Compare	25
pftd::utils::Container< T, C >	26
pftd::Cub	28
pftd::Entity	30
pftd::utils::parser::SaveFileParser::EntityInfo	35
pftd::FollowPath	37
pftd::FZC	37
pftd::GameScene	39
pftd::IcicleStabber	43
pftd::GameScene::Inventory	45
pftd::GameScene::InventoryItem	47
pftd::gr::Label	49
pftd::Level	52
pftd::utils::parser::LevelConfigParser	57
pftd::MenuScene	59
pftd::Level::Nest	61
pftd::Object	63
pftd::utils::parser::Parser	66
pftd::Projectile	70
pftd::utils::Random	72
pftd::RegularSeal	74
pftd::Renderer	75
pftd::ResourceManager	79
pftd::utils::parser::SaveFileParser	83
pftd::Scene	86
pftd::SceneError	90
pftd::Seal	91
pftd::utils::Serializable	96
pftd::Snowball	97
pftd::Snowballer	99
pftd::gr::Sprite	101
pftd::Level::Stats	104

pftd::utils::parser::SaveFileParser::StatsInfo	107
pftd::Tower	108
pftd::utils::Vec2< T >	113
pftd::ZombieSeal	118

4. fejezet

Fájlmutató

4.1. Fájllista

Az összes fájl listája rövid leírásokkal:

include/all.hpp	121
include/app.hpp	122
include/resources.hpp	138
include/scene.hpp	139
include/game/level.hpp	123
include/objects/clickable.hpp	125
include/objects/object_base.hpp	137
include/objects/entities/entity_base.hpp	126
include/objects/entities/projectiles/projectile_base.hpp	127
include/objects/entities/projectiles/snowball.hpp	127
include/objects/entities/seals/cub.hpp	128
include/objects/entities/seals/followpath.hpp	129
include/objects/entities/seals/fortified_zombie_cub.hpp	129
include/objects/entities/seals/regular.hpp	130
include/objects/entities/seals/seal_base.hpp	130
include/objects/entities/seals/zombie.hpp	131
include/objects/entities/towers/iciclestabber.hpp	132
include/objects/entities/towers/snowballer.hpp	133
include/objects/entities/towers/tower_base.hpp	133
include/objects/gui/button.hpp	134
include/objects/gui/image.hpp	135
include/objects/gui/label.hpp	136
include/scenes/game.hpp	140
include/scenes/menu.hpp	142
include/utils/hetero_collection.hpp	142
include/utils/logger.hpp	143
include/utils/parsers.hpp	144
include/utils/random_gen.hpp	146
include/utils/serializable.hpp	147
include/utils/substitute_types.hpp	148

5. fejezet

Névterek dokumentációja

5.1. pftd névtér-referencia

Névterek

- namespace [gr](#)
- namespace [utils](#)

Adatszerkezetek

- class [App](#)
- class [Clickable](#)
- class [Cub](#)
- class [Entity](#)
- struct [FollowPath](#)
- class [FZC](#)
- class [GameScene](#)
- class [IcicleStabber](#)
- class [Level](#)
- class [MenuScene](#)
- class [Object](#)
- class [Projectile](#)
- class [RegularSeal](#)
- class [Renderer](#)
- class [ResourceManager](#)
- class [Scene](#)
- struct [SceneError](#)
- class [Seal](#)
- class [Snowball](#)
- class [Snowballer](#)
- class [Tower](#)
- class [ZombieSeal](#)

Típusdefiníciók

- using [EndPoint](#) = [utils::Vec2f](#)

5.1.1. Típusdefiníciók dokumentációja

5.1.1.1. EndPoint

```
using pftd::EndPoint = typedef utils::Vec2f
```

5.2. pftd::gr névtér-referencia

Adatszerkezetek

- class [Button](#)
- class [Label](#)
- class [Sprite](#)

5.3. pftd::utils névtér-referencia

Névterek

- namespace [parser](#)

Adatszerkezetek

- class [Container](#)
- class [Random](#)
- class [Serializable](#)
- struct [Vec2](#)

Típusdefiníciók

- using [Vec2i](#) = [Vec2](#)< int >
- using [Vec2f](#) = [Vec2](#)< float >

5.3.1. Típusdefiníciók dokumentációja

5.3.1.1. Vec2f

```
using pftd::utils::Vec2f = typedef Vec2<float>
```

5.3.1.2. Vec2i

```
using pftd::utils::Vec2i = typedef Vec2<int>
```

5.4. pftd::utils::parser névtér-referencia

Adatszerkezetek

- class [LevelConfigParser](#)
- class [Parser](#)
- class [SaveFileParser](#)

6. fejezet

Adatszerkezetek dokumentációja

6.1. pftd::App osztályreferencia

```
#include <app.hpp>
```

Publikus tagfüggvények

- `App (App const &)=delete`
- `App (App &&)=delete`
- `~App ()`
Felszabadítja a kezelt erőforrásokat.
- `void run ()`
A fő programciklus.
- `void addScene (std::string id, Scene *scene, bool active=false)`
Hozzáad egy új nézetet.
- `bool changeScene (std::string id)`
Nézetet vált (aktívvá tesz egy másikat).
- `auto isRunning () const`
m_running getter.
- `int getWindowWidth () const`
- `int getWindowHeight () const`
- `App & operator= (App const &)=delete`

Statikus publikus tagfüggvények

- `static App * getInstance ()`
A singleton instance megszerzése.
- `static App * create (unsigned int width, unsigned int height, std::string const &windowTitle)`
Inicializálja a különböző programelemeket.
- `static void destroy ()`
Törli a singleton instance-t.

Privát tagfüggvények

- `App ()=default`

Privát attribútumok

- bool `m_running` = false
- `Renderer * m_renderer` = nullptr
- `std::map< std::string, Scene * > m_scenes`
- `std::string m_activeSceneID`

Statikus privát attribútumok

- static `App * m_instance`

6.1.1. Részletes leírás

Az egész játékért felelő osztály.

6.1.2. Konstruktorkok és destruktorkok dokumentációja

6.1.2.1. `App()` [1/3]

```
pftd::App::App (  
    App const & ) [delete]
```

6.1.2.2. `App()` [2/3]

```
pftd::App::App (  
    App && ) [delete]
```

6.1.2.3. `~App()`

```
pftd::App::~~App ( )
```

Felszabadítja a kezelt erőforrásokat.

Törli a renderer-t, az erőforrás-kezelőt és a nézeteket.

6.1.2.4. `App()` [3/3]

```
pftd::App::App ( ) [private], [default]
```


6.1.3. Tagfüggvények dokumentációja

6.1.3.1. addScene()

```
void pftd::App::addScene (
    std::string id,
    Scene * scene,
    bool active = false )
```

Hozzáad egy új nézetet.

Ha az `active` igaz, akkor meghívja a `changeScene`-t a megadott `id`-ra. Megjegyzés: Ha már létezik ilyen nézet a megadott azonosítóval, akkor `SceneError`-t dob.

Paraméterek

<i>id</i>	A nézet azonosítója.
<i>scene</i>	Maga a nézet osztály egy példánya (pl.: MenuScene , GameScene).
<i>active</i>	Ez legyen e az aktív nézet.

6.1.3.2. changeScene()

```
bool pftd::App::changeScene (
    std::string id )
```

Nézetet vált (aktívvá tesz egy másikat).

Ha megegyezik az eddigi és a kért nézet, akkor nem csinál semmit. Egyébként meg az eddigi nézetet deaktiválja, és az újat pedig aktiválja. Megjegyzés: Ha már létezik ilyen nézet a megadott azonosítóval, akkor `SceneError`-t dob.

Paraméterek

<i>id</i>	Az új nézet azonosítója.
-----------	--------------------------

Visszatérési érték

Sikerült e váltani.

6.1.3.3. create()

```
static App * pftd::App::create (
    unsigned int width,
```

```
unsigned int height,  
std::string const & windowTitle ) [static]
```

Inicializálja a különböző programelemeket.

Megjegyzés: Meg kell hívni a `destroy()`-t, ha már nincs szükség rá!

Paraméterek

<i>width</i>	Az ablak szélessége.
<i>height</i>	Az ablak magassága.
<i>windowTitle</i>	Az ablak címe.

Visszatérési érték

Az osztályhoz tartozó singleton instance.

6.1.3.4. `destroy()`

```
static void pftd::App::destroy ( ) [inline], [static]
```

Törli a singleton instance-t.

6.1.3.5. `getInstance()`

```
static App * pftd::App::getInstance ( ) [inline], [static]
```

A singleton instance megszerzése.

Visszatérési érték

Az osztályhoz tartozó singleton instance.

6.1.3.6. `getWindowHeight()`

```
int pftd::App::getWindowHeight ( ) const [inline]
```

Visszatérési érték

A programhoz tartozó ablak magassága.

6.1.3.7. getWidth()

```
int pftd::App::getWidth ( ) const [inline]
```

Visszatérési érték

A programhoz tartozó ablak szélessége.

6.1.3.8. isRunning()

```
auto pftd::App::isRunning ( ) const [inline]
```

m_running getter.

Visszatérési érték

Fut e a program.

6.1.3.9. operator=()

```
App & pftd::App::operator= (
    App const & ) [delete]
```

6.1.3.10. run()

```
void pftd::App::run ( )
```

A fő programciklus.

6.1.4. Adatmezők dokumentációja

6.1.4.1. m_activeSceneID

```
std::string pftd::App::m_activeSceneID [private]
```

Az aktív nézet azonosítója.

6.1.4.2. m_instance

```
App* pftd::App::m_instance [static], [private]
```

A singleton instance.

6.1.4.3. m_renderer

```
Renderer* pftd::App::m_renderer = nullptr [private]
```

A renderer.

6.1.4.4. m_running

```
bool pftd::App::m_running = false [private]
```

Fut e a program.

6.1.4.5. m_scenes

```
std::map<std::string, Scene*> pftd::App::m_scenes [private]
```

A nézetek: [id, nézet].

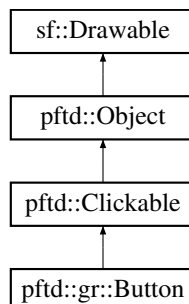
Ez a dokumentáció az osztályról a következő fájl alapján készült:

- include/app.hpp

6.2. pftd::gr::Button osztályreferencia

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

A pftd::gr::Button osztály származási diagramja:



Publikus tagfüggvények

- `Button` (`sf::Text` &&`label`, `utils::Vec2f` const &`position`, `utils::Vec2f` const &`size`, bool `active=true`, int `zIndex=0`)
- virtual `~Button` ()
- void `setSound` (`std::string` const &`src`)
Kattintás hangeffekt beállítása.
- void `setBackground` (`std::string` const &`src`)
Háttérkép beállítása.
- `std::string` `getLabel` () const
Ráírt szöveg.
- virtual void `handleClick` (`utils::Vec2i` const &`clickCoords`) override
Kattintás kezelése.
- virtual void `draw` (`sf::RenderTarget` &`target`, `sf::RenderStates` `states`) const override
Objektum megjelenítése.

Védett attribútumok

- `Label` `label`

Privát attribútumok

- `sf::FloatRect` `m_rect`
- `Sprite` * `m_background` = nullptr
- `sf::Sound` `m_clickSound`

További örökölt tagok

6.2.1. Részletes leírás

Gomb GUI elem.

6.2.2. Konstruktorkok és destruktorkok dokumentációja

6.2.2.1. Button()

```
pftd::gr::Button::Button (
    sf::Text && label,
    utils::Vec2f const & position,
    utils::Vec2f const & size,
    bool active = true,
    int zIndex = 0 )
```

Paraméterek

<code>label</code>	Szöveg objektum: maga a szöveg, betűtípus és betűméret.
<code>position</code>	Pozíció.
<code>size</code>	Méret.
<code>active</code>	Aktív e? Rá lehet e kattintani.
<code>zIndex</code>	Z koordináta.

6.2.2.2. ~Button()

```
virtual pftd::gr::Button::~~Button ( ) [virtual]
```

6.2.3. Tagfüggvények dokumentációja

6.2.3.1. draw()

```
virtual void pftd::gr::Button::draw (
    sf::RenderTarget & target,
    sf::RenderStates states ) const [override], [virtual]
```

Objektum megjelenítése.

Paraméterek

<i>target</i>	Hol legyen megjelenítve? Ez általában egy ablak.
<i>states</i>	Render-eléshez szükséges egyéb állapotok.

Megvalósítja a következőket: [pftd::Object](#).

6.2.3.2. getLabel()

```
std::string pftd::gr::Button::getLabel ( ) const [inline]
```

Ráírt szöveg.

Visszatérési érték

A tárolt címkére írt szöveg.

6.2.3.3. handleClick()

```
virtual void pftd::gr::Button::handleClick (
    utils::Vec2i const & clickCoords ) [override], [virtual]
```

Kattintás kezelése.

Paraméterek

<i>clickCoords</i>	Kurzor koordinátái kattintáskor.
--------------------	----------------------------------

Újraimplementált ősök: [pftd::Clickable](#).

6.2.3.4. setBackground()

```
void pftd::gr::Button::setBackground (
    std::string const & src )
```

Háttérkép beállítása.

Paraméterek

<i>src</i>	Háttérkép elérési útvonala.
------------	-----------------------------

6.2.3.5. setSound()

```
void pftd::gr::Button::setSound (
    std::string const & src )
```

Kattintás hangeffekt beállítása.

Paraméterek

<i>src</i>	Hangfájl elérési útvonala.
------------	----------------------------

6.2.4. Adatmezők dokumentációja

6.2.4.1. label

```
Label pftd::gr::Button::label [protected]
```

Címke.

6.2.4.2. m_background

```
Sprite* pftd::gr::Button::m_background = nullptr [private]
```

Háttér.

6.2.4.3. m_clickSound

```
sf::Sound pftd::gr::Button::m_clickSound [private]
```

Kattintás hangeffekt.

6.2.4.4. m_rect

```
sf::FloatRect pftd::gr::Button::m_rect [private]
```

Minden gomb egy téglalap: ennek a pozíciója és mérete.

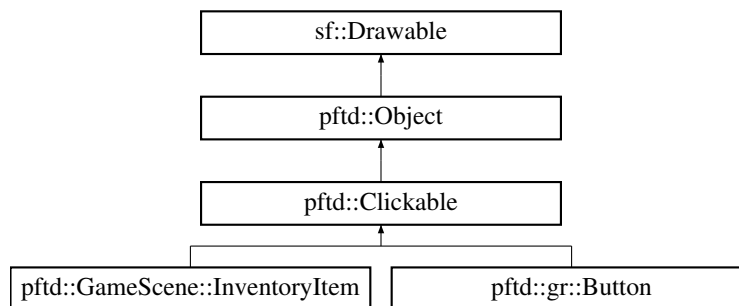
Ez a dokumentáció az osztályról a következő fájl alapján készült:

- include/objects/gui/[button.hpp](#)

6.3. pftd::Clickable osztályreferencia

```
#include <clickable.hpp>
```

A pftd::Clickable osztály származási diagramja:



Publikus tagfüggvények

- [Clickable](#) (bool active=true)
- [Clickable](#) (utils::Vec2f const &position, utils::Vec2f const &size, int zIndex=0, bool active=true)
- virtual void [setCallback](#) (std::function< void()> callback)
Callback beállítása.
- virtual void [handleClick](#) (utils::Vec2i const &clickCoords)
Kattintás kezelése.

Adatmezők

- bool [isActive](#)

Védett attribútumok

- std::function< void()> [m_callback](#)

6.3.1. Részletes leírás

Kattintható (interaktálható) objektum.

6.3.2. Konstruktorkok és destruktorkok dokumentációja

6.3.2.1. Clickable() [1/2]

```
pftd::Clickable::Clickable (
    bool active = true )
```

6.3.2.2. Clickable() [2/2]

```
pftd::Clickable::Clickable (
    utils::Vec2f const & position,
    utils::Vec2f const & size,
    int zIndex = 0,
    bool active = true )
```

6.3.3. Tagfüggvények dokumentációja

6.3.3.1. handleClick()

```
virtual void pftd::Clickable::handleClick (
    utils::Vec2i const & clickCoords ) [virtual]
```

Kattintás kezelése.

Paraméterek

<i>clickCoords</i>	Kurzor koordinátái kattintáskor.
--------------------	----------------------------------

Újraimplementáló leszármazottak: [pftd::gr::Button](#).

6.3.3.2. setCallback()

```
virtual void pftd::Clickable::setCallback (
    std::function< void()> callback ) [virtual]
```

Callback beállítása.

Paraméterek

<i>callback</i>	A kattintásra lefuttatni kívánt függvény.
-----------------	---

6.3.4. Adatmezők dokumentációja

6.3.4.1. isActive

```
bool pftd::Clickable::isActive
```

Aktív e.

6.3.4.2. m_callback

```
std::function<void()> pftd::Clickable::m_callback [protected]
```

Callback: akkor fut le, ha az objektumra rákattintunk.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- include/objects/[clickable.hpp](#)

6.4. pftd::Object::Compare struktúrareferencia

```
#include <object_base.hpp>
```

Publikus tagfüggvények

- bool [operator\(\)](#) ([Object](#) const &o1, [Object](#) const &o2) const
- bool [operator\(\)](#) ([Object](#) const *o1, [Object](#) const *o2) const

6.4.1. Részletes leírás

Priority queue miatt szükséges segédosztály.

6.4.2. Tagfüggvények dokumentációja

6.4.2.1. operator>() [1/2]

```
bool pftd::Object::Compare::operator() (
    Object const & o1,
    Object const & o2 ) const [inline]
```

6.4.2.2. operator>() [2/2]

```
bool pftd::Object::Compare::operator() (
    Object const * o1,
    Object const * o2 ) const [inline]
```

Ez a dokumentáció a struktúráról a következő fájl alapján készült:

- include/objects/object_base.hpp

6.5. pftd::utils::Container< T, C > osztálysablon-referencia

```
#include <hetero_collection.hpp>
```

Publikus tagfüggvények

- [Container](#) ()=default
- virtual [~Container](#) ()
- virtual T *const [append](#) (T *elem)
Új elem hozzáadása.
- std::size_t [size](#) () const
A tároló mérete.
- C const & [getContainer](#) () const
Konstans m_container getter.

Privát attribútumok

- C [m_container](#)

6.5.1. Részletes leírás

```
template<typename T, typename C = std::vector<T*>>
class pftd::utils::Container< T, C >
```

Olyan osztályok, amelyek tárolóként (is) használhatók. Ilyen például: [FollowPath](#), [Inventory](#).

6.5.2. Konstruktorkok és destruktorkok dokumentációja

6.5.2.1. Container()

```
template<typename T , typename C = std::vector<T*>>
pftd::utils::Container< T, C >::Container ( ) [default]
```

6.5.2.2. ~Container()

```
template<typename T , typename C = std::vector<T*>>
virtual pftd::utils::Container< T, C >::~~Container ( ) [inline], [virtual]
```

6.5.3. Tagfüggvények dokumentációja

6.5.3.1. append()

```
template<typename T , typename C = std::vector<T*>>
virtual T *const pftd::utils::Container< T, C >::append (
    T * elem ) [inline], [virtual]
```

Új elem hozzáadása.

Paraméterek

<i>elem</i>	A hozzáadni kívánt dolog.
-------------	---------------------------

Visszatérési érték

Az imént hozzáadott dolog.

6.5.3.2. getContainer()

```
template<typename T , typename C = std::vector<T*>>
C const & pftd::utils::Container< T, C >::getContainer ( ) const [inline]
```

Konstans `m_container` getter.

Visszatérési érték

A tároló.

6.5.3.3. size()

```
template<typename T , typename C = std::vector<T*>>
std::size_t pftd::utils::Container< T, C >::size ( ) const [inline]
```

A tároló mérete.

Visszatérési érték

A tároló elemeinek száma.

6.5.4. Adatmezők dokumentációja

6.5.4.1. m_container

```
template<typename T , typename C = std::vector<T*>>
C pftd::utils::Container< T, C >::m_container [private]
```

A tároló.

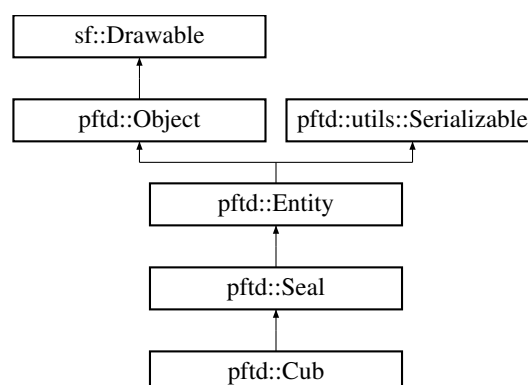
Ez a dokumentáció az osztályról a következő fájl alapján készült:

- [include/utills/hetero_collection.hpp](#)

6.6. pftd::Cub osztályreferencia

```
#include <cub.hpp>
```

A pftd::Cub osztály származási diagramja:



Publikus tagfüggvények

- [Cub](#) ([FollowPath](#) const &followpath)
A textúra hard code-olt, a kezdő pozíciót pedig a `followPath` határozza meg.
- [Seal](#) * [clone](#) () const override
Egy dinamikus memóriában foglalt másolatot készít a fókából.

További örökölt tagok

6.6.1. Részletes leírás

Kicsiny.

6.6.2. Konstruktorok és destruktorok dokumentációja

6.6.2.1. Cub()

```
pftd::Cub::Cub (
    FollowPath const & followpath )
```

A textúra hard code-olt, a kezdő pozíciót pedig a `followPath` határozza meg.

Paraméterek

<code>followpath</code>	Az útvonal amin végig kell menjen (majd pedig vissza).
-------------------------	--

6.6.3. Tagfüggvények dokumentációja

6.6.3.1. clone()

```
Seal * pftd::Cub::clone ( ) const [override], [virtual]
```

Egy dinamikus memóriában foglalt másolatot készít a főkából.

Visszatérési érték

A másolat.

Megvalósítja a következőket: [pftd::Seal](#).

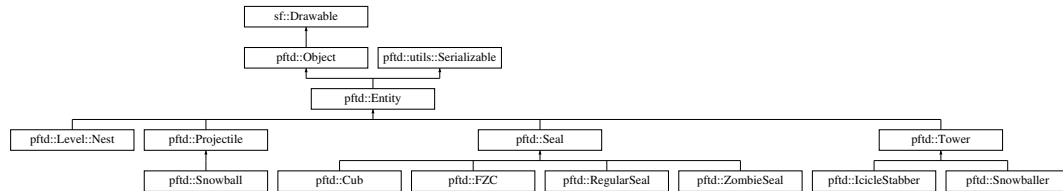
Ez a dokumentáció az osztályról a következő fájl alapján készült:

- [include/objects/entities/seals/cub.hpp](#)

6.7. pftd::Entity osztályreferencia

```
#include <entity_base.hpp>
```

A pftd::Entity osztály származási diagramja:



Publikus tagfüggvények

- **Entity** (std::string const &spriteSheetSrc, **utils::Vec2i** spriteSize, **utils::Vec2f** const &position, **utils::Vec2f** const &size, int zIndex=0)
- **Entity** (sf::Texture const &texture, **utils::Vec2i** spriteSize, **utils::Vec2f** const &position, **utils::Vec2f** const &size, int zIndex=0)
- **Entity** (std::string const &spriteSrc, **utils::Vec2f** const &position, **utils::Vec2f** const &size, int zIndex=0)
- **Entity** (**Entity** const &other)
- virtual ~**Entity** ()
- virtual **Entity** * **clone** () const =0

Egy dinamikus memóriában foglalt másolatot készít az entitásról.

- virtual void **update** (float dt)

Update.

- virtual void **draw** (sf::RenderTarget &target, sf::RenderStates states) const override

Objektum megjelenítése.

- void **setPosition** (**utils::Vec2f** position)

Pozíció beállítása.

- auto **getPosition** () const

position getter.

- **gr::Sprite** const * **getSprite** () const

Konstans currentSprite getter.

- **gr::Sprite** * **getSprite** ()

currentSprite getter.

Adatmezők

- bool **isAnimated** = true

Védett tagfüggvények

- virtual void **advanceAnimationFrame** ()

Animáció: képkocka léptetése.

Védett attribútumok

- sf::Texture const & [spriteSheet](#)
- [utils::Vec2i](#) [cellSize](#)
- size_t const [CELL_N](#)
- size_t [currentCell](#) = 0U
- [gr::Sprite](#) * [currentSprite](#) = nullptr
- float [frameDurationSec](#) = 1.0f
- float [totalElapsedSec](#) = 0.0f

6.7.1. Részletes leírás

Mindenféle "actor" őssztálya: tornyok, fólák és lövedékek.

6.7.2. Konstruktorkok és destruktorkok dokumentációja

6.7.2.1. Entity() [1/4]

```
pftd::Entity::Entity (
    std::string const & spriteSheetSrc,
    utils::Vec2i spriteSize,
    utils::Vec2f const & position,
    utils::Vec2f const & size,
    int zIndex = 0 )
```

6.7.2.2. Entity() [2/4]

```
pftd::Entity::Entity (
    sf::Texture const & texture,
    utils::Vec2i spriteSize,
    utils::Vec2f const & position,
    utils::Vec2f const & size,
    int zIndex = 0 )
```

6.7.2.3. Entity() [3/4]

```
pftd::Entity::Entity (
    std::string const & spriteSrc,
    utils::Vec2f const & position,
    utils::Vec2f const & size,
    int zIndex = 0 )
```

6.7.2.4. Entity() [4/4]

```
pftd::Entity::Entity (
    Entity const & other )
```

6.7.2.5. ~Entity()

```
virtual pftd::Entity::~~Entity ( ) [virtual]
```

6.7.3. Tagfüggvények dokumentációja

6.7.3.1. advanceAnimationFrame()

```
virtual void pftd::Entity::advanceAnimationFrame ( ) [protected], [virtual]
```

Animáció: képkocka léptetése.

Újraimplementáló leszármazottak: [pftd::Seal](#) és [pftd::Tower](#).

6.7.3.2. clone()

```
virtual Entity * pftd::Entity::clone ( ) const [pure virtual]
```

Egy dinamikus memóriában foglalt másolatot készít az entitásról.

Visszatérési érték

A másolat.

Megvalósítják a következők: [pftd::Snowball](#), [pftd::Cub](#), [pftd::FZC](#), [pftd::RegularSeal](#), [pftd::ZombieSeal](#), [pftd::IcicleStabber](#), [pftd::Snowballer](#), [pftd::Seal](#) és [pftd::Tower](#).

6.7.3.3. draw()

```
virtual void pftd::Entity::draw (
    sf::RenderTarget & target,
    sf::RenderStates states ) const [override], [virtual]
```

Objektum megjelenítése.

Paraméterek

<i>target</i>	Hol legyen megjelenítve? Ez általában egy ablak.
<i>states</i>	Render-eléshez szükséges egyéb állapotok.

Megvalósítja a következőket: [pftd::Object](#).

6.7.3.4. getPosition()

```
auto pftd::Entity::getPosition ( ) const [inline]
position getter.
```

Visszatérési érték

Az entitás jelenlegi pozíciója.

6.7.3.5. getSprite() [1/2]

```
gr::Sprite * pftd::Entity::getSprite ( ) [inline]
currentSprite getter.
```

Visszatérési érték

A jelenlegi sprite.

6.7.3.6. getSprite() [2/2]

```
gr::Sprite const * pftd::Entity::getSprite ( ) const [inline]
Konstans currentSprite getter.
```

Visszatérési érték

A jelenlegi sprite.

6.7.3.7. setPosition()

```
void pftd::Entity::setPosition (
    utils::Vec2f position )
```

Pozíció beállítása.

Paraméterek

<i>position</i>	Az új pozíció.
-----------------	----------------

6.7.3.8. update()

```
virtual void pftd::Entity::update (  
    float dt ) [virtual]
```

Update.

Paraméterek

<i>dt</i>	Delta idő.
-----------	------------

Újraimplementáló leszármazottak: [pftd::Projectile](#), [pftd::Seal](#), [pftd::IcicleStabber](#), [pftd::Snowballer](#) és [pftd::Tower](#).

6.7.4. Adatmezők dokumentációja**6.7.4.1. CELL_N**

```
size_t const pftd::Entity::CELL_N [protected]
```

Képkockák száma.

6.7.4.2. cellSize

```
utils::Vec2i pftd::Entity::cellSize [protected]
```

Képkockák mérete (fix).

6.7.4.3. currentCell

```
size_t pftd::Entity::currentCell = 0U [protected]
```

A sprite sheet aktív képkockája.

6.7.4.4. currentSprite

```
gr::Sprite* pftd::Entity::currentSprite = nullptr [protected]
```

A jelenlegi képkocka által meghatározott sprite.

6.7.4.5. frameDurationSec

```
float pftd::Entity::frameDurationSec = 1.0f [protected]
```

Ennyi ideig "tart" egy képkocka, ezután váltani kell.

6.7.4.6. isAnimated

```
bool pftd::Entity::isAnimated = true
```

Animált e?

6.7.4.7. spriteSheet

```
sf::Texture const& pftd::Entity::spriteSheet [protected]
```

A sprite sheet.

6.7.4.8. totalElapsedSec

```
float pftd::Entity::totalElapsedSec = 0.0f [protected]
```

Eddig eltelt idő.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- [include/objects/entities/entity_base.hpp](#)

6.8. pftd::utils::parser::SaveFileParser::EntityInfo struktúrareferencia

```
#include <parsers.hpp>
```

Adatmezők

- [EntityType](#) entityType
- [utils::Vec2f](#) position
- union {
 - struct {
 - unsigned int [sealID](#)
 - unsigned int [hp](#)
- unsigned int [towerID](#)

6.8.1. Részletes leírás

Mentett játék betöltéséhez használt entitás infó.

6.8.2. Adatmezők dokumentációja

6.8.2.1.

```
union { ... } pftd::utils::parser::SaveFileParser::EntityInfo::@1
```

Típustól függő egyéb adat.

6.8.2.2. entityType

```
EntityType pftd::utils::parser::SaveFileParser::EntityInfo::entityType
```

Típusa.

6.8.2.3. hp

```
unsigned int pftd::utils::parser::SaveFileParser::EntityInfo::hp
```

6.8.2.4. position

```
utils::Vec2f pftd::utils::parser::SaveFileParser::EntityInfo::position
```

Pozíciója.

6.8.2.5. sealID

```
unsigned int pftd::utils::parser::SaveFileParser::EntityInfo::sealID
```

6.8.2.6. towerID

```
unsigned int pftd::utils::parser::SaveFileParser::EntityInfo::towerID
```

Lövedék típusra valló adatok.

Torony típusra valló adatok.

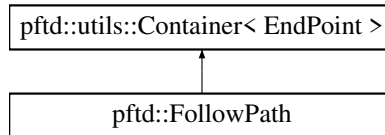
Ez a dokumentáció a struktúráról a következő fájl alapján készült:

- include/utils/[parsers.hpp](#)

6.9. pftd::FollowPath struktúráreferencia

```
#include <followpath.hpp>
```

A pftd::FollowPath osztály származási diagramja:



Publikus tagfüggvények

- [FollowPath\(\)](#)
A [Level](#), amihez tartozik, "építi fel."

6.9.1. Részletes leírás

Az ellenfelek által követett út.

6.9.2. Konstruktorkok és destruktorkok dokumentációja

6.9.2.1. FollowPath()

```
pftd::FollowPath::FollowPath ( )
```

A [Level](#), amihez tartozik, "építi fel."

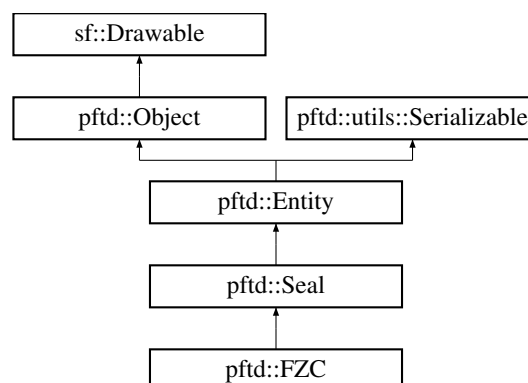
Ez a dokumentáció a struktúráról a következő fájl alapján készült:

- include/objects/entities/seals/[followpath.hpp](#)

6.10. pftd::FZC osztályreferencia

```
#include <fortified_zombie_cub.hpp>
```

A pftd::FZC osztály származási diagramja:



Publikus tagfüggvények

- [FZC](#) ([FollowPath](#) const &followpath)
- [Seal](#) * [clone](#) () const override

Egy dinamikus memóriában foglalt másolatot készít a fókából.

További örökölt tagok

6.10.1. Részletes leírás

Páncélos zombi kicsiny.

6.10.2. Konstruktorok és destruktorok dokumentációja

6.10.2.1. FZC()

```
pftd::FZC::FZC (
    FollowPath const & followpath )
```

Paraméterek

<i>followpath</i>	Az útvonal amin végig kell menjen (majd pedig vissza).
-------------------	--

6.10.3. Tagfüggvények dokumentációja

6.10.3.1. clone()

```
Seal * pftd::FZC::clone ( ) const [override], [virtual]
```

Egy dinamikus memóriában foglalt másolatot készít a fókából.

Visszatérési érték

A másolat.

Megvalósítja a következőket: [pftd::Seal](#).

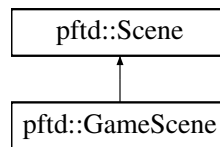
Ez a dokumentáció az osztályról a következő fájl alapján készült:

- [include/objects/entities/seals/fortified_zombie_cub.hpp](#)

6.11. pftd::GameScene osztályreferencia

```
#include <game.hpp>
```

A pftd::GameScene osztály származási diagramja:



Adatszerkezetek

- struct [Inventory](#)
- struct [InventoryItem](#)

Publikus tagfüggvények

- [GameScene](#) ()
- [~GameScene](#) ()
- void [onEvent](#) (sf::Event const &event) override
Delegált események kezelése.
- void [update](#) (float dt) override
Update.
- bool [toggleActive](#) (bool [isActive](#)) override
Nézet aktiválása/deaktiválása.
- void [startGame](#) ()
Játék elindítása.
- void [updateScore](#) ()
Pontszám felirat frissítése.
- void [updateWealth](#) ()
Pénz felirat frissítése.

Privát attribútumok

- [gr::Button](#) * [m_saveButt](#) = nullptr
- [gr::Label](#) * [m_moneyCounter](#) = nullptr
- [gr::Label](#) * [m_scoreCounter](#) = nullptr
- sf::Sound [m_hornSound](#) = sf::Sound([ResourceManager::getInstance\(\)](#)->getSound("res/audio/ready_for_↵ battle.mp3"))
- sf::Sound [m_gameoverSound](#) = sf::Sound([ResourceManager::getInstance\(\)](#)->getSound("res/audio/gameover.↵ mp3"))
- [Inventory](#) * [m_inventory](#) = nullptr
- [Level](#) * [m_level](#) = nullptr

További örökölt tagok

6.11.1. Részletes leírás

Játék nézet.

6.11.2. Konstruktorok és destruktorok dokumentációja

6.11.2.1. `GameScene()`

```
pftd::GameScene::GameScene ( )
```

6.11.2.2. `~GameScene()`

```
pftd::GameScene::~~GameScene ( )
```

6.11.3. Tagfüggvények dokumentációja

6.11.3.1. `onEvent()`

```
void pftd::GameScene::onEvent (
    sf::Event const & event ) [override], [virtual]
```

Delegált események kezelése.

Paraméterek

<i>event</i>	Kezelni kívánt esemény.
--------------	-------------------------

Megvalósítja a következőket: [pftd::Scene](#).

6.11.3.2. `startGame()`

```
void pftd::GameScene::startGame ( )
```

Játék elindítása.

Inicializálja a játék kezdetét: létrehozza a "szintet."

6.11.3.3. toggleActive()

```
bool pftd::GameScene::toggleActive (
    bool isActive ) [override], [virtual]
```

Nézet aktiválása/deaktiválása.

Paraméterek

<i>isActive</i>	Aktiválás e?
-----------------	--------------

Visszatérési érték

Történt e állapot változás.

Újraimplementált ősök: [pftd::Scene](#).

6.11.3.4. update()

```
void pftd::GameScene::update (
    float dt ) [override], [virtual]
```

Update.

Paraméterek

<i>dt</i>	Delta idő.
-----------	------------

Megvalósítja a következőket: [pftd::Scene](#).

6.11.3.5. updateScore()

```
void pftd::GameScene::updateScore ( )
```

Pontszám felirat frissítése.

6.11.3.6. updateWealth()

```
void pftd::GameScene::updateWealth ( )
```

Pénz felirat frissítése.

6.11.4. Adatmezők dokumentációja

6.11.4.1. m_gameoverSound

```
sf::Sound pftd::GameScene::m_gameoverSound = sf::Sound{ResourceManager::getInstance()->getSound("res/audio/gameover.mp3")} [private]
```

Játék vége hangeffekt.

6.11.4.2. m_hornSound

```
sf::Sound pftd::GameScene::m_hornSound = sf::Sound{ResourceManager::getInstance()->getSound("res/audio/ready_for_battle.mp3")} [private]
```

Kürt hang. Ez a játék elindításánál játszódik be.

6.11.4.3. m_inventory

```
Inventory* pftd::GameScene::m_inventory = nullptr [private]
```

Az "eszköztár."

6.11.4.4. m_level

```
Level* pftd::GameScene::m_level = nullptr [private]
```

Játéklogikát tartalmazó "szint."

6.11.4.5. m_moneyCounter

```
gr::Label* pftd::GameScene::m_moneyCounter = nullptr [private]
```

Pánzszámláló felirat.

6.11.4.6. m_saveButt

```
gr::Button* pftd::GameScene::m_saveButt = nullptr [private]
```

Mentés gomb.

6.11.4.7. m_scoreCounter

```
gr::Label* pftd::GameScene::m_scoreCounter = nullptr [private]
```

Pontszámláló felirat.

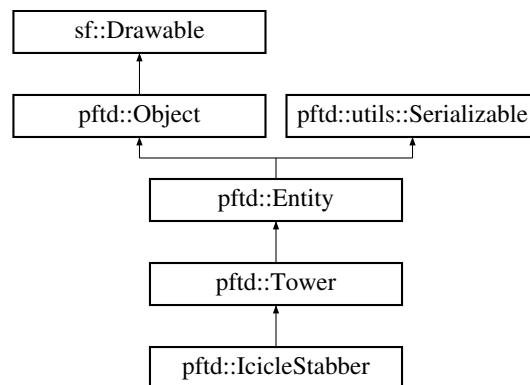
Ez a dokumentáció az osztályról a következő fájl alapján készült:

- include/scenes/[game.hpp](#)

6.12. pftd::IcicleStabber osztályreferencia

```
#include <iciclestabber.hpp>
```

A pftd::IcicleStabber osztály származási diagramja:



Publikus tagfüggvények

- [IcicleStabber](#) ([utils::Vec2f](#) const &[position](#), int [zIndex](#)=0)
- [IcicleStabber](#) ()
- [~IcicleStabber](#) ()=default
- [Tower](#) * [clone](#) () const override
Egy dinamikus memóriában foglalt másolatot készít a toronyról.
- void [update](#) (float dt) override
Update.

További örökölt tagok

6.12.1. Részletes leírás

Jégcsap kardos.

6.12.2. Konstruktorkok és destruktorkok dokumentációja

6.12.2.1. `IcicleStabber()` [1/2]

```
pftd::IcicleStabber::IcicleStabber (
    utils::Vec2f const & position,
    int zIndex = 0 )
```

6.12.2.2. `IcicleStabber()` [2/2]

```
pftd::IcicleStabber::IcicleStabber ( )
```

6.12.2.3. `~IcicleStabber()`

```
pftd::IcicleStabber::~~IcicleStabber ( ) [default]
```

6.12.3. Tagfüggvények dokumentációja

6.12.3.1. `clone()`

```
Tower * pftd::IcicleStabber::clone ( ) const [override], [virtual]
```

Egy dinamikus memóriában foglalt másolatot készít a toronyról.

Visszatérési érték

A másolat.

Megvalósítja a következőket: [pftd::Tower](#).

6.12.3.2. `update()`

```
void pftd::IcicleStabber::update (
    float dt ) [override], [virtual]
```

Update.

Paraméterek

<i>dt</i>	Delta idő.
-----------	------------

Újraimplementált ősök: [pftd::Tower](#).

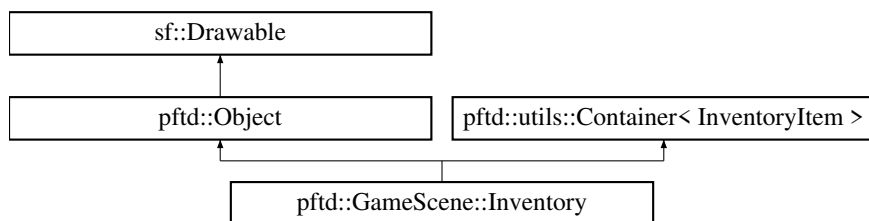
Ez a dokumentáció az osztályról a következő fájl alapján készült:

- [include/objects/entities/towers/iclestabber.hpp](#)

6.13. pftd::GameScene::Inventory struktúrareferencia

```
#include <game.hpp>
```

A pftd::GameScene::Inventory osztály származási diagramja:



Publikus tagfüggvények

- [Inventory](#) (std::string const &backgroundImageSrc)
- [~Inventory](#) ()=default
- void [draw](#) (sf::RenderTarget &target, sf::RenderStates states) const override
Objektum megjelenítése.

Adatmezők

- [gr::Sprite background](#)

További örökölt tagok

6.13.1. Részletes leírás

"Eszköztár": innen lehet megvenni a tornyokat.

6.13.2. Konstruktorkok és destruktorkok dokumentációja

6.13.2.1. Inventory()

```
pftd::GameScene::Inventory::Inventory (
    std::string const & backgroundImageSrc )
```

Paraméterek

<i>backgroundImageSrc</i>	Háttérkép elérési útvonala.
---------------------------	-----------------------------

6.13.2.2. ~Inventory()

```
pftd::GameScene::Inventory::~~Inventory ( ) [default]
```

6.13.3. Tagfüggvények dokumentációja**6.13.3.1. draw()**

```
void pftd::GameScene::Inventory::draw (
    sf::RenderTarget & target,
    sf::RenderStates states ) const [override], [virtual]
```

Objektum megjelenítése.

Paraméterek

<i>target</i>	Hol legyen megjelenítve? Ez általában egy ablak.
<i>states</i>	Render-eléshez szükséges egyéb állapotok.

Megvalósítja a következőket: [pftd::Object](#).

6.13.4. Adatmezők dokumentációja**6.13.4.1. background**

```
gr::Sprite pftd::GameScene::Inventory::background
```

Háttér.

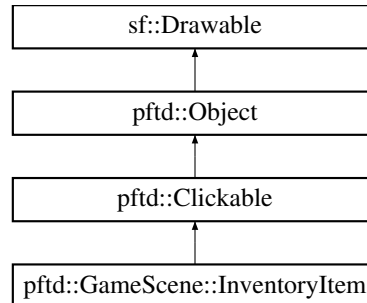
Ez a dokumentáció a struktúráról a következő fájl alapján készült:

- [include/scenes/game.hpp](#)

6.14. pftd::GameScene::InventoryItem struktúrareferencia

```
#include <game.hpp>
```

A pftd::GameScene::InventoryItem osztály származási diagramja:



Publikus tagfüggvények

- `InventoryItem (Tower *tower, Level *const level, utils::Vec2f const &position, utils::Vec2f const &size)`
- `~InventoryItem ()`
- `virtual void draw (sf::RenderTarget &target, sf::RenderStates states) const override`
Objektum megjelenítése.

Adatmezők

- `gr::Sprite frame`
- `gr::Sprite icon`
- `Tower * towerToSpawn = nullptr`
- `gr::Label priceLabel`

További örökölt tagok

6.14.1. Részletes leírás

Az "eszköztárban" tárolt torony.

6.14.2. Konstruktorkok és destruktorkok dokumentációja

6.14.2.1. InventoryItem()

```

pftd::GameScene::InventoryItem::InventoryItem (
    Tower * tower,
    Level *const level,
    utils::Vec2f const & position,
    utils::Vec2f const & size )
  
```

6.14.2.2. ~InventoryItem()

```
pftd::GameScene::InventoryItem::~~InventoryItem ( )
```

6.14.3. Tagfüggvények dokumentációja

6.14.3.1. draw()

```
virtual void pftd::GameScene::InventoryItem::draw (
    sf::RenderTarget & target,
    sf::RenderStates states ) const [override], [virtual]
```

Objektum megjelenítése.

Paraméterek

<i>target</i>	Hol legyen megjelenítve? Ez általában egy ablak.
<i>states</i>	Render-eléshez szükséges egyéb állapotok.

Megvalósítja a következőket: [pftd::Object](#).

6.14.4. Adatmezők dokumentációja

6.14.4.1. frame

```
gr::Sprite pftd::GameScene::InventoryItem::frame
```

Keret.

6.14.4.2. icon

```
gr::Sprite pftd::GameScene::InventoryItem::icon
```

Ikon.

6.14.4.3. priceLabel

```
gr::Label pftd::GameScene::InventoryItem::priceLabel
```

Árcímke (felirat).

6.14.4.4. towerToSpawn

```
Tower* pftd::GameScene::InventoryItem::towerToSpawn = nullptr
```

Torony amit meg szeretnénk venni.

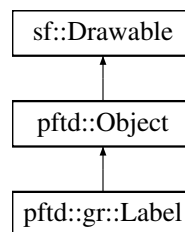
Ez a dokumentáció a struktúráról a következő fájl alapján készült:

- include/scenes/game.hpp

6.15. pftd::gr::Label osztályreferencia

```
#include <label.hpp>
```

A pftd::gr::Label osztály származási diagramja:



Publikus tagfüggvények

- **Label** (sf::Text const &text, sf::Color const &color=sf::Color::White)
- **Label** (sf::Text const &text, **utils::Vec2f** position, int zIndex=0, sf::Color const &color=sf::Color::White)
- void **setLabel** (std::string const &newLabel)
Új szöveg beállítása.
- void **setOutline** (sf::Color color, float thickness=1.0f)
Szöveg outline beállítása.
- sf::Text & **getText** ()
m_text getter.
- sf::Text const & **getText** () const
Konstans m_text getter.
- void **draw** (sf::RenderTarget &target, sf::RenderStates states) const override
Objektum megjelenítése.

Privát attribútumok

- sf::Text **m_text**

További örökölt tagok

6.15.1. Részletes leírás

Felirat GUI elem.

6.15.2. Konstruktorok és destruktorok dokumentációja

6.15.2.1. Label() [1/2]

```
pftd::gr::Label::Label (
    sf::Text const & text,
    sf::Color const & color = sf::Color::White )
```

Paraméterek

<i>text</i>	Szöveg objektum: maga a szöveg, betűtípus és betűméret.
<i>color</i>	Betűszín.

6.15.2.2. Label() [2/2]

```
pftd::gr::Label::Label (
    sf::Text const & text,
    utils::Vec2f position,
    int zIndex = 0,
    sf::Color const & color = sf::Color::White )
```

Paraméterek

<i>text</i>	Szöveg objektum: maga a szöveg, betűtípus és betűméret.
<i>position</i>	Pozíció.
<i>zIndex</i>	Z koordináta.
<i>color</i>	Betűszín.

6.15.3. Tagfüggvények dokumentációja

6.15.3.1. draw()

```
void pftd::gr::Label::draw (
    sf::RenderTarget & target,
    sf::RenderStates states ) const [inline], [override], [virtual]
```

Objektum megjelenítése.

Paraméterek

<i>target</i>	Hol legyen megjelenítve? Ez általában egy ablak.
<i>states</i>	Render-eléshez szükséges egyéb állapotok.

Megvalósítja a következőket: [pftd::Object](#).

6.15.3.2. getText() [1/2]

```
sf::Text & pftd::gr::Label::getText ( ) [inline]
```

m_text getter.

Visszatérési érték

A tárolt szöveg objektum.

6.15.3.3. getText() [2/2]

```
sf::Text const & pftd::gr::Label::getText ( ) const [inline]
```

Konstans m_text getter.

Visszatérési érték

A tárolt szöveg objektum.

6.15.3.4. setLabel()

```
void pftd::gr::Label::setLabel (
    std::string const & newLabel ) [inline]
```

Új szöveg beállítása.

Paraméterek

<i>newLabel</i>	Az új szöveg.
-----------------	---------------

6.15.3.5. setOutline()

```
void pftd::gr::Label::setOutline (
    sf::Color color,
    float thickness = 1.0f ) [inline]
```

Szöveg outline beállítása.

Paraméterek

<i>color</i>	Színe.
<i>thickness</i>	Vastagsága.

6.15.4. Adatmezők dokumentációja

6.15.4.1. m_text

```
sf::Text pftd::gr::Label::m_text [private]
```

Tárolt szöveg objektum.

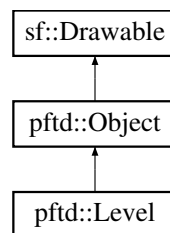
Ez a dokumentáció az osztályról a következő fájl alapján készült:

- [include/objects/gui/label.hpp](#)

6.16. pftd::Level osztályreferencia

```
#include <level.hpp>
```

A pftd::Level osztály származási diagramja:



Adatszerkezetek

- struct [Nest](#)
- struct [Stats](#)

Publikus tagfüggvények

- `Level (Stats stats=Stats{})`
- `virtual ~Level ()`
- `virtual void loseHP (int hpLost=1)`
HP vesztes.
- `virtual bool placeTower ()`
Kiválasztott torony lehelyezése.
- `void deselectTower ()`
Kiválasztott torony törlése.
- `void selectTower (Tower *newTower)`
Torony kiválasztása.
- `void update (float dt)`
Update.
- `bool isGameOver () const`
Vége van e játéknak?
- `virtual void draw (sf::RenderTarget &target, sf::RenderStates states) const override`
Objektum megjelenítése.

Adatmezők

- `Stats stats`
- `Tower * selectedTower = nullptr`

Védett tagfüggvények

- `void spawnSeal ()`
Ellenfél spawnolás.

Védett attribútumok

- `Nest * nest = nullptr`
- `FollowPath followPath`
- `std::vector< Tower * > towers`
- `std::vector< Seal * > seals`
- `std::vector< Projectile * > projectiles`
- `utils::parser::LevelConfigParser config`

Privát attribútumok

- `float m_accuTimeSpawnSec = 0.0f`

6.16.1. Részletes leírás

Játéklogikát és entitásokat tartalmazó osztály.

6.16.2. Konstruktorkok és destruktorkok dokumentációja

6.16.2.1. Level()

```
pftd::Level::Level (
    Stats stats = Stats{} )
```

Paraméterek

<i>stats</i>	Kezdeti játékállás.
--------------	---------------------

6.16.2.2. ~Level()

```
virtual pftd::Level::~~Level ( ) [virtual]
```

6.16.3. Tagfüggvények dokumentációja**6.16.3.1. deselectTower()**

```
void pftd::Level::deselectTower ( )
```

Kiválasztott torony törlése.

6.16.3.2. draw()

```
virtual void pftd::Level::draw (
    sf::RenderTarget & target,
    sf::RenderStates states ) const [override], [virtual]
```

Objektum megjelenítése.

Paraméterek

<i>target</i>	Hol legyen megjelenítve? Ez általában egy ablak.
<i>states</i>	Render-eléshez szükséges egyéb állapotok.

Megvalósítja a következőket: [pftd::Object](#).

6.16.3.3. isGameOver()

```
bool pftd::Level::isGameOver ( ) const
```

Vége van e játéknak?

Visszatérési érték

Vesztett e a játékos?

6.16.3.4. loseHP()

```
virtual void pftd::Level::loseHP (
    int hpLost = 1 ) [virtual]
```

HP vesztes.

Paraméterek

<i>hpLost</i>	Ennyi HP-t veszít a játékos.
---------------	------------------------------

6.16.3.5. placeTower()

```
virtual bool pftd::Level::placeTower ( ) [virtual]
```

Kiválasztott torony lehelyezése.

Visszatérési érték

Sikerült e lehelyezni?

6.16.3.6. selectTower()

```
void pftd::Level::selectTower (
    Tower * newTower )
```

Torony kiválasztása.

Ezt a játék nézet "eszköztárja" adja át.

Paraméterek

<i>newTower</i>	Lehelyezni kívánt torony.
-----------------	---------------------------

6.16.3.7. spawnSeal()

```
void pftd::Level::spawnSeal ( ) [protected]
```

Ellenfél spawnolás.

6.16.3.8. update()

```
void pftd::Level::update (
    float dt )
```

Update.

Paraméterek

<i>dt</i>	Delta idő.
-----------	------------

6.16.4. Adatmezők dokumentációja

6.16.4.1. config

```
utils::parser::LevelConfigParser pftd::Level::config [protected]
```

Inicializáláshoz szükséges konfiguráció.

6.16.4.2. followPath

```
FollowPath pftd::Level::followPath [protected]
```

Ellenségek által követett útvonal.

6.16.4.3. m_accuTimeSpawnSec

```
float pftd::Level::m_accuTimeSpawnSec = 0.0f [private]
```

Előző spawn óta eltelt idő.

6.16.4.4. nest

```
Nest* pftd::Level::nest = nullptr [protected]
```

A fészek (cél).

6.16.4.5. projectiles

```
std::vector<Projectile*> pftd::Level::projectiles [protected]
```

Lövedékek.

6.16.4.6. seals

```
std::vector<Seal*> pftd::Level::seals [protected]
```

Élő ellenfelek.

6.16.4.7. selectedTower

```
Tower* pftd::Level::selectedTower = nullptr
```

Lehelyezni kívánt torony.

6.16.4.8. stats

```
Stats pftd::Level::stats
```

Játékos statisztika.

6.16.4.9. towers

```
std::vector<Tower*> pftd::Level::towers [protected]
```

Lehelyezett tornyok.

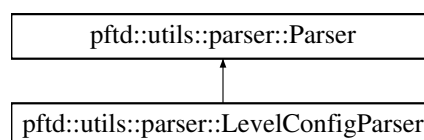
Ez a dokumentáció az osztályról a következő fájl alapján készült:

- include/game/level.hpp

6.17. pftd::utils::parser::LevelConfigParser osztályreferencia

```
#include <parsers.hpp>
```

A pftd::utils::parser::LevelConfigParser osztály származási diagramja:



Publikus tagfüggvények

- [LevelConfigParser](#) (std::string const &sourceFile)
- [~LevelConfigParser](#) ()=default
- void [parse](#) () override
 - A *parse* főciklus.
- std::vector< [Vec2f](#) > [getAttribute](#) (std::string name) const

Privát tagfüggvények

- `std::string _getAttribute ()`

Privát attribútumok

- `std::string m_lastAttribute`
- `std::unordered_map< std::string, std::vector< Vec2f > > m_attribs`

További örökölt tagok

6.17.1. Részletes leírás

Egy szint alap beállításait tudjuk vele betölteni: az ellenfelek által követett útvonalat és a fészek helyzetét.

6.17.2. Konstruktorkok és destruktorkok dokumentációja

6.17.2.1. LevelConfigParser()

```
pftd::utils::parser::LevelConfigParser::LevelConfigParser (
    std::string const & sourceFile )
```

6.17.2.2. ~LevelConfigParser()

```
pftd::utils::parser::LevelConfigParser::~~LevelConfigParser ( ) [default]
```

6.17.3. Tagfüggvények dokumentációja

6.17.3.1. _getAttribute()

```
std::string pftd::utils::parser::LevelConfigParser::_getAttribute ( ) [private]
```

6.17.3.2. getAttribute()

```
std::vector< Vec2f > pftd::utils::parser::LevelConfigParser::getAttribute (
    std::string name ) const
```

6.17.3.3. parse()

```
void pftd::utils::parser::LevelConfigParser::parse ( ) [override], [virtual]
```

A parse főciklus.

Megvalósítja a következőket: [pftd::utils::parser::Parser](#).

6.17.4. Adatmezők dokumentációja

6.17.4.1. m_attribs

```
std::unordered_map<std::string, std::vector<Vec2f> > pftd::utils::parser::LevelConfigParser↔
::m_attribs [private]
```

6.17.4.2. m_lastAttribute

```
std::string pftd::utils::parser::LevelConfigParser::m_lastAttribute [private]
```

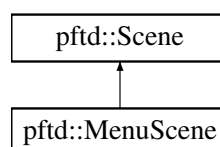
Ez a dokumentáció az osztályról a következő fájl alapján készült:

- [include/utills/parsers.hpp](#)

6.18. pftd::MenuScene osztályreferencia

```
#include <menu.hpp>
```

A pftd::MenuScene osztály származási diagramja:



Publikus tagfüggvények

- `MenuScene()`
- `~MenuScene()`
- void `onEvent` (`sf::Event const &event`) override
Delegált események kezelése.
- void `update` (`float dt`) override
Update.

Privát attribútumok

- `std::vector< gr::Button * > m_buttons`

További örökölt tagok

6.18.1. Részletes leírás

Menü nézet.

6.18.2. Konstruktorkok és destruktorkok dokumentációja

6.18.2.1. MenuScene()

```
pftd::MenuScene::MenuScene ( )
```

6.18.2.2. ~MenuScene()

```
pftd::MenuScene::~MenuScene ( )
```

6.18.3. Tagfüggvények dokumentációja

6.18.3.1. onEvent()

```
void pftd::MenuScene::onEvent (
    sf::Event const & event ) [override], [virtual]
```

Delegált események kezelése.

Paraméterek

<i>event</i>	Kezelni kívánt esemény.
--------------	-------------------------

Megvalósítja a következőket: [pftd::Scene](#).

6.18.3.2. update()

```
void pftd::MenuScene::update (
    float dt ) [override], [virtual]
```

Update.

Paraméterek

<i>dt</i>	Delta idő.
-----------	------------

Megvalósítja a következőket: [pftd::Scene](#).

6.18.4. Adatmezők dokumentációja

6.18.4.1. m_buttons

```
std::vector<gr::Button*> pftd::MenuScene::m_buttons [private]
```

Gombok.

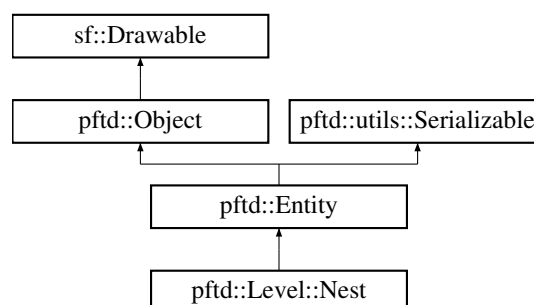
Ez a dokumentáció az osztályról a következő fájl alapján készült:

- include/scenes/[menu.hpp](#)

6.19. pftd::Level::Nest struktúrareferencia

```
#include <level.hpp>
```

A pftd::Level::Nest osztály származási diagramja:



Publikus tagfüggvények

- `Nest (utils::Vec2f const &position)`
A pozíció kívül minden más hard code-olt.
- `Nest (Nest const &other)=default`
- `~Nest ()`

Adatmezők

- `int const radiusPixel = 110`

További örökölt tagok

6.19.1. Részletes leírás

Fészek.

6.19.2. Konstruktorok és destruktorok dokumentációja

6.19.2.1. Nest() [1/2]

```
pftd::Level::Nest::Nest (
    utils::Vec2f const & position ) [explicit]
```

A pozíció kívül minden más hard code-olt.

Paraméterek

<code>position</code>	Pozíció.
-----------------------	----------

6.19.2.2. Nest() [2/2]

```
pftd::Level::Nest::Nest (
    Nest const & other ) [default]
```

6.19.2.3. ~Nest()

```
pftd::Level::Nest::~~Nest ( )
```


6.19.3. Adatmezők dokumentációja

6.19.3.1. radiusPixel

```
int const pftd::Level::Nest::radiusPixel = 110
```

Az a környezete, amelybe tornyot nem lehet tenni.

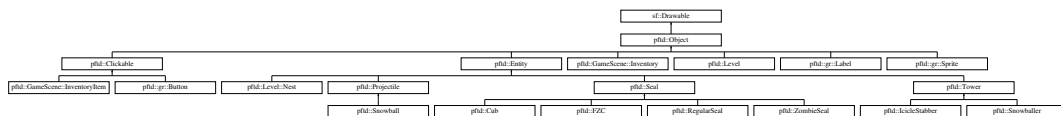
Ez a dokumentáció a struktúráról a következő fájl alapján készült:

- include/game/[level.hpp](#)

6.20. pftd::Object osztályreferencia

```
#include <object_base.hpp>
```

A pftd::Object osztály származási diagramja:



Adatszerkezetek

- struct [Compare](#)

Publikus tagfüggvények

- [Object](#) ()=default
- [Object](#) (utils::Vec2f position, utils::Vec2f size, int zIndex=0)
- [Object](#) (Object const &)=default
- [Object](#) (Object &&) noexcept=default
- virtual [~Object](#) ()=default
- virtual void [draw](#) (sf::RenderTarget &target, sf::RenderStates states) const =0
Objektum megjelenítése.

Védett attribútumok

- int [zIndex](#) = 0
- utils::Vec2f [position](#)
- utils::Vec2f [size](#)

6.20.1. Részletes leírás

Programelemek.

6.20.2. Konstruktorkok és destruktorkok dokumentációja

6.20.2.1. Object() [1/4]

```
pftd::Object::Object ( ) [default]
```

6.20.2.2. Object() [2/4]

```
pftd::Object::Object (
    utils::Vec2f position,
    utils::Vec2f size,
    int zIndex = 0 ) [inline]
```

Paraméterek

<i>position</i>	Pozíció koordinátái.
<i>size</i>	Méret.
<i>zIndex</i>	Z koordináta: "mélységi index."

6.20.2.3. Object() [3/4]

```
pftd::Object::Object (
    Object const & ) [default]
```

6.20.2.4. Object() [4/4]

```
pftd::Object::Object (
    Object && ) [default], [noexcept]
```

6.20.2.5. ~Object()

```
virtual pftd::Object::~Object ( ) [virtual], [default]
```

6.20.3. Tagfüggvények dokumentációja

6.20.3.1. draw()

```
virtual void pftd::Object::draw (
    sf::RenderTarget & target,
    sf::RenderStates states ) const [pure virtual]
```

Objektum megjelenítése.

Paraméterek

<i>target</i>	Hol legyen megjelenítve? Ez általában egy ablak.
<i>states</i>	Render-eléshez szükséges egyéb állapotok.

Megvalósítják a következők: [pftd::Level](#), [pftd::Entity](#), [pftd::gr::Button](#), [pftd::gr::Sprite](#), [pftd::gr::Label](#), [pftd::GameScene::InventoryItem](#) és [pftd::GameScene::Inventory](#).

6.20.4. Adatmezők dokumentációja

6.20.4.1. position

```
utils::Vec2f pftd::Object::position [protected]
```

Pozíció.

6.20.4.2. size

```
utils::Vec2f pftd::Object::size [protected]
```

Méret (szélesség x magasság).

6.20.4.3. zIndex

```
int pftd::Object::zIndex = 0 [protected]
```

Z koordináta.

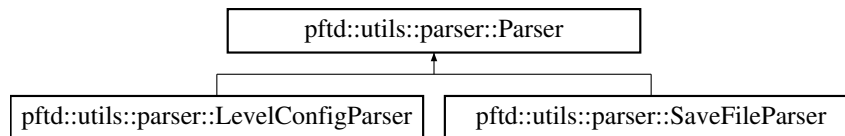
Ez a dokumentáció az osztályról a következő fájl alapján készült:

- [include/objects/object_base.hpp](#)

6.21. pftd::utils::parser::Parser osztályreferencia

```
#include <parsers.hpp>
```

A pftd::utils::parser::Parser osztály származási diagramja:



Publikus tagfüggvények

- **Parser** (std::string const &sourceFile, std::string label="")
- virtual **~Parser** ()
- void **reset** ()
Reseteli az IO flageket, és a stream elejére mozgatja a kurzort.
- bool **isLabelValid** (bool **skip**=true)
Megmondja, hogy tényleg a megadott karaktersorral kezdődik-e a stream.
- void **skip** (size_t howMany, char until='\n')
Ugorjon át valamennyi karaktert, de legfeljebb egy megadottig.
- void **skip** (char while_)
Ugorjon át mindent amíg a következő karakter egy megadott.
- template<typename T >
 T **get** ()
Beolvas egy megadott típusú adatot.
- char **peekAhead** ()
Megnézi, hogy milyen karakter van a kurzornál (lásd: megjegyzés).
- virtual void **parse** ()=0
A parse főciklus.

Védett attribútumok

- std::ifstream **sourceStream**
- std::string **validLabel**
- char **commentDenoter** = '#'

Privát tagfüggvények

- void **_skipLine** ()
Egész sor átugrása.
- void **_skipWhitespace** ()
Whitespace-ek átugrása.

6.21.1. Konstruktorkok és destruktorkok dokumentációja

6.21.1.1. Parser()

```
pftd::utils::parser::Parser::Parser (
    std::string const & sourceFile,
    std::string label = "" )
```

Paraméterek

<i>sourceFile</i>	A fájlnek az elérési útvonala, ahonnan olvasni szeretnénk.
<i>label</i>	A karaktersor, amellyel kötelezően kezdődnie kell a fájlnek.

6.21.1.2. ~Parser()

```
virtual pftd::utils::parser::Parser::~~Parser ( ) [virtual]
```

6.21.2. Tagfüggvények dokumentációja**6.21.2.1. _skipLine()**

```
void pftd::utils::parser::Parser::_skipLine ( ) [private]
```

Egész sor átugrása.

6.21.2.2. _skipWhitespace()

```
void pftd::utils::parser::Parser::_skipWhitespace ( ) [private]
```

Whitespace-ek átugrása.

6.21.2.3. get()

```
template<typename T >  
T pftd::utils::parser::Parser::get ( ) [inline]
```

Beolvas egy megadott típusú adatot.

Megjegyzés: a kommenteket átugorja.

Visszatérési érték

A beolvasott adat.

6.21.2.4. isLabelValid()

```
bool pftd::utils::parser::Parser::isLabelValid (  
    bool skip = true )
```

Megmondja, hogy tényleg a megadott karaktersorral kezdődik e a stream.

Paraméterek

<i>skip</i>	Ha beolvasta a karaktersort, akkor skippelje e, vagy visszamozogjon.
-------------	--

Visszatérési érték

Helyes e a karaktersor.

6.21.2.5. parse()

```
virtual void pftd::utils::parser::Parser::parse ( ) [pure virtual]
```

A parse főciklus.

Megvalósítják a következők: [pftd::utils::parser::LevelConfigParser](#) és [pftd::utils::parser::SaveFileParser](#).

6.21.2.6. peekAhead()

```
char pftd::utils::parser::Parser::peekAhead ( )
```

Megnézi, hogy milyen karakter van a kurzornál (lásd: megjegyzés).

Megjegyzés: átugorja a white space-eket és a kommenteket, és csak az azutáni karaktert nézi meg.

Visszatérési érték

A kurzornál levő karakter.

6.21.2.7. reset()

```
void pftd::utils::parser::Parser::reset ( )
```

Reseteli az IO flageket, és a stream elejére mozgatja a kurzort.

6.21.2.8. skip() [1/2]

```
void pftd::utils::parser::Parser::skip (
    char while_ )
```

Ugorjon át mindent amíg a következő karakter egy megadott.

Paraméterek

<i>while</i> ↔	Addig ugor át mindent, amíg ez a következő karakter.
—	

6.21.2.9. skip() [2/2]

```
void pftd::utils::parser::Parser::skip (
    size_t howMany,
    char until = '\n' )
```

Ugorjon át valamennyi karaktert, de legfeljebb egy megadottig.

Paraméterek

<i>howMany</i>	Ennyi karaktert ugrik át legfeljebb.
<i>until</i>	Eddig a karakterig ugrik át mindent, ha még nem ugrott át <i>howMany</i> számút (ezt is beleértve).

6.21.3. Adatmezők dokumentációja**6.21.3.1. commentDenoter**

```
char pftd::utils::parser::Parser::commentDenoter = '#' [protected]
```

Kommentet jelölő karakter (ezt követően a sorban minden karakter a komment része).

6.21.3.2. sourceStream

```
std::ifstream pftd::utils::parser::Parser::sourceStream [protected]
```

Input stream.

6.21.3.3. validLabel

```
std::string pftd::utils::parser::Parser::validLabel [protected]
```

Elvart karaktersor, amivel a streamnek kezdődnie kell.

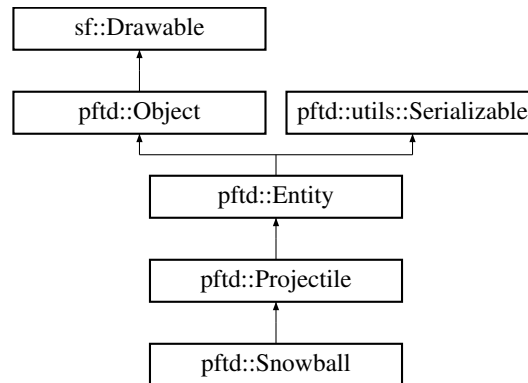
Ez a dokumentáció az osztályról a következő fájl alapján készült:

- include/utils/[parsers.hpp](#)

6.22. pftd::Projectile osztályreferencia

```
#include <projectile_base.hpp>
```

A pftd::Projectile osztály származási diagramja:



Publikus tagfüggvények

- virtual `~Projectile()`=default
- virtual void `update` (float dt) override
Update.

Adatmezők

- `utils::Vec2f` `direction`
- float `linearSpeed`
- float `angularVelocityRadPerSec`

Védett tagfüggvények

- `Projectile` (std::string const &spriteSrc, `utils::Vec2f` const &position, `utils::Vec2f` const &size, `utils::Vec2f` const &direction, float speed, float angularSpeed=0.0f, int zIndex=0)

További örökölt tagok

6.22.1. Konstruktorkok és destruktorkok dokumentációja

6.22.1.1. ~Projectile()

```
virtual pftd::Projectile::~~Projectile ( ) [virtual], [default]
```


6.22.1.2. Projectile()

```
pftd::Projectile::Projectile (
    std::string const & spriteSrc,
    utils::Vec2f const & position,
    utils::Vec2f const & size,
    utils::Vec2f const & direction,
    float speed,
    float angularSpeed = 0.0f,
    int zIndex = 0 ) [protected]
```

Paraméterek

<i>spriteSrc</i>	Sprite eleérési útvonal.
<i>position</i>	Pozíció.
<i>size</i>	Méret.
<i>direction</i>	Írány.
<i>speed</i>	Lineáris gyorsaság.
<i>angularSpeed</i>	Forgási sebesség (radián / mp).
<i>zIndex</i>	Z koordináta.

6.22.2. Tagfüggvények dokumentációja

6.22.2.1. update()

```
virtual void pftd::Projectile::update (
    float dt ) [override], [virtual]
```

Update.

Paraméterek

<i>dt</i>	Delta idő.
-----------	------------

Újraimplementált ősök: [pftd::Entity](#).

6.22.3. Adatmezők dokumentációja

6.22.3.1. angularVelocityRadPerSec

```
float pftd::Projectile::angularVelocityRadPerSec
```

Forgási sebesség.

6.22.3.2. direction

```
utils::Vec2f pftd::Projectile::direction
```

Írány (normalizált).

6.22.3.3. linearSpeed

```
float pftd::Projectile::linearSpeed
```

Lineáris gyorsaság.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- [include/objects/entities/projectiles/projectile_base.hpp](#)

6.23. pftd::utils::Random osztályreferencia

```
#include <random_gen.hpp>
```

Publikus tagfüggvények

- [Random](#) ()=delete

Statikus publikus tagfüggvények

- `template<typename RandomDist >`
`static auto generate (RandomDist distribution)`
[Random](#) szám generálása megadott eloszlás szerint.

Statikus publikus attribútumok

- `static std::default_random_engine randomEngine`

6.23.1. Részletes leírás

[Random](#) szám generáláshoz használt segédosztály. Ez egy teljesen statikus osztály.

6.23.2. Konstruktorkok és destruktorkok dokumentációja

6.23.2.1. Random()

```
pftd::utils::Random::Random ( ) [delete]
```

6.23.3. Tagfüggvények dokumentációja

6.23.3.1. generate()

```
template<typename RandomDist >  
static auto pftd::utils::Random::generate (  
    RandomDist distribution ) [inline], [static]
```

[Random](#) szám generálása megadott eloszlás szerint.

Paraméterek

<i>distribution</i>	Generálandó számok eloszlása.
---------------------	-------------------------------

Visszatérési érték

Generált szám.

6.23.4. Adatmezők dokumentációja

6.23.4.1. randomEngine

```
std::default_random_engine pftd::utils::Random::randomEngine [static]
```

Felhasznált random engine. (Az alapértelmezett.)

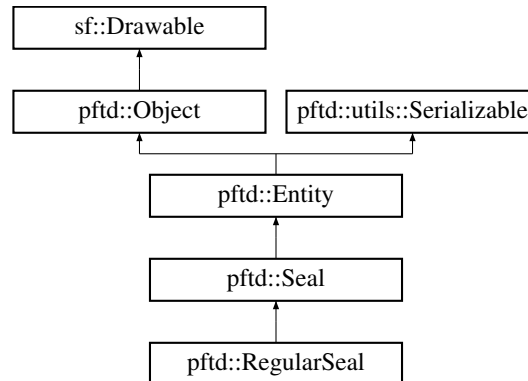
Ez a dokumentáció az osztályról a következő fájl alapján készült:

- [include/utils/random_gen.hpp](#)

6.24. pftd::RegularSeal osztályreferencia

```
#include <regular.hpp>
```

A pftd::RegularSeal osztály származási diagramja:



Publikus tagfüggvények

- [RegularSeal](#) ([FollowPath](#) const &followpath)
- [Seal](#) * [clone](#) () const override

Egy dinamikus memóriában foglalt másolatot készít a főkából.

További örökölt tagok

6.24.1. Részletes leírás

"Normális" foka.

6.24.2. Konstruktorkok és destruktorkok dokumentációja

6.24.2.1. RegularSeal()

```
pftd::RegularSeal::RegularSeal (
    FollowPath const & followpath )
```

Paraméterek

<i>followpath</i>	Az útvonal amin végig kell menjen (majd pedig vissza).
-------------------	--

6.24.3. Tagfüggvények dokumentációja

6.24.3.1. clone()

```
Seal * pftd::RegularSeal::clone ( ) const [override], [virtual]
```

Egy dinamikus memóriában foglalt másolatot készít a fókából.

Visszatérési érték

A másolat.

Megvalósítja a következőket: [pftd::Seal](#).

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- [include/objects/entities/seals/regular.hpp](#)

6.25. pftd::Renderer osztályreferencia

```
#include <app.hpp>
```

Publikus tagfüggvények

- [Renderer](#) (unsigned int width, unsigned int height, std::string const &windowTitle)
Létrehozza az ablakot a rendering kontextussal együtt.
- [Renderer](#) ([Renderer](#) const &)=delete
- [Renderer](#) ([Renderer](#) &&)=delete
- [~Renderer](#) ()
Felszabadítja az ablak erőforrásait.
- void [render](#) ()
Átmásolja a frame buffer-re az `m_queue`-ban tárolt objektumokat.
- void [clear](#) () const
Törli a frame buffer-t.
- void [display](#) () const
Megjeleníti a frame buffer tartalmát.
- void [pushQueue](#) ([Object](#) const *o)
Hozzáad az `m_queue`-hoz egy megjelenítendő objektumot.
- sf::RenderWindow * [getWindow](#) () const
`m_window` getter.

Privát attribútumok

- sf::RenderWindow * [m_window](#) = nullptr
- unsigned int [m_width](#)
- unsigned int [m_height](#)
- std::priority_queue< [Object](#) const *, std::vector< [Object](#) const * >, [Object::Compare](#) > [m_queue](#)

Barátok

- class [App](#)

6.25.1. Részletes leírás

A renderer.

6.25.2. Konstruktorok és destruktorok dokumentációja

6.25.2.1. `Renderer()` [1/3]

```
pftd::Renderer::Renderer (
    unsigned int width,
    unsigned int height,
    std::string const & windowTitle )
```

Létrehozza az ablakot a rendering kontextussal együtt.

Paraméterek

<i>width</i>	Az ablak szélessége.
<i>height</i>	Az ablak magassága.
<i>windowTitle</i>	Az ablak címe.

6.25.2.2. `Renderer()` [2/3]

```
pftd::Renderer::Renderer (
    Renderer const & ) [delete]
```

6.25.2.3. `Renderer()` [3/3]

```
pftd::Renderer::Renderer (
    Renderer && ) [delete]
```

6.25.2.4. `~Renderer()`

```
pftd::Renderer::~~Renderer ( )
```

Felszabadítja az ablak erőforrásait.

6.25.3. Tagfüggvények dokumentációja

6.25.3.1. clear()

```
void pftd::Renderer::clear ( ) const [inline]
```

Törli a frame buffer-t.

6.25.3.2. display()

```
void pftd::Renderer::display ( ) const [inline]
```

Megjeleníti a frame buffer tartalmát.

6.25.3.3. getWindow()

```
sf::RenderWindow * pftd::Renderer::getWindow ( ) const [inline]
```

m_window getter.

Visszatérési érték

A kezelt ablak.

6.25.3.4. pushQueue()

```
void pftd::Renderer::pushQueue (
    Object const * o ) [inline]
```

Hozzáad az m_queue-hoz egy megjelenítendő objektumot.

Paraméterek

<i>o</i>	Az objektum amit megjeleníteni szeretnénk.
----------	--

6.25.3.5. render()

```
void pftd::Renderer::render ( )
```

Átmásolja a frame buffer-re az `m_queue`-ban tárolt objektumokat.

6.25.4. Barát és kapcsolódó függvények dokumentációja

6.25.4.1. App

```
friend class App [friend]
```

6.25.5. Adatmezők dokumentációja

6.25.5.1. m_height

```
unsigned int pftd::Renderer::m_height [private]
```

Az ablak magassága.

6.25.5.2. m_queue

```
std::priority_queue<Object const*, std::vector<Object const*>, Object::Compare> pftd::↵  
Renderer::m_queue [private]
```

A render queue.

6.25.5.3. m_width

```
unsigned int pftd::Renderer::m_width [private]
```

Az ablak szélessége.

6.25.5.4. m_window

```
sf::RenderWindow* pftd::Renderer::m_window = nullptr [private]
```

A kezelt ablak.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- `include/app.hpp`

6.26. pftd::ResourceManager osztályreferencia

```
#include <resources.hpp>
```

Publikus tagfüggvények

- `ResourceManager (ResourceManager const &)=delete`
- `ResourceManager (ResourceManager &&)=delete`
- `~ResourceManager ()`
Felszabadítja a tárolt erőforrásokat.
- `bool loadDefaultFont (std::string const &path)`
Az alapértelmezett (valójában csak ez van) betűtípus betöltése.
- `sf::Texture const & getTexture (std::string const &source)`
Új vagy tárolt textúra megszerzése.
- `sf::SoundBuffer const & getSound (std::string const &source)`
Új vagy tárolt audio buffer megszerzése.
- `sf::Font const & getDefaultFont () const`
m_defaultFont getter.

Statikus publikus tagfüggvények

- `static ResourceManager * getInstance ()`
A singleton instance megszerzése.
- `static ResourceManager * create ()`
Létrehozza a singleton instance-t.
- `static void destroy ()`
Törli a singleton instance-t.

Privát tagfüggvények

- `ResourceManager ()=default`

Privát attribútumok

- `sf::Font m_defaultFont`
- `std::unordered_map< std::string, sf::Texture > m_textures`
- `std::unordered_map< std::string, sf::SoundBuffer > m_sounds`

Statikus privát attribútumok

- `static ResourceManager * m_instance`

6.26.1. Részletes leírás

Az erőforrás-kezelő.

6.26.2. Konstruktorok és destruktorok dokumentációja

6.26.2.1. ResourceManager() [1/3]

```
pftd::ResourceManager::ResourceManager (
    ResourceManager const & ) [delete]
```

6.26.2.2. ResourceManager() [2/3]

```
pftd::ResourceManager::ResourceManager (
    ResourceManager && ) [delete]
```

6.26.2.3. ~ResourceManager()

```
pftd::ResourceManager::~~ResourceManager ( )
```

Felszabadítja a tárolt erőforrásokat.

6.26.2.4. ResourceManager() [3/3]

```
pftd::ResourceManager::ResourceManager ( ) [private], [default]
```

6.26.3. Tagfüggvények dokumentációja

6.26.3.1. create()

```
static ResourceManager * pftd::ResourceManager::create ( ) [static]
```

Létrehozza a singleton instance-t.

Megjegyzés: Meg kell hívni a `destroy()`-t, ha már nincs szükség rá!

Visszatérési érték

Az osztályhoz tartozó singleton instance.

6.26.3.2. destroy()

```
static void pftd::ResourceManager::destroy ( ) [inline], [static]
```

Törli a singleton instance-t.

6.26.3.3. getDefaultFont()

```
sf::Font const & pftd::ResourceManager::getDefaultFont ( ) const [inline]
```

m_defaultFont getter.

Visszatérési érték

Az alapértelmezett betűtípus.

6.26.3.4. getInstance()

```
static ResourceManager * pftd::ResourceManager::getInstance ( ) [inline], [static]
```

A singleton instance megszerzése.

Visszatérési érték

Az osztályhoz tartozó singleton instance.

6.26.3.5. getSound()

```
sf::SoundBuffer const & pftd::ResourceManager::getSound (
    std::string const & source )
```

Új vagy tárolt audio buffer megszerzése.

Megjegyzés: LoadError-t dob, ha nem találja.

Paraméterek

<i>source</i>	A hangfájl elérési útvonala / azonosítója (megegyezik).
---------------	---

Visszatérési érték

A kért audio buffer.

6.26.3.6. `getTexture()`

```
sf::Texture const & pftd::ResourceManager::getTexture (
    std::string const & source )
```

Új vagy tárolt textúra megszerzése.

Megjegyzés: `LoadError`-t dob, ha nem találja.

Paraméterek

<i>source</i>	A textúra elérési útvonala / azonosítója (megegyezik).
---------------	--

Visszatérési érték

A kért textúra.

6.26.3.7. `loadDefaultFont()`

```
bool pftd::ResourceManager::loadDefaultFont (
    std::string const & path )
```

Az alapértelmezett (valójában csak ez van) betűtípus betöltése.

Paraméterek

<i>path</i>	A betűtípus elérési útvonala.
-------------	-------------------------------

Visszatérési érték

Sikerült e betölteni.

6.26.4. Adatmezők dokumentációja

6.26.4.1. `m_defaultFont`

```
sf::Font pftd::ResourceManager::m_defaultFont [private]
```

Az alapértelmezett betűtípus.

6.26.4.2. m_instance

```
ResourceManager* pftd::ResourceManager::m_instance [static], [private]
```

A singleton instance.

6.26.4.3. m_sounds

```
std::unordered_map<std::string, sf::SoundBuffer> pftd::ResourceManager::m_sounds [private]
```

Audio: [source, hang buffer példány]

6.26.4.4. m_textures

```
std::unordered_map<std::string, sf::Texture> pftd::ResourceManager::m_textures [private]
```

Textúrák: [source, textúra példány]

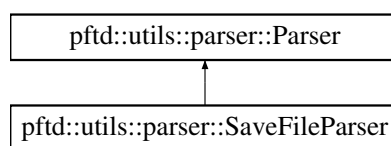
Ez a dokumentáció az osztályról a következő fájl alapján készült:

- include/[resources.hpp](#)

6.27. pftd::utils::parser::SaveFileParser osztályreferencia

```
#include <parsers.hpp>
```

A pftd::utils::parser::SaveFileParser osztály származási diagramja:



Adatszerkezetek

- struct [EntityInfo](#)
- struct [StatsInfo](#)

Publikus típusok

- enum class [EntityType](#) { [TOWER](#) = 0U , [SEAL](#) }

Publikus tagfüggvények

- `SaveFileParser` (`std::string const &sourceFile`)
- `~SaveFileParser` ()=default
- void `parse` () override
A parse fűciklus.
- `StatsInfo` `getStats` () const
Mentett statisztika beolvasása: pont, pénz, HP.
- `std::vector< EntityInfo >` `getEntities` () const
Mentett entitások beolvasása.

Privát attribútumok

- `StatsInfo` `m_readStats`
- `std::vector< EntityInfo >` `m_entities`

További örökölt tagok

6.27.1. Részletes leírás

Mentett játékállás betöltésére szolgál.

6.27.2. Enumeráció-tagok dokumentációja

6.27.2.1. EntityType

```
enum class pftd::utils::parser::SaveFileParser::EntityType [strong]
```

Mentett játék betöltéséhez használt entitás típusok.

Enumeráció-értékek

TOWER	
SEAL	

6.27.3. Konstruktorkok és destruktorkok dokumentációja

6.27.3.1. SaveFileParser()

```
pftd::utils::parser::SaveFileParser::SaveFileParser (
    std::string const & sourceFile )
```

6.27.3.2. ~SaveFileParser()

```
pftd::utils::parser::SaveFileParser::~~SaveFileParser ( ) [default]
```

6.27.4. Tagfüggvények dokumentációja

6.27.4.1. getEntities()

```
std::vector< EntityInfo > pftd::utils::parser::SaveFileParser::getEntities ( ) const
```

Mentett entitások beolvasása.

Visszatérési érték

Ezeket az entitásokat (azoknak a lényeges információját) tartalmazó lista.

6.27.4.2. getStats()

```
StatsInfo pftd::utils::parser::SaveFileParser::getStats ( ) const
```

Mentett statisztika beolvasása: pont, pénz, HP.

Visszatérési érték

A mentett statisztikát tartalmazó struktúra.

6.27.4.3. parse()

```
void pftd::utils::parser::SaveFileParser::parse ( ) [override], [virtual]
```

A parse főciklus.

Megvalósítja a következőket: [pftd::utils::parser::Parser](#).

6.27.5. Adatmezők dokumentációja

6.27.5.1. m_entities

```
std::vector<EntityInfo> pftd::utils::parser::SaveFileParser::m_entities [private]
```

A beolvasott entitás infók.

6.27.5.2. m_readStats

```
StatsInfo pftd::utils::parser::SaveFileParser::m_readStats [private]
```

A beolvasott statisztika.

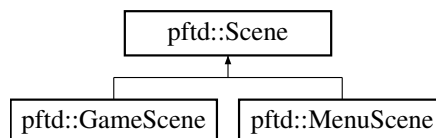
Ez a dokumentáció az osztályról a következő fájl alapján készült:

- include/utils/parsers.hpp

6.28. pftd::Scene osztályreferencia

```
#include <scene.hpp>
```

A pftd::Scene osztály származási diagramja:



Publikus tagfüggvények

- `Scene()`=default
- `Scene(Scene const &)=delete`
- `Scene(Scene &&) noexcept=delete`
- `virtual ~Scene()`
Törli a kezelt objektumokat.
- `ObjPtrVec const &getObjects() const`
objects getter (konstans).
- `ObjPtrVec &getObjects()`
objects getter.
- `void setMusic(std::string const &source, float volume=100.0f)`
Zene beállítása (loopen megy amíg aktív a nézet).
- `virtual bool toggleActive(bool active)`
Nézet aktiválása/deaktiválása.
- `virtual void onEvent(sf::Event const &event)=0`
Delegált események kezelése.
- `virtual void update(float dt)=0`
Update.

Védett attribútumok

- bool `isActive` = false
- `ObjPtrVec` objects
- `sf::Music * backgroundMusic` = nullptr

Privát típusok

- using `ObjPtrVec` = `std::vector< Object * >`

6.28.1. Részletes leírás

Nézet ősosztály.

6.28.2. Típusdefiníció-tagok dokumentációja

6.28.2.1. ObjPtrVec

```
using pftd::Scene::ObjPtrVec = std::vector<Object*> [private]
```

6.28.3. Konstruktorok és destruktorok dokumentációja

6.28.3.1. Scene() [1/3]

```
pftd::Scene::Scene ( ) [default]
```

6.28.3.2. Scene() [2/3]

```
pftd::Scene::Scene (
    Scene const & ) [delete]
```

6.28.3.3. Scene() [3/3]

```
pftd::Scene::Scene (
    Scene && ) [delete], [noexcept]
```

6.28.3.4. ~Scene()

```
virtual pftd::Scene::~~Scene ( ) [virtual]
```

Törli a kezelt objektumokat.

6.28.4. Tagfüggvények dokumentációja

6.28.4.1. getObjects() [1/2]

```
ObjPtrVec & pftd::Scene::getObjects ( ) [inline]
```

`objects` getter.

Visszatérési érték

A kezelt elemek listája.

6.28.4.2. getObjects() [2/2]

```
ObjPtrVec const & pftd::Scene::getObjects ( ) const [inline]
```

`objects` getter (konstans).

Visszatérési érték

A kezelt elemek listája.

6.28.4.3. onEvent()

```
virtual void pftd::Scene::onEvent (
    sf::Event const & event ) [pure virtual]
```

Delegált események kezelése.

Paraméterek

<i>event</i>	Kezelni kívánt esemény.
--------------	-------------------------

Megvalósítják a következők: [pftd::GameScene](#) és [pftd::MenuScene](#).

6.28.4.4. setMusic()

```
void pftd::Scene::setMusic (
    std::string const & source,
    float volume = 100.0f )
```

Zene beállítása (loopen megy amíg aktív a nézet).

Paraméterek

<i>source</i>	A hangfájl elérési útvonala.
<i>volume</i>	A zene hangereje.

6.28.4.5. toggleActive()

```
virtual bool pftd::Scene::toggleActive (
    bool active ) [virtual]
```

Nézet aktiválása/deaktiválása.

Paraméterek

<i>active</i>	Aktiválás e?
---------------	--------------

Visszatérési érték

Történt e állapot változás.

Újrimplementáló leszármazottak: [pftd::GameScene](#).

6.28.4.6. update()

```
virtual void pftd::Scene::update (
    float dt ) [pure virtual]
```

Update.

Paraméterek

<i>dt</i>	Delta idő.
-----------	------------

Megvalósítják a következők: [pftd::GameScene](#) és [pftd::MenuScene](#).

6.28.5. Adatmezők dokumentációja

6.28.5.1. backgroundMusic

```
sf::Music* pftd::Scene::backgroundMusic = nullptr [protected]
```

Háttérzene.

6.28.5.2. isActive

```
bool pftd::Scene::isActive = false [protected]
```

Aktív e.

6.28.5.3. objects

```
ObjPtrVec pftd::Scene::objects [protected]
```

Kezelt objektumok.

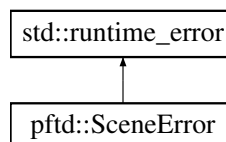
Ez a dokumentáció az osztályról a következő fájl alapján készült:

- [include/scene.hpp](#)

6.29. pftd::SceneError struktúrareferencia

```
#include <app.hpp>
```

A pftd::SceneError osztály származási diagramja:



6.29.1. Részletes leírás

Nézethiba.

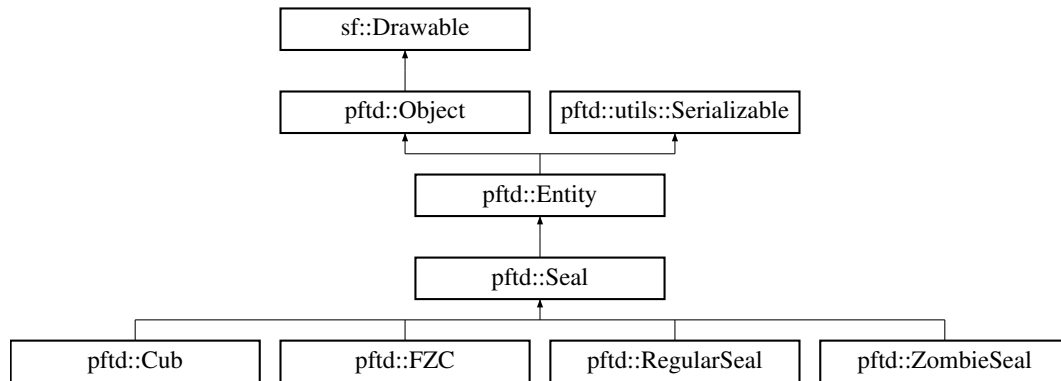
Ez a dokumentáció a struktúráról a következő fájl alapján készült:

- [include/app.hpp](#)

6.30. pftd::Seal osztályreferencia

```
#include <seal_base.hpp>
```

A pftd::Seal osztály származási diagramja:



Publikus tagfüggvények

- virtual `~Seal()` = default
- void `lerpPath()`
LERP paraméter frissítése.
- virtual `Seal * clone()` const override=0
Egy dinamikus memóriában foglalt másolatot készít a főkából.
- void `update(float dt)` override
Update.
- bool `hasCompletedPath()` const
returned getter.
- bool `hasReachedNest()` const
reachedNest getter.
- void `damage(int hpLost=1)`
Lesebzés.

Adatmezők

- bool `isCurrentlyStealing` = false
- int `hp`
- float `speed`
- unsigned int `value`

Védett tagfüggvények

- `Seal(FollowPath const &followPath, std::string const &spriteSrc, utils::Vec2f const &size, int hp, float speed, unsigned int value, int zIndex=0)`

Védett attribútumok

- `FollowPath` const & `followPath`
- float `lerpParam` = 0.0f
- bool `reachedNest` = false
- bool `returned` = false
- `EndPoint` `nextPoint`

Privát tagfüggvények

- void `advanceAnimationFrame` () override
Animáció: képkocka léptetése.

6.30.1. Részletes leírás

Fóka.

6.30.2. Konstruktorkok és destruktorkok dokumentációja

6.30.2.1. ~Seal()

```
virtual pftd::Seal::~~Seal ( ) [virtual], [default]
```

6.30.2.2. Seal()

```
pftd::Seal::Seal (
    FollowPath const & followPath,
    std::string const & spriteSrc,
    utils::Vec2f const & size,
    int hp,
    float speed,
    unsigned int value,
    int zIndex = 0 ) [protected]
```

Paraméterek

<i>followPath</i>	Az útvonal, amit követnie kell.
<i>spriteSrc</i>	A sprite-jának az elérési útvonala.
<i>size</i>	Méret.
<i>hp</i>	HP.
<i>speed</i>	Gyorsaság.
<i>value</i>	Érték: ennyi pénzt kap a játékos ha megöli.
<i>zIndex</i>	Z koordináta.

6.30.3. Tagfüggvények dokumentációja

6.30.3.1. advanceAnimationFrame()

```
void pftd::Seal::advanceAnimationFrame ( ) [override], [private], [virtual]
```

Animáció: képkocka léptetése.

Újrimplementált ősök: [pftd::Entity](#).

6.30.3.2. clone()

```
virtual Seal * pftd::Seal::clone ( ) const [override], [pure virtual]
```

Egy dinamikus memóriában foglalt másolatot készít a fókából.

Visszatérési érték

A másolat.

Megvalósítja a következőket: [pftd::Entity](#).

Megvalósítják a következők: [pftd::Cub](#), [pftd::FZC](#), [pftd::RegularSeal](#) és [pftd::ZombieSeal](#).

6.30.3.3. damage()

```
void pftd::Seal::damage (
    int hpLost = 1 )
```

Lesebzés.

Paraméterek

<i>hpLost</i>	Ennyi HP-ja fog lemenni a fókának.
---------------	------------------------------------

6.30.3.4. hasCompletedPath()

```
bool pftd::Seal::hasCompletedPath ( ) const [inline]
```

returned getter.

Visszatérési érték

Elment e a fészekig és vissza a kezdőhelyre.

6.30.3.5. hasReachedNest()

```
bool pftd::Seal::hasReachedNest ( ) const [inline]

reachedNest getter.
```

Visszatérési érték

Elment e a fészekig.

6.30.3.6. lerpPath()

```
void pftd::Seal::lerpPath ( )
```

LERP paraméter frissítése.

A `followPath` pontjai közt lineárisan interpolál.

6.30.3.7. update()

```
void pftd::Seal::update (
    float dt ) [override], [virtual]
```

Update.

Paraméterek

<i>dt</i>	Delta idő.
-----------	------------

Újraimplementált ősök: [pftd::Entity](#).

6.30.4. Adatmezők dokumentációja**6.30.4.1. followPath**

```
FollowPath const& pftd::Seal::followPath [protected]
```

Az útvonal amin végigmegy.

6.30.4.2. hp

```
int pftd::Seal::hp
```

HP.

6.30.4.3. isCurrentlyStealing

```
bool pftd::Seal::isCurrentlyStealing = false
```

Éppen lopás közben van e?

6.30.4.4. lerpParam

```
float pftd::Seal::lerpParam = 0.0f [protected]
```

0-tól 1-ig terjedő paraméter az interpolációhoz.

6.30.4.5. nextPoint

```
EndPoint pftd::Seal::nextPoint [protected]
```

A következő pont az útvonalon.

6.30.4.6. reachedNest

```
bool pftd::Seal::reachedNest = false [protected]
```

Elérte e a fészket.

6.30.4.7. returned

```
bool pftd::Seal::returned = false [protected]
```

Visszatért e a kezdő ponthoz.

6.30.4.8. speed

```
float pftd::Seal::speed
```

Gyorsaság.

6.30.4.9. value

```
unsigned int pftd::Seal::value
```

Pénz amit ad halál után.

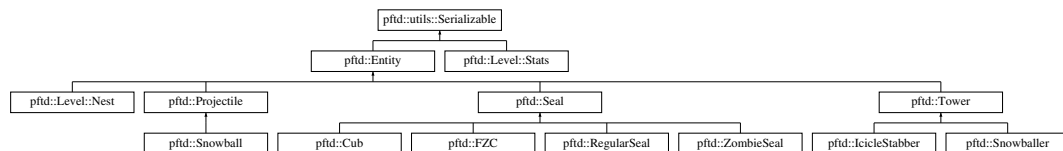
Ez a dokumentáció az osztályról a következő fájl alapján készült:

- include/objects/entities/seals/[seal_base.hpp](#)

6.31. pftd::utils::Serializable osztályreferencia

```
#include <serializable.hpp>
```

A pftd::utils::Serializable osztály származási diagramja:



Publikus tagfüggvények

- virtual void [serialize](#) (std::ostream &out)=0
Kiír egy streamre.
- virtual void [deserialize](#) (std::istream &in)=0
Beolvas egy streamről.

6.31.1. Részletes leírás

Szerializálható osztályok.

6.31.2. Tagfüggvények dokumentációja

6.31.2.1. deserialize()

```
virtual void pftd::utils::Serializable::deserialize (
    std::istream & in ) [pure virtual]
```

Beolvas egy streamről.

Paraméterek

<i>in</i>	A stream.
-----------	-----------

Megvalósítják a következők: [pftd::Level::Stats](#).

6.31.2.2. serialize()

```
virtual void pftd::utils::Serializable::serialize (
    std::ostream & out ) [pure virtual]
```

Kiír egy streamre.

Paraméterek

<i>out</i>	A stream.
------------	-----------

Megvalósítják a következők: [pftd::Level::Stats](#).

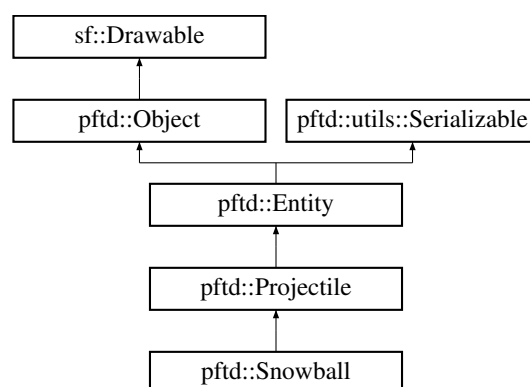
Ez a dokumentáció az osztályról a következő fájl alapján készült:

- include/utis/[serializable.hpp](#)

6.32. pftd::Snowball osztályreferencia

```
#include <snowball.hpp>
```

A pftd::Snowball osztály származási diagramja:



Publikus tagfüggvények

- [Snowball](#) ([utils::Vec2f](#) const &[position](#), [utils::Vec2f](#) [direction](#), float speed=50.0f)
A textúra hard code-olt.
- [Entity](#) * [clone](#) () const override
Egy dinamikus memóriában foglalt másolatot készít az entitásról.

További örökölt tagok

6.32.1. Részletes leírás

Hógolyó lövedék.

6.32.2. Konstruktorok és destruktorok dokumentációja

6.32.2.1. Snowball()

```
pftd::Snowball::Snowball (
    utils::Vec2f const & position,
    utils::Vec2f direction,
    float speed = 50.0f )
```

A textúra hard code-olt.

Paraméterek

<i>position</i>	Pozíció.
<i>direction</i>	Írány.
<i>speed</i>	Repülési gyorsaság.

6.32.3. Tagfüggvények dokumentációja

6.32.3.1. clone()

```
Entity * pftd::Snowball::clone ( ) const [override], [virtual]
```

Egy dinamikus memóriában foglalt másolatot készít az entitásról.

Visszatérési érték

A másolat.

Megvalósítja a következőket: [pftd::Entity](#).

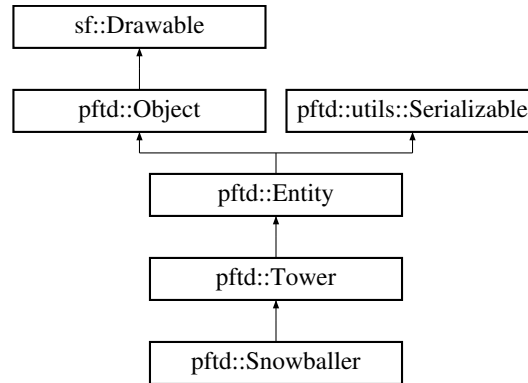
Ez a dokumentáció az osztályról a következő fájl alapján készült:

- [include/objects/entities/projectiles/snowball.hpp](#)

6.33. pftd::Snowballer osztályreferencia

```
#include <snowballer.hpp>
```

A pftd::Snowballer osztály származási diagramja:



Publikus tagfüggvények

- `Snowballer (utils::Vec2f const &position, int zIndex=0)`
- `Snowballer ()`
- `~Snowballer ()=default`
- `Tower * clone () const` override
Egy dinamikus memóriában foglalt másolatot készít a toronyról.
- `void update (float dt)` override
Update.
- `void attack ()` override
Megtámadja a `Tower::target`-et, ha van.

További örökölt tagok

6.33.1. Részletes leírás

Hógolyó dobáló.

6.33.2. Konstruktorkok és destruktorkok dokumentációja

6.33.2.1. Snowballer() [1/2]

```

pftd::Snowballer::Snowballer (
    utils::Vec2f const & position,
    int zIndex = 0 )
  
```

6.33.2.2. Snowballer() [2/2]

```
pftd::Snowballer::Snowballer ( )
```

6.33.2.3. ~Snowballer()

```
pftd::Snowballer::~~Snowballer ( ) [default]
```

6.33.3. Tagfüggvények dokumentációja

6.33.3.1. attack()

```
void pftd::Snowballer::attack ( ) [override], [virtual]
```

Megtámadja a `Tower::target`-et, ha van.

Újraimplementált ősök: `pftd::Tower`.

6.33.3.2. clone()

```
Tower * pftd::Snowballer::clone ( ) const [override], [virtual]
```

Egy dinamikus memóriában foglalt másolatot készít a toronyról.

Visszatérési érték

A másolat.

Megvalósítja a következőket: `pftd::Tower`.

6.33.3.3. update()

```
void pftd::Snowballer::update (
    float dt ) [override], [virtual]
```

Update.

Paraméterek

<i>dt</i>	Delta idő.
-----------	------------

Újraimplementált ősök: [pftd::Tower](#).

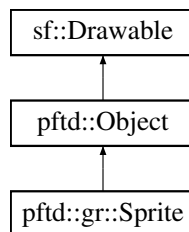
Ez a dokumentáció az osztályról a következő fájl alapján készült:

- `include/objects/entities/towers/snowballer.hpp`

6.34. pftd::gr::Sprite osztályreferencia

```
#include <image.hpp>
```

A pftd::gr::Sprite osztály származási diagramja:



Publikus tagfüggvények

- [Sprite](#) (sf::Texture const &texture, [utils::Vec2f](#) const &position, [utils::Vec2f](#) const &size, int zIndex=-1)
- [Sprite](#) (sf::Texture const &texture, sf::IntRect const &textureRect, [utils::Vec2f](#) const &position, [utils::Vec2f](#) const &size, int zIndex=-1)
- [Sprite](#) (std::string const &imageSrc, [utils::Vec2f](#) const &position, [utils::Vec2f](#) const &size, int zIndex=-1)
- [Sprite](#) ([Sprite](#) const &other)
- [Sprite](#) ([Sprite](#) &&other) noexcept
- [~Sprite](#) ()=default
- void [modColor](#) (sf::Color const &color)
[Sprite](#) szín modulációjának beállítása.
- void [flipY](#) ()
Y tengely mentén tükröz.
- void [setSpriteRect](#) (sf::IntRect const &textureRect)
A textúra csak egy részletének használata [Sprite](#)-ként.
- void [draw](#) (sf::RenderTarget &target, sf::RenderStates states) const override
Objektum megjelenítése.

Privát attribútumok

- sf::Texture const & [m_texture](#)
- sf::Sprite [m_sprite](#)

További örökölt tagok

6.34.1. Részletes leírás

Saját sprite osztály. Használható kép GUI elemként.

6.34.2. Konstruktorok és destruktorok dokumentációja

6.34.2.1. Sprite() [1/5]

```
pftd::gr::Sprite::Sprite (
    sf::Texture const & texture,
    utils::Vec2f const & position,
    utils::Vec2f const & size,
    int zIndex = -1 )
```

6.34.2.2. Sprite() [2/5]

```
pftd::gr::Sprite::Sprite (
    sf::Texture const & texture,
    sf::IntRect const & textureRect,
    utils::Vec2f const & position,
    utils::Vec2f const & size,
    int zIndex = -1 )
```

6.34.2.3. Sprite() [3/5]

```
pftd::gr::Sprite::Sprite (
    std::string const & imageSrc,
    utils::Vec2f const & position,
    utils::Vec2f const & size,
    int zIndex = -1 )
```

6.34.2.4. Sprite() [4/5]

```
pftd::gr::Sprite::Sprite (
    Sprite const & other )
```


6.34.2.5. Sprite() [5/5]

```
pftd::gr::Sprite::Sprite (
    Sprite && other ) [noexcept]
```

6.34.2.6. ~Sprite()

```
pftd::gr::Sprite::~~Sprite ( ) [default]
```

6.34.3. Tagfüggvények dokumentációja

6.34.3.1. draw()

```
void pftd::gr::Sprite::draw (
    sf::RenderTarget & target,
    sf::RenderStates states ) const [override], [virtual]
```

Objektum megjelenítése.

Paraméterek

<i>target</i>	Hol legyen megjelenítve? Ez általában egy ablak.
<i>states</i>	Render-eléshez szükséges egyéb állapotok.

Megvalósítja a következőket: [pftd::Object](#).

6.34.3.2. flipY()

```
void pftd::gr::Sprite::flipY ( )
```

Y tengely mentén tükröz.

6.34.3.3. modColor()

```
void pftd::gr::Sprite::modColor (
    sf::Color const & color )
```

[Sprite](#) szín modulációjának beállítása.

Megjegyzés: Az SFML által biztosított modulációt használja.

Paraméterek

<i>color</i>	Szín, amivel szorzunk.
--------------	------------------------

6.34.3.4. setSpriteRect()

```
void pftd::gr::Sprite::setSpriteRect (
    sf::IntRect const & textureRect )
```

A textúra csak egy részletének használata Sprite-ként.

Paraméterek

<i>textureRect</i>	A tárolt textúra része egy téglalappal megadva.
--------------------	---

6.34.4. Adatmezők dokumentációja**6.34.4.1. m_sprite**

```
sf::Sprite pftd::gr::Sprite::m_sprite [private]
```

[Sprite](#).

6.34.4.2. m_texture

```
sf::Texture const& pftd::gr::Sprite::m_texture [private]
```

Sprite-hoz tartozó textúra.

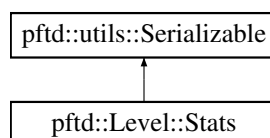
Ez a dokumentáció az osztályról a következő fájl alapján készült:

- `include/objects/gui/image.hpp`

6.35. pftd::Level::Stats struktúrareferencia

```
#include <level.hpp>
```

A pftd::Level::Stats osztály származási diagramja:



Publikus tagfüggvények

- [Stats](#) ()
- [Stats](#) (int maxHp, int currentHp, unsigned int [score](#), unsigned int wealth)
- void [serialize](#) (std::ostream &out) override
Kiír egy streamre.
- void [deserialize](#) (std::istream &in) override
Beolvas egy streamről.

Adatmezők

- int const [MAX_HP](#) = 3
- int [hp](#) = [MAX_HP](#)
- unsigned int [score](#) = 0U
- unsigned int [money](#) = 100U

6.35.1. Részletes leírás

Játékállás / statisztika.

6.35.2. Konstruktorkok és destruktorkok dokumentációja

6.35.2.1. Stats() [1/2]

```
pftd::Level::Stats::Stats ( ) [explicit]
```

6.35.2.2. Stats() [2/2]

```
pftd::Level::Stats::Stats (
    int maxHp,
    int currentHp,
    unsigned int score,
    unsigned int wealth ) [explicit]
```

6.35.3. Tagfüggvények dokumentációja

6.35.3.1. deserialize()

```
void pftd::Level::Stats::deserialize (
    std::istream & in ) [override], [virtual]
```

Beolvas egy streamről.

Paraméterek

<i>in</i>	A stream.
-----------	-----------

Megvalósítja a következőket: [pftd::utils::Serializable](#).

6.35.3.2. serialize()

```
void pftd::Level::Stats::serialize (
    std::ostream & out ) [override], [virtual]
```

Kiír egy streamre.

Paraméterek

<i>out</i>	A stream.
------------	-----------

Megvalósítja a következőket: [pftd::utils::Serializable](#).

6.35.4. Adatmezők dokumentációja**6.35.4.1. hp**

```
int pftd::Level::Stats::hp = MAX_HP
```

Jelenlegi HP.

6.35.4.2. MAX_HP

```
int const pftd::Level::Stats::MAX_HP = 3
```

A maximum HP.

6.35.4.3. money

```
unsigned int pftd::Level::Stats::money = 100U
```

Pénz.

6.35.4.4. score

```
unsigned int pftd::Level::Stats::score = 0U
```

Pontszám.

Ez a dokumentáció a struktúráról a következő fájl alapján készült:

- include/game/[level.hpp](#)

6.36. pftd::utils::parser::SaveFileParser::StatsInfo struktúrareferencia

```
#include <parsers.hpp>
```

Adatmezők

- unsigned int [score](#)
- unsigned int [wealth](#)
- unsigned int [hp](#)

6.36.1. Részletes leírás

Mentett játék betöltéséhez használt statisztika osztály.

6.36.2. Adatmezők dokumentációja

6.36.2.1. hp

```
unsigned int pftd::utils::parser::SaveFileParser::StatsInfo::hp
```

6.36.2.2. score

```
unsigned int pftd::utils::parser::SaveFileParser::StatsInfo::score
```

6.36.2.3. wealth

```
unsigned int pftd::utils::parser::SaveFileParser::StatsInfo::wealth
```

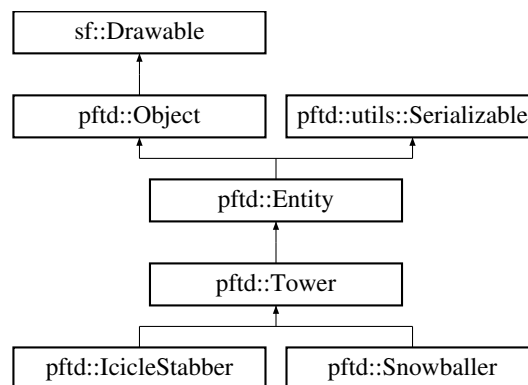
Ez a dokumentáció a struktúráról a következő fájl alapján készült:

- include/utils/parsers.hpp

6.37. pftd::Tower osztályreferencia

```
#include <tower_base.hpp>
```

A pftd::Tower osztály származási diagramja:



Publikus tagfüggvények

- **Tower** (std::string const &spriteSheetSrc, utils::Vec2i spriteSize, float attackSpeedSec, utils::Vec2f const &position, utils::Vec2f const &size, int zIndex=0)
- **Tower** (std::string const &spriteSrc, float attackSpeedSec, utils::Vec2f const &position, utils::Vec2f const &size, int zIndex=0)
- **Tower** (Tower const &)
- virtual ~Tower ()=default
- void setProjSpawnCb (ProjSpawnFunc callback)
Lövedék kilövéséhez használt callback függvény beállítása.
- virtual Tower * clone () const override=0
Egy dinamikus memóriában foglalt másolatot készít a toronyról.
- virtual void attack ()
Megtámadja a target-et, ha van.
- virtual bool lookForTarget (std::vector< Seal * > const &enemies)
Megfelelő célpontot keres.
- virtual void update (float dt) override
Update.

Adatmezők

- float `radiusPixel` = 120.0f
- float `attackRangePixel` = 100.0f
- float `attackSpeedSec`
- unsigned int `attackDamage` = 1U
- unsigned int `price` = 0U
- bool `instantAttack` = false

Védett tagfüggvények

- void `advanceAnimationFrame` () override
Animáció: képkocka léptetése.

Védett attribútumok

- `Seal * target` = nullptr
- `ProjSpawnFunc spawnProjectile`
- float `attackTimerSec` = 0.0f

Privát típusok

- using `ProjSpawnFunc` = std::function< void(`Projectile` *)>

6.37.1. Részletes leírás

Torony (pingvin).

6.37.2. Típusdefiníció-tagok dokumentációja

6.37.2.1. ProjSpawnFunc

```
using pftd::Tower::ProjSpawnFunc = std::function<void(Projectile*)> [private]
```

6.37.3. Konstruktorkok és destruktorkok dokumentációja

6.37.3.1. Tower() [1/3]

```
pftd::Tower::Tower (
    std::string const & spriteSheetSrc,
    utils::Vec2i spriteSize,
    float attackSpeedSec,
    utils::Vec2f const & position,
    utils::Vec2f const & size,
    int zIndex = 0 )
```

6.37.3.2. Tower() [2/3]

```
pftd::Tower::Tower (
    std::string const & spriteSrc,
    float attackSpeedSec,
    utils::Vec2f const & position,
    utils::Vec2f const & size,
    int zIndex = 0 )
```

6.37.3.3. Tower() [3/3]

```
pftd::Tower::Tower (
    Tower const & )
```

6.37.3.4. ~Tower()

```
virtual pftd::Tower::~~Tower ( ) [virtual], [default]
```

6.37.4. Tagfüggvények dokumentációja

6.37.4.1. advanceAnimationFrame()

```
void pftd::Tower::advanceAnimationFrame ( ) [override], [protected], [virtual]
```

Animáció: képkocka léptetése.

Újrimplementált ősök: [pftd::Entity](#).

6.37.4.2. attack()

```
virtual void pftd::Tower::attack ( ) [virtual]
```

Megtámadja a `target`-et, ha van.

Újrimplementáló leszármazottak: [pftd::Snowballer](#).

6.37.4.3. clone()

```
virtual Tower * pftd::Tower::clone ( ) const [override], [pure virtual]
```

Egy dinamikus memóriában foglalt másolatot készít a toronyról.

Visszatérési érték

A másolat.

Megvalósítja a következőket: [pftd::Entity](#).

Megvalósítják a következők: [pftd::IcicleStabber](#) és [pftd::Snowballer](#).

6.37.4.4. lookForTarget()

```
virtual bool pftd::Tower::lookForTarget (
    std::vector< Seal * > const & enemies ) [virtual]
```

Megfelelő célpontot keres.

Megjegyzés: csak akkor keres, ha a `target` egy nullptr.

Paraméterek

<i>enemies</i>	Ellenfelek, amik közt célpontot keres.
----------------	--

6.37.4.5. setProjSpawnCb()

```
void pftd::Tower::setProjSpawnCb (
    ProjSpawnFunc callback )
```

Lövedék kilövéséhez használt callback függvény beállítása.

Paraméterek

<i>callback</i>	A függvény.
-----------------	-------------

6.37.4.6. update()

```
virtual void pftd::Tower::update (  
    float dt ) [override], [virtual]
```

Update.

Paraméterek

<i>dt</i>	Delta idő.
-----------	------------

Újrimplementált ősök: [pftd::Entity](#).

Újrimplementáló leszármazottak: [pftd::IcicleStabber](#) és [pftd::Snowballer](#).

6.37.5. Adatmezők dokumentációja**6.37.5.1. attackDamage**

```
unsigned int pftd::Tower::attackDamage = 1U
```

Támadás ereje: mennyi HP-t visz le.

6.37.5.2. attackRangePixel

```
float pftd::Tower::attackRangePixel = 100.0f
```

Az a környezete, amiben célpontot keres.

6.37.5.3. attackSpeedSec

```
float pftd::Tower::attackSpeedSec
```

Támadás gyorsasága.

6.37.5.4. attackTimerSec

```
float pftd::Tower::attackTimerSec = 0.0f [protected]
```

Támadások közti idő mérésére szolgáló számláló.

6.37.5.5. instantAttack

```
bool pftd::Tower::instantAttack = false
```

Instant támadó (true)? Vagy lövedékkel (false)?

6.37.5.6. price

```
unsigned int pftd::Tower::price = 0U
```

Az ára, amikor az "eszköztárban" van.

6.37.5.7. radiusPixel

```
float pftd::Tower::radiusPixel = 120.0f
```

Az a környezete, amibe másik tornyot nem lehet lehelyezni.

6.37.5.8. spawnProjectile

```
ProjSpawnFunc pftd::Tower::spawnProjectile [protected]
```

Callback függvény, ha a torony távolsági (valamit kilő).

6.37.5.9. target

```
Seal* pftd::Tower::target = nullptr [protected]
```

Célpont.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- [include/objects/entities/towers/tower_base.hpp](#)

6.38. pftd::utils::Vec2< T > struktúrasablon-referencia

```
#include <substitute_types.hpp>
```

Publikus tagfüggvények

- `Vec2 ()`
- `Vec2 (T x, T y)`
- `Vec2 (Vec2 const &right)`
- `Vec2 normalize () const`
Vektor normalizálása.
- `Vec2 operator+ (Vec2 const &right) const`
- `Vec2 operator- (Vec2 const &right) const`
- `Vec2 operator/ (T scalar) const`
- `Vec2 operator* (T scalar) const`
- `bool operator== (Vec2 other) const`
- `bool operator!= (Vec2 other) const`
- `Vec2 & operator= (Vec2 const &right)`

Statikus publikus tagfüggvények

- `static float distance (Vec2 const &v1, Vec2 const &v2)`
Két vektor végpontja közti euklideszi távolság.

Adatmezők

- `T x`
- `T y`

Barátok

- `std::ostream & operator<< (std::ostream &out, Vec2 const &vec)`

6.38.1. Részletes leírás

```
template<typename T>
struct pftd::utils::Vec2< T >
```

`sf::Vector2` helyett használt matematikai vektor implementáció.

6.38.2. Konstruktorkok és destruktorkok dokumentációja

6.38.2.1. Vec2() [1/3]

```
template<typename T >
pftd::utils::Vec2< T >::Vec2 ( ) [inline]
```

6.38.2.2. Vec2() [2/3]

```
template<typename T >
pftd::utils::Vec2< T >::Vec2 (
    T x,
    T y ) [inline]
```

6.38.2.3. Vec2() [3/3]

```
template<typename T >
pftd::utils::Vec2< T >::Vec2 (
    Vec2< T > const & right ) [inline]
```

6.38.3. Tagfüggvények dokumentációja**6.38.3.1. distance()**

```
template<typename T >
static float pftd::utils::Vec2< T >::distance (
    Vec2< T > const & v1,
    Vec2< T > const & v2 ) [inline], [static]
```

Két vektor végpontja közti euklideszi távolság.

Paraméterek

<i>v1</i>	Egyik vektor.
<i>v2</i>	Másik vektor.

Visszatérési érték

A távolság.

6.38.3.2. normalize()

```
template<typename T >
Vec2 pftd::utils::Vec2< T >::normalize ( ) const [inline]
```

Vektor normalizálása.

Ez a metódus nem módosítja a Vec2-t, hanem egy újat hoz létre belőle.

Visszatérési érték

Egy új, normalizált Vec2.

6.38.3.3. operator"!=()

```
template<typename T >
bool pftd::utils::Vec2< T >::operator!= (
    Vec2< T > other ) const [inline]
```

6.38.3.4. operator*()

```
template<typename T >
Vec2 pftd::utils::Vec2< T >::operator* (
    T scalar ) const [inline]
```

6.38.3.5. operator+()

```
template<typename T >
Vec2 pftd::utils::Vec2< T >::operator+ (
    Vec2< T > const & right ) const [inline]
```

6.38.3.6. operator-()

```
template<typename T >
Vec2 pftd::utils::Vec2< T >::operator- (
    Vec2< T > const & right ) const [inline]
```

6.38.3.7. operator/()

```
template<typename T >
Vec2 pftd::utils::Vec2< T >::operator/ (
    T scalar ) const [inline]
```

6.38.3.8. operator=()

```
template<typename T >
Vec2 & pftd::utils::Vec2< T >::operator= (
    Vec2< T > const & right ) [inline]
```

6.38.3.9. operator==()

```
template<typename T >
bool pftd::utils::Vec2< T >::operator==(
    Vec2< T > other ) const [inline]
```

6.38.4. Barát és kapcsolódó függvények dokumentációja

6.38.4.1. operator<<

```
template<typename T >
std::ostream & operator<< (
    std::ostream & out,
    Vec2< T > const & vec ) [friend]
```

6.38.5. Adatmezők dokumentációja

6.38.5.1. x

```
template<typename T >
T pftd::utils::Vec2< T >::x
```

X koordináta.

6.38.5.2. y

```
template<typename T >
T pftd::utils::Vec2< T >::y
```

Y koordináta.

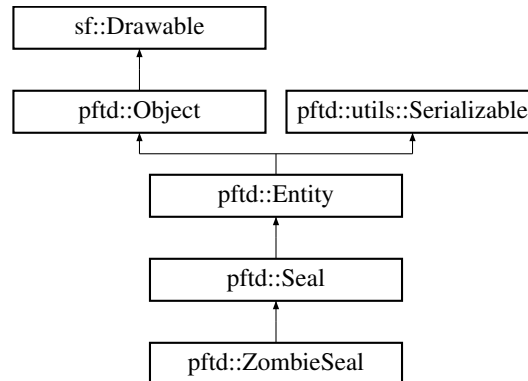
Ez a dokumentáció a struktúráról a következő fájl alapján készült:

- include/utils/[substitute_types.hpp](#)

6.39. pftd::ZombieSeal osztályreferencia

```
#include <zombie.hpp>
```

A pftd::ZombieSeal osztály származási diagramja:



Publikus tagfüggvények

- [ZombieSeal](#) ([FollowPath](#) const &followpath)
- [Seal](#) * [clone](#) () const override

Egy dinamikus memóriában foglalt másolatot készít a főkából.

További örökölt tagok

6.39.1. Részletes leírás

Zombi fóka.

6.39.2. Konstruktorkok és destruktorkok dokumentációja

6.39.2.1. ZombieSeal()

```
pftd::ZombieSeal::ZombieSeal (
    FollowPath const & followpath )
```

Paraméterek

<i>followpath</i>	Az útvonal amin végig kell menjen (majd pedig vissza).
-------------------	--

6.39.3. Tagfüggvények dokumentációja

6.39.3.1. clone()

```
Seal * pftd::ZombieSeal::clone ( ) const [override], [virtual]
```

Egy dinamikus memóriában foglalt másolatot készít a fókából.

Visszatérési érték

A másolat.

Megvalósítja a következőket: [pftd::Seal](#).

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- [include/objects/entities/seals/zombie.hpp](#)

7. fejezet

Fájlok dokumentációja

7.1. include/all.hpp fájlreferencia

```
#include <cmath>
#include <vector>
#include <functional>
#include <optional>
#include <utility>
#include <unordered_map>
#include <map>
#include <queue>
#include <algorithm>
#include <ctime>
#include <ios>
#include <limits>
#include <filesystem>
#include <stdexcept>
#include <random>
#include <fstream>
#include "SFML/Graphics.hpp"
#include "SFML/Audio.hpp"
#include "utils/logger.hpp"
#include "utils/substitute_types.hpp"
```

7.2. all.hpp

[Ugrás a fájl dokumentációjához.](#)

```
1 #pragma once
2
3 // Ez egy precompiled header fájl.
4
5 // STL
6 #include <cmath>
7 #include <vector>
8 #include <functional>
9 #include <optional>
10 #include <utility>
11 #include <unordered_map>
12 #include <map>
13 #include <queue>
14 #include <algorithm>
15 #include <ctime>
```

```
16 #include <ios>
17 #include <limits>
18 #include <filesystem>
19 #include <stdexcept>
20 #include <random>
21 #include <fstream>
22
23 // SFML
24 #if not defined(_PFTD_TEST) && not defined(CPORTA)
25 #include "SFML/Graphics.hpp"
26 #include "SFML/Audio.hpp"
27 #endif
28
29 // Own
30 #include "utils/logger.hpp"
31 #include "utils/substitute_types.hpp"
```

7.3. include/app.hpp fájlreferencia

```
#include "objects/object_base.hpp"
#include "scene.hpp"
```

Adatszerkezetek

- class [pftd::Renderer](#)
- class [pftd::App](#)
- struct [pftd::SceneError](#)

Névterek

- namespace [pftd](#)

7.4. app.hpp

[Ugrás a fájl dokumentációjához.](#)

```
1 #pragma once
2
3 #include "objects/object_base.hpp"
4 #include "scene.hpp"
5
6 namespace pftd {
7
8     class Renderer final
9     {
10     public:
11         friend class App;
12         Renderer(unsigned int width, unsigned int height, std::string const& windowTitle);
13         Renderer(Renderer const&) = delete;
14         Renderer(Renderer&&) = delete;
15         ~Renderer();
16
17         void render();
18
19         void clear()const { m_window->clear(); }
20
21         void display()const { m_window->display(); }
22
23         void pushQueue(Object const* o) { m_queue.push(o); }
24
25         sf::RenderWindow* getWindow()const { return m_window; }
26
27     private:
```

```

61     sf::RenderWindow* m_window = nullptr;
62
64     unsigned int m_width;
65
67     unsigned int m_height;
68
70     std::priority_queue<Object const*, std::vector<Object const*>, Object::Compare> m_queue;
71
72 };
73
75 class App final
76 {
77 public:
78     App(App const&) = delete;
79     App(App&&) = delete;
80
86     ~App();
87
91     void run();
92
103     void addScene(std::string id, Scene* scene, bool active = false);
104
114     bool changeScene(std::string id);
115
121     auto isRunning()const { return this->m_running; }
122
128     int getWindowWidth()const { return this->m_renderer->m_width; }
129
135     int getWindowHeight()const { return this->m_renderer->m_height; }
136
137     App& operator=(App const&) = delete;
138
144     static App* getInstance() { return m_instance; }
145
156     static App* create(unsigned int width, unsigned int height, std::string const& windowTitle);
157
161     static void destroy() { delete m_instance; }
162
163 private:
164     // Az 'm_instance' létrehozásához van használva.
165     App() = default;
166
168     static App* m_instance;
169
171     bool m_running = false;
172
174     Renderer* m_renderer = nullptr;
175
177     std::map<std::string, Scene*> m_scenes;
178
180     std::string m_activeSceneID;
181
182 };
183
185 struct SceneError : public std::runtime_error
186 {
187     using std::runtime_error::runtime_error;
188 };
189
190 }

```

7.5. include/game/level.hpp fájlreferencia

```

#include "all.hpp"
#include "objects/entities/projectiles/projectile_base.hpp"
#include "objects/entities/seals/seal_base.hpp"
#include "objects/entities/towers/tower_base.hpp"
#include "utils/serializable.hpp"
#include "utils/parsers.hpp"

```

Adatszerkezetek

- class `pftd::Level`
- struct `pftd::Level::Nest`
- struct `pftd::Level::Stats`

Névterek

- namespace `pftd`

7.6. level.hpp

[Ugrás a fájl dokumentációjához.](#)

```

1
2 #include "all.hpp"
3
4 #include "objects/entities/projectiles/projectile_base.hpp"
5 #include "objects/entities/seals/seal_base.hpp"
6 #include "objects/entities/towers/tower_base.hpp"
7 #include "utils/serializable.hpp"
8 #include "utils/parsers.hpp"
9
10 namespace pftd
11 {
12
13     class Level : public Object
14     {
15     public:
16         struct Nest : public Entity
17         {
18             int const radiusPixel = 110;
19
20             explicit Nest(utils::Vec2f const& position);
21             Nest(Nest const& other) = default;
22
23             ~Nest();
24         };
25
26         struct Stats : public utils::Serializable
27         {
28             int const MAX_HP = 3;
29
30             int hp = MAX_HP;
31
32             unsigned int score = 0U;
33
34             unsigned int money = 100U;
35
36             explicit Stats();
37             explicit Stats(int maxHp, int currentHp, unsigned int score, unsigned int wealth);
38
39             void serialize(std::ostream& out) override;
40             void deserialize(std::istream& in) override;
41         };
42
43         Stats stats;
44
45         Tower* selectedTower = nullptr;
46
47         Level(Stats stats = Stats{});
48         virtual ~Level();
49
50         virtual void loseHP(int hpLost = 1);
51
52         virtual bool placeTower();
53
54         void deselectTower();
55
56         void selectTower(Tower* newTower);
57
58         void update(float dt);
59
60         bool isGameOver() const;
61
62         virtual void draw(sf::RenderTarget& target, sf::RenderStates states) const override;
63     protected:
64         Nest* nest = nullptr;
65
66         FollowPath followPath;
67
68         std::vector<Tower*> towers;
69
70         std::vector<Seal*> seals;
71
72         std::vector<Projectile*> projectiles;

```

```
135
137     utils::parser::LevelConfigParser config;
138
142     void spawnSeal();
143
144 private:
146     float m_accuTimeSpawnSec = 0.0f;
147
148 };
149
150 }
```

7.7. include/objects/clickable.hpp fájlreferencia

```
#include "all.hpp"
#include "objects/object_base.hpp"
#include "utils/substitute_types.hpp"
```

Adatszerkezetek

- class [pftd::Clickable](#)

Névterek

- namespace [pftd](#)

7.8. clickable.hpp

[Ugrás a fájl dokumentációjához.](#)

```
1 #pragma once
2
3 #include "all.hpp"
4
5 #include "objects/object_base.hpp"
6 #include "utils/substitute_types.hpp"
7
8 namespace pftd
9 {
10
12 class Clickable : public Object
13 {
14 public:
16     bool isActive;
17
18     Clickable(bool active = true);
19     Clickable(utils::Vec2f const& position, utils::Vec2f const& size, int zIndex = 0, bool active =
20         true);
21
26     virtual void setCallback(std::function<void()> callback);
27
33     virtual void handleClick(utils::Vec2i const& clickCoords);
34
35 protected:
37     std::function<void()> m_callback;
38
39 };
40
41 ;
```

7.9. include/objects/entities/entity_base.hpp fájlreferencia

```
#include "all.hpp"
#include "objects/gui/image.hpp"
#include "objects/object_base.hpp"
#include "utils/serializable.hpp"
```

Adatszerkezetek

- class `pftd::Entity`

Névterek

- namespace `pftd`

7.10. entity_base.hpp

[Ugrás a fájl dokumentációjához.](#)

```
1 #pragma once
2
3 #include "all.hpp"
4
5 #include "objects/gui/image.hpp"
6 #include "objects/object_base.hpp"
7 #include "utils/serializable.hpp"
8
9 namespace pftd
10 {
11
12     class Entity : public Object, public utils::Serializable
13     {
14     public:
15         bool isAnimated = true;
16
17         Entity(std::string const& spriteSheetSrc, utils::Vec2i spriteSize, utils::Vec2f const& position,
18             utils::Vec2f const& size, int zIndex = 0);
19         Entity(sf::Texture const& texture, utils::Vec2i spriteSize, utils::Vec2f const& position,
20             utils::Vec2f const& size, int zIndex = 0);
21         Entity(std::string const& spriteSrc, utils::Vec2f const& position, utils::Vec2f const& size, int
22             zIndex = 0);
23         Entity(Entity const& other);
24         virtual ~Entity();
25
26         virtual Entity* clone() const = 0;
27
28         virtual void update(float dt);
29
30         virtual void draw(sf::RenderTarget& target, sf::RenderStates states) const override;
31
32         void setPosition(utils::Vec2f position);
33
34         auto getPosition()const { return position; }
35
36         gr::Sprite const* getSprite()const { return currentSprite; }
37
38         gr::Sprite* getSprite() { return currentSprite; }
39
40     protected:
41         sf::Texture const& spriteSheet; // Feltételezzük, hogy egy sorban vannak a sprite-ok
42
43         utils::Vec2i cellSize;
44
45         size_t const CELL_N;
46
47         size_t currentCell = 0U;
48
49         gr::Sprite* currentSprite = nullptr;
50
51     }
```



```
92     float frameDurationSec = 1.0f;
93
94     float totalElapsedSec = 0.0f; // Animációhoz van használva elsősorban.
95
96     virtual void advanceAnimationFrame();
97 };
98
99
100 }
```

7.11. include/objects/entities/projectiles/projectile_base.hpp fájlreferencia

```
#include "objects/entities/entity_base.hpp"
#include "utils/substitute_types.hpp"
```

Adatszerkezetek

- class `pftd::Projectile`

Névterek

- namespace `pftd`

7.12. projectile_base.hpp

[Ugrás a fájl dokumentációjához.](#)

```
1 #pragma once
2
3 #include "objects/entities/entity_base.hpp"
4 #include "utils/substitute_types.hpp"
5
6 namespace pftd
7 {
8
9     class Projectile : public Entity
10     {
11     public:
12         utils::Vec2f direction;
13
14         float linearSpeed;
15
16         float angularVelocityRadPerSec;
17
18         virtual ~Projectile() = default;
19
20         virtual void update(float dt) override;
21
22     protected:
23         Projectile(std::string const& spriteSrc, utils::Vec2f const& position,
24             utils::Vec2f const& size, utils::Vec2f const& direction, float speed, float angularSpeed = 0.0f,
25             int zIndex = 0);
26
27     };
28
29 }
```

7.13. include/objects/entities/projectiles/snowball.hpp fájlreferencia

```
#include "objects/entities/projectiles/projectile_base.hpp"
```

Adatszerkezetek

- class `pftd::Snowball`

Névterek

- namespace `pftd`

7.14. snowball.hpp

[Ugrás a fájl dokumentációjához.](#)

```
1 #pragma once
2
3 #include "objects/entities/projectiles/projectile_base.hpp"
4
5 namespace pftd {
6
7     class Snowball : public Projectile
8     {
9     public:
10         Snowball(utils::Vec2f const& position, utils::Vec2f direction, float speed = 50.0f);
11
12         Entity* clone() const override;
13
14     };
15 }
```

7.15. include/objects/entities/seals/cub.hpp fájlreferencia

```
#include "objects/entities/seals/seal_base.hpp"
```

Adatszerkezetek

- class `pftd::Cub`

Névterek

- namespace `pftd`

7.16. cub.hpp

[Ugrás a fájl dokumentációjához.](#)

```
1 #pragma once
2
3 #include "objects/entities/seals/seal_base.hpp"
4
5 namespace pftd {
6
7     class Cub : public Seal
8     {
9     public:
10         Cub(FollowPath const& followpath);
11
12         Seal* clone() const override;
13
14     };
15 }
```

7.17. include/objects/entities/seals/followpath.hpp fájlreferencia

```
#include "utils/hetero_collection.hpp"
```

Adatszerkezetek

- struct [pftd::FollowPath](#)

Névterek

- namespace [pftd](#)

Típusdefiníciók

- using [pftd::EndPoint](#) = [utils::Vec2f](#)

7.18. followpath.hpp

[Ugrás a fájl dokumentációjához.](#)

```
1 #pragma once
2
3 #include "utils/hetero_collection.hpp"
4
5 namespace pftd {
6
7     using EndPoint = utils::Vec2f;
8
9     struct FollowPath : public utils::Container<EndPoint>
10     {
11     public:
12         FollowPath();
13     };
14
15 }
```

7.19. include/objects/entities/seals/fortified_zombie_cub.hpp fájlreferencia

```
#include "objects/entities/seals/seal_base.hpp"
```

Adatszerkezetek

- class [pftd::FZC](#)

Névterek

- namespace [pftd](#)

7.20. fortified_zombie_cub.hpp

[Ugrás a fájl dokumentációjához.](#)

```
1 #pragma once
2
3 #include "objects/entities/seals/seal_base.hpp"
4
5 namespace pftd {
6
7     class FZC : public Seal
8     {
9     public:
10         FZC(FollowPath const& followpath);
11
12         Seal* clone() const override;
13
14     };
15
16 }
```

7.21. include/objects/entities/seals/regular.hpp fájlreferencia

```
#include "objects/entities/seals/seal_base.hpp"
```

Adatszerkezetek

- class [pftd::RegularSeal](#)

Névterek

- namespace [pftd](#)

7.22. regular.hpp

[Ugrás a fájl dokumentációjához.](#)

```
1 #pragma once
2
3 #include "objects/entities/seals/seal_base.hpp"
4
5 namespace pftd {
6
7     class RegularSeal : public Seal
8     {
9     public:
10         RegularSeal(FollowPath const& followpath);
11
12         Seal* clone() const override;
13
14     };
15
16 }
```

7.23. include/objects/entities/seals/seal_base.hpp fájlreferencia

```
#include "objects/entities/entity_base.hpp"
#include "objects/entities/seals/followpath.hpp"
#include "utils/substitute_types.hpp"
```

Adatszerkezetek

- class `pftd::Seal`

Névterek

- namespace `pftd`

7.24. seal_base.hpp

[Ugrás a fájl dokumentációjához.](#)

```

1 #pragma once
2
3 #include "objects/entities/entity_base.hpp"
4 #include "objects/entities/seals/followpath.hpp"
5 #include "utils/substitute_types.hpp"
6
7 namespace pftd
8 {
9
10 class Seal : public Entity
11 {
12 public:
13     bool isCurrentlyStealing = false;
14
15     int hp;
16
17     float speed;
18
19     unsigned int value;
20
21     virtual ~Seal() = default;
22
23     void lerpPath();
24
25     virtual Seal* clone() const override = 0;
26
27     void update(float dt) override;
28
29     bool hasCompletedPath()const { return returned; }
30
31     bool hasReachedNest()const { return reachedNest; }
32
33     void damage(int hpLost = 1);
34
35 protected:
36     Seal(FollowPath const& followPath, std::string const& spriteSrc, utils::Vec2f const& size, int hp,
37         float speed, unsigned int value, int zIndex = 0);
38
39     FollowPath const& followPath;
40
41     // Interpolációhoz
42
43     float lerpParam = 0.0f;
44
45     bool reachedNest = false;
46
47     bool returned = false;
48
49     EndPoint nextPoint;
50
51 private:
52     void advanceAnimationFrame() override;
53
54 };
55
56 }
```

7.25. include/objects/entities/seals/zombie.hpp fájlreferencia

```
#include "objects/entities/seals/seal_base.hpp"
```

Adatszerkezetek

- class [pftd::ZombieSeal](#)

Névterek

- namespace [pftd](#)

7.26. zombie.hpp

[Ugrás a fájl dokumentációjához.](#)

```
1 #pragma once
2
3 #include "objects/entities/seals/seal_base.hpp"
4
5 namespace pftd {
6
7     class ZombieSeal : public Seal
8     {
9     public:
10         ZombieSeal(FollowPath const& followpath);
11
12         Seal* clone() const override;
13
14     };
15
16 }
```

7.27. include/objects/entities/towers/iciclestabber.hpp fájlreferencia

```
#include "objects/entities/towers/tower_base.hpp"
```

Adatszerkezetek

- class [pftd::IcicleStabber](#)

Névterek

- namespace [pftd](#)

7.28. iciclestabber.hpp

[Ugrás a fájl dokumentációjához.](#)

```
1 #pragma once
2
3 #include "objects/entities/towers/tower_base.hpp"
4
5 namespace pftd {
6
7     class IcicleStabber final : public Tower
8     {
9     public:
10         IcicleStabber(utils::Vec2f const& position, int zIndex = 0);
11         IcicleStabber();
12         ~IcicleStabber() = default;
13
14         Tower* clone() const override;
15
16         void update(float dt) override;
17
18     };
19
20 }
```

7.29. include/objects/entities/towers/snowballer.hpp fájlreferencia

```
#include "objects/entities/towers/tower_base.hpp"
```

Adatszerkezetek

- class [pftd::Snowballer](#)

Névterek

- namespace [pftd](#)

7.30. snowballer.hpp

[Ugrás a fájl dokumentációjához.](#)

```
1 #pragma once
2
3 #include "objects/entities/towers/tower_base.hpp"
4
5 namespace pftd {
6
7     class Snowballer final : public Tower
8     {
9     public:
10         Snowballer(utils::Vec2f const& position, int zIndex = 0);
11         Snowballer();
12         ~Snowballer() = default;
13
14         Tower* clone() const override;
15
16         void update(float dt) override;
17
18         void attack() override;
19
20     };
21
22 }
```

7.31. include/objects/entities/towers/tower_base.hpp fájlreferencia

```
#include "objects/entities/entity_base.hpp"
#include "objects/entities/projectiles/projectile_base.hpp"
#include "objects/entities/seals/seal_base.hpp"
```

Adatszerkezetek

- class [pftd::Tower](#)

Névterek

- namespace [pftd](#)

7.32. tower_base.hpp

[Ugrás a fájl dokumentációjához.](#)

```

1 #pragma once
2
3 #include "objects/entities/entity_base.hpp"
4 #include "objects/entities/projectiles/projectile_base.hpp"
5 #include "objects/entities/seals/seal_base.hpp"
6
7 namespace pftd {
8
9     class Tower : public Entity
10     {
11     public:
12         using ProjSpawnFunc = std::function<void(Projectile*)>;
13         float radiusPixel = 120.0f;
14
15         float attackRangePixel = 100.0f;
16
17         float attackSpeedSec;
18
19         unsigned int attackDamage = 1U;
20
21         unsigned int price = 0U;
22
23         bool instantAttack = false;
24
25         // TODO: config struct for the data above as a param
26         Tower(std::string const& spriteSheetSrc, utils::Vec2i spriteSize, float attackSpeedSec, utils::Vec2f
27         const& position, utils::Vec2f const& size, int zIndex = 0);
28         Tower(std::string const& spriteSrc, float attackSpeedSec, utils::Vec2f const& position, utils::Vec2f
29         const& size, int zIndex = 0);
30         Tower(Tower const&);
31         virtual ~Tower() = default;
32
33         void setProjSpawnCb(ProjSpawnFunc callback);
34
35         virtual Tower* clone() const override = 0;
36
37         virtual void attack();
38
39         virtual bool lookForTarget(std::vector<Seal*> const& enemies);
40
41         virtual void update(float dt) override;
42
43     protected:
44         Seal* target = nullptr;
45
46         ProjSpawnFunc spawnProjectile;
47
48         float attackTimerSec = 0.0f;
49
50         void advanceAnimationFrame() override;
51     };
52 }

```

7.33. include/objects/gui/button.hpp fájlreferencia

```

#include "all.hpp"
#include "objects/clickable.hpp"
#include "objects/gui/image.hpp"
#include "objects/gui/label.hpp"
#include "objects/object_base.hpp"

```

Adatszerkezetek

- class `pftd::gr::Button`

Névterek

- namespace [pftd](#)
- namespace [pftd::gr](#)

7.34. button.hpp

[Ugrás a fájl dokumentációjához.](#)

```

1 #pragma once
2
3 #include "all.hpp"
4
5 #include "objects/clickable.hpp"
6 #include "objects/gui/image.hpp"
7 #include "objects/gui/label.hpp"
8 #include "objects/object_base.hpp"
9
10 namespace pftd {
11     namespace gr {
12
13         class Button : public Clickable
14         {
15         public:
16             Button(sf::Text&& label, utils::Vec2f const& position, utils::Vec2f const& size, bool active = true,
17                 int zIndex = 0);
18             virtual ~Button();
19
20             void setSound(std::string const& src);
21
22             void setBackground(std::string const& src);
23
24             std::string getLabel() const { return static_cast<std::string>(label.getText().getString()); };
25
26             virtual void handleClick(utils::Vec2i const& clickCoords) override;
27
28             virtual void draw(sf::RenderTarget& target, sf::RenderStates states) const override;
29
30 protected:
31             Label label;
32
33 private:
34             sf::FloatRect m_rect;
35
36             Sprite* m_background = nullptr;
37
38             sf::Sound m_clickSound;
39
40     };
41 }
42 
```

7.35. include/objects/gui/image.hpp fájlreferencia

```

#include "all.hpp"
#include "objects/object_base.hpp"

```

Adatszerkezetek

- class [pftd::gr::Sprite](#)

Névterek

- namespace [pftd](#)
- namespace [pftd::gr](#)

7.36. image.hpp

[Ugrás a fájl dokumentációjához.](#)

```
1 #pragma once
2
3 #include "all.hpp"
4
5 #include "objects/object_base.hpp"
6
7 namespace pftd
8 {
9     namespace gr
10    {
11
12        class Sprite final : public Object
13        {
14        public:
15            Sprite(sf::Texture const& texture, utils::Vec2f const& position, utils::Vec2f const& size, int zIndex
16                = -1);
17            Sprite(sf::Texture const& texture, sf::IntRect const& textureRect, utils::Vec2f const& position,
18                utils::Vec2f const& size, int zIndex = -1);
19            Sprite(std::string const& imageSrc, utils::Vec2f const& position, utils::Vec2f const& size, int
20                zIndex = -1);
21
22            Sprite(Sprite const& other);
23            Sprite(Sprite&& other) noexcept;
24
25            ~Sprite() = default;
26
27            void modColor(sf::Color const& color);
28
29            void flipY();
30
31            void setSpriteRect(sf::IntRect const& textureRect);
32
33            void draw(sf::RenderTarget& target, sf::RenderStates states) const override;
34
35        private:
36            sf::Texture const& m_texture;
37
38            sf::Sprite m_sprite;
39
40    };
41
42 }
```

7.37. include/objects/gui/label.hpp fájlreferencia

```
#include "all.hpp"
#include "objects/object_base.hpp"
```

Adatszerkezetek

- class [pftd::gr::Label](#)

Névterek

- namespace [pftd](#)
- namespace [pftd::gr](#)

7.38. label.hpp

[Ugrás a fájl dokumentációjához.](#)

```

1 #pragma once
2
3 #include "all.hpp"
4
5 #include "objects/object_base.hpp"
6
7 namespace pftd {
8 namespace gr {
9
11 class Label final : public Object
12 {
13 public:
14     Label(sf::Text const& text, sf::Color const& color = sf::Color::White);
15
16     Label(sf::Text const& text, utils::Vec2f position, int zIndex = 0, sf::Color const& color =
17         sf::Color::White);
18
19     // Ezek mind inline-ok. VVV
20
21     void setLabel(std::string const& newLabel) { m_text.setString(newLabel); }
22
23     void setOutline(sf::Color color, float thickness = 1.0f) { m_text.setOutlineColor(color);
24         m_text.setOutlineThickness(thickness); }
25
26     sf::Text& getText() { return m_text; }
27
28     sf::Text const& getText()const { return m_text; }
29
30     void draw(sf::RenderTarget& target, sf::RenderStates states)const override
31     {
32         target.draw(m_text, states);
33     }
34
35 private:
36     sf::Text m_text;
37 };
38
39 }
```

7.39. include/objects/object_base.hpp fájlreferencia

```
#include "all.hpp"
```

Adatszerkezetek

- class [pftd::Object](#)
- struct [pftd::Object::Compare](#)

Névterek

- namespace [pftd](#)

7.40. object_base.hpp

[Ugrás a fájl dokumentációjához.](#)

```
1 #pragma once
2
3 #include "all.hpp"
4
5 namespace pftd
6 {
7
8     class Object : public sf::Drawable
9     {
10     public:
11         Object() = default;
12
13         Object(utls::Vec2f position, utls::Vec2f size, int zIndex = 0):
14             zIndex{zIndex}, position{position}, size{size}
15         {}
16
17         Object(Object const&) = default;
18         Object(Object&&) noexcept = default;
19
20         virtual ~Object() = default;
21
22         virtual void draw(sf::RenderTarget& target, sf::RenderStates states) const = 0;
23
24         struct Compare
25         {
26             bool operator()(Object const& o1, Object const& o2) const
27             {
28                 return o1.zIndex > o2.zIndex;
29             };
30
31             bool operator()(Object const* o1, Object const* o2) const
32             {
33                 return this->operator()(*o1, *o2);
34             };
35         };
36     protected:
37         int zIndex = 0;
38
39         utls::Vec2f position;
40
41         utls::Vec2f size;
42     };
43 }
```

7.41. include/resources.hpp fájlreferencia

```
#include "SFML/Graphics.hpp"
#include "SFML/Audio.hpp"
#include "objects/gui/image.hpp"
```

Adatszerkezetek

- class [pftd::ResourceManager](#)

Névterek

- namespace [pftd](#)

7.42. resources.hpp

[Ugrás a fájl dokumentációjához.](#)

```

1 #pragma once
2
3 #include "SFML/Graphics.hpp"
4 #include "SFML/Audio.hpp"
5
6 #include "objects/gui/image.hpp"
7
8 namespace pftd
9 {
10
11     class ResourceManager final
12     {
13     public:
14         ResourceManager(ResourceManager const&) = delete;
15         ResourceManager(ResourceManager&&) = delete;
16
17         ~ResourceManager();
18
19         bool loadDefaultFont(std::string const& path);
20
21         sf::Texture const& getTexture(std::string const& source);
22
23         sf::SoundBuffer const& getSound(std::string const& source);
24
25         sf::Font const& getDefaultFont() const { return m_defaultFont; }
26
27         static ResourceManager* getInstance() { return m_instance; }
28
29         static ResourceManager* create();
30
31         static void destroy() { delete m_instance; }
32
33     private:
34         // Az 'm_instance' létrehozásához van használva.
35         ResourceManager() = default;
36
37         static ResourceManager* m_instance;
38
39         sf::Font m_defaultFont;
40
41         std::unordered_map<std::string, sf::Texture> m_textures;
42
43         std::unordered_map<std::string, sf::SoundBuffer> m_sounds;
44     };
45 }

```

7.43. include/scene.hpp fájlreferencia

```

#include "all.hpp"
#include "SFML/Window/Event.hpp"
#include "objects/object_base.hpp"

```

Adatszerkezetek

- class [pftd::Scene](#)

Névterek

- namespace [pftd](#)

7.44. scene.hpp

[Ugrás a fájl dokumentációjához.](#)

```

1 #pragma once
2
3 #include "all.hpp"
4 #include "SFML/Window/Event.hpp"
5 #include "objects/object_base.hpp"
6
7 namespace pftd
8 {
9
10
11 class Scene
12 {
13     using ObjPtrVec = std::vector<Object*>;
14 public:
15     Scene() = default;
16
17     Scene(Scene const&) = delete;
18     Scene(Scene&&) noexcept = delete;
19
20
21     virtual ~Scene();
22
23
24     ObjPtrVec const& getObjects()const { return objects; }
25
26     ObjPtrVec& getObjects() { return objects; }
27
28     void setMusic(std::string const& source, float volume = 100.0f);
29
30     virtual bool toggleActive(bool active);
31
32     virtual void onEvent(sf::Event const& event) = 0;
33
34     virtual void update(float dt) = 0;
35
36 protected:
37     bool isActive = false;
38
39     ObjPtrVec objects;
40
41     sf::Music* backgroundMusic = nullptr;
42 };
43
44 }
```

7.45. include/scenes/game.hpp fájlreferencia

```

#include "objects/gui/image.hpp"
#include "objects/gui/button.hpp"
#include "utils/hetero_collection.hpp"
#include "game/level.hpp"
#include "resources.hpp"
#include "scene.hpp"
```

Adatszerkezetek

- class [pftd::GameScene](#)
- struct [pftd::GameScene::InventoryItem](#)
- struct [pftd::GameScene::Inventory](#)

Névterek

- namespace [pftd](#)

7.46. game.hpp

[Ugrás a fájl dokumentációjához.](#)

```

1 #pragma once
2
3 #include "objects/gui/image.hpp"
4 #include "objects/gui/button.hpp"
5 #include "utils/hetero_collection.hpp"
6 #include "game/level.hpp"
7 #include "resources.hpp"
8 #include "scene.hpp"
9
10 namespace pftd
11 {
12
13     class GameScene final : public Scene
14     {
15     public:
16         struct InventoryItem : public Clickable
17         {
18             gr::Sprite frame;
19
20             gr::Sprite icon;
21
22             Tower* towerToSpawn = nullptr;
23
24             gr::Label priceLabel;
25
26             InventoryItem(Tower* tower, Level * const level, utils::Vec2f const& position, utils::Vec2f
const& size);
27             ~InventoryItem();
28
29             virtual void draw(sf::RenderTarget& target, sf::RenderStates states) const override;
30         };
31
32         struct Inventory final : public Object, public utils::Container<InventoryItem>
33         {
34             gr::Sprite background;
35
36             Inventory(std::string const& backgroundImageSrc);
37             ~Inventory() = default;
38
39             void draw(sf::RenderTarget& target, sf::RenderStates states) const override;
40         };
41
42         GameScene();
43         ~GameScene();
44
45         void onEvent(sf::Event const& event) override;
46
47         void update(float dt) override;
48
49         bool toggleActive(bool isActive) override;
50
51         void startGame();
52
53         void updateScore();
54
55         void updateWealth();
56     private:
57         //gr::Sprite* m_mapBackground = nullptr;
58
59         gr::Button* m_saveButt = nullptr;
60
61         gr::Label* m_moneyCounter = nullptr;
62
63         gr::Label* m_scoreCounter = nullptr;
64
65         sf::Sound m_hornSound =
sf::Sound{ResourceManager::getInstance()->getSound("res/audio/ready_for_battle.mp3")};
66
67         sf::Sound m_gameoverSound =
sf::Sound{ResourceManager::getInstance()->getSound("res/audio/gameover.mp3")};
68
69         Inventory* m_inventory = nullptr;
70
71         Level* m_level = nullptr;
72     };
73 }
74

```

7.47. include/scenes/menu.hpp fájlreferencia

```
#include "objects/gui/button.hpp"  
#include "scene.hpp"
```

Adatszerkezetek

- class [pftd::MenuScene](#)

Névterek

- namespace [pftd](#)

7.48. menu.hpp

[Ugrás a fájl dokumentációjához.](#)

```
1 #pragma once  
2  
3 #include "objects/gui/button.hpp"  
4 #include "scene.hpp"  
5  
6 namespace pftd  
7 {  
8  
9     class MenuScene final : public Scene  
10    {  
11    public:  
12        MenuScene();  
13        ~MenuScene();  
14  
15        void onEvent(sf::Event const& event) override;  
16  
17        void update(float dt) override;  
18  
19    private:  
20        std::vector<gr::Button*> m_buttons;  
21    };  
22  
23 }  
24  
25 }
```

7.49. include/utils/hetero_collection.hpp fájlreferencia

```
#include "all.hpp"
```

Adatszerkezetek

- class [pftd::utils::Container< T, C >](#)

Névterek

- namespace [pftd](#)
- namespace [pftd::utils](#)

7.50. hetero_collection.hpp

[Ugrás a fájl dokumentációjához.](#)

```

1  #pragma once
2
3  #include "all.hpp"
4
5  namespace pftd {
6  namespace utils {
7
12 template<typename T, typename C = std::vector<T*>
13 class Container
14 {
15 public:
16     Container() = default;
17     virtual ~Container()
18     {
19         for(auto& elem : m_container) {
20             delete elem;
21         }
22     }
23
30     virtual T* const append(T* elem)
31     {
32         try {
33             m_container.push_back(elem);
34         } catch(std::runtime_error err) {
35             throw std::runtime_error{err.what()}; // TODO: ezt ne így, nincs sok értelme
36         }
37
38         return elem;
39     }
40
41     // ? Nem biztos, hogy kellene fog bármikor is.
42     // template<typename Pred>
43     // void remove(Pred);
44
50     std::size_t size()const { return m_container.size(); }
51
57     C const& getContainer()const { return m_container; }
58
59 private:
60     C m_container;
61 };
62
63 };
64
65 }
66 }

```

7.51. include/utils/logger.hpp fájlreferencia

Makródefiníciók

- #define `print(x)`
- #define `where()`

7.51.1. Makródefiníciók dokumentációja

7.51.1.1. print

```

#define print(
    x )

```

7.51.1.2. where

```
#define where( )
```

7.52. logger.hpp

[Ugrás a fájl dokumentációjához.](#)

```
1 #pragma once
2
3 // Zseniálisan egyszerű logger makrókkal.
4
5 #ifdef _PFTD_DEBUG
6
7 #include <iostream>
8
9 #define print(x) std::cout << x << '\n'
10 #define where() print("[CALLED IN] " << __PRETTY_FUNCTION__)
11
12 #else
13
14 #define print(x)
15 #define where()
16
17 #endif
```

7.53. include/utils/parsers.hpp fájlreferencia

```
#include "all.hpp"
```

Adatszerkezetek

- class [pftd::utils::parser::Parser](#)
- class [pftd::utils::parser::LevelConfigParser](#)
- class [pftd::utils::parser::SaveFileParser](#)
- struct [pftd::utils::parser::SaveFileParser::StatsInfo](#)
- struct [pftd::utils::parser::SaveFileParser::EntityInfo](#)

Névterek

- namespace [pftd](#)
- namespace [pftd::utils](#)
- namespace [pftd::utils::parser](#)

7.54. parsers.hpp

[Ugrás a fájl dokumentációjához.](#)

```

1  #pragma once
2
3  #include "all.hpp"
4
5  namespace pftd
6  {
7      namespace utils
8      {
9          namespace parser
10         {
11
12             class Parser
13             {
14             public:
15                 Parser(std::string const& sourceFile, std::string label = "");
16
17                 virtual ~Parser();
18
19                 void reset();
20
21                 bool isLabelValid(bool skip = true);
22
23                 void skip(size_t howMany, char until = '\n');
24
25                 void skip(char while_);
26
27                 template<typename T> T get()
28                 {
29                     this->_skipWhitespace();
30                     while(sourceStream.peek() == commentDenoter) {
31                         this->_skipLine();
32                         this->_skipWhitespace();
33                     }
34
35                     T got;
36                     sourceStream >> got;
37                     return got;
38                 }
39
40                 char peekAhead();
41
42                 virtual void parse() = 0;
43
44             protected:
45                 std::ifstream sourceStream;
46
47                 std::string validLabel;
48
49                 char commentDenoter = '#';
50
51                 //virtual bool validate() const = 0;
52
53             private:
54                 void _skipLine();
55
56                 void _skipWhitespace();
57
58         };
59
60         class LevelConfigParser final : public Parser
61         {
62         public:
63             LevelConfigParser(std::string const& sourceFile);
64             ~LevelConfigParser() = default;
65
66             void parse() override;
67             std::vector<Vec2f> getAttribute(std::string name) const;
68
69         private:
70             std::string m_lastAttribute;
71             std::unordered_map<std::string, std::vector<Vec2f>> m_attribs;
72
73             std::string _getAttribute();
74
75         };
76
77         class SaveFileParser final : public Parser
78         {
79         public:
80             ///? Ezeknek még helyett kell keresni a végleges verzióig.
81             struct StatsInfo
82             {

```

```

141     unsigned int score, wealth, hp;
142 };
143
144 enum class EntityType
145 {
146     TOWER = 0U,
147     SEAL,
148     //PROJECTILE
149 };
150
151 struct EntityInfo
152 {
153     EntityType entityType;
154     utils::Vec2f position;
155
156     union {
157         struct {
158             unsigned int sealID;
159             unsigned int hp;
160         };
161
162         // struct {
163         //     unsigned int projID;
164         //     utils::Vec2f direction;
165         //     float speed;
166         // };
167
168         unsigned int towerID;
169     };
170 };
171
172 //public:
173 SaveFileParser(std::string const& sourceFile);
174 ~SaveFileParser() = default;
175
176 void parse() override;
177
178 StatsInfo getStats() const;
179
180 std::vector<EntityInfo> getEntities() const;
181
182 private:
183     StatsInfo m_readStats;
184
185     std::vector<EntityInfo> m_entities;
186 };
187
188 }
189
190 }
191
192 }

```

7.55. include/utils/random_gen.hpp fájlreferencia

```
#include "all.hpp"
```

Adatszerkezetek

- class [pftd::utils::Random](#)

Névterek

- namespace [pftd](#)
- namespace [pftd::utils](#)

7.56. random_gen.hpp

[Ugrás a fájl dokumentációjához.](#)

```
1 #pragma once
2
3 #include "all.hpp"
4
5 namespace pftd {
6 namespace utils {
7
8     class Random
9     {
10     public:
11         Random() = delete;
12
13         static std::default_random_engine randomEngine;
14
15         template<typename RandomDist>
16         static auto generate(RandomDist distribution)
17         {
18             return distribution(randomEngine);
19         }
20     };
21 };
22
23 }
```

7.57. include/utils/serializable.hpp fájlreferencia

```
#include "all.hpp"
```

Adatszerkezetek

- class [pftd::utils::Serializable](#)

Névterek

- namespace [pftd](#)
- namespace [pftd::utils](#)

7.58. serializable.hpp

[Ugrás a fájl dokumentációjához.](#)

```
1 #pragma once
2
3 #include "all.hpp"
4
5 namespace pftd {
6 {
7 namespace utils
8 {
9
10     class Serializable
11     {
12     public:
13         virtual void serialize(std::ostream& out) = 0;
14
15         virtual void deserialize(std::istream& in) = 0;
16     };
17 };
18
19 }
```

7.59. include/utils/substitute_types.hpp fájlreferencia

```
#include <cmath>
#include <ostream>
```

Adatszerkezetek

- struct `pftd::utils::Vec2< T >`

Névterek

- namespace `pftd`
- namespace `pftd::utils`

Típusdefiníciók

- using `pftd::utils::Vec2i` = `Vec2< int >`
- using `pftd::utils::Vec2f` = `Vec2< float >`

7.60. substitute_types.hpp

[Ugrás a fájl dokumentációjához.](#)

```
1 #pragma once
2
3 #include <cmath>
4 #include <ostream>
5
6 namespace pftd
7 {
8     namespace utils
9     {
10
11         template<typename T>
12         struct Vec2
13         {
14             T x;
15             T y;
16
17             Vec2() {}
18             Vec2(T x, T y):
19                 x{x}, y{y}
20             {}
21             Vec2(Vec2 const& right):
22                 x{right.x}, y{right.y}
23             {}
24
25             static float distance(Vec2 const& v1, Vec2 const& v2)
26             {
27                 return std::sqrt((v2.x-v1.x)*(v2.x-v1.x) + (v2.y-v1.y)*(v2.y-v1.y));
28             }
29
30             Vec2 normalize() const
31             {
32                 return *this / Vec2::distance(*this, {0, 0});
33             }
34
35             Vec2 operator+(Vec2 const& right) const
36             {
37                 return {this->x + right.x, this->y + right.y};
38             }
39
40             Vec2 operator-(Vec2 const& right) const
41             {
42                 return {this->x - right.x, this->y - right.y};
43             }
44         };
45     };
46 }
```

```
60         return {this->x - right.x, this->y - right.y};
61     }
62
63     Vec2 operator/(T scalar) const
64 {
65         return {this->x / scalar, this->y / scalar};
66     }
67     Vec2 operator*(T scalar) const
68 {
69         return {this->x * scalar, this->y * scalar};
70     }
71     bool operator==(Vec2 other) const
72 {
73         return this->x == other.x && this->y == other.y;
74     }
75     bool operator!=(Vec2 other) const
76 {
77         return !(this == other);
78     }
79
80     Vec2& operator=(Vec2 const& right)
81     {
82         if(&right != this) {
83             this->x = right.x;
84             this->y = right.y;
85         }
86         return *this;
87     }
88
89     friend std::ostream& operator<<(std::ostream& out, Vec2 const& vec)
90     {
91         out << "(" << vec.x << " " << vec.y << ")";
92         return out;
93     }
94 };
95
96 // Az SFML is ilyet csinál.
97 using Vec2i = Vec2<int>;
98 using Vec2f = Vec2<float>;
99
100 }
101 }
```


Tárgymutató

- [_getAttribute](#)
pftd::utils::parser::LevelConfigParser, [58](#)
 - [_skipLine](#)
pftd::utils::parser::Parser, [67](#)
 - [_skipWhitespace](#)
pftd::utils::parser::Parser, [67](#)
- [~App](#)
pftd::App, [14](#)
- [~Button](#)
pftd::gr::Button, [20](#)
- [~Container](#)
pftd::utils::Container< T, C >, [27](#)
- [~Entity](#)
pftd::Entity, [32](#)
- [~GameScene](#)
pftd::GameScene, [40](#)
- [~IcicleStabber](#)
pftd::IcicleStabber, [44](#)
- [~Inventory](#)
pftd::GameScene::Inventory, [46](#)
- [~InventoryItem](#)
pftd::GameScene::InventoryItem, [47](#)
- [~Level](#)
pftd::Level, [54](#)
- [~LevelConfigParser](#)
pftd::utils::parser::LevelConfigParser, [58](#)
- [~MenuScene](#)
pftd::MenuScene, [60](#)
- [~Nest](#)
pftd::Level::Nest, [62](#)
- [~Object](#)
pftd::Object, [64](#)
- [~Parser](#)
pftd::utils::parser::Parser, [67](#)
- [~Projectile](#)
pftd::Projectile, [70](#)
- [~Renderer](#)
pftd::Renderer, [76](#)
- [~ResourceManager](#)
pftd::ResourceManager, [80](#)
- [~SaveFileParser](#)
pftd::utils::parser::SaveFileParser, [85](#)
- [~Scene](#)
pftd::Scene, [87](#)
- [~Seal](#)
pftd::Seal, [92](#)
- [~Snowballer](#)
pftd::Snowballer, [100](#)
- [~Sprite](#)
pftd::gr::Sprite, [103](#)
- [~Tower](#)
pftd::Tower, [110](#)
- [addScene](#)
pftd::App, [15](#)
- [advanceAnimationFrame](#)
pftd::Entity, [32](#)
pftd::Seal, [93](#)
pftd::Tower, [110](#)
- [angularVelocityRadPerSec](#)
pftd::Projectile, [71](#)
- [App](#)
pftd::App, [14](#)
pftd::Renderer, [78](#)
- [append](#)
pftd::utils::Container< T, C >, [27](#)
- [attack](#)
pftd::Snowballer, [100](#)
pftd::Tower, [110](#)
- [attackDamage](#)
pftd::Tower, [112](#)
- [attackRangePixel](#)
pftd::Tower, [112](#)
- [attackSpeedSec](#)
pftd::Tower, [112](#)
- [attackTimerSec](#)
pftd::Tower, [112](#)
- [background](#)
pftd::GameScene::Inventory, [46](#)
- [backgroundMusic](#)
pftd::Scene, [90](#)
- [Button](#)
pftd::gr::Button, [19](#)
- [CELL_N](#)
pftd::Entity, [34](#)
- [cellSize](#)
pftd::Entity, [34](#)
- [changeScene](#)
pftd::App, [15](#)
- [clear](#)
pftd::Renderer, [77](#)
- [Clickable](#)
pftd::Clickable, [23](#)
- [clone](#)
pftd::Cub, [29](#)
pftd::Entity, [32](#)
pftd::FZC, [38](#)

pftd::IcicleStabber, [44](#)
 pftd::RegularSeal, [75](#)
 pftd::Seal, [93](#)
 pftd::Snowball, [98](#)
 pftd::Snowballer, [100](#)
 pftd::Tower, [111](#)
 pftd::ZombieSeal, [119](#)
 commentDenoter
 pftd::utils::parser::Parser, [69](#)
 config
 pftd::Level, [56](#)
 Container
 pftd::utils::Container< T, C >, [27](#)
 create
 pftd::App, [15](#)
 pftd::ResourceManager, [80](#)
 Cub
 pftd::Cub, [29](#)
 currentCell
 pftd::Entity, [34](#)
 currentSprite
 pftd::Entity, [34](#)

 damage
 pftd::Seal, [93](#)
 deselectTower
 pftd::Level, [54](#)
 deserialize
 pftd::Level::Stats, [105](#)
 pftd::utils::Serializable, [96](#)
 destroy
 pftd::App, [16](#)
 pftd::ResourceManager, [80](#)
 direction
 pftd::Projectile, [71](#)
 display
 pftd::Renderer, [77](#)
 distance
 pftd::utils::Vec2< T >, [115](#)
 draw
 pftd::Entity, [32](#)
 pftd::GameScene::Inventory, [46](#)
 pftd::GameScene::InventoryItem, [48](#)
 pftd::gr::Button, [20](#)
 pftd::gr::Label, [50](#)
 pftd::gr::Sprite, [103](#)
 pftd::Level, [54](#)
 pftd::Object, [65](#)

 EndPoint
 pftd, [10](#)
 Entity
 pftd::Entity, [31](#)
 EntityType
 pftd::utils::parser::SaveFileParser, [84](#)
 entityType
 pftd::utils::parser::SaveFileParser::EntityInfo, [36](#)

 flipY

 pftd::gr::Sprite, [103](#)
 FollowPath
 pftd::FollowPath, [37](#)
 followPath
 pftd::Level, [56](#)
 pftd::Seal, [94](#)
 frame
 pftd::GameScene::InventoryItem, [48](#)
 frameDurationSec
 pftd::Entity, [34](#)
 FZC
 pftd::FZC, [38](#)

 GameScene
 pftd::GameScene, [40](#)
 generate
 pftd::utils::Random, [73](#)
 get
 pftd::utils::parser::Parser, [67](#)
 getAttribute
 pftd::utils::parser::LevelConfigParser, [58](#)
 getContainer
 pftd::utils::Container< T, C >, [27](#)
 getDefaultFont
 pftd::ResourceManager, [81](#)
 getEntities
 pftd::utils::parser::SaveFileParser, [85](#)
 getInstance
 pftd::App, [16](#)
 pftd::ResourceManager, [81](#)
 getLabel
 pftd::gr::Button, [20](#)
 getObjects
 pftd::Scene, [88](#)
 getPosition
 pftd::Entity, [33](#)
 getSound
 pftd::ResourceManager, [81](#)
 getSprite
 pftd::Entity, [33](#)
 getStats
 pftd::utils::parser::SaveFileParser, [85](#)
 getText
 pftd::gr::Label, [51](#)
 getTexture
 pftd::ResourceManager, [82](#)
 getWindow
 pftd::Renderer, [77](#)
 getWindowHeight
 pftd::App, [16](#)
 getWindowWidth
 pftd::App, [16](#)

 handleClick
 pftd::Clickable, [23](#)
 pftd::gr::Button, [20](#)
 hasCompletedPath
 pftd::Seal, [93](#)
 hasReachedNest

- pftd::Seal, [94](#)
- hp
 - pftd::Level::Stats, [106](#)
 - pftd::Seal, [94](#)
 - pftd::utils::parser::SaveFileParser::EntityInfo, [36](#)
 - pftd::utils::parser::SaveFileParser::StatsInfo, [107](#)
- IcicleStabber
 - pftd::IcicleStabber, [43, 44](#)
- icon
 - pftd::GameScene::InventoryItem, [48](#)
- include/all.hpp, [121](#)
- include/app.hpp, [122](#)
- include/game/level.hpp, [123, 124](#)
- include/objects/clickable.hpp, [125](#)
- include/objects/entities/entity_base.hpp, [126](#)
- include/objects/entities/projectiles/projectile_base.hpp, [127](#)
- include/objects/entities/projectiles/snowball.hpp, [127, 128](#)
- include/objects/entities/seals/cub.hpp, [128](#)
- include/objects/entities/seals/followpath.hpp, [129](#)
- include/objects/entities/seals/fortified_zombie_cub.hpp, [129, 130](#)
- include/objects/entities/seals/regular.hpp, [130](#)
- include/objects/entities/seals/seal_base.hpp, [130, 131](#)
- include/objects/entities/seals/zombie.hpp, [131, 132](#)
- include/objects/entities/towers/iciclestabber.hpp, [132](#)
- include/objects/entities/towers/snowballer.hpp, [133](#)
- include/objects/entities/towers/tower_base.hpp, [133, 134](#)
- include/objects/gui/button.hpp, [134, 135](#)
- include/objects/gui/image.hpp, [135, 136](#)
- include/objects/gui/label.hpp, [136, 137](#)
- include/objects/object_base.hpp, [137, 138](#)
- include/resources.hpp, [138, 139](#)
- include/scene.hpp, [139, 140](#)
- include/scenes/game.hpp, [140, 141](#)
- include/scenes/menu.hpp, [142](#)
- include/utils/hetero_collection.hpp, [142, 143](#)
- include/utils/logger.hpp, [143, 144](#)
- include/utils/parsers.hpp, [144, 145](#)
- include/utils/random_gen.hpp, [146, 147](#)
- include/utils/serializable.hpp, [147](#)
- include/utils/substitute_types.hpp, [148](#)
- instantAttack
 - pftd::Tower, [113](#)
- Inventory
 - pftd::GameScene::Inventory, [45](#)
- InventoryItem
 - pftd::GameScene::InventoryItem, [47](#)
- isActive
 - pftd::Clickable, [25](#)
 - pftd::Scene, [90](#)
- isAnimated
 - pftd::Entity, [35](#)
- isCurrentlyStealing
 - pftd::Seal, [95](#)
- isGameOver
 - pftd::Level, [54](#)
- isLabelValid
 - pftd::utils::parser::Parser, [67](#)
- isRunning
 - pftd::App, [17](#)
- Label
 - pftd::gr::Label, [50](#)
- label
 - pftd::gr::Button, [21](#)
- lerpParam
 - pftd::Seal, [95](#)
- lerpPath
 - pftd::Seal, [94](#)
- Level
 - pftd::Level, [53](#)
- LevelConfigParser
 - pftd::utils::parser::LevelConfigParser, [58](#)
- linearSpeed
 - pftd::Projectile, [72](#)
- loadDefaultFont
 - pftd::ResourceManager, [82](#)
- logger.hpp
 - print, [143](#)
 - where, [143](#)
- lookForTarget
 - pftd::Tower, [111](#)
- loseHP
 - pftd::Level, [54](#)
- m_accuTimeSpawnSec
 - pftd::Level, [56](#)
- m_activeSceneID
 - pftd::App, [17](#)
- m_attribs
 - pftd::utils::parser::LevelConfigParser, [59](#)
- m_background
 - pftd::gr::Button, [21](#)
- m_buttons
 - pftd::MenuScene, [61](#)
- m_callback
 - pftd::Clickable, [25](#)
- m_clickSound
 - pftd::gr::Button, [21](#)
- m_container
 - pftd::utils::Container< T, C >, [28](#)
- m_defaultFont
 - pftd::ResourceManager, [82](#)
- m_entities
 - pftd::utils::parser::SaveFileParser, [85](#)
- m_gameoverSound
 - pftd::GameScene, [42](#)
- m_height
 - pftd::Renderer, [78](#)
- m_hornSound
 - pftd::GameScene, [42](#)
- m_instance
 - pftd::App, [17](#)
 - pftd::ResourceManager, [82](#)

- m_inventory
 - pftd::GameScene, 42
- m_lastAttribute
 - pftd::utils::parser::LevelConfigParser, 59
- m_level
 - pftd::GameScene, 42
- m_moneyCounter
 - pftd::GameScene, 42
- m_queue
 - pftd::Renderer, 78
- m_readStats
 - pftd::utils::parser::SaveFileParser, 86
- m_rect
 - pftd::gr::Button, 22
- m_renderer
 - pftd::App, 18
- m_running
 - pftd::App, 18
- m_saveButt
 - pftd::GameScene, 42
- m_scenes
 - pftd::App, 18
- m_scoreCounter
 - pftd::GameScene, 42
- m_sounds
 - pftd::ResourceManager, 83
- m_sprite
 - pftd::gr::Sprite, 104
- m_text
 - pftd::gr::Label, 52
- m_texture
 - pftd::gr::Sprite, 104
- m_textures
 - pftd::ResourceManager, 83
- m_width
 - pftd::Renderer, 78
- m_window
 - pftd::Renderer, 78
- MAX_HP
 - pftd::Level::Stats, 106
- MenuScene
 - pftd::MenuScene, 60
- modColor
 - pftd::gr::Sprite, 103
- money
 - pftd::Level::Stats, 106
- Nest
 - pftd::Level::Nest, 62
- nest
 - pftd::Level, 56
- nextPoint
 - pftd::Seal, 95
- normalize
 - pftd::utils::Vec2< T >, 115
- Object
 - pftd::Object, 64
- objects
 - pftd::Scene, 90
- ObjPtrVec
 - pftd::Scene, 87
- onEvent
 - pftd::GameScene, 40
 - pftd::MenuScene, 60
 - pftd::Scene, 88
- operator!=
 - pftd::utils::Vec2< T >, 115
- operator<<
 - pftd::utils::Vec2< T >, 117
- operator*
 - pftd::utils::Vec2< T >, 116
- operator()
 - pftd::Object::Compare, 25, 26
- operator+
 - pftd::utils::Vec2< T >, 116
- operator-
 - pftd::utils::Vec2< T >, 116
- operator/
 - pftd::utils::Vec2< T >, 116
- operator=
 - pftd::App, 17
 - pftd::utils::Vec2< T >, 116
- operator==
 - pftd::utils::Vec2< T >, 116
- parse
 - pftd::utils::parser::LevelConfigParser, 59
 - pftd::utils::parser::Parser, 68
 - pftd::utils::parser::SaveFileParser, 85
- Parser
 - pftd::utils::parser::Parser, 66
- peekAhead
 - pftd::utils::parser::Parser, 68
- pftd, 9
 - EndPoint, 10
- pftd::App, 13
 - ~App, 14
 - addScene, 15
 - App, 14
 - changeScene, 15
 - create, 15
 - destroy, 16
 - getInstance, 16
 - getWindowHeight, 16
 - getWindowWidth, 16
 - isRunning, 17
 - m_activeSceneID, 17
 - m_instance, 17
 - m_renderer, 18
 - m_running, 18
 - m_scenes, 18
 - operator=, 17
 - run, 17
- pftd::Clickable, 22
 - Clickable, 23
 - handleClick, 23
 - isActive, 25

- m_callback, 25
 - setCallback, 23
- pftd::Cub, 28
 - clone, 29
 - Cub, 29
- pftd::Entity, 30
 - ~Entity, 32
 - advanceAnimationFrame, 32
 - CELL_N, 34
 - cellSize, 34
 - clone, 32
 - currentCell, 34
 - currentSprite, 34
 - draw, 32
 - Entity, 31
 - frameDurationSec, 34
 - getPosition, 33
 - getSprite, 33
 - isAnimated, 35
 - setPosition, 33
 - spriteSheet, 35
 - totalElapsedSec, 35
 - update, 34
- pftd::FollowPath, 37
 - FollowPath, 37
- pftd::FZC, 37
 - clone, 38
 - FZC, 38
- pftd::GameScene, 39
 - ~GameScene, 40
 - GameScene, 40
 - m_gameoverSound, 42
 - m_hornSound, 42
 - m_inventory, 42
 - m_level, 42
 - m_moneyCounter, 42
 - m_saveButt, 42
 - m_scoreCounter, 42
 - onEvent, 40
 - startGame, 40
 - toggleActive, 40
 - update, 41
 - updateScore, 41
 - updateWealth, 41
- pftd::GameScene::Inventory, 45
 - ~Inventory, 46
 - background, 46
 - draw, 46
 - Inventory, 45
- pftd::GameScene::InventoryItem, 47
 - ~InventoryItem, 47
 - draw, 48
 - frame, 48
 - icon, 48
 - InventoryItem, 47
 - priceLabel, 48
 - towerToSpawn, 48
- pftd::gr, 10
 - m_callback, 25
 - setCallback, 23
- pftd::gr::Button, 18
 - ~Button, 20
 - Button, 19
 - draw, 20
 - getLabel, 20
 - handleClick, 20
 - label, 21
 - m_background, 21
 - m_clickSound, 21
 - m_rect, 22
 - setBackground, 21
 - setSound, 21
- pftd::gr::Label, 49
 - draw, 50
 - getText, 51
 - Label, 50
 - m_text, 52
 - setLabel, 51
 - setOutline, 51
- pftd::gr::Sprite, 101
 - ~Sprite, 103
 - draw, 103
 - flipY, 103
 - m_sprite, 104
 - m_texture, 104
 - modColor, 103
 - setSpriteRect, 104
 - Sprite, 102
- pftd::IcicleStabber, 43
 - ~IcicleStabber, 44
 - clone, 44
 - IcicleStabber, 43, 44
 - update, 44
- pftd::Level, 52
 - ~Level, 54
 - config, 56
 - deselectTower, 54
 - draw, 54
 - followPath, 56
 - isGameOver, 54
 - Level, 53
 - loseHP, 54
 - m_accuTimeSpawnSec, 56
 - nest, 56
 - placeTower, 55
 - projectiles, 56
 - seals, 56
 - selectedTower, 57
 - selectTower, 55
 - spawnSeal, 55
 - stats, 57
 - towers, 57
 - update, 55
- pftd::Level::Nest, 61
 - ~Nest, 62
 - Nest, 62
 - radiusPixel, 63
- pftd::Level::Stats, 104

- deserialize, 105
- hp, 106
- MAX_HP, 106
- money, 106
- score, 106
- serialize, 106
- Stats, 105
- pftd::MenuScene, 59
 - ~MenuScene, 60
 - m_buttons, 61
 - MenuScene, 60
 - onEvent, 60
 - update, 61
- pftd::Object, 63
 - ~Object, 64
 - draw, 65
 - Object, 64
 - position, 65
 - size, 65
 - zIndex, 65
- pftd::Object::Compare, 25
 - operator(), 25, 26
- pftd::Projectile, 70
 - ~Projectile, 70
 - angularVelocityRadPerSec, 71
 - direction, 71
 - linearSpeed, 72
 - Projectile, 70
 - update, 71
- pftd::RegularSeal, 74
 - clone, 75
 - RegularSeal, 74
- pftd::Renderer, 75
 - ~Renderer, 76
 - App, 78
 - clear, 77
 - display, 77
 - getWindow, 77
 - m_height, 78
 - m_queue, 78
 - m_width, 78
 - m_window, 78
 - pushQueue, 77
 - render, 77
 - Renderer, 76
- pftd::ResourceManager, 79
 - ~ResourceManager, 80
 - create, 80
 - destroy, 80
 - getDefaultFont, 81
 - getInstance, 81
 - getSound, 81
 - getTexture, 82
 - loadDefaultFont, 82
 - m_defaultFont, 82
 - m_instance, 82
 - m_sounds, 83
 - m_textures, 83
 - ResourceManager, 80
- pftd::Scene, 86
 - ~Scene, 87
 - backgroundMusic, 90
 - getObjects, 88
 - isActive, 90
 - objects, 90
 - ObjPtrVec, 87
 - onEvent, 88
 - Scene, 87
 - setMusic, 89
 - toggleActive, 89
 - update, 89
- pftd::SceneError, 90
- pftd::Seal, 91
 - ~Seal, 92
 - advanceAnimationFrame, 93
 - clone, 93
 - damage, 93
 - followPath, 94
 - hasCompletedPath, 93
 - hasReachedNest, 94
 - hp, 94
 - isCurrentlyStealing, 95
 - lerpParam, 95
 - lerpPath, 94
 - nextPoint, 95
 - reachedNest, 95
 - returned, 95
 - Seal, 92
 - speed, 95
 - update, 94
 - value, 95
- pftd::Snowball, 97
 - clone, 98
 - Snowball, 98
- pftd::Snowballer, 99
 - ~Snowballer, 100
 - attack, 100
 - clone, 100
 - Snowballer, 99
 - update, 100
- pftd::Tower, 108
 - ~Tower, 110
 - advanceAnimationFrame, 110
 - attack, 110
 - attackDamage, 112
 - attackRangePixel, 112
 - attackSpeedSec, 112
 - attackTimerSec, 112
 - clone, 111
 - instantAttack, 113
 - lookForTarget, 111
 - price, 113
 - ProjSpawnFunc, 109
 - radiusPixel, 113
 - setProjSpawnCb, 111
 - spawnProjectile, 113

- target, 113
- Tower, 109, 110
- update, 112
- pftd::utils, 10
 - Vec2f, 10
 - Vec2i, 10
- pftd::utils::Container< T, C >, 26
 - ~Container, 27
 - append, 27
 - Container, 27
 - getContainer, 27
 - m_container, 28
 - size, 27
- pftd::utils::parser, 11
- pftd::utils::parser::LevelConfigParser, 57
 - _getAttribute, 58
 - ~LevelConfigParser, 58
 - getAttribute, 58
 - LevelConfigParser, 58
 - m_attribs, 59
 - m_lastAttribute, 59
 - parse, 59
- pftd::utils::parser::Parser, 66
 - _skipLine, 67
 - _skipWhitespace, 67
 - ~Parser, 67
 - commentDenoter, 69
 - get, 67
 - isLabelValid, 67
 - parse, 68
 - Parser, 66
 - peekAhead, 68
 - reset, 68
 - skip, 68, 69
 - sourceStream, 69
 - validLabel, 69
- pftd::utils::parser::SaveFileParser, 83
 - ~SaveFileParser, 85
 - EntityType, 84
 - getEntities, 85
 - getStats, 85
 - m_entities, 85
 - m_readStats, 86
 - parse, 85
 - SaveFileParser, 84
 - SEAL, 84
 - TOWER, 84
- pftd::utils::parser::SaveFileParser::EntityInfo, 35
 - entityType, 36
 - hp, 36
 - position, 36
 - sealID, 36
 - towerID, 36
- pftd::utils::parser::SaveFileParser::StatsInfo, 107
 - hp, 107
 - score, 107
 - wealth, 107
- pftd::utils::Random, 72
 - generate, 73
 - Random, 72
 - randomEngine, 73
- pftd::utils::Serializable, 96
 - deserialize, 96
 - serialize, 97
- pftd::utils::Vec2< T >, 113
 - distance, 115
 - normalize, 115
 - operator!=, 115
 - operator<<, 117
 - operator*, 116
 - operator+, 116
 - operator-, 116
 - operator/, 116
 - operator=, 116
 - operator==, 116
 - Vec2, 114, 115
 - x, 117
 - y, 117
- pftd::ZombieSeal, 118
 - clone, 119
 - ZombieSeal, 118
- placeTower
 - pftd::Level, 55
- position
 - pftd::Object, 65
 - pftd::utils::parser::SaveFileParser::EntityInfo, 36
- price
 - pftd::Tower, 113
- priceLabel
 - pftd::GameScene::InventoryItem, 48
- print
 - logger.hpp, 143
- Projectile
 - pftd::Projectile, 70
- projectiles
 - pftd::Level, 56
- ProjSpawnFunc
 - pftd::Tower, 109
- pushQueue
 - pftd::Renderer, 77
- radiusPixel
 - pftd::Level::Nest, 63
 - pftd::Tower, 113
- Random
 - pftd::utils::Random, 72
- randomEngine
 - pftd::utils::Random, 73
- reachedNest
 - pftd::Seal, 95
- RegularSeal
 - pftd::RegularSeal, 74
- render
 - pftd::Renderer, 77
- Renderer
 - pftd::Renderer, 76
- reset

pftd::utils::parser::Parser, 68
 ResourceManager
 pftd::ResourceManager, 80
 returned
 pftd::Seal, 95
 run
 pftd::App, 17
 SaveFileParser
 pftd::utils::parser::SaveFileParser, 84
 Scene
 pftd::Scene, 87
 score
 pftd::Level::Stats, 106
 pftd::utils::parser::SaveFileParser::StatsInfo, 107
 SEAL
 pftd::utils::parser::SaveFileParser, 84
 Seal
 pftd::Seal, 92
 sealID
 pftd::utils::parser::SaveFileParser::EntityInfo, 36
 seals
 pftd::Level, 56
 selectedTower
 pftd::Level, 57
 selectTower
 pftd::Level, 55
 serialize
 pftd::Level::Stats, 106
 pftd::utils::Serializable, 97
 setBackground
 pftd::gr::Button, 21
 setCallback
 pftd::Clickable, 23
 setLabel
 pftd::gr::Label, 51
 setMusic
 pftd::Scene, 89
 setOutline
 pftd::gr::Label, 51
 setPosition
 pftd::Entity, 33
 setProjSpawnCb
 pftd::Tower, 111
 setSound
 pftd::gr::Button, 21
 setSpriteRect
 pftd::gr::Sprite, 104
 size
 pftd::Object, 65
 pftd::utils::Container< T, C >, 27
 skip
 pftd::utils::parser::Parser, 68, 69
 Snowball
 pftd::Snowball, 98
 Snowballer
 pftd::Snowballer, 99
 sourceStream
 pftd::utils::parser::Parser, 69
 spawnProjectile
 pftd::Tower, 113
 spawnSeal
 pftd::Level, 55
 speed
 pftd::Seal, 95
 Sprite
 pftd::gr::Sprite, 102
 spriteSheet
 pftd::Entity, 35
 startGame
 pftd::GameScene, 40
 Stats
 pftd::Level::Stats, 105
 stats
 pftd::Level, 57
 target
 pftd::Tower, 113
 toggleActive
 pftd::GameScene, 40
 pftd::Scene, 89
 totalElapsedSec
 pftd::Entity, 35
 TOWER
 pftd::utils::parser::SaveFileParser, 84
 Tower
 pftd::Tower, 109, 110
 towerID
 pftd::utils::parser::SaveFileParser::EntityInfo, 36
 towers
 pftd::Level, 57
 towerToSpawn
 pftd::GameScene::InventoryItem, 48
 update
 pftd::Entity, 34
 pftd::GameScene, 41
 pftd::IcicleStabber, 44
 pftd::Level, 55
 pftd::MenuScene, 61
 pftd::Projectile, 71
 pftd::Scene, 89
 pftd::Seal, 94
 pftd::Snowballer, 100
 pftd::Tower, 112
 updateScore
 pftd::GameScene, 41
 updateWealth
 pftd::GameScene, 41
 validLabel
 pftd::utils::parser::Parser, 69
 value
 pftd::Seal, 95
 Vec2
 pftd::utils::Vec2< T >, 114, 115
 Vec2f
 pftd::utils, 10

Vec2i

pftd::utils, [10](#)

wealth

pftd::utils::parser::SaveFileParser::StatsInfo, [107](#)

where

logger.hpp, [143](#)

x

pftd::utils::Vec2< T >, [117](#)

y

pftd::utils::Vec2< T >, [117](#)

zIndex

pftd::Object, [65](#)

ZombieSeal

pftd::ZombieSeal, [118](#)