

r-type
0.0.0

Generated by Doxygen 1.11.0

1 r-type	1
1.1 R-Type	1
1.1.1 Supported Platforms	1
1.1.2 Project Structure	1
1.1.3 Prerequisites	2
1.1.4 Clone the project	2
1.1.5 Build and Run	2
1.1.5.1 Unix (Linux, macOS)	2
1.1.5.2 Windows	2
1.1.6 Documentation	2
1.1.7 External Libraries	2
1.1.8 Commit Norms	2
2 Namespace Index	5
2.1 Namespace List	5
3 Hierarchical Index	7
3.1 Class Hierarchy	7
4 Class Index	9
4.1 Class List	9
5 File Index	11
5.1 File List	11
6 Namespace Documentation	13
6.1 cli Namespace Reference	13
6.1.1 Typedef Documentation	13
6.1.1.1 json	13
6.2 cli::Config Namespace Reference	14
6.3 cli::Config::Audio Namespace Reference	14
6.3.1 Variable Documentation	14
6.3.1.1 DEFAULT_AUDIO_MUTED	14
6.3.1.2 DEFAULT_AUDIO_VOLUME	14
6.4 cli::Config::Window Namespace Reference	14
6.4.1 Variable Documentation	14
6.4.1.1 DEFAULT_WINDOW_FRAME_LIMIT	14
6.4.1.2 DEFAULT_WINDOW_FULLSCREEN	15
6.4.1.3 DEFAULT_WINDOW_HEIGHT	15
6.4.1.4 DEFAULT_WINDOW_WIDTH	15
6.5 cli::Paths Namespace Reference	15
6.6 cli::Paths::Audio Namespace Reference	15
6.6.1 Variable Documentation	15
6.6.1.1 AUDIO_BATTLE_THEME	15
6.6.1.2 AUDIO_COIN	15

6.6.1.3 AUDIO_TITLE	16
6.7 cli::Paths::Fonts Namespace Reference	16
6.7.1 Variable Documentation	16
6.7.1.1 FONTS_RTYPE	16
6.8 ecs Namespace Reference	16
6.8.1 Typedef Documentation	17
6.8.1.1 Entity	17
6.8.2 Variable Documentation	17
6.8.2.1 INVALID_ENTITY	17
6.9 eng Namespace Reference	17
6.9.1 Typedef Documentation	18
6.9.1.1 scene_id_t	18
6.9.2 Enumeration Type Documentation	18
6.9.2.1 EventType	18
6.9.2.2 Key	18
6.10 srv Namespace Reference	20
6.10.1 Typedef Documentation	20
6.10.1.1 json	20
6.11 utl Namespace Reference	20
6.11.1 Enumeration Type Documentation	20
6.11.1.1 LogLevel	20
6.11.2 Function Documentation	21
6.11.2.1 readFile()	21
7 Class Documentation	23
7.1 cli::AGameClient Class Reference	23
7.1.1 Detailed Description	24
7.1.2 Constructor & Destructor Documentation	25
7.1.2.1 ~AGameClient()	25
7.1.3 Member Function Documentation	25
7.1.3.1 getName()	25
7.1.3.2 setName()	25
7.1.4 Member Data Documentation	25
7.1.4.1 m_name	25
7.2 srv::AGameServer Class Reference	26
7.2.1 Detailed Description	27
7.2.2 Constructor & Destructor Documentation	28
7.2.2.1 ~AGameServer()	28
7.2.3 Member Function Documentation	28
7.2.3.1 getName()	28
7.2.3.2 setName()	28
7.2.4 Member Data Documentation	28
7.2.4.1 m_name	28

7.3 cli::ArgsConfig Struct Reference	29
7.3.1 Detailed Description	29
7.3.2 Member Function Documentation	29
7.3.2.1 fromFile()	29
7.3.3 Member Data Documentation	30
7.3.3.1 exit	30
7.3.3.2 frameLimit	30
7.3.3.3 fullscreen	30
7.3.3.4 height	30
7.3.3.5 width	31
7.4 srv::ArgsConfig Struct Reference	31
7.4.1 Detailed Description	32
7.4.2 Member Function Documentation	32
7.4.2.1 fromFile()	32
7.4.3 Member Data Documentation	32
7.4.3.1 exit	32
7.4.3.2 host	32
7.4.3.3 port	33
7.5 cli::ArgsHandler Class Reference	33
7.5.1 Detailed Description	34
7.5.2 Constructor & Destructor Documentation	34
7.5.2.1 ArgsHandler() [1/3]	34
7.5.2.2 ~ArgsHandler()	34
7.5.2.3 ArgsHandler() [2/3]	34
7.5.2.4 ArgsHandler() [3/3]	34
7.5.3 Member Function Documentation	34
7.5.3.1 operator=() [1/2]	34
7.5.3.2 operator=() [2/2]	34
7.5.3.3 ParseArgs()	35
7.5.3.4 ParseEnv()	35
7.6 srv::ArgsHandler Class Reference	36
7.6.1 Detailed Description	36
7.6.2 Constructor & Destructor Documentation	37
7.6.2.1 ArgsHandler() [1/3]	37
7.6.2.2 ~ArgsHandler()	37
7.6.2.3 ArgsHandler() [2/3]	37
7.6.2.4 ArgsHandler() [3/3]	37
7.6.3 Member Function Documentation	37
7.6.3.1 operator=() [1/2]	37
7.6.3.2 operator=() [2/2]	37
7.6.3.3 ParseArgs()	38
7.6.3.4 ParseEnv()	38
7.7 ecs::Audio Struct Reference	39

7.7.1 Detailed Description	39
7.7.2 Member Data Documentation	39
7.7.2.1 id	39
7.7.2.2 loop	40
7.7.2.3 path	40
7.7.2.4 volume	40
7.8 eng::Audio Struct Reference	40
7.8.1 Detailed Description	41
7.8.2 Member Data Documentation	41
7.8.2.1 loop	41
7.8.2.2 name	41
7.8.2.3 path	41
7.8.2.4 volume	41
7.9 ecs::AudioSystem Class Reference	42
7.9.1 Detailed Description	42
7.9.2 Constructor & Destructor Documentation	42
7.9.2.1 AudioSystem() [1/3]	42
7.9.2.2 ~AudioSystem()	43
7.9.2.3 AudioSystem() [2/3]	43
7.9.2.4 AudioSystem() [3/3]	43
7.9.3 Member Function Documentation	43
7.9.3.1 operator=() [1/2]	43
7.9.3.2 operator=() [2/2]	43
7.9.3.3 update()	43
7.10 eng::AudioSystem Class Reference	44
7.10.1 Detailed Description	46
7.10.2 Constructor & Destructor Documentation	46
7.10.2.1 AudioSystem() [1/3]	46
7.10.2.2 ~AudioSystem()	46
7.10.2.3 AudioSystem() [2/3]	46
7.10.2.4 AudioSystem() [3/3]	46
7.10.3 Member Function Documentation	47
7.10.3.1 isEnabled()	47
7.10.3.2 operator=() [1/2]	47
7.10.3.3 operator=() [2/2]	47
7.10.3.4 setEnable()	47
7.10.3.5 update()	47
7.10.4 Member Data Documentation	47
7.10.4.1 m_isEnable	47
7.10.4.2 m_renderer	48
7.11 cli::Client Class Reference	48
7.11.1 Detailed Description	49
7.11.2 Constructor & Destructor Documentation	49

7.11.2.1 Client() [1/3]	49
7.11.2.2 ~Client()	50
7.11.2.3 Client() [2/3]	50
7.11.2.4 Client() [3/3]	50
7.11.3 Member Function Documentation	50
7.11.3.1 operator=() [1/2]	50
7.11.3.2 operator=() [2/2]	51
7.11.4 Member Data Documentation	51
7.11.4.1 m_engine	51
7.11.4.2 m_game	51
7.12 utl::Clock Class Reference	51
7.12.1 Detailed Description	53
7.12.2 Member Typedef Documentation	53
7.12.2.1 Duration	53
7.12.2.2 TimePoint	53
7.12.3 Constructor & Destructor Documentation	53
7.12.3.1 Clock() [1/3]	53
7.12.3.2 ~Clock()	53
7.12.3.3 Clock() [2/3]	54
7.12.3.4 Clock() [3/3]	54
7.12.4 Member Function Documentation	54
7.12.4.1 getDeltaSeconds()	54
7.12.4.2 getElapsed()	55
7.12.4.3 now()	55
7.12.4.4 operator=() [1/2]	55
7.12.4.5 operator=() [2/2]	56
7.12.4.6 pause()	56
7.12.4.7 restart()	56
7.12.4.8 resume()	57
7.12.5 Friends And Related Symbol Documentation	57
7.12.5.1 operator<<	57
7.12.6 Member Data Documentation	57
7.12.6.1 m_isPaused	57
7.12.6.2 m_pausedDuration	57
7.12.6.3 m_pausedTime	58
7.12.6.4 m_start	58
7.13 ecs::Color Struct Reference	58
7.13.1 Detailed Description	59
7.13.2 Member Data Documentation	59
7.13.2.1 a	59
7.13.2.2 b	59
7.13.2.3 g	59
7.13.2.4 id	59

7.13.2.5 r	59
7.14 eng::Color Struct Reference	60
7.14.1 Detailed Description	60
7.14.2 Member Data Documentation	60
7.14.2.1 a	60
7.14.2.2 b	60
7.14.2.3 g	61
7.14.2.4 r	61
7.15 eng::Engine Class Reference	61
7.15.1 Detailed Description	62
7.15.2 Constructor & Destructor Documentation	62
7.15.2.1 Engine() [1/3]	62
7.15.2.2 ~Engine()	62
7.15.2.3 Engine() [2/3]	63
7.15.2.4 Engine() [3/3]	63
7.15.3 Member Function Documentation	63
7.15.3.1 addSystem()	63
7.15.3.2 getAudio()	63
7.15.3.3 getClock()	64
7.15.3.4 getNetworkClient()	64
7.15.3.5 getRegistry()	64
7.15.3.6 getRenderer()	65
7.15.3.7 operator=() [1/2]	65
7.15.3.8 operator=() [2/2]	65
7.15.3.9 updateSystems()	65
7.15.4 Member Data Documentation	66
7.15.4.1 m_audio	66
7.15.4.2 m_clock	66
7.15.4.3 m_networkClient	66
7.15.4.4 m_registry	66
7.15.4.5 m_renderer	66
7.15.4.6 m_systems	66
7.16 cli::EnvConfig Struct Reference	67
7.16.1 Detailed Description	67
7.17 srv::EnvConfig Struct Reference	67
7.17.1 Detailed Description	67
7.18 eng::Event Struct Reference	68
7.18.1 Detailed Description	68
7.18.2 Member Data Documentation	68
7.18.2.1 key	68
7.18.2.2 type	68
7.19 ecs::Font Struct Reference	69
7.19.1 Detailed Description	69

7.19.2 Member Data Documentation	69
7.19.2.1 id	69
7.19.2.2 path	70
7.20 eng::Font Struct Reference	70
7.20.1 Detailed Description	70
7.20.2 Member Data Documentation	71
7.20.2.1 name	71
7.20.2.2 path	71
7.21 ecs::FontSystem Class Reference	71
7.21.1 Detailed Description	72
7.21.2 Constructor & Destructor Documentation	72
7.21.2.1 FontSystem() [1/3]	72
7.21.2.2 ~FontSystem()	72
7.21.2.3 FontSystem() [2/3]	72
7.21.2.4 FontSystem() [3/3]	72
7.21.3 Member Function Documentation	72
7.21.3.1 operator=() [1/2]	72
7.21.3.2 operator=() [2/2]	73
7.21.3.3 update()	73
7.22 eng::FontSystem Class Reference	73
7.22.1 Detailed Description	75
7.22.2 Constructor & Destructor Documentation	75
7.22.2.1 FontSystem() [1/3]	75
7.22.2.2 ~FontSystem()	75
7.22.2.3 FontSystem() [2/3]	75
7.22.2.4 FontSystem() [3/3]	75
7.22.3 Member Function Documentation	76
7.22.3.1 isEnabled()	76
7.22.3.2 operator=() [1/2]	76
7.22.3.3 operator=() [2/2]	76
7.22.3.4 setEnable()	76
7.22.3.5 update()	76
7.22.4 Member Data Documentation	76
7.22.4.1 m_isEnable	76
7.22.4.2 m_renderer	77
7.23 eng::IAudio Class Reference	77
7.23.1 Detailed Description	78
7.23.2 Constructor & Destructor Documentation	78
7.23.2.1 ~IAudio()	78
7.23.3 Member Function Documentation	78
7.23.3.1 createAudio()	78
7.23.3.2 playAudio()	79
7.23.3.3 setLoop()	79

7.23.3.4	setVolume()	79
7.24	cli::IGameClient Class Reference	79
7.24.1	Detailed Description	80
7.24.2	Constructor & Destructor Documentation	80
7.24.2.1	~IGameClient()	80
7.24.3	Member Function Documentation	80
7.24.3.1	getName()	80
7.24.3.2	setName()	80
7.25	srv::IGameServer Class Reference	81
7.25.1	Detailed Description	82
7.25.2	Constructor & Destructor Documentation	82
7.25.2.1	~IGameServer()	82
7.25.3	Member Function Documentation	82
7.25.3.1	getName()	82
7.25.3.2	setName()	82
7.26	eng::SFMLAudio::Impl Struct Reference	83
7.26.1	Detailed Description	83
7.26.2	Member Data Documentation	84
7.26.2.1	musics	84
7.27	eng::SFMLRendererr::Impl Struct Reference	84
7.27.1	Detailed Description	85
7.27.2	Member Data Documentation	85
7.27.2.1	fonts	85
7.27.2.2	texts	85
7.27.2.3	window	85
7.28	eng::INetworkClient Class Reference	85
7.28.1	Detailed Description	86
7.28.2	Constructor & Destructor Documentation	86
7.28.2.1	~INetworkClient()	86
7.29	srv::INetworkServer Class Reference	86
7.29.1	Detailed Description	86
7.29.2	Constructor & Destructor Documentation	87
7.29.2.1	~INetworkServer()	87
7.30	ecs::Registry::IPool Class Reference	87
7.30.1	Detailed Description	88
7.30.2	Constructor & Destructor Documentation	88
7.30.2.1	~IPool()	88
7.30.3	Member Function Documentation	88
7.30.3.1	remove()	88
7.31	eng::IRenderer Class Reference	88
7.31.1	Detailed Description	90
7.31.2	Constructor & Destructor Documentation	91
7.31.2.1	~IRenderer()	91

7.31.3 Member Function Documentation	91
7.31.3.1 clearWindow()	91
7.31.3.2 closeWindow()	91
7.31.3.3 createFont()	91
7.31.3.4 createText()	91
7.31.3.5 createWindow()	91
7.31.3.6 displayWindow()	92
7.31.3.7 drawText()	92
7.31.3.8 pollEvent()	92
7.31.3.9 setFrameLimit()	92
7.31.3.10 setTextColor()	92
7.31.3.11 setTextContent()	93
7.31.3.12 setTextPosition()	93
7.31.3.13 windowIsOpen()	93
7.32 eng::ISystem Class Reference	94
7.32.1 Detailed Description	95
7.32.2 Constructor & Destructor Documentation	95
7.32.2.1 ~ISystem()	95
7.32.3 Member Function Documentation	95
7.32.3.1 isEnabled()	95
7.32.3.2 update()	95
7.33 utl::Logger Class Reference	95
7.33.1 Detailed Description	97
7.33.2 Member Enumeration Documentation	97
7.33.2.1 ColorIndex	97
7.33.3 Constructor & Destructor Documentation	97
7.33.3.1 Logger() [1/3]	97
7.33.3.2 Logger() [2/3]	97
7.33.3.3 Logger() [3/3]	98
7.33.3.4 ~Logger()	98
7.33.4 Member Function Documentation	98
7.33.4.1 formatLogMessage()	98
7.33.4.2 getColorForDuration()	98
7.33.4.3 init()	99
7.33.4.4 log()	99
7.33.4.5 logExecutionTime()	100
7.33.4.6 operator=() [1/2]	100
7.33.4.7 operator=() [2/2]	101
7.33.5 Member Data Documentation	101
7.33.5.1 LOG_LEVEL_COLOR	101
7.33.5.2 LOG_LEVEL_STRING	101
7.34 ecs::Registry::Pool< T > Class Template Reference	102
7.34.1 Detailed Description	104

7.34.2 Member Function Documentation	104
7.34.2.1 add()	104
7.34.2.2 get()	104
7.34.2.3 has()	104
7.34.2.4 remove()	104
7.34.3 Member Data Documentation	105
7.34.3.1 data	105
7.35 ecs::Registry Class Reference	105
7.35.1 Detailed Description	106
7.35.2 Constructor & Destructor Documentation	106
7.35.2.1 Registry() [1/3]	106
7.35.2.2 ~Registry()	107
7.35.2.3 Registry() [2/3]	107
7.35.2.4 Registry() [3/3]	107
7.35.3 Member Function Documentation	107
7.35.3.1 addComponent()	107
7.35.3.2 createEntity()	108
7.35.3.3 getAll()	108
7.35.3.4 getComponent()	109
7.35.3.5 getPool()	109
7.35.3.6 hasComponent()	110
7.35.3.7 onComponentAdded()	110
7.35.3.8 onEntityCreated()	111
7.35.3.9 operator=() [1/2]	111
7.35.3.10 operator=() [2/2]	111
7.35.3.11 removeComponent()	111
7.35.4 Member Data Documentation	112
7.35.4.1 m_components	112
7.35.4.2 m_entities	112
7.35.4.3 m_lastEntity	112
7.35.4.4 m_onComponentAddedCallbacks	112
7.35.4.5 m_onEntityCreatedCallbacks	112
7.36 eng::Scene Class Reference	113
7.36.1 Detailed Description	114
7.36.2 Constructor & Destructor Documentation	114
7.36.2.1 Scene()	114
7.36.2.2 ~Scene()	114
7.36.3 Member Function Documentation	114
7.36.3.1 getDisplay()	114
7.36.3.2 getId()	114
7.36.3.3 getName()	114
7.36.3.4 setDisplay()	115
7.36.3.5 setName()	115

7.36.4 Member Data Documentation	115
7.36.4.1 m_display	115
7.36.4.2 m_id	115
7.36.4.3 m_name	115
7.37 eng::SceneManager Class Reference	116
7.37.1 Detailed Description	117
7.37.2 Constructor & Destructor Documentation	117
7.37.2.1 SceneManager() [1/3]	117
7.37.2.2 ~SceneManager()	117
7.37.2.3 SceneManager() [2/3]	117
7.37.2.4 SceneManager() [3/3]	117
7.37.3 Member Function Documentation	118
7.37.3.1 addScene()	118
7.37.3.2 getCurrentScene()	118
7.37.3.3 getScene()	118
7.37.3.4 operator=() [1/2]	118
7.37.3.5 operator=() [2/2]	118
7.37.3.6 switchToScene()	118
7.37.4 Member Data Documentation	118
7.37.4.1 m_currentSceneId	118
7.37.4.2 m_scenes	118
7.38 srv::Server Class Reference	119
7.38.1 Detailed Description	119
7.38.2 Constructor & Destructor Documentation	119
7.38.2.1 Server() [1/3]	119
7.38.2.2 ~Server()	120
7.38.2.3 Server() [2/3]	120
7.38.2.4 Server() [3/3]	120
7.38.3 Member Function Documentation	120
7.38.3.1 operator=() [1/2]	120
7.38.3.2 operator=() [2/2]	120
7.39 eng::SFMLAudio Class Reference	121
7.39.1 Detailed Description	123
7.39.2 Constructor & Destructor Documentation	123
7.39.2.1 SFMLAudio() [1/3]	123
7.39.2.2 ~SFMLAudio()	123
7.39.2.3 SFMLAudio() [2/3]	123
7.39.2.4 SFMLAudio() [3/3]	123
7.39.3 Member Function Documentation	124
7.39.3.1 createAudio()	124
7.39.3.2 operator=() [1/2]	124
7.39.3.3 operator=() [2/2]	124
7.39.3.4 playAudio()	124

7.39.3.5	setLoop()	124
7.39.3.6	setVolume()	125
7.39.4	Member Data Documentation	125
7.39.4.1	pImpl	125
7.40	eng::SFMLRenderer Class Reference	125
7.40.1	Detailed Description	128
7.40.2	Constructor & Destructor Documentation	128
7.40.2.1	SFMLRenderer() [1/3]	128
7.40.2.2	~SFMLRenderer()	128
7.40.2.3	SFMLRenderer() [2/3]	129
7.40.2.4	SFMLRenderer() [3/3]	129
7.40.3	Member Function Documentation	129
7.40.3.1	clearWindow()	129
7.40.3.2	closeWindow()	129
7.40.3.3	createFont()	129
7.40.3.4	createText()	129
7.40.3.5	createWindow()	130
7.40.3.6	displayWindow()	130
7.40.3.7	drawText()	130
7.40.3.8	operator=() [1/2]	130
7.40.3.9	operator=() [2/2]	130
7.40.3.10	pollEvent()	131
7.40.3.11	setFrameLimit()	131
7.40.3.12	setTextColor()	131
7.40.3.13	setTextContent()	131
7.40.3.14	setTextPosition()	132
7.40.3.15	windowIsOpen()	132
7.40.4	Member Data Documentation	132
7.40.4.1	m_impl	132
7.41	ecs::Sprite Struct Reference	132
7.41.1	Detailed Description	133
7.41.2	Member Data Documentation	133
7.41.2.1	id	133
7.41.2.2	path	134
7.42	ecs::Text Struct Reference	134
7.42.1	Detailed Description	135
7.42.2	Member Data Documentation	135
7.42.2.1	content	135
7.42.2.2	fontSize	135
7.42.2.3	id	135
7.43	eng::Text Struct Reference	135
7.43.1	Detailed Description	136
7.43.2	Member Data Documentation	137

7.43.2.1	color	137
7.43.2.2	content	137
7.43.2.3	fontName	137
7.43.2.4	name	137
7.43.2.5	size	137
7.43.2.6	x	137
7.43.2.7	y	138
7.44	ecs::TextSyStem Class Reference	138
7.44.1	Detailed Description	139
7.44.2	Member Typedef Documentation	139
7.44.2.1	DrawCallback	139
7.44.3	Constructor & Destructor Documentation	139
7.44.3.1	TextSyStem() [1/3]	139
7.44.3.2	~TextSyStem()	139
7.44.3.3	TextSyStem() [2/3]	139
7.44.3.4	TextSyStem() [3/3]	139
7.44.4	Member Function Documentation	139
7.44.4.1	operator=() [1/2]	139
7.44.4.2	operator=() [2/2]	140
7.44.4.3	setDrawCallback()	140
7.44.4.4	update()	140
7.44.5	Member Data Documentation	140
7.44.5.1	m_drawCallback	140
7.45	eng::TextSyStem Class Reference	141
7.45.1	Detailed Description	143
7.45.2	Constructor & Destructor Documentation	143
7.45.2.1	TextSyStem() [1/3]	143
7.45.2.2	~TextSyStem()	143
7.45.2.3	TextSyStem() [2/3]	143
7.45.2.4	TextSyStem() [3/3]	143
7.45.3	Member Function Documentation	144
7.45.3.1	isEnabled()	144
7.45.3.2	operator=() [1/2]	144
7.45.3.3	operator=() [2/2]	144
7.45.3.4	setEnabled()	144
7.45.3.5	update()	144
7.45.4	Member Data Documentation	145
7.45.4.1	m_isEnable	145
7.45.4.2	m_renderer	145
7.46	ecs::Transform Struct Reference	145
7.46.1	Detailed Description	146
7.46.2	Member Data Documentation	146
7.46.2.1	id	146

7.46.2.2 rotation	146
7.46.2.3 x	146
7.46.2.4 y	146
7.47 ecs::Velocity Struct Reference	147
7.47.1 Detailed Description	147
7.47.2 Member Data Documentation	147
7.47.2.1 id	147
7.47.2.2 x	148
7.47.2.3 y	148
8 File Documentation	149
8.1 /home/masina/Projects/Epitech/rtype/client/include/Client/ArgsHandler.hpp File Reference	149
8.1.1 Detailed Description	150
8.2 ArgsHandler.hpp	151
8.3 /home/masina/Projects/Epitech/rtype/server/include/Server/ArgsHandler.hpp File Reference	151
8.3.1 Detailed Description	153
8.4 ArgsHandler.hpp	153
8.5 /home/masina/Projects/Epitech/rtype/client/include/Client/Client.hpp File Reference	153
8.5.1 Detailed Description	154
8.6 Client.hpp	155
8.7 /home/masina/Projects/Epitech/rtype/client/include/Client/Common.hpp File Reference	155
8.7.1 Detailed Description	157
8.8 Common.hpp	157
8.9 /home/masina/Projects/Epitech/rtype/client/include/Client/Generated/Version.hpp File Reference	158
8.9.1 Macro Definition Documentation	158
8.9.1.1 BUILD_TYPE	158
8.9.1.2 GIT_COMMIT_HASH	158
8.9.1.3 GIT_TAG	159
8.9.1.4 PROJECT_NAME	159
8.9.1.5 PROJECT_VERSION	159
8.9.1.6 PROJECT_VERSION_MAJOR	159
8.9.1.7 PROJECT_VERSION_MINOR	159
8.9.1.8 PROJECT_VERSION_PATCH	159
8.10 Version.hpp	160
8.11 /home/masina/Projects/Epitech/rtype/server/include/Server/Generated/Version.hpp File Reference	160
8.11.1 Macro Definition Documentation	161
8.11.1.1 BUILD_TYPE	161
8.11.1.2 GIT_COMMIT_HASH	161
8.11.1.3 GIT_TAG	161
8.11.1.4 PROJECT_NAME	161

8.11.1.5 PROJECT_VERSION	161
8.11.1.6 PROJECT_VERSION_MAJOR	161
8.11.1.7 PROJECT_VERSION_MINOR	161
8.11.1.8 PROJECT_VERSION_PATCH	162
8.12 Version.hpp	162
8.13 /home/masina/Projects/Epitech/rtype/client/src/argsHandler.cpp File Reference	162
8.13.1 Macro Definition Documentation	163
8.13.1.1 APP_EXTENSION	163
8.13.2 Variable Documentation	163
8.13.2.1 HELP_MESSAGE	163
8.13.2.2 VERSION_MESSAGE	163
8.14 argsHandler.cpp	163
8.15 /home/masina/Projects/Epitech/rtype/server/src/argsHandler.cpp File Reference	165
8.15.1 Macro Definition Documentation	165
8.15.1.1 APP_EXTENSION	165
8.15.2 Variable Documentation	166
8.15.2.1 HELP_MESSAGE	166
8.15.2.2 VERSION_MESSAGE	166
8.16 argsHandler.cpp	166
8.17 /home/masina/Projects/Epitech/rtype/client/src/client.cpp File Reference	167
8.18 client.cpp	168
8.19 /home/masina/Projects/Epitech/rtype/client/src/main.cpp File Reference	169
8.19.1 Function Documentation	170
8.19.1.1 main()	170
8.20 main.cpp	170
8.21 /home/masina/Projects/Epitech/rtype/server/src/main.cpp File Reference	171
8.21.1 Function Documentation	171
8.21.1.1 main()	171
8.22 main.cpp	172
8.23 /home/masina/Projects/Epitech/rtype/modules/Audio/SFMLAudio/include/SFMLAudio/SFMLAudio.hpp File Reference	172
8.23.1 Detailed Description	173
8.24 SFMLAudio.hpp	173
8.25 /home/masina/Projects/Epitech/rtype/modules/Audio/SFMLAudio/src/SFMLAudio.cpp File Reference	174
8.26 SFMLAudio.cpp	174
8.27 /home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.hpp File Reference	175
8.27.1 Detailed Description	176
8.28 Component.hpp	177
8.29 /home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Entity.hpp File Reference	177
8.29.1 Detailed Description	179
8.30 Entity.hpp	179

8.31	/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Registry.hpp File Reference	179
8.31.1	Detailed Description	180
8.32	Registry.hpp	181
8.33	/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Systems.hpp File Reference	182
8.33.1	Detailed Description	183
8.34	Systems.hpp	183
8.35	/home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/Systems.hpp File Reference	185
8.35.1	Detailed Description	186
8.36	Systems.hpp	186
8.37	/home/masina/Projects/Epitech/rtype/modules/ECS/src/registry.cpp File Reference	188
8.38	registry.cpp	188
8.39	/home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/Engine.hpp File Reference	188
8.39.1	Detailed Description	189
8.40	Engine.hpp	189
8.41	/home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/Scene.hpp File Reference	190
8.41.1	Detailed Description	192
8.42	Scene.hpp	192
8.43	/home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/SceneManager.hpp File Reference	192
8.43.1	Detailed Description	193
8.44	SceneManager.hpp	193
8.45	/home/masina/Projects/Epitech/rtype/modules/Engine/src/Engine.cpp File Reference	194
8.46	Engine.cpp	194
8.47	/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/AGameClient.hpp File Reference	195
8.47.1	Detailed Description	195
8.48	AGameClient.hpp	196
8.49	/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/AGameServer.hpp File Reference	196
8.49.1	Detailed Description	197
8.50	AGameServer.hpp	197
8.51	/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IAudio.hpp File Reference	197
8.51.1	Detailed Description	198
8.52	IAudio.hpp	198
8.53	/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IGameClient.hpp File Reference	199
8.53.1	Detailed Description	199
8.54	IGameClient.hpp	200
8.55	/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IGameServer.hpp File Reference	200

8.55.1 Detailed Description	201
8.56 IGameServer.hpp	201
8.57 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/INetworkClient.hpp File Reference	201
8.57.1 Detailed Description	202
8.58 INetworkClient.hpp	202
8.59 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/INetworkServer.hpp File Reference	203
8.59.1 Detailed Description	203
8.60 INetworkServer.hpp	203
8.61 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IRenderer.hpp File Reference	204
8.61.1 Detailed Description	205
8.62 IRenderer.hpp	205
8.63 /home/masina/Projects/Epitech/rtype/modules/Renderer/SFMLRenderer/include/SFMLRenderer/SFMLRenderer.hpp File Reference	207
8.63.1 Detailed Description	208
8.64 SFMLRenderer.hpp	208
8.65 /home/masina/Projects/Epitech/rtype/modules/Renderer/SFMLRenderer/src/SFMLRenderer.cpp File Reference	209
8.65.1 Function Documentation	210
8.65.1.1 scancodeToKey()	210
8.66 SFMLRenderer.cpp	210
8.67 /home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/Clock.hpp File Reference	213
8.67.1 Detailed Description	214
8.68 Clock.hpp	214
8.69 /home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/Logger.hpp File Reference	216
8.70 Logger.hpp	217
8.71 /home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/Utils.hpp File Reference	218
8.71.1 Detailed Description	219
8.72 Utils.hpp	219
8.73 /home/masina/Projects/Epitech/rtype/modules/Utils/src/logger.cpp File Reference	219
8.74 logger.cpp	220
8.75 /home/masina/Projects/Epitech/rtype/modules/Utils/src/utils.cpp File Reference	220
8.76 utils.cpp	220
8.77 /home/masina/Projects/Epitech/rtype/README.md File Reference	221
8.78 /home/masina/Projects/Epitech/rtype/server/include/Server/Server.hpp File Reference	221
8.78.1 Detailed Description	222
8.79 Server.hpp	222
8.80 /home/masina/Projects/Epitech/rtype/server/src/server.cpp File Reference	223
8.81 server.cpp	223

Chapter 1

r-type

1.1 R-Type

The Goal of this project is to implement a multithreaded server and a graphical client for a game called R-Type, using an engine of your own design.

1.1.1 Supported Platforms

Platform	Compiler	Status
Linux	g++	
macOS	g++	
Windows	MSVC	

1.1.2 Project Structure

```
R-Type
assets          # Game assets (images, sounds, etc.)
client          # Client source code
documentation   # Project documentation
modules         # Static libraries for the project
scripts         # Build and utility scripts
server          # Server source code
tests           # Unit and integration tests
third-party     # External libraries as submodules
flowchart LR
    subgraph App
        subgraph client [Client]
            A[Client]
            A -->|.a/.lib| E[IGameClient]
            A -->|.a/.lib| B[Engine]
            B -->|.a/.lib| D[IRenderer]
            B -->|.a/.lib| F[INetworkClient]
            B -->|.a/.lib| G[IAudio]
            B -->|.a/.lib| K[ECS]
        end

        subgraph server [Server]
            H[Server]
            H -->|.a/.lib| I[INetworkServer]
            H -->|.a/.lib| J[IGameServer]
        end

        A <==>|TCP/UDP| H
    end
```

1.1.3 Prerequisites

Make sure you have the following dependencies installed on your system:

- [CMake 4.0.0](#)
- [C++23](#)

1.1.4 Clone the project

Important

When cloning the project, you should also initialize the submodules:

```
git clone --recurse-submodules git@github.com:bobis33/R-Type.git
```

If you already cloned the project, you can initialize the submodules with:

```
git submodule update --init --recursive
```

1.1.5 Build and Run

1.1.5.1 Unix (Linux, macOS)

```
./scripts/unix/build.sh release
## Or
cmake -S . -B cmake-build-release -G "Ninja" -DCMAKE_BUILD_TYPE=Release -DCMAKE_CXX_COMPILER=g++
-DCMAKE_C_COMPILER=gcc
cmake --build cmake-build-release -- -j4
## Then
./cmake-build-release/r-type_client ## client
./cmake-build-release/r-type_server ## server
```

1.1.5.2 Windows

```
cmake -S . -B cmake-build-release -G "Visual Studio 17 2022" -A x64 -DCMAKE_BUILD_TYPE=Release
cmake --build cmake-build-release --config Release
## Then
cmake-build-release\bin\r-type_client.exe ## client
cmake-build-release\bin\r-type_server.exe ## server
```

1.1.6 Documentation

API documentation is generated using Doxygen and deployed on [GitHub Pages](#). You can find the same documentation as PDF [here](#). More specific documentation for each part of the project can be found in their respective directories:

- [Client documentation](#)
- [Server documentation](#)

1.1.7 External Libraries

All dependencies are included as submodules in the [third-party](#) directory.

1.1.8 Commit Norms

Commit Type	Description
build	Changes that affect the build system or external dependencies (npm, make, etc.)
ci	Changes related to integration files and scripts or configuration (Travis, Ansible, BrowserStack, etc.)
feat	Addition of a new feature
fix	Bug fix
perf	Performance improvements
refactor	Modification that neither adds a new feature nor improves performance
style	Change that does not affect functionality or semantics (indentation, formatting, adding space, renaming a variable, etc.)
docs	Writing or updating documentation
test	Addition or modification of tests

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

cli	13
cli::Config	14
cli::Config::Audio	14
cli::Config::Window	14
cli::Paths	15
cli::Paths::Audio	15
cli::Paths::Fonts	16
ecs	16
eng	17
srv	20
utl	20

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

cli::ArgsConfig	29
srv::ArgsConfig	31
cli::ArgsHandler	33
srv::ArgsHandler	36
ecs::Audio	39
eng::Audio	40
ecs::AudioSystem	42
cli::Client	48
utl::Clock	51
ecs::Color	58
eng::Color	60
eng::Engine	61
cli::EnvConfig	67
srv::EnvConfig	67
eng::Event	68
ecs::Font	69
eng::Font	70
ecs::FontSystem	71
eng::IAudio	77
eng::SFMLAudio	121
cli::IGameClient	79
cli::AGameClient	23
srv::IGameServer	81
srv::AGameServer	26
eng::SFMLAudio::Impl	83
eng::SFMLRenderer::Impl	84
eng::INetworkClient	85
srv::INetworkServer	86
ecs::Registry::IPool	87
ecs::Registry::Pool< T >	102
eng::IRenderer	88
eng::SFMLRenderer	125
eng::ISystem	94
eng::AudioSystem	44

eng::FontSystem	73
eng::TextSyStem	141
utl::Logger	95
ecs::Registry	105
eng::Scene	113
eng::SceneManager	116
srv::Server	119
ecs::Sprite	132
ecs::Text	134
eng::Text	135
ecs::TextSyStem	138
ecs::Transform	145
ecs::Velocity	147

Chapter 4

Class Index

4.1 Class List

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

cli::AGameClient	
Abstraction for the games	23
srv::AGameServer	
Abstraction for the games	26
cli::ArgsConfig	29
srv::ArgsConfig	31
cli::ArgsHandler	
Class to handle command line arguments	33
srv::ArgsHandler	
Class to handle command line arguments	36
ecs::Audio	39
eng::Audio	40
ecs::AudioSystem	
Class for managing entities and their components	42
eng::AudioSystem	
Class for managing entities and their components	44
cli::Client	
Class for the client	48
utl::Clock	
Class for clock	51
ecs::Color	58
eng::Color	60
eng::Engine	
Class for the game engine	61
cli::EnvConfig	67
srv::EnvConfig	67
eng::Event	68
ecs::Font	69
eng::Font	70
ecs::FontSystem	
Class for managing entities and their components	71
eng::FontSystem	
Class for managing entities and their components	73
eng::IAudio	
Interface for the audio	77

cli::IGameClient	
Interface for the games	79
srv::IGameServer	
Interface for the games	81
eng::SFMLAudio::Impl	83
eng::SFMLRenderer::Impl	84
eng::INetworkClient	
Interface for the client network	85
srv::INetworkServer	
Interface for the server network	86
ecs::Registry::IPool	87
eng::IRenderer	
Interface for the renderer	88
eng::ISystem	94
utl::Logger	95
ecs::Registry::Pool< T >	102
ecs::Registry	
Class for managing entities and their components	105
eng::Scene	
Class for scene and manage entities	113
eng::SceneManager	
Class for managing scenes	116
srv::Server	
Class for the server	119
eng::SFMLAudio	
Class for audio management	121
eng::SFMLRenderer	
Class for the R-Type game	125
ecs::Sprite	132
ecs::Text	134
eng::Text	135
ecs::TextSyStem	
Class for managing entities and their components	138
eng::TextSyStem	
Class for managing entities and their components	141
ecs::Transform	145
ecs::Velocity	147

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

/home/masina/Projects/Epitech/rtype/client/include/Client/ ArgsHandler.hpp	
This file contains the ArgsHandler class declaration	149
/home/masina/Projects/Epitech/rtype/client/include/Client/ Client.hpp	
This file contains the Client class declaration	153
/home/masina/Projects/Epitech/rtype/client/include/Client/ Common.hpp	
This file contains common definitions and constants	155
/home/masina/Projects/Epitech/rtype/client/include/Client/Generated/ Version.hpp	158
/home/masina/Projects/Epitech/rtype/client/src/ argsHandler.cpp	162
/home/masina/Projects/Epitech/rtype/client/src/ client.cpp	167
/home/masina/Projects/Epitech/rtype/client/src/ main.cpp	169
/home/masina/Projects/Epitech/rtype/modules/Audio/SFMLAudio/include/SFMLAudio/ SFMLAudio.hpp	
SFMLAudio class declaration	172
/home/masina/Projects/Epitech/rtype/modules/Audio/SFMLAudio/src/ SFMLAudio.cpp	174
/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/ Component.hpp	
This file contains the component definitions	175
/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/ Entity.hpp	
This file contains the entity definitions	177
/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/ Registry.hpp	
This file contains the Registry class declaration	179
/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/ Systems.hpp	
This file contains the system definitions	182
/home/masina/Projects/Epitech/rtype/modules/ECS/src/ registry.cpp	188
/home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/ Engine.hpp	
This file contains the Engine class declaration	188
/home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/ Scene.hpp	
This file contains the Scene class	190
/home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/ SceneManager.hpp	
This file contains the SceneManager class declaration	192
/home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/ Systems.hpp	
This file contains the system definitions	185
/home/masina/Projects/Epitech/rtype/modules/Engine/src/ Engine.cpp	194
/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/ AGameClient.hpp	
This file contains the game abstract class	195
/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/ AGameServer.hpp	
This file contains the game abstract class	196

/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/ IAudio.hpp	
This file contains the Audio interface	197
/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/ IGameClient.hpp	
This file contains the Game interface	199
/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/ IGameServer.hpp	
This file contains the Game interface	200
/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/ INetworkClient.hpp	
This file contains the client network interface	201
/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/ INetworkServer.hpp	
This file contains the server network interface	203
/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/ IRenderer.hpp	
This file contains the IRenderer class declaration	204
/home/masina/Projects/Epitech/rtype/modules/Renderer/SFMLRenderer/include/SFMLRenderer/ SFMLRenderer.h	
SFMLRenderer class declaration with PImpl	207
/home/masina/Projects/Epitech/rtype/modules/Renderer/SFMLRenderer/src/ SFMLRenderer.cpp	209
/home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/ Clock.hpp	
This file contains the Clock class	213
/home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/ Logger.hpp	216
/home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/ Utils.hpp	
This file contains utility functions	218
/home/masina/Projects/Epitech/rtype/modules/Utils/src/ logger.cpp	219
/home/masina/Projects/Epitech/rtype/modules/Utils/src/ utils.cpp	220
/home/masina/Projects/Epitech/rtype/server/include/Server/ ArgsHandler.hpp	
This file contains the ArgsHandler class declaration	151
/home/masina/Projects/Epitech/rtype/server/include/Server/ Server.hpp	
This file contains the Server class declaration	221
/home/masina/Projects/Epitech/rtype/server/include/Server/Generated/ Version.hpp	160
/home/masina/Projects/Epitech/rtype/server/src/ argsHandler.cpp	165
/home/masina/Projects/Epitech/rtype/server/src/ main.cpp	171
/home/masina/Projects/Epitech/rtype/server/src/ server.cpp	223

Chapter 6

Namespace Documentation

6.1 cli Namespace Reference

Namespaces

- namespace [Config](#)
- namespace [Paths](#)

Classes

- class [AGameClient](#)
Abstraction for the games.
- struct [ArgsConfig](#)
- class [ArgsHandler](#)
Class to handle command line arguments.
- class [Client](#)
Class for the client.
- struct [EnvConfig](#)
- class [IGameClient](#)
Interface for the games.

Typedefs

- using [json](#) = nlohmann::json

6.1.1 Typedef Documentation

6.1.1.1 json

using [cli::json](#) = nlohmann::json

Definition at line [16](#) of file [ArgsHandler.hpp](#).

6.2 cli::Config Namespace Reference

Namespaces

- namespace [Audio](#)
- namespace [Window](#)

6.3 cli::Config::Audio Namespace Reference

Variables

- constexpr auto [DEFAULT_AUDIO_VOLUME](#) = 50
- constexpr auto [DEFAULT_AUDIO_MUTED](#) = false

6.3.1 Variable Documentation

6.3.1.1 DEFAULT_AUDIO_MUTED

auto cli::Config::Audio::DEFAULT_AUDIO_MUTED = false [inline], [constexpr]

Definition at line 36 of file [Common.hpp](#).

6.3.1.2 DEFAULT_AUDIO_VOLUME

auto cli::Config::Audio::DEFAULT_AUDIO_VOLUME = 50 [inline], [constexpr]

Definition at line 35 of file [Common.hpp](#).

6.4 cli::Config::Window Namespace Reference

Variables

- constexpr auto [DEFAULT_WINDOW_WIDTH](#) = 960
- constexpr auto [DEFAULT_WINDOW_HEIGHT](#) = 540
- constexpr auto [DEFAULT_WINDOW_FRAME_LIMIT](#) = 240
- constexpr auto [DEFAULT_WINDOW_FULLSCREEN](#) = false

6.4.1 Variable Documentation

6.4.1.1 DEFAULT_WINDOW_FRAME_LIMIT

auto cli::Config::Window::DEFAULT_WINDOW_FRAME_LIMIT = 240 [inline], [constexpr]

Definition at line 30 of file [Common.hpp](#).

6.4.1.2 DEFAULT_WINDOW_FULLSCREEN

```
auto cli::Config::Window::DEFAULT_WINDOW_FULLSCREEN = false [inline], [constexpr]
```

Definition at line 31 of file [Common.hpp](#).

6.4.1.3 DEFAULT_WINDOW_HEIGHT

```
auto cli::Config::Window::DEFAULT_WINDOW_HEIGHT = 540 [inline], [constexpr]
```

Definition at line 29 of file [Common.hpp](#).

6.4.1.4 DEFAULT_WINDOW_WIDTH

```
auto cli::Config::Window::DEFAULT_WINDOW_WIDTH = 960 [inline], [constexpr]
```

Definition at line 28 of file [Common.hpp](#).

6.5 cli::Paths Namespace Reference

Namespaces

- namespace [Audio](#)
- namespace [Fonts](#)

6.6 cli::Paths::Audio Namespace Reference

Variables

- constexpr auto [AUDIO_TITLE](#) = "assets/audio/title.mp3"
- constexpr auto [AUDIO_COIN](#) = "assets/audio/coin.mp3"
- constexpr auto [AUDIO_BATTLE_THEME](#) = "assets/audio/battle_theme.mp3"

6.6.1 Variable Documentation

6.6.1.1 AUDIO_BATTLE_THEME

```
auto cli::Paths::Audio::AUDIO_BATTLE_THEME = "assets/audio/battle_theme.mp3" [inline], [constexpr]
```

Definition at line 17 of file [Common.hpp](#).

6.6.1.2 AUDIO_COIN

```
auto cli::Paths::Audio::AUDIO_COIN = "assets/audio/coin.mp3" [inline], [constexpr]
```

Definition at line 16 of file [Common.hpp](#).

6.6.1.3 AUDIO_TITLE

auto cli::Paths::Audio::AUDIO_TITLE = "assets/audio/title.mp3" [inline], [constexpr]

Definition at line 15 of file [Common.hpp](#).

Referenced by [cli::Client::Client\(\)](#).

6.7 cli::Paths::Fonts Namespace Reference

Variables

- constexpr auto [FONTS_RTYPE](#) = "assets/fonts/r-type.otf"

6.7.1 Variable Documentation

6.7.1.1 FONTS_RTYPE

auto cli::Paths::Fonts::FONTS_RTYPE = "assets/fonts/r-type.otf" [inline], [constexpr]

Definition at line 21 of file [Common.hpp](#).

Referenced by [cli::Client::Client\(\)](#).

6.8 ecs Namespace Reference

Classes

- struct [Audio](#)
- class [AudioSystem](#)
Class for managing entities and their components.
- struct [Color](#)
- struct [Font](#)
- class [FontSystem](#)
Class for managing entities and their components.
- class [Registry](#)
Class for managing entities and their components.
- struct [Sprite](#)
- struct [Text](#)
- class [TextSyStem](#)
Class for managing entities and their components.
- struct [Transform](#)
- struct [Velocity](#)

Typedefs

- using [Entity](#) = std::uint32_t

Variables

- constexpr [Entity](#) `INVALID_ENTITY` = 0

6.8.1 Typedef Documentation

6.8.1.1 Entity

using [ecs::Entity](#) = std::uint32_t

Definition at line 13 of file [Entity.hpp](#).

6.8.2 Variable Documentation

6.8.2.1 INVALID_ENTITY

[Entity](#) `ecs::INVALID_ENTITY` = 0 [constexpr]

Definition at line 14 of file [Entity.hpp](#).

6.9 eng Namespace Reference

Classes

- struct [Audio](#)
- class [AudioSystem](#)
 - Class for managing entities and their components.
- struct [Color](#)
- class [Engine](#)
 - Class for the game engine.
- struct [Event](#)
- struct [Font](#)
- class [FontSystem](#)
 - Class for managing entities and their components.
- class [IAudio](#)
 - Interface for the audio.
- class [INetworkClient](#)
 - Interface for the client network.
- class [IRenderer](#)
 - Interface for the renderer.
- class [ISystem](#)
- class [Scene](#)
 - class for scene and manage entities
- class [SceneManager](#)
 - Class for managing scenes.
- class [SFMLAudio](#)
 - Class for audio management.
- class [SFMLRenderer](#)
 - Class for the R-Type game.
- struct [Text](#)
- class [TextSyStem](#)
 - Class for managing entities and their components.

Typedefs

- using [scene_id_t](#) = unsigned int

Enumerations

- enum class [Key](#) {
[Unknown](#) , [Escape](#) , [Space](#) , [Up](#) ,
[Down](#) , [Left](#) , [Right](#) , [A](#) ,
[B](#) , [C](#) , [D](#) , [E](#) ,
[F](#) , [G](#) , [H](#) , [I](#) ,
[J](#) , [K](#) , [L](#) , [M](#) ,
[N](#) , [O](#) , [P](#) , [Q](#) ,
[R](#) , [S](#) , [T](#) , [U](#) ,
[V](#) , [W](#) , [X](#) , [Y](#) ,
[Z](#) , [Num0](#) , [Num1](#) , [Num2](#) ,
[Num3](#) , [Num4](#) , [Num5](#) , [Num6](#) ,
[Num7](#) , [Num8](#) , [Num9](#) }
- enum class [EventType](#) { [Closed](#) , [KeyPressed](#) , [KeyReleased](#) , [None](#) }

6.9.1 Typedef Documentation

6.9.1.1 scene_id_t

using [eng::scene_id_t](#) = unsigned int

Definition at line 14 of file [Scene.hpp](#).

6.9.2 Enumeration Type Documentation

6.9.2.1 EventType

enum class [eng::EventType](#) [strong]

Enumerator

Closed	
KeyPressed	
KeyReleased	
None	

Definition at line 89 of file [IRenderer.hpp](#).

6.9.2.2 Key

enum class [eng::Key](#) [strong]

Enumerator

Unknown	
Escape	
Space	
Up	
Down	
Left	
Right	
A	
B	
C	
D	
E	
F	
G	
H	
I	
J	
K	
L	
M	
N	
O	
P	
Q	
R	
S	
T	
U	
V	
W	
X	
Y	
Z	
Num0	
Num1	
Num2	
Num3	
Num4	
Num5	
Num6	
Num7	
Num8	
Num9	

Definition at line 43 of file [IRenderer.hpp](#).

6.10 srv Namespace Reference

Classes

- class [AGameServer](#)
Abstraction for the games.
- struct [ArgsConfig](#)
- class [ArgsHandler](#)
Class to handle command line arguments.
- struct [EnvConfig](#)
- class [IGameServer](#)
Interface for the games.
- class [INetworkServer](#)
Interface for the server network.
- class [Server](#)
Class for the server.

Typedefs

- using [json](#) = nlohmann::json

6.10.1 Typedef Documentation

6.10.1.1 json

using [srv::json](#) = nlohmann::json

Definition at line 16 of file [ArgsHandler.hpp](#).

6.11 utl Namespace Reference

Classes

- class [Clock](#)
Class for clock.
- class [Logger](#)

Enumerations

- enum class [LogLevel](#) : uint8_t { [INFO](#) , [WARNING](#) }

Functions

- std::vector< char > [readFile](#) (const std::string &filename)

6.11.1 Enumeration Type Documentation

6.11.1.1 LogLevel

enum class [utl::LogLevel](#) : uint8_t [strong]

Enumerator

INFO	
WARNING	

Definition at line 11 of file [Logger.hpp](#).

6.11.2 Function Documentation

6.11.2.1 readFile()

```
std::vector< char > utl::readFile (  
    const std::string & filename)  [nodiscard]
```

Definition at line 5 of file [utils.cpp](#).

Chapter 7

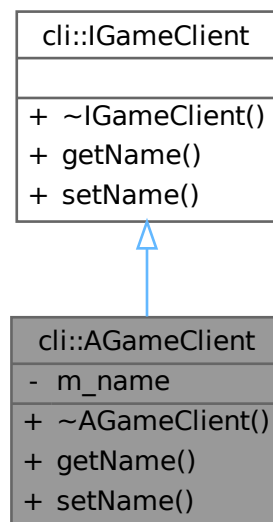
Class Documentation

7.1 cli::AGameClient Class Reference

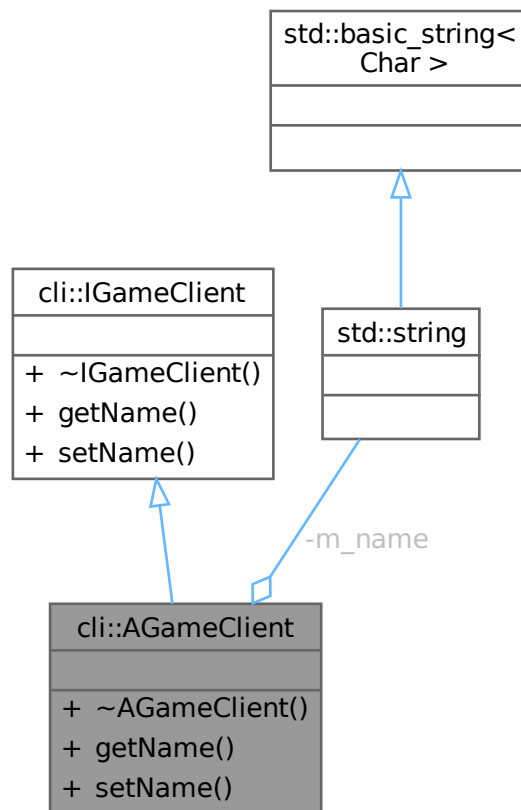
Abstraction for the games.

```
#include <AGameClient.hpp>
```

Inheritance diagram for cli::AGameClient:



Collaboration diagram for `cli::AGameClient`:



Public Member Functions

- `~AGameClient()` override=default
- `std::string & getName()` override
- `void setName (const std::string &newName)` override

Public Member Functions inherited from `cli::IGameClient`

- virtual `~IGameClient()`=default

Private Attributes

- `std::string m_name = "default_name"`

7.1.1 Detailed Description

Abstraction for the games.

Definition at line 21 of file `AGameClient.hpp`.

7.1.2 Constructor & Destructor Documentation

7.1.2.1 ~AGameClient()

cli::AGameClient::~~AGameClient () [override], [default]

7.1.3 Member Function Documentation

7.1.3.1 getName()

std::string & cli::AGameClient::getName () [inline], [nodiscard], [override], [virtual]

Reimplemented from [cli::IGameClient](#).

Definition at line 26 of file [AGameClient.hpp](#).

References [m_name](#).

7.1.3.2 setName()

void cli::AGameClient::setName (
 const std::string & newName) [inline], [override], [virtual]

Reimplemented from [cli::IGameClient](#).

Definition at line 27 of file [AGameClient.hpp](#).

References [m_name](#).

7.1.4 Member Data Documentation

7.1.4.1 m_name

std::string cli::AGameClient::m_name = "default_name" [private]

Definition at line 30 of file [AGameClient.hpp](#).

Referenced by [getName\(\)](#), and [setName\(\)](#).

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

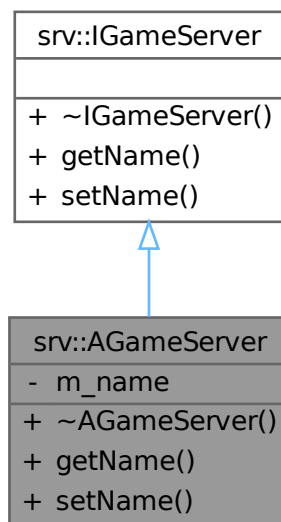
- [/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/AGameClient.hpp](#)

7.2 srv::AGameServer Class Reference

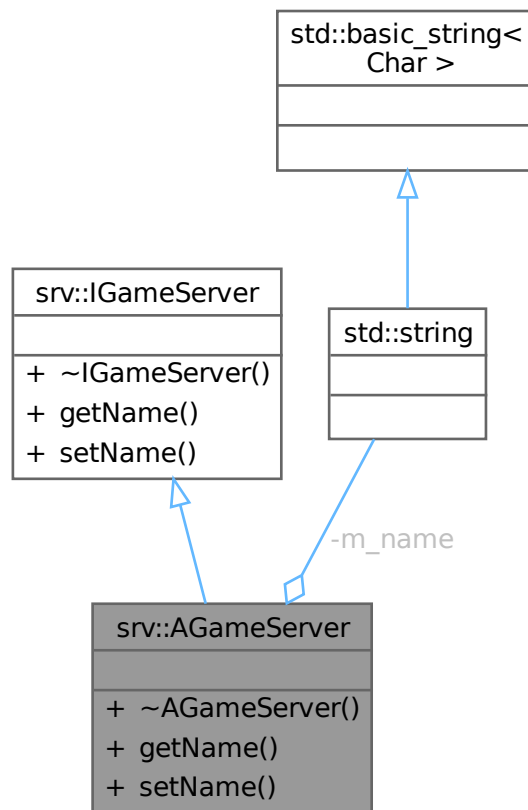
Abstraction for the games.

```
#include <AGameServer.hpp>
```

Inheritance diagram for srv::AGameServer:



Collaboration diagram for `srv::AGameServer`:



Public Member Functions

- `~AGameServer()` override=default
- `std::string & getName()` override
- `void setName (const std::string &newName)` override

Public Member Functions inherited from `srv::IGameServer`

- virtual `~IGameServer()`=default

Private Attributes

- `std::string m_name` = "default_name"

7.2.1 Detailed Description

Abstraction for the games.

Definition at line 21 of file `AGameServer.hpp`.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 ~AGameServer()

srv::AGameServer::~~AGameServer () [override], [default]

7.2.3 Member Function Documentation

7.2.3.1 getName()

std::string & srv::AGameServer::getName () [inline], [nodiscard], [override], [virtual]

Reimplemented from [srv::IGameServer](#).

Definition at line 26 of file [AGameServer.hpp](#).

References [m_name](#).

7.2.3.2 setName()

void srv::AGameServer::setName (
 const std::string & newName) [inline], [override], [virtual]

Reimplemented from [srv::IGameServer](#).

Definition at line 27 of file [AGameServer.hpp](#).

References [m_name](#).

7.2.4 Member Data Documentation

7.2.4.1 m_name

std::string srv::AGameServer::m_name = "default_name" [private]

Definition at line 30 of file [AGameServer.hpp](#).

Referenced by [getName\(\)](#), and [setName\(\)](#).

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

- [/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/AGameServer.hpp](#)

7.3 cli::ArgsConfig Struct Reference

#include <ArgsHandler.hpp>

Collaboration diagram for cli::ArgsConfig:

cli::ArgsConfig
+ exit
+ width
+ height
+ frameLimit
+ fullscreen
+ fromFile()

Static Public Member Functions

- static [ArgsConfig fromFile](#) (const std::string &path)

Public Attributes

- bool [exit](#) = false
- unsigned int [width](#) = [Config::Window::DEFAULT_WINDOW_WIDTH](#)
- unsigned int [height](#) = [Config::Window::DEFAULT_WINDOW_HEIGHT](#)
- unsigned int [frameLimit](#) = [Config::Window::DEFAULT_WINDOW_FRAME_LIMIT](#)
- bool [fullscreen](#) = [Config::Window::DEFAULT_WINDOW_FULLSCREEN](#)

7.3.1 Detailed Description

Definition at line 18 of file [ArgsHandler.hpp](#).

7.3.2 Member Function Documentation

7.3.2.1 fromFile()

[cli::ArgsConfig](#) cli::ArgsConfig::fromFile (
 const std::string & path) [static]

Definition at line 26 of file [argsHandler.cpp](#).

References [frameLimit](#), [fullscreen](#), [height](#), and [width](#).

Referenced by [cli::ArgsHandler::ParseArgs\(\)](#).

Here is the caller graph for this function:



7.3.3 Member Data Documentation

7.3.3.1 `exit`

```
bool cli::ArgsConfig::exit = false
```

Definition at line 20 of file [ArgsHandler.hpp](#).

Referenced by [main\(\)](#).

7.3.3.2 `frameLimit`

```
unsigned int cli::ArgsConfig::frameLimit = Config::Window::DEFAULT_WINDOW_FRAME_LIMIT
```

Definition at line 23 of file [ArgsHandler.hpp](#).

Referenced by [cli::Client::Client\(\)](#), and [fromFile\(\)](#).

7.3.3.3 `fullscreen`

```
bool cli::ArgsConfig::fullscreen = Config::Window::DEFAULT_WINDOW_FULLSCREEN
```

Definition at line 24 of file [ArgsHandler.hpp](#).

Referenced by [cli::Client::Client\(\)](#), and [fromFile\(\)](#).

7.3.3.4 `height`

```
unsigned int cli::ArgsConfig::height = Config::Window::DEFAULT_WINDOW_HEIGHT
```

Definition at line 22 of file [ArgsHandler.hpp](#).

Referenced by [cli::Client::Client\(\)](#), and [fromFile\(\)](#).

7.3.3.5 width

unsigned int cli::ArgsConfig::width = [Config::Window::DEFAULT_WINDOW_WIDTH](#)

Definition at line 21 of file [ArgsHandler.hpp](#).

Referenced by [cli::Client::Client\(\)](#), and [fromFile\(\)](#).

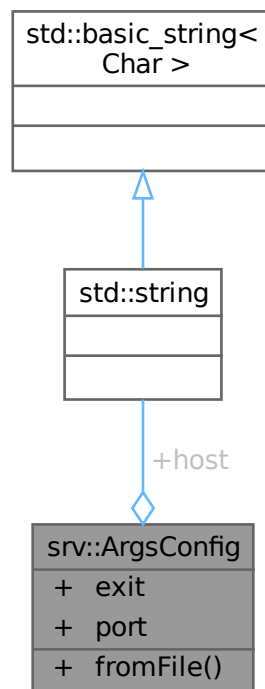
The documentation for this struct was generated from the following files:

- [/home/masina/Projects/Epitech/rtype/client/include/Client/ArgsHandler.hpp](#)
- [/home/masina/Projects/Epitech/rtype/client/src/argsHandler.cpp](#)

7.4 srv::ArgsConfig Struct Reference

`#include <ArgsHandler.hpp>`

Collaboration diagram for `srv::ArgsConfig`:



Static Public Member Functions

- static [ArgsConfig fromFile](#) (const std::string &path)

Public Attributes

- bool `exit` = false
- std::string `host` = "0.0.0.0"
- unsigned int `port` = 2560

7.4.1 Detailed Description

Definition at line 18 of file [ArgsHandler.hpp](#).

7.4.2 Member Function Documentation

7.4.2.1 fromFile()

[srv::ArgsConfig](#) `srv::ArgsConfig::fromFile` (
 const std::string & path) [static]

Definition at line 26 of file [argsHandler.cpp](#).

References [host](#), and [port](#).

Referenced by [srv::ArgsHandler::ParseArgs\(\)](#).

Here is the caller graph for this function:



7.4.3 Member Data Documentation

7.4.3.1 exit

bool `srv::ArgsConfig::exit` = false

Definition at line 20 of file [ArgsHandler.hpp](#).

Referenced by [main\(\)](#).

7.4.3.2 host

std::string `srv::ArgsConfig::host` = "0.0.0.0"

Definition at line 21 of file [ArgsHandler.hpp](#).

Referenced by [fromFile\(\)](#).

7.4.3.3 port

```
unsigned int srv::ArgsConfig::port = 2560
```

Definition at line 22 of file [ArgsHandler.hpp](#).

Referenced by [fromFile\(\)](#).

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

- [/home/masina/Projects/Epitech/rtype/server/include/Server/ArgsHandler.hpp](#)
- [/home/masina/Projects/Epitech/rtype/server/src/argsHandler.cpp](#)

7.5 cli::ArgsHandler Class Reference

Class to handle command line arguments.

```
#include <ArgsHandler.hpp>
```

Collaboration diagram for cli::ArgsHandler:

cli::ArgsHandler
<ul style="list-style-type: none"> + ArgsHandler() + ~ArgsHandler() + ArgsHandler() + operator=() + ArgsHandler() + operator=() + ParseArgs() + ParseEnv()

Public Member Functions

- [ArgsHandler](#) ()=default
- [~ArgsHandler](#) ()=default
- [ArgsHandler](#) (const [ArgsHandler](#) &)=delete
- [ArgsHandler](#) & operator= (const [ArgsHandler](#) &)=delete
- [ArgsHandler](#) ([ArgsHandler](#) &&)=delete
- [ArgsHandler](#) & operator= ([ArgsHandler](#) &&)=delete

Static Public Member Functions

- static [ArgsConfig ParseArgs](#) (int argc, const char *const argv[])
- static [EnvConfig ParseEnv](#) (const char *const env[])

7.5.1 Detailed Description

Class to handle command line arguments.

Definition at line 36 of file [ArgsHandler.hpp](#).

7.5.2 Constructor & Destructor Documentation

7.5.2.1 ArgsHandler() [1/3]

cli::ArgsHandler::ArgsHandler () [default]

7.5.2.2 ~ArgsHandler()

cli::ArgsHandler::~~ArgsHandler () [default]

7.5.2.3 ArgsHandler() [2/3]

cli::ArgsHandler::ArgsHandler (
const [ArgsHandler](#) &) [delete]

7.5.2.4 ArgsHandler() [3/3]

cli::ArgsHandler::ArgsHandler (
[ArgsHandler](#) &&) [delete]

7.5.3 Member Function Documentation

7.5.3.1 operator=() [1/2]

[ArgsHandler](#) & cli::ArgsHandler::operator= (
[ArgsHandler](#) &&) [delete]

7.5.3.2 operator=() [2/2]

[ArgsHandler](#) & cli::ArgsHandler::operator= (
const [ArgsHandler](#) &) [delete]

7.5.3.3 ParseArgs()

```
cli::ArgsConfig cli::ArgsHandler::ParseArgs (
    int argc,
    const char *const argv[]) [static]
```

Definition at line 61 of file [argsHandler.cpp](#).

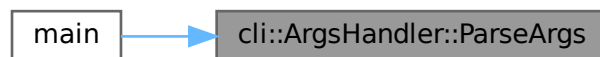
References [cli::ArgsConfig::fromFile\(\)](#), [HELP_MESSAGE](#), [utl::INFO](#), [utl::Logger::log\(\)](#), and [VERSION_MESSAGE](#).

Referenced by [main\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.5.3.4 ParseEnv()

```
cli::EnvConfig cli::ArgsHandler::ParseEnv (
    const char *const env[]) [static]
```

Definition at line 115 of file [argsHandler.cpp](#).

Referenced by [main\(\)](#).

Here is the caller graph for this function:



The documentation for this class was generated from the following files:

- [/home/masina/Projects/Epitech/rtype/client/include/Client/ArgsHandler.hpp](#)
- [/home/masina/Projects/Epitech/rtype/client/src/argsHandler.cpp](#)

7.6 `srv::ArgsHandler` Class Reference

Class to handle command line arguments.

`#include <ArgsHandler.hpp>`

Collaboration diagram for `srv::ArgsHandler`:

<code>srv::ArgsHandler</code>
<div>+ <code>ArgsHandler()</code> + <code>~ArgsHandler()</code> + <code>ArgsHandler()</code> + <code>operator=()</code> + <code>ArgsHandler()</code> + <code>operator=()</code> + <code>ParseArgs()</code> + <code>ParseEnv()</code></div>

Public Member Functions

- [ArgsHandler](#) ()=default
- [~ArgsHandler](#) ()=default
- [ArgsHandler](#) (const [ArgsHandler](#) &)=delete
- [ArgsHandler](#) & [operator=](#) (const [ArgsHandler](#) &)=delete
- [ArgsHandler](#) ([ArgsHandler](#) &&)=delete
- [ArgsHandler](#) & [operator=](#) ([ArgsHandler](#) &&)=delete

Static Public Member Functions

- static [ArgsConfig](#) [ParseArgs](#) (int argc, const char *const argv[])
- static [EnvConfig](#) [ParseEnv](#) (const char *const env[])

7.6.1 Detailed Description

Class to handle command line arguments.

Definition at line 35 of file [ArgsHandler.hpp](#).

7.6.2 Constructor & Destructor Documentation

7.6.2.1 `ArgsHandler()` [1/3]

`srv::ArgsHandler::ArgsHandler ()` [default]

7.6.2.2 `~ArgsHandler()`

`srv::ArgsHandler::~~ArgsHandler ()` [default]

7.6.2.3 `ArgsHandler()` [2/3]

`srv::ArgsHandler::ArgsHandler (`
 `const ArgsHandler &)` [delete]

7.6.2.4 `ArgsHandler()` [3/3]

`srv::ArgsHandler::ArgsHandler (`
 `ArgsHandler &&)` [delete]

7.6.3 Member Function Documentation

7.6.3.1 `operator=()` [1/2]

`ArgsHandler & srv::ArgsHandler::operator= (`
 `ArgsHandler &&)` [delete]

7.6.3.2 `operator=()` [2/2]

`ArgsHandler & srv::ArgsHandler::operator= (`
 `const ArgsHandler &)` [delete]

7.6.3.3 ParseArgs()

```

srv::ArgsConfig srv::ArgsHandler::ParseArgs (
    int argc,
    const char *const argv[]) [static]

```

Definition at line 49 of file [argsHandler.cpp](#).

References [srv::ArgsConfig::fromFile\(\)](#), [HELP_MESSAGE](#), [utl::INFO](#), [utl::Logger::log\(\)](#), and [VERSION_MESSAGE](#).

Referenced by [main\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.6.3.4 ParseEnv()

```

srv::EnvConfig srv::ArgsHandler::ParseEnv (
    const char *const env[]) [static]

```

Definition at line 102 of file [argsHandler.cpp](#).

Referenced by [main\(\)](#).

Here is the caller graph for this function:



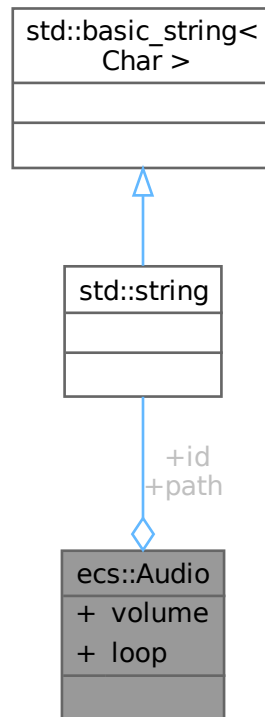
The documentation for this class was generated from the following files:

- [/home/masina/Projects/Epitech/rtype/server/include/Server/ArgsHandler.hpp](#)
- [/home/masina/Projects/Epitech/rtype/server/src/argsHandler.cpp](#)

7.7 ecs::Audio Struct Reference

```
#include <Component.hpp>
```

Collaboration diagram for ecs::Audio:



Public Attributes

- `std::string` [id](#)
- `std::string` [path](#)
- `float` [volume](#)
- `bool` [loop](#)

7.7.1 Detailed Description

Definition at line [13](#) of file [Component.hpp](#).

7.7.2 Member Data Documentation

7.7.2.1 id

`std::string` `ecs::Audio::id`

Definition at line [15](#) of file [Component.hpp](#).

7.7.2.2 loop

bool ecs::Audio::loop

Definition at line 18 of file [Component.hpp](#).

7.7.2.3 path

std::string ecs::Audio::path

Definition at line 16 of file [Component.hpp](#).

7.7.2.4 volume

float ecs::Audio::volume

Definition at line 17 of file [Component.hpp](#).

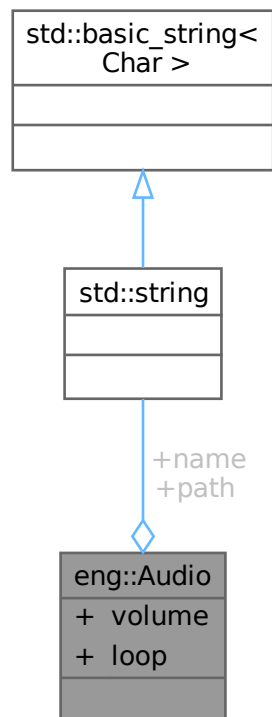
The documentation for this struct was generated from the following file:

- [/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.hpp](#)

7.8 eng::Audio Struct Reference

#include <IRenderer.hpp>

Collaboration diagram for eng::Audio:



Public Attributes

- std::string [path](#)
- float [volume](#)
- bool [loop](#)
- std::string [name](#)

7.8.1 Detailed Description

Definition at line [13](#) of file [IRenderer.hpp](#).

7.8.2 Member Data Documentation

7.8.2.1 loop

bool eng::Audio::loop

Definition at line [17](#) of file [IRenderer.hpp](#).

7.8.2.2 name

std::string eng::Audio::name

Definition at line [18](#) of file [IRenderer.hpp](#).

7.8.2.3 path

std::string eng::Audio::path

Definition at line [15](#) of file [IRenderer.hpp](#).

7.8.2.4 volume

float eng::Audio::volume

Definition at line [16](#) of file [IRenderer.hpp](#).

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

- [/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IRenderer.hpp](#)

7.9 ecs::AudioSystem Class Reference

Class for managing entities and their components.

```
#include <Systems.hpp>
```

Collaboration diagram for ecs::AudioSystem:

ecs::AudioSystem
<ul style="list-style-type: none"> + AudioSystem() + ~AudioSystem() + AudioSystem() + operator=() + AudioSystem() + operator=() + update()

Public Member Functions

- [AudioSystem](#) ()=default
- [~AudioSystem](#) ()=default
- [AudioSystem](#) (const [AudioSystem](#) &)=delete
- [AudioSystem](#) & operator= (const [AudioSystem](#) &)=delete
- [AudioSystem](#) ([AudioSystem](#) &&)=delete
- [AudioSystem](#) & operator= ([AudioSystem](#) &&)=delete
- void [update](#) ([Registry](#) ®istry, float dt)

7.9.1 Detailed Description

Class for managing entities and their components.

Definition at line 79 of file [Systems.hpp](#).

7.9.2 Constructor & Destructor Documentation

7.9.2.1 AudioSystem() [1/3]

ecs::AudioSystem::AudioSystem () [default]

7.9.2.2 ~AudioSystem()

ecs::AudioSystem::~~AudioSystem () [default]

7.9.2.3 AudioSystem() [2/3]

ecs::AudioSystem::AudioSystem (
 const [AudioSystem](#) &) [delete]

7.9.2.4 AudioSystem() [3/3]

ecs::AudioSystem::AudioSystem (
 [AudioSystem](#) &&) [delete]

7.9.3 Member Function Documentation

7.9.3.1 operator=() [1/2]

[AudioSystem](#) & ecs::AudioSystem::operator= (
 [AudioSystem](#) &&) [delete]

7.9.3.2 operator=() [2/2]

[AudioSystem](#) & ecs::AudioSystem::operator= (
 const [AudioSystem](#) &) [delete]

7.9.3.3 update()

void ecs::AudioSystem::update (
 [Registry](#) & registry,
 float dt) [inline]

Definition at line 90 of file [Systems.hpp](#).

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

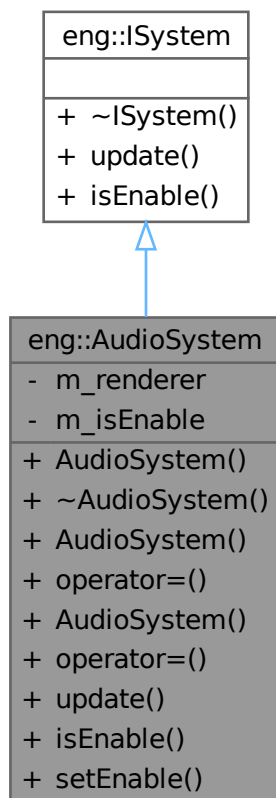
- [/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Systems.hpp](#)

7.10 eng::AudioSystem Class Reference

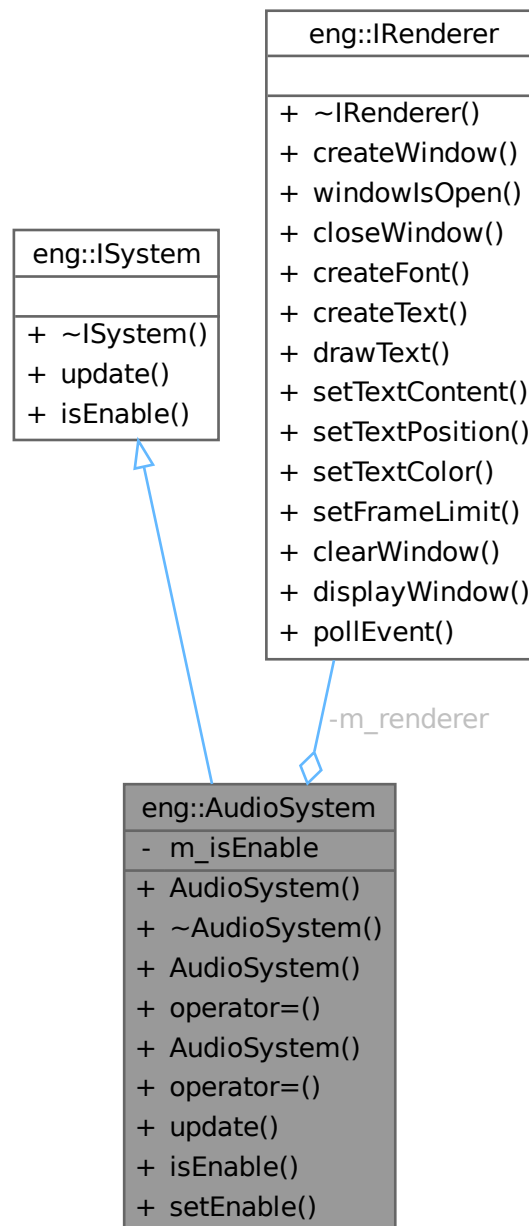
Class for managing entities and their components.

```
#include <Systems.hpp>
```

Inheritance diagram for eng::AudioSystem:



Collaboration diagram for eng::AudioSystem:



Public Member Functions

- [AudioSystem](#) ([IRenderer](#) &renderer)
- [~AudioSystem](#) () override=default
- [AudioSystem](#) (const [AudioSystem](#) &)=delete
- [AudioSystem](#) & operator= (const [AudioSystem](#) &)=delete
- [AudioSystem](#) ([AudioSystem](#) &&)=delete
- [AudioSystem](#) & operator= ([AudioSystem](#) &&)=delete

- void [update](#) ([ecs::Registry](#) ®istry, float dt) override
- bool [isEnabled](#) () override
- void [setEnabled](#) (const bool enable)

Public Member Functions inherited from [eng::ISystem](#)

- virtual [~ISystem](#) ()=default

Private Attributes

- [IRenderer](#) & [m_renderer](#)
- bool [m_isEnable](#) = true

7.10.1 Detailed Description

Class for managing entities and their components.

Definition at line 96 of file [Systems.hpp](#).

7.10.2 Constructor & Destructor Documentation

7.10.2.1 [AudioSystem](#)() [1/3]

```
eng::AudioSystem::AudioSystem (
    IRenderer & renderer) [inline], [explicit]
```

Definition at line 99 of file [Systems.hpp](#).

7.10.2.2 [~AudioSystem](#)()

```
eng::AudioSystem::~~AudioSystem () [override], [default]
```

7.10.2.3 [AudioSystem](#)() [2/3]

```
eng::AudioSystem::AudioSystem (
    const AudioSystem & ) [delete]
```

7.10.2.4 [AudioSystem](#)() [3/3]

```
eng::AudioSystem::AudioSystem (
    AudioSystem && ) [delete]
```

7.10.3 Member Function Documentation

7.10.3.1 isEnabled()

bool eng::AudioSystem::isEnabled () [inline], [override], [virtual]

Implements [eng::ISystem](#).

Definition at line 108 of file [Systems.hpp](#).

7.10.3.2 operator=() [1/2]

[AudioSystem](#) & eng::AudioSystem::operator= (
 [AudioSystem](#) &&) [delete]

7.10.3.3 operator=() [2/2]

[AudioSystem](#) & eng::AudioSystem::operator= (
 const [AudioSystem](#) &) [delete]

7.10.3.4 setEnable()

void eng::AudioSystem::setEnable (
 const bool enable) [inline]

Definition at line 109 of file [Systems.hpp](#).

7.10.3.5 update()

void eng::AudioSystem::update (
 [ecs::Registry](#) & registry,
 float dt) [inline], [override], [virtual]

Implements [eng::ISystem](#).

Definition at line 107 of file [Systems.hpp](#).

7.10.4 Member Data Documentation

7.10.4.1 m_isEnable

bool eng::AudioSystem::m_isEnable = true [private]

Definition at line 113 of file [Systems.hpp](#).

7.10.4.2 m_renderer

`IRenderer& eng::AudioSystem::m_renderer` [private]

Definition at line 112 of file [Systems.hpp](#).

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

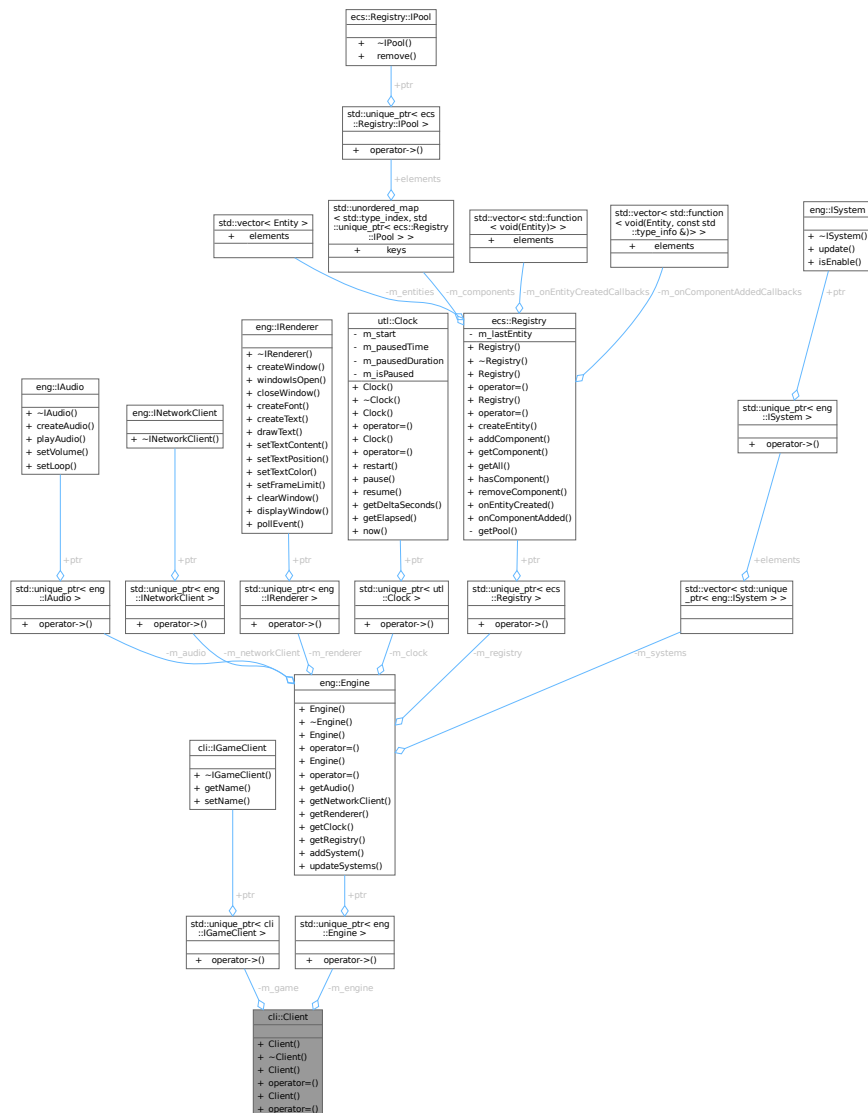
- [/home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/Systems.hpp](#)

7.11 cli::Client Class Reference

Class for the client.

`#include <Client.hpp>`

Collaboration diagram for `cli::Client`:



Public Member Functions

- [Client](#) (const [ArgsConfig](#) &cfg)
- [~Client](#) ()=default
- [Client](#) (const [Client](#) &)=delete
- [Client](#) & [operator=](#) (const [Client](#) &)=delete
- [Client](#) ([Client](#) &&)=delete
- [Client](#) & [operator=](#) ([Client](#) &&)=delete

Private Attributes

- std::unique_ptr< [IGameClient](#) > [m_game](#)
- std::unique_ptr< [eng::Engine](#) > [m_engine](#)

7.11.1 Detailed Description

Class for the client.

Definition at line 23 of file [Client.hpp](#).

7.11.2 Constructor & Destructor Documentation

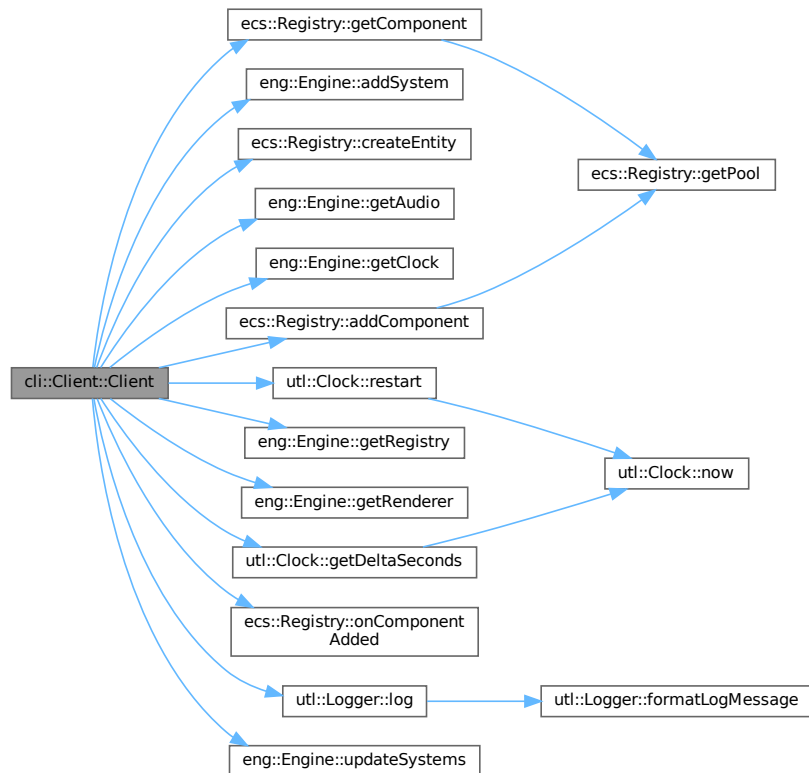
7.11.2.1 Client() [1/3]

```
cli::Client::Client (
    const ArgsConfig & cfg) [explicit]
```

Definition at line 10 of file [client.cpp](#).

References [ecs::Registry::addComponent\(\)](#), [eng::Engine::addSystem\(\)](#), [cli::Paths::Audio::AUDIO_TITLE](#), [BUILD_TYPE](#), [eng::Closed](#), [ecs::Registry::createEntity\(\)](#), [eng::Escape](#), [eng::Text::fontName](#), [cli::Paths::Fonts::FONTS_F](#), [cli::ArgsConfig::frameLimit](#), [cli::ArgsConfig::fullscreen](#), [eng::Engine::getAudio\(\)](#), [eng::Engine::getClock\(\)](#), [ecs::Registry::getComponent\(\)](#), [utl::Clock::getDeltaSeconds\(\)](#), [eng::Engine::getRegistry\(\)](#), [eng::Engine::getRenderer\(\)](#), [GIT_COMMIT_HASH](#), [GIT_TAG](#), [cli::ArgsConfig::height](#), [utl::INFO](#), [eng::KeyPressed](#), [utl::Logger::log\(\)](#), [m_engine](#), [ecs::Registry::onComponentAdded\(\)](#), [eng::Font::path](#), [PROJECT_NAME](#), [PROJECT_VERSION](#), [eng::Color::r](#), [utl::Clock::restart\(\)](#), [eng::Engine::updateSystems\(\)](#), and [cli::ArgsConfig::width](#).

Here is the call graph for this function:



7.11.2.2 ~Client()

cli::Client::~~Client () [default]

7.11.2.3 Client() [2/3]

cli::Client::Client (
 const [Client](#) &) [delete]

7.11.2.4 Client() [3/3]

cli::Client::Client (
 [Client](#) &&) [delete]

7.11.3 Member Function Documentation

7.11.3.1 operator=() [1/2]

[Client](#) & cli::Client::operator= (
 [Client](#) &&) [delete]

7.11.3.2 operator=() [2/2]

```
Client & cli::Client::operator= (  
    const Client & ) [delete]
```

7.11.4 Member Data Documentation

7.11.4.1 m_engine

```
std::unique_ptr<eng::Engine> cli::Client::m_engine [private]
```

Definition at line 37 of file [Client.hpp](#).

Referenced by [Client\(\)](#).

7.11.4.2 m_game

```
std::unique_ptr<IGameClient> cli::Client::m_game [private]
```

Definition at line 36 of file [Client.hpp](#).

The documentation for this class was generated from the following files:

- [/home/masina/Projects/Epitech/rtype/client/include/Client/Client.hpp](#)
- [/home/masina/Projects/Epitech/rtype/client/src/client.cpp](#)

7.12 utl::Clock Class Reference

Class for clock.

```
#include <Clock.hpp>
```

Collaboration diagram for `utl::Clock`:

utl::Clock
<ul style="list-style-type: none"> - m_start - m_pausedTime - m_pausedDuration - m_isPaused
<ul style="list-style-type: none"> + Clock() + ~Clock() + Clock() + operator=() + Clock() + operator=() + restart() + pause() + resume() + getDeltaSeconds() + getElapsed() + now()

Public Types

- using [TimePoint](#) = `std::chrono::time_point<std::chrono::high_resolution_clock>`

Public Member Functions

- [Clock](#) (const bool startNow=true)
- [~Clock](#) ()=default
- [Clock](#) (const [Clock](#) &)=delete
- [Clock](#) & [operator=](#) (const [Clock](#) &)=delete
- [Clock](#) ([Clock](#) &&)=delete
- [Clock](#) & [operator=](#) ([Clock](#) &&)=delete
- void [restart](#) ()
- void [pause](#) ()
- void [resume](#) ()
- float [getDeltaSeconds](#) () const
- template<typename [Duration](#) = `std::chrono::seconds`>
auto [getElapsed](#) () const

Static Public Member Functions

- static [TimePoint](#) [now](#) ()

Private Types

- using [Duration](#) = std::chrono::high_resolution_clock::duration

Private Attributes

- [TimePoint](#) m_start
- [TimePoint](#) m_pausedTime
- [Duration](#) m_pausedDuration
- bool m_isPaused {false}

Friends

- std::ostream & [operator<<](#) (std::ostream &os, const [Clock](#) &clock)

7.12.1 Detailed Description

Class for clock.

Definition at line 20 of file [Clock.hpp](#).

7.12.2 Member Typedef Documentation

7.12.2.1 Duration

using [utl::Clock::Duration](#) = std::chrono::high_resolution_clock::duration [private]

Definition at line 78 of file [Clock.hpp](#).

7.12.2.2 TimePoint

using [utl::Clock::TimePoint](#) = std::chrono::time_point<std::chrono::high_resolution_clock>

Definition at line 24 of file [Clock.hpp](#).

7.12.3 Constructor & Destructor Documentation

7.12.3.1 Clock() [1/3]

[utl::Clock::Clock](#) (
 const bool startNow = true) [inline], [explicit]

Definition at line 26 of file [Clock.hpp](#).

7.12.3.2 ~Clock()

[utl::Clock::~~Clock](#) () [default]

7.12.3.3 Clock() [2/3]

```
utl::Clock::Clock (  
    const Clock & ) [delete]
```

7.12.3.4 Clock() [3/3]

```
utl::Clock::Clock (  
    Clock && ) [delete]
```

7.12.4 Member Function Documentation

7.12.4.1 getDeltaSeconds()

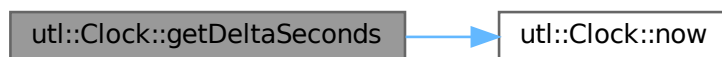
```
float utl::Clock::getDeltaSeconds () const [inline], [nodiscard]
```

Definition at line 63 of file [Clock.hpp](#).

References [m_isPaused](#), [m_pausedDuration](#), [m_pausedTime](#), [m_start](#), and [now\(\)](#).

Referenced by [cli::Client::Client\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.12.4.2 getElapsed()

```
template<typename Duration = std::chrono::seconds>
auto utl::Clock::getElapsed () const [inline], [nodiscard]
```

Definition at line 72 of file [Clock.hpp](#).

References [m_pausedDuration](#), [m_start](#), and [now\(\)](#).

Here is the call graph for this function:



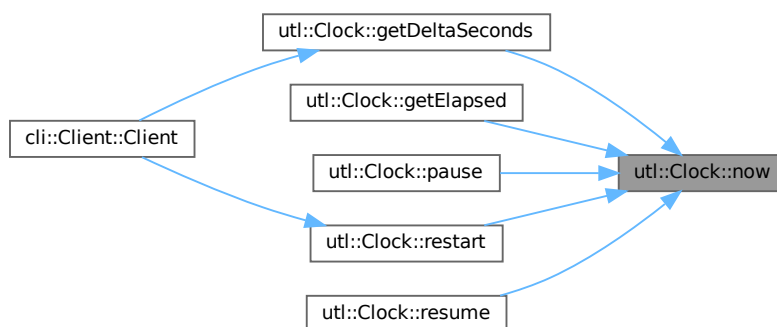
7.12.4.3 now()

```
static TimePoint utl::Clock::now () [inline], [static]
```

Definition at line 40 of file [Clock.hpp](#).

Referenced by [getDeltaSeconds\(\)](#), [getElapsed\(\)](#), [pause\(\)](#), [restart\(\)](#), and [resume\(\)](#).

Here is the caller graph for this function:



7.12.4.4 operator=() [1/2]

```
Clock & utl::Clock::operator= (
    Clock && ) [delete]
```

7.12.4.5 `operator=()` [2/2]

`Clock` & utl::Clock::operator= (
 const `Clock` &) [delete]

7.12.4.6 `pause()`

void utl::Clock::pause () [inline]

Definition at line 47 of file `Clock.hpp`.

References `m_isPaused`, `m_pausedTime`, and `now()`.

Here is the call graph for this function:

7.12.4.7 `restart()`

void utl::Clock::restart () [inline]

Definition at line 41 of file `Clock.hpp`.

References `m_isPaused`, `m_pausedDuration`, `m_start`, and `now()`.

Referenced by `cli::Client::Client()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.12.4.8 resume()

void utl::Clock::resume () [inline]

Definition at line 55 of file [Clock.hpp](#).

References [m_isPaused](#), [m_pausedDuration](#), [m_pausedTime](#), and [now\(\)](#).

Here is the call graph for this function:



7.12.5 Friends And Related Symbol Documentation

7.12.5.1 operator<<

```
std::ostream & operator<< (  
    std::ostream & os,  
    const Clock & clock) [friend]
```

Definition at line 34 of file [Clock.hpp](#).

7.12.6 Member Data Documentation

7.12.6.1 m_isPaused

```
bool utl::Clock::m_isPaused {false} [private]
```

Definition at line 83 of file [Clock.hpp](#).

Referenced by [getDeltaSeconds\(\)](#), [pause\(\)](#), [restart\(\)](#), and [resume\(\)](#).

7.12.6.2 m_pausedDuration

```
Duration utl::Clock::m_pausedDuration [private]
```

Definition at line 82 of file [Clock.hpp](#).

Referenced by [getDeltaSeconds\(\)](#), [getElapsed\(\)](#), [restart\(\)](#), and [resume\(\)](#).

7.12.6.3 m_pausedTime

[TimePoint](#) utl::Clock::m_pausedTime [private]

Definition at line 81 of file [Clock.hpp](#).

Referenced by [getDeltaSeconds\(\)](#), [pause\(\)](#), and [resume\(\)](#).

7.12.6.4 m_start

[TimePoint](#) utl::Clock::m_start [private]

Definition at line 80 of file [Clock.hpp](#).

Referenced by [getDeltaSeconds\(\)](#), [getElapsed\(\)](#), and [restart\(\)](#).

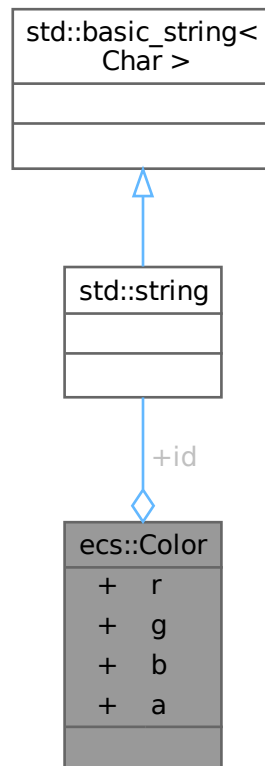
The documentation for this class was generated from the following file:

- [/home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/Clock.hpp](#)

7.13 ecs::Color Struct Reference

#include <Component.hpp>

Collaboration diagram for `ecs::Color`:



Public Attributes

- `std::string` [id](#)
- `int` [r](#)
- `int` [g](#)
- `int` [b](#)
- `int` [a](#)

7.13.1 Detailed Description

Definition at line 20 of file [Component.hpp](#).

7.13.2 Member Data Documentation

7.13.2.1 [a](#)

`int ecs::Color::a`

Definition at line 26 of file [Component.hpp](#).

7.13.2.2 [b](#)

`int ecs::Color::b`

Definition at line 25 of file [Component.hpp](#).

7.13.2.3 [g](#)

`int ecs::Color::g`

Definition at line 24 of file [Component.hpp](#).

7.13.2.4 [id](#)

`std::string ecs::Color::id`

Definition at line 22 of file [Component.hpp](#).

7.13.2.5 [r](#)

`int ecs::Color::r`

Definition at line 23 of file [Component.hpp](#).

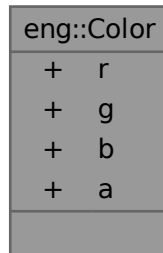
The documentation for this struct was generated from the following file:

- [/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.hpp](#)

7.14 eng::Color Struct Reference

#include <IRenderer.hpp>

Collaboration diagram for eng::Color:



Public Attributes

- `uint8_t` [r](#)
- `uint8_t` [g](#)
- `uint8_t` [b](#)
- `uint8_t` [a](#)

7.14.1 Detailed Description

Definition at line 20 of file [IRenderer.hpp](#).

7.14.2 Member Data Documentation

7.14.2.1 [a](#)

`uint8_t` [eng::Color::a](#)

Definition at line 25 of file [IRenderer.hpp](#).

Referenced by [eng::SFMLRenderer::clearWindow\(\)](#), [eng::SFMLRenderer::createText\(\)](#), and [eng::SFMLRenderer::setTextColor\(\)](#).

7.14.2.2 [b](#)

`uint8_t` [eng::Color::b](#)

Definition at line 24 of file [IRenderer.hpp](#).

Referenced by [eng::SFMLRenderer::clearWindow\(\)](#), [eng::SFMLRenderer::createText\(\)](#), and [eng::SFMLRenderer::setTextColor\(\)](#).

7.14.2.3 g

uint8_t eng::Color::g

Definition at line 23 of file [IRenderer.hpp](#).

Referenced by [eng::SFMLRenderer::clearWindow\(\)](#), [eng::SFMLRenderer::createText\(\)](#), and [eng::SFMLRenderer::setTextColor\(\)](#).

7.14.2.4 r

uint8_t eng::Color::r

Definition at line 22 of file [IRenderer.hpp](#).

Referenced by [eng::SFMLRenderer::clearWindow\(\)](#), [cli::Client::Client\(\)](#), [eng::SFMLRenderer::createText\(\)](#), and [eng::SFMLRenderer::setTextColor\(\)](#).

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

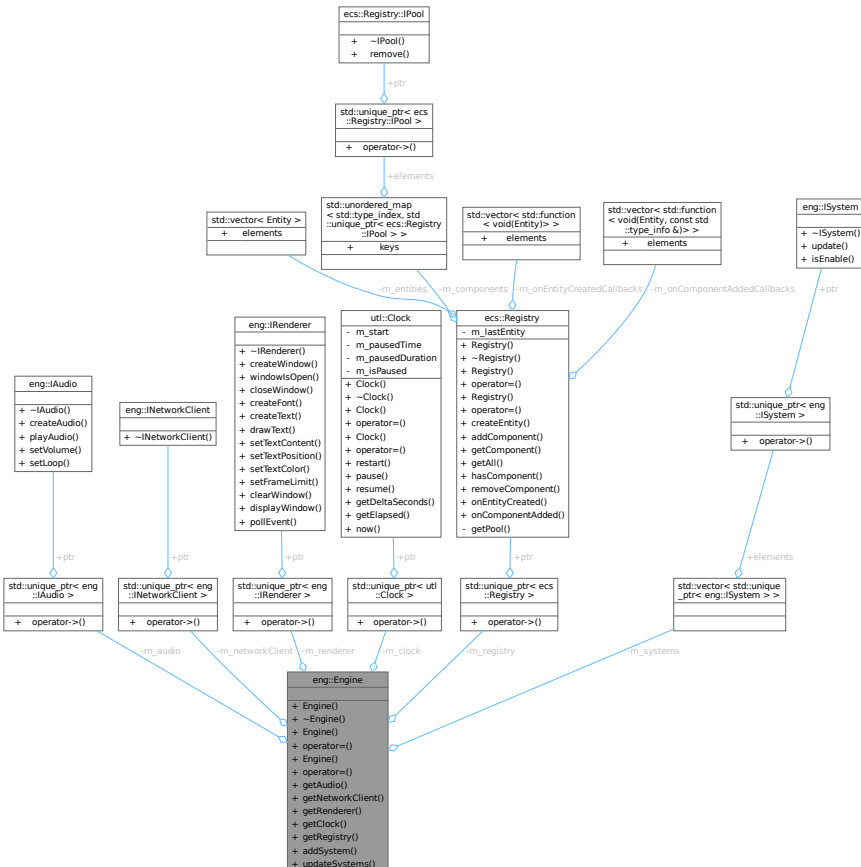
- [/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IRenderer.hpp](#)

7.15 eng::Engine Class Reference

Class for the game engine.

#include <Engine.hpp>

Collaboration diagram for eng::Engine:



Public Member Functions

- [Engine](#) (const std::function< std::unique_ptr< [IAudio](#) >()> &audioFactory, const std::function< std::unique_ptr< [INetworkClient](#) >()> &networkFactory, const std::function< std::unique_ptr< [IRenderer](#) >()> &rendererFactory)
- [~Engine](#) ()=default
- [Engine](#) (const [Engine](#) &)=delete
- [Engine](#) & operator= (const [Engine](#) &)=delete
- [Engine](#) ([Engine](#) &&)=delete
- [Engine](#) & operator= ([Engine](#) &&)=delete
- std::unique_ptr< [IAudio](#) > & [getAudio](#) ()
- std::unique_ptr< [INetworkClient](#) > & [getNetworkClient](#) ()
- std::unique_ptr< [IRenderer](#) > & [getRenderer](#) ()
- std::unique_ptr< [utl::Clock](#) > & [getClock](#) ()
- std::unique_ptr< [ecs::Registry](#) > & [getRegistry](#) ()
- void [addSystem](#) (std::unique_ptr< [ISystem](#) > system)
- void [updateSystems](#) (const float dt) const

Private Attributes

- std::unique_ptr< [IAudio](#) > [m_audio](#)
- std::unique_ptr< [INetworkClient](#) > [m_networkClient](#)
- std::unique_ptr< [IRenderer](#) > [m_renderer](#)
- std::unique_ptr< [utl::Clock](#) > [m_clock](#)
- std::unique_ptr< [ecs::Registry](#) > [m_registry](#)
- std::vector< std::unique_ptr< [ISystem](#) > > [m_systems](#)

7.15.1 Detailed Description

Class for the game engine.

Definition at line 27 of file [Engine.hpp](#).

7.15.2 Constructor & Destructor Documentation

7.15.2.1 Engine() [1/3]

```
eng::Engine::Engine (
    const std::function< std::unique_ptr< IAudio >()> & audioFactory,
    const std::function< std::unique_ptr< INetworkClient >()> & networkFactory,
    const std::function< std::unique_ptr< IRenderer >()> & rendererFactory)
```

Definition at line 4 of file [Engine.cpp](#).

7.15.2.2 ~Engine()

```
eng::Engine::~Engine () [default]
```

7.15.2.3 Engine() [2/3]

```
eng::Engine::Engine (  
    const Engine & ) [delete]
```

7.15.2.4 Engine() [3/3]

```
eng::Engine::Engine (  
    Engine && ) [delete]
```

7.15.3 Member Function Documentation

7.15.3.1 addSystem()

```
void eng::Engine::addSystem (  
    std::unique_ptr< ISystem > system) [inline]
```

Definition at line 47 of file [Engine.hpp](#).

References [m_systems](#).

Referenced by [cli::Client::Client\(\)](#).

Here is the caller graph for this function:



7.15.3.2 getAudio()

```
std::unique_ptr< IAudio > & eng::Engine::getAudio () [inline]
```

Definition at line 41 of file [Engine.hpp](#).

References [m_audio](#).

Referenced by [cli::Client::Client\(\)](#).

Here is the caller graph for this function:



7.15.3.3 `getClock()`

```
std::unique_ptr< utl::Clock > & eng::Engine::getClock () [inline]
```

Definition at line 44 of file [Engine.hpp](#).

References [m_clock](#).

Referenced by [cli::Client::Client\(\)](#).

Here is the caller graph for this function:



7.15.3.4 `getNetworkClient()`

```
std::unique_ptr< INetworkClient > & eng::Engine::getNetworkClient () [inline]
```

Definition at line 42 of file [Engine.hpp](#).

References [m_networkClient](#).

7.15.3.5 `getRegistry()`

```
std::unique_ptr< ecs::Registry > & eng::Engine::getRegistry () [inline]
```

Definition at line 45 of file [Engine.hpp](#).

References [m_registry](#).

Referenced by [cli::Client::Client\(\)](#).

Here is the caller graph for this function:



7.15.3.6 getRenderer()

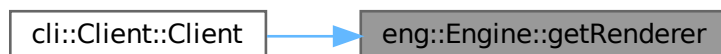
```
std::unique_ptr< IRenderer > & eng::Engine::getRenderer () [inline]
```

Definition at line 43 of file [Engine.hpp](#).

References [m_renderer](#).

Referenced by [cli::Client::Client\(\)](#).

Here is the caller graph for this function:



7.15.3.7 operator=() [1/2]

```
Engine & eng::Engine::operator= (  
    const Engine & ) [delete]
```

7.15.3.8 operator=() [2/2]

```
Engine & eng::Engine::operator= (  
    Engine && ) [delete]
```

7.15.3.9 updateSystems()

```
void eng::Engine::updateSystems (  
    const float dt) const [inline]
```

Definition at line 48 of file [Engine.hpp](#).

References [m_registry](#), and [m_systems](#).

Referenced by [cli::Client::Client\(\)](#).

Here is the caller graph for this function:



7.15.4 Member Data Documentation

7.15.4.1 m_audio

`std::unique_ptr<IAudio> eng::Engine::m_audio [private]`

Definition at line 57 of file [Engine.hpp](#).

Referenced by [getAudio\(\)](#).

7.15.4.2 m_clock

`std::unique_ptr<utl::Clock> eng::Engine::m_clock [private]`

Definition at line 61 of file [Engine.hpp](#).

Referenced by [getClock\(\)](#).

7.15.4.3 m_networkClient

`std::unique_ptr<INetworkClient> eng::Engine::m_networkClient [private]`

Definition at line 58 of file [Engine.hpp](#).

Referenced by [getNetworkClient\(\)](#).

7.15.4.4 m_registry

`std::unique_ptr<ecs::Registry> eng::Engine::m_registry [private]`

Definition at line 62 of file [Engine.hpp](#).

Referenced by [getRegistry\(\)](#), and [updateSystems\(\)](#).

7.15.4.5 m_renderer

`std::unique_ptr<IRenderer> eng::Engine::m_renderer [private]`

Definition at line 59 of file [Engine.hpp](#).

Referenced by [getRenderer\(\)](#).

7.15.4.6 m_systems

`std::vector<std::unique_ptr<ISystem> > eng::Engine::m_systems [private]`

Definition at line 63 of file [Engine.hpp](#).

Referenced by [addSystem\(\)](#), and [updateSystems\(\)](#).

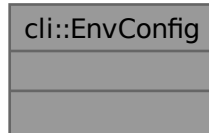
The documentation for this class was generated from the following files:

- [/home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/Engine.hpp](#)
- [/home/masina/Projects/Epitech/rtype/modules/Engine/src/Engine.cpp](#)

7.16 cli::EnvConfig Struct Reference

#include <ArgsHandler.hpp>

Collaboration diagram for cli::EnvConfig:



7.16.1 Detailed Description

Definition at line 27 of file [ArgsHandler.hpp](#).

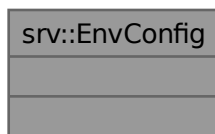
The documentation for this struct was generated from the following file:

- [/home/masina/Projects/Epitech/rtype/client/include/Client/ArgsHandler.hpp](#)

7.17 srv::EnvConfig Struct Reference

#include <ArgsHandler.hpp>

Collaboration diagram for srv::EnvConfig:



7.17.1 Detailed Description

Definition at line 26 of file [ArgsHandler.hpp](#).

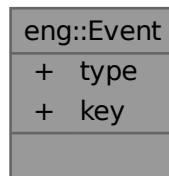
The documentation for this struct was generated from the following file:

- [/home/masina/Projects/Epitech/rtype/server/include/Server/ArgsHandler.hpp](#)

7.18 eng::Event Struct Reference

#include <IRenderer.hpp>

Collaboration diagram for eng::Event:



Public Attributes

- [EventType](#) type = [EventType::None](#)
- [Key](#) key = [Key::Unknown](#)

7.18.1 Detailed Description

Definition at line 97 of file [IRenderer.hpp](#).

7.18.2 Member Data Documentation

7.18.2.1 key

[Key](#) eng::Event::key = [Key::Unknown](#)

Definition at line 100 of file [IRenderer.hpp](#).

7.18.2.2 type

[EventType](#) eng::Event::type = [EventType::None](#)

Definition at line 99 of file [IRenderer.hpp](#).

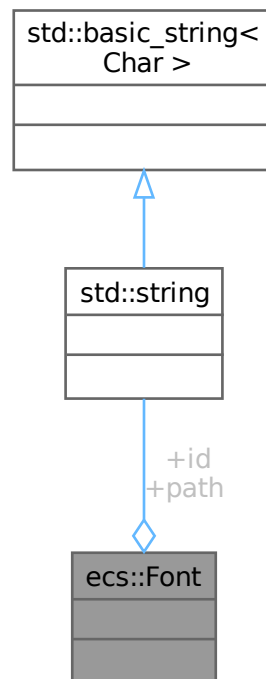
The documentation for this struct was generated from the following file:

- [/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IRenderer.hpp](#)

7.19 ecs::Font Struct Reference

#include <Component.hpp>

Collaboration diagram for ecs::Font:



Public Attributes

- `std::string` [id](#)
- `std::string` [path](#)

7.19.1 Detailed Description

Definition at line [28](#) of file [Component.hpp](#).

7.19.2 Member Data Documentation

7.19.2.1 id

`std::string` `ecs::Font::id`

Definition at line [30](#) of file [Component.hpp](#).

7.19.2.2 path

std::string ecs::Font::path

Definition at line 31 of file [Component.hpp](#).

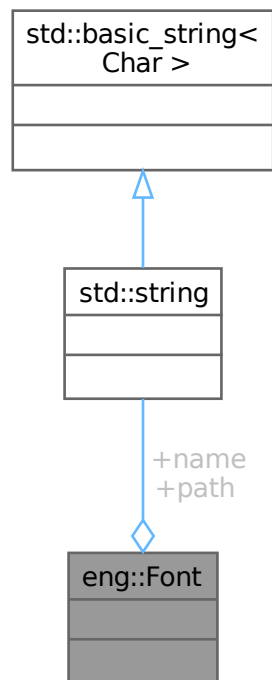
The documentation for this struct was generated from the following file:

- [/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.hpp](#)

7.20 eng::Font Struct Reference

```
#include <IRenderer.hpp>
```

Collaboration diagram for eng::Font:



Public Attributes

- std::string [path](#)
- std::string [name](#)

7.20.1 Detailed Description

Definition at line 27 of file [IRenderer.hpp](#).

7.20.2 Member Data Documentation

7.20.2.1 name

std::string eng::Font::name

Definition at line 30 of file [IRenderer.hpp](#).

Referenced by [eng::SFMLRenderer::createFont\(\)](#).

7.20.2.2 path

std::string eng::Font::path

Definition at line 29 of file [IRenderer.hpp](#).

Referenced by [cli::Client::Client\(\)](#), and [eng::SFMLRenderer::createFont\(\)](#).

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

- [/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IRenderer.hpp](#)

7.21 ecs::FontSystem Class Reference

Class for managing entities and their components.

```
#include <Systems.hpp>
```

Collaboration diagram for ecs::FontSystem:

ecs::FontSystem
<ul style="list-style-type: none"> + FontSystem() + ~FontSystem() + FontSystem() + operator=() + FontSystem() + operator=() + update()

Public Member Functions

- [FontSystem](#) ()=default
- [~FontSystem](#) ()=default
- [FontSystem](#) (const [FontSystem](#) &)=delete
- [FontSystem](#) & operator= (const [FontSystem](#) &)=delete
- [FontSystem](#) ([FontSystem](#) &&)=delete
- [FontSystem](#) & operator= ([FontSystem](#) &&)=delete
- void [update](#) ([Registry](#) ®istry)

7.21.1 Detailed Description

Class for managing entities and their components.

Definition at line 58 of file [Systems.hpp](#).

7.21.2 Constructor & Destructor Documentation

7.21.2.1 [FontSystem](#)() [1/3]

```
ecs::FontSystem::FontSystem () [default]
```

7.21.2.2 [~FontSystem](#)()

```
ecs::FontSystem::~~FontSystem () [default]
```

7.21.2.3 [FontSystem](#)() [2/3]

```
ecs::FontSystem::FontSystem (
    const FontSystem & ) [delete]
```

7.21.2.4 [FontSystem](#)() [3/3]

```
ecs::FontSystem::FontSystem (
    FontSystem && ) [delete]
```

7.21.3 Member Function Documentation

7.21.3.1 operator=() [1/2]

```
FontSystem & ecs::FontSystem::operator= (
    const FontSystem & ) [delete]
```

7.21.3.2 operator=() [2/2]

[FontSystem](#) & ecs::FontSystem::operator= (
[FontSystem](#) &&) [delete]

7.21.3.3 update()

void ecs::FontSystem::update (
[Registry](#) & registry) [inline]

Definition at line 69 of file [Systems.hpp](#).

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

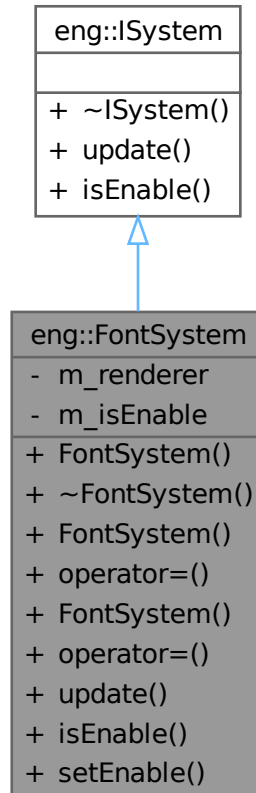
- [/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Systems.hpp](#)

7.22 eng::FontSystem Class Reference

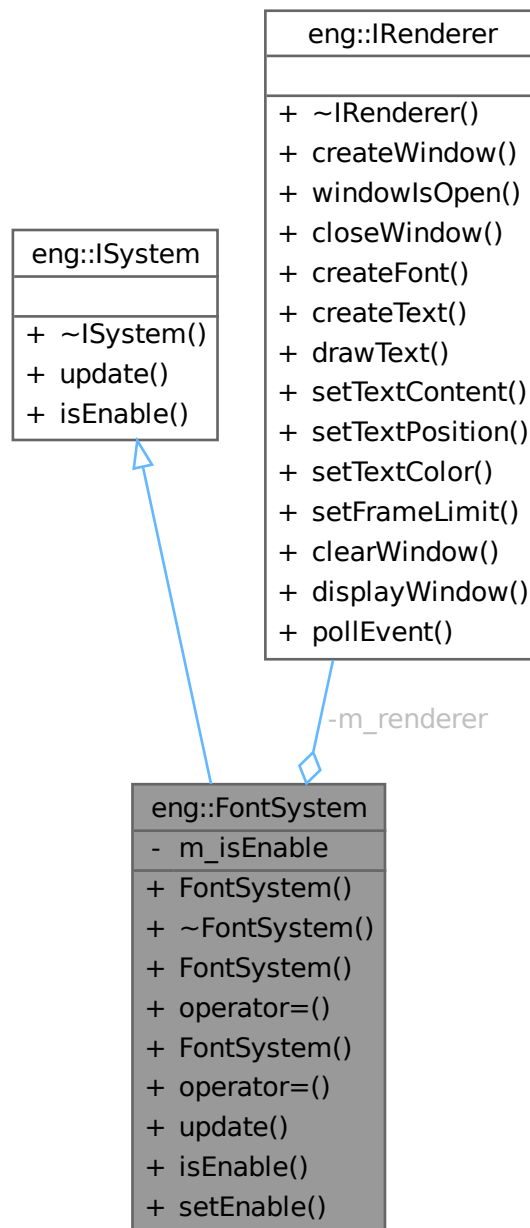
Class for managing entities and their components.

#include <Systems.hpp>

Inheritance diagram for eng::FontSystem:



Collaboration diagram for eng::FontSystem:



Public Member Functions

- `FontSystem (IRenderer &renderer)`
- `~FontSystem ()` override=default
- `FontSystem (const FontSystem &)=delete`
- `FontSystem & operator= (const FontSystem &)=delete`
- `FontSystem (FontSystem &&)=delete`
- `FontSystem & operator= (FontSystem &&)=delete`

- void [update](#) ([ecs::Registry](#) ®istry, const float dt) override
- bool [isEnabled](#) () override
- void [setEnabled](#) (const bool enable)

Public Member Functions inherited from [eng::ISystem](#)

- virtual [~ISystem](#) ()=default

Private Attributes

- [IRenderer](#) & [m_renderer](#)
- bool [m_isEnabled](#) = true

7.22.1 Detailed Description

Class for managing entities and their components.

Definition at line 71 of file [Systems.hpp](#).

7.22.2 Constructor & Destructor Documentation

7.22.2.1 FontSystem() [1/3]

```
eng::FontSystem::FontSystem (
    IRenderer & renderer)    [inline], [explicit]
```

Definition at line 74 of file [Systems.hpp](#).

7.22.2.2 ~FontSystem()

```
eng::FontSystem::~FontSystem ()    [override], [default]
```

7.22.2.3 FontSystem() [2/3]

```
eng::FontSystem::FontSystem (
    const FontSystem & )    [delete]
```

7.22.2.4 FontSystem() [3/3]

```
eng::FontSystem::FontSystem (
    FontSystem && )    [delete]
```

7.22.3 Member Function Documentation

7.22.3.1 isEnabled()

bool eng::FontSystem::isEnabled () [inline], [override], [virtual]

Implements [eng::ISystem](#).

Definition at line 83 of file [Systems.hpp](#).

7.22.3.2 operator=() [1/2]

[FontSystem](#) & eng::FontSystem::operator= (
const [FontSystem](#) &) [delete]

7.22.3.3 operator=() [2/2]

[FontSystem](#) & eng::FontSystem::operator= (
[FontSystem](#) &&) [delete]

7.22.3.4 setEnable()

void eng::FontSystem::setEnable (
const bool enable) [inline]

Definition at line 84 of file [Systems.hpp](#).

7.22.3.5 update()

void eng::FontSystem::update (
[ecs::Registry](#) & registry,
const float dt) [inline], [override], [virtual]

Implements [eng::ISystem](#).

Definition at line 82 of file [Systems.hpp](#).

7.22.4 Member Data Documentation

7.22.4.1 m_isEnable

bool eng::FontSystem::m_isEnable = true [private]

Definition at line 88 of file [Systems.hpp](#).

7.22.4.2 m_renderer

[IRenderer](#)& eng::FontSystem::m_renderer [private]

Definition at line 87 of file [Systems.hpp](#).

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

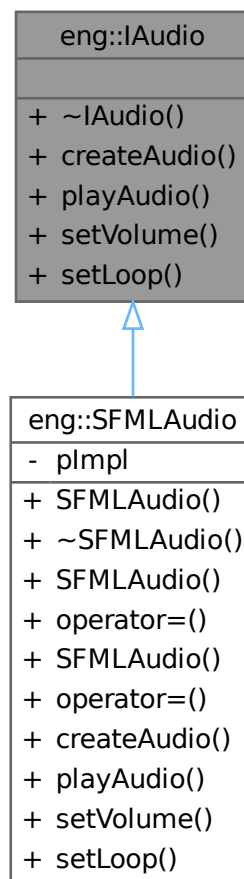
- [/home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/Systems.hpp](#)

7.23 eng::IAudio Class Reference

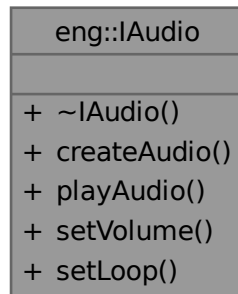
Interface for the audio.

```
#include <IAudio.hpp>
```

Inheritance diagram for eng::IAudio:



Collaboration diagram for `eng::IAudio`:



Public Member Functions

- virtual `~IAudio()`=default
- virtual void `createAudio` (const std::string &path, float volume, bool loop, const std::string &name)=0
- virtual void `playAudio` (const std::string &name)=0
- virtual void `setVolume` (const std::string &name, float volume)=0
- virtual void `setLoop` (const std::string &name, bool loop)=0

7.23.1 Detailed Description

Interface for the audio.

Definition at line 17 of file [IAudio.hpp](#).

7.23.2 Constructor & Destructor Documentation

7.23.2.1 ~IAudio()

virtual `eng::IAudio::~IAudio()` [virtual], [default]

7.23.3 Member Function Documentation

7.23.3.1 createAudio()

```
virtual void eng::IAudio::createAudio (
    const std::string & path,
    float volume,
    bool loop,
    const std::string & name) [pure virtual]
```

Implemented in [eng::SFMLAudio](#).

7.23.3.2 playAudio()

```
virtual void eng::IAudio::playAudio (
    const std::string & name) [pure virtual]
```

Implemented in [eng::SFMLAudio](#).

7.23.3.3 setLoop()

```
virtual void eng::IAudio::setLoop (
    const std::string & name,
    bool loop) [pure virtual]
```

Implemented in [eng::SFMLAudio](#).

7.23.3.4 setVolume()

```
virtual void eng::IAudio::setVolume (
    const std::string & name,
    float volume) [pure virtual]
```

Implemented in [eng::SFMLAudio](#).

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

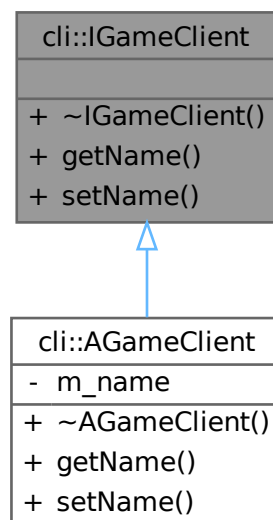
- [/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IAudio.hpp](#)

7.24 cli::IGameClient Class Reference

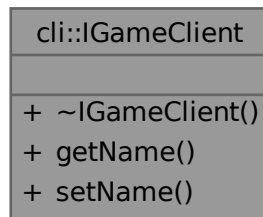
Interface for the games.

```
#include <IGameClient.hpp>
```

Inheritance diagram for cli::IGameClient:



Collaboration diagram for cli::IGameClient:



Public Member Functions

- virtual [~IGameClient](#) ()=default
- virtual `std::string` & [getName](#) ()
- virtual void [setName](#) (const `std::string` &newName)

7.24.1 Detailed Description

Interface for the games.

Definition at line 17 of file [IGameClient.hpp](#).

7.24.2 Constructor & Destructor Documentation

7.24.2.1 ~IGameClient()

virtual `cli::IGameClient::~~IGameClient` () [virtual], [default]

7.24.3 Member Function Documentation

7.24.3.1 getName()

virtual `std::string` & `cli::IGameClient::getName` () [nodiscard], [virtual]

Reimplemented in [cli::AGameClient](#).

7.24.3.2 setName()

virtual void `cli::IGameClient::setName` (
 const `std::string` & newName) [virtual]

Reimplemented in [cli::AGameClient](#).

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

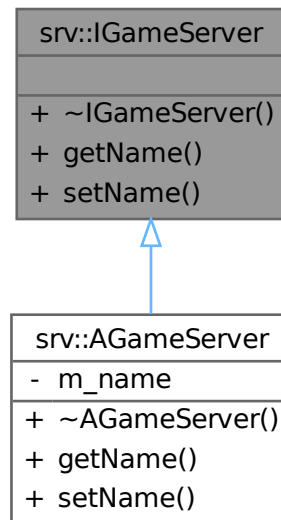
- [/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IGameClient.hpp](#)

7.25 srv::IGameServer Class Reference

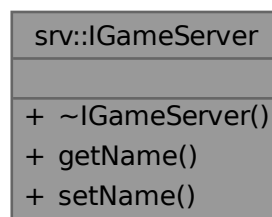
Interface for the games.

```
#include <IGameServer.hpp>
```

Inheritance diagram for srv::IGameServer:



Collaboration diagram for srv::IGameServer:



Public Member Functions

- virtual `~IGameServer()`=default
- virtual `std::string & getName()`
- virtual void `setName(const std::string &newName)`

7.25.1 Detailed Description

Interface for the games.

Definition at line 17 of file [IGameServer.hpp](#).

7.25.2 Constructor & Destructor Documentation

7.25.2.1 ~IGameServer()

virtual srv::IGameServer::~IGameServer () [virtual], [default]

7.25.3 Member Function Documentation

7.25.3.1 getName()

virtual std::string & srv::IGameServer::getName () [nodiscard], [virtual]

Reimplemented in [srv::AGameServer](#).

7.25.3.2 setName()

virtual void srv::IGameServer::setName (
const std::string & newName) [virtual]

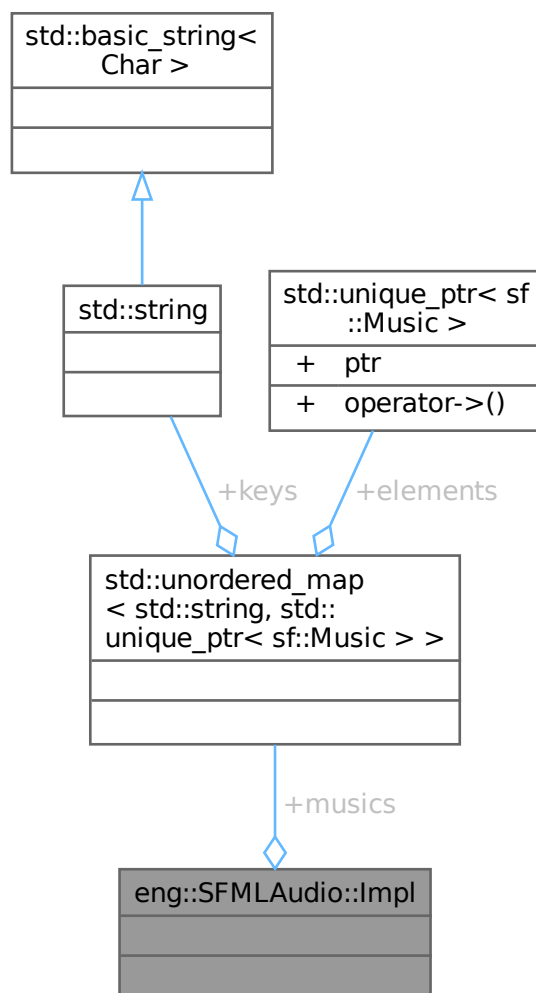
Reimplemented in [srv::AGameServer](#).

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

- [/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IGameServer.hpp](#)

7.26 eng::SFMLAudio::Impl Struct Reference

Collaboration diagram for eng::SFMLAudio::Impl:



Public Attributes

- `std::unordered_map< std::string, std::unique_ptr< sf::Music > >` [musics](#)

7.26.1 Detailed Description

Definition at line 9 of file [SFMLAudio.cpp](#).

7.26.2 Member Data Documentation

7.26.2.1 musics

`std::unordered_map<std::string, std::unique_ptr<sf::Music> > eng::SFMLAudio::Impl::musics`

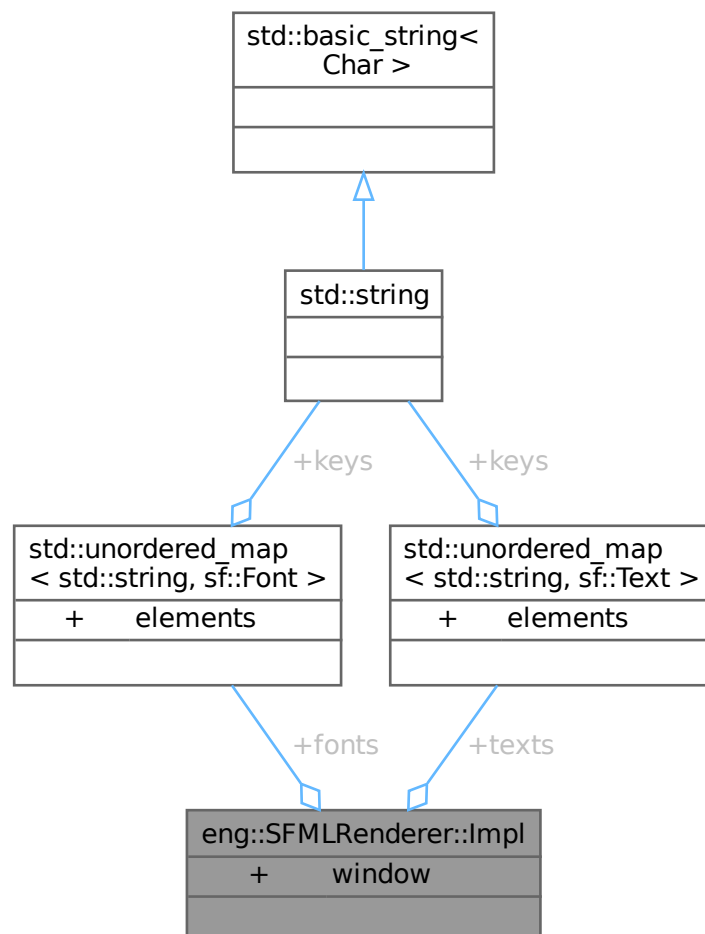
Definition at line 11 of file [SFMLAudio.cpp](#).

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

- [/home/masina/Projects/Epitech/rtype/modules/Audio/SFMLAudio/src/SFMLAudio.cpp](#)

7.27 eng::SFMLRenderer::Impl Struct Reference

Collaboration diagram for `eng::SFMLRenderer::Impl`:



Public Attributes

- sf::RenderWindow [window](#)
- std::unordered_map< std::string, sf::Font > [fonts](#)
- std::unordered_map< std::string, sf::Text > [texts](#)

7.27.1 Detailed Description

Definition at line 11 of file [SFMLRenderrer.cpp](#).

7.27.2 Member Data Documentation

7.27.2.1 fonts

std::unordered_map<std::string, sf::Font> eng::SFMLRenderrer::Impl::fonts

Definition at line 14 of file [SFMLRenderrer.cpp](#).

7.27.2.2 texts

std::unordered_map<std::string, sf::Text> eng::SFMLRenderrer::Impl::texts

Definition at line 15 of file [SFMLRenderrer.cpp](#).

7.27.2.3 window

sf::RenderWindow eng::SFMLRenderrer::Impl::window

Definition at line 13 of file [SFMLRenderrer.cpp](#).

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

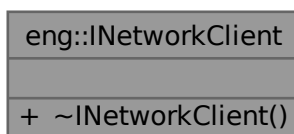
- [/home/masina/Projects/Epitech/rtype/modules/Renderer/SFMLRenderrer/src/SFMLRenderrer.cpp](#)

7.28 eng::INetworkClient Class Reference

Interface for the client network.

```
#include <INetworkClient.hpp>
```

Collaboration diagram for eng::INetworkClient:



Public Member Functions

- virtual [~INetworkClient](#) ()=default

7.28.1 Detailed Description

Interface for the client network.

Definition at line 17 of file [INetworkClient.hpp](#).

7.28.2 Constructor & Destructor Documentation

7.28.2.1 ~INetworkClient()

virtual eng::INetworkClient::~~INetworkClient () [virtual], [default]

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

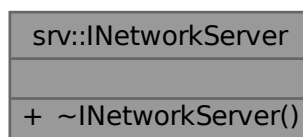
- /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/[INetworkClient.hpp](#)

7.29 srv::INetworkServer Class Reference

Interface for the server network.

```
#include <INetworkServer.hpp>
```

Collaboration diagram for srv::INetworkServer:



Public Member Functions

- virtual [~INetworkServer](#) ()=default

7.29.1 Detailed Description

Interface for the server network.

Definition at line 17 of file [INetworkServer.hpp](#).

7.29.2 Constructor & Destructor Documentation

7.29.2.1 ~INetworkServer()

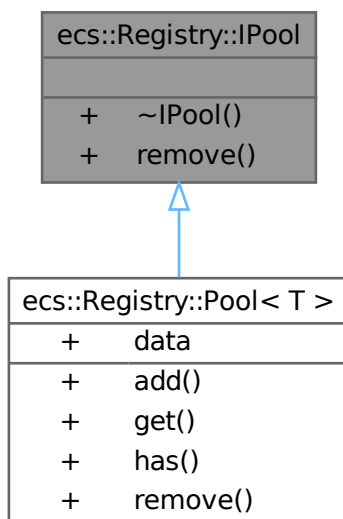
virtual srv::INetworkServer::~INetworkServer () [virtual], [default]

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

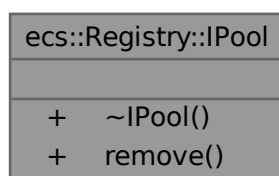
- /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/[INetworkServer.hpp](#)

7.30 ecs::Registry::IPool Class Reference

Inheritance diagram for ecs::Registry::IPool:



Collaboration diagram for ecs::Registry::IPool:



Public Member Functions

- virtual [~IPool](#) ()=default
- virtual void [remove](#) ([Entity](#) e)=0

7.30.1 Detailed Description

Definition at line [83](#) of file [Registry.hpp](#).

7.30.2 Constructor & Destructor Documentation

7.30.2.1 ~IPool()

virtual [ecs::Registry::IPool::~IPool](#) () [virtual], [default]

7.30.3 Member Function Documentation

7.30.3.1 remove()

virtual void [ecs::Registry::IPool::remove](#) (
[Entity](#) e) [pure virtual]

Implemented in [ecs::Registry::Pool< T >](#).

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

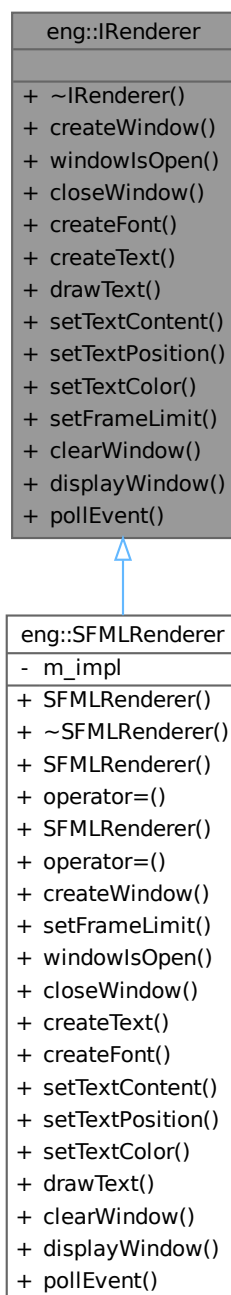
- [/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Registry.hpp](#)

7.31 eng::IRenderer Class Reference

Interface for the renderer.

```
#include <IRenderer.hpp>
```

Inheritance diagram for eng::IRenderer:



Collaboration diagram for `eng::IRenderer`:

eng::IRenderer
<ul style="list-style-type: none"> + ~IRenderer() + createWindow() + windowIsOpen() + closeWindow() + createFont() + createText() + drawText() + setTextContent() + setTextPosition() + setTextColor() + setFrameLimit() + clearWindow() + displayWindow() + pollEvent()

Public Member Functions

- virtual [~IRenderer](#) ()=default
- virtual void [createWindow](#) (const std::string &title, unsigned int height, unsigned int width, unsigned int frameLimit, bool fullscreen)=0
- virtual bool [windowIsOpen](#) () const =0
- virtual void [closeWindow](#) ()=0
- virtual void [createFont](#) ([Font](#) font)=0
- virtual void [createText](#) ([Text](#) text)=0
- virtual void [drawText](#) (const std::string &name)=0
- virtual void [setTextContent](#) (const std::string &name, const std::string &content)=0
- virtual void [setTextPosition](#) (const std::string &name, int x, int y)=0
- virtual void [setTextColor](#) (const std::string &name, [Color](#) color)=0
- virtual void [setFrameLimit](#) (unsigned int frameLimit)=0
- virtual void [clearWindow](#) ([Color](#) color)=0
- virtual void [displayWindow](#) ()=0
- virtual bool [pollEvent](#) ([Event](#) &event)=0

7.31.1 Detailed Description

Interface for the renderer.

Definition at line 108 of file [IRenderer.hpp](#).

7.31.2 Constructor & Destructor Documentation

7.31.2.1 ~IRenderer()

virtual eng::IRenderer::~IRenderer () [virtual], [default]

7.31.3 Member Function Documentation

7.31.3.1 clearWindow()

virtual void eng::IRenderer::clearWindow (
 [Color](#) color) [pure virtual]

Implemented in [eng::SFMLRenderer](#).

7.31.3.2 closeWindow()

virtual void eng::IRenderer::closeWindow () [pure virtual]

Implemented in [eng::SFMLRenderer](#).

7.31.3.3 createFont()

virtual void eng::IRenderer::createFont (
 [Font](#) font) [pure virtual]

Implemented in [eng::SFMLRenderer](#).

7.31.3.4 createText()

virtual void eng::IRenderer::createText (
 [Text](#) text) [pure virtual]

Implemented in [eng::SFMLRenderer](#).

7.31.3.5 createWindow()

virtual void eng::IRenderer::createWindow (
 const std::string & title,
 unsigned int height,
 unsigned int width,
 unsigned int frameLimit,
 bool fullscreen) [pure virtual]

Implemented in [eng::SFMLRenderer](#).

7.31.3.6 displayWindow()

virtual void eng::IRenderer::displayWindow () [pure virtual]

Implemented in [eng::SFMLRenderer](#).

7.31.3.7 drawText()

virtual void eng::IRenderer::drawText (
const std::string & name) [pure virtual]

Implemented in [eng::SFMLRenderer](#).

Referenced by [eng::TextSyStem::update\(\)](#).

Here is the caller graph for this function:



7.31.3.8 pollEvent()

virtual bool eng::IRenderer::pollEvent (
[Event](#) & event) [nodiscard], [pure virtual]

Implemented in [eng::SFMLRenderer](#).

7.31.3.9 setFrameLimit()

virtual void eng::IRenderer::setFrameLimit (
unsigned int frameLimit) [pure virtual]

Implemented in [eng::SFMLRenderer](#).

7.31.3.10 setTextColor()

virtual void eng::IRenderer::setTextColor (
const std::string & name,
[Color](#) color) [pure virtual]

Implemented in [eng::SFMLRenderer](#).

Referenced by [eng::TextSyStem::update\(\)](#).

Here is the caller graph for this function:



7.31.3.11 setTextContent()

```
virtual void eng::IRenderer::setTextContent (  
    const std::string & name,  
    const std::string & content) [pure virtual]
```

Implemented in [eng::SFMLRenderer](#).

Referenced by [eng::TextSyStem::update\(\)](#).

Here is the caller graph for this function:



7.31.3.12 setPosition()

```
virtual void eng::IRenderer::setPosition (  
    const std::string & name,  
    int x,  
    int y) [pure virtual]
```

Implemented in [eng::SFMLRenderer](#).

Referenced by [eng::TextSyStem::update\(\)](#).

Here is the caller graph for this function:



7.31.3.13 windowIsOpen()

```
virtual bool eng::IRenderer::windowIsOpen () const [nodiscard], [pure virtual]
```

Implemented in [eng::SFMLRenderer](#).

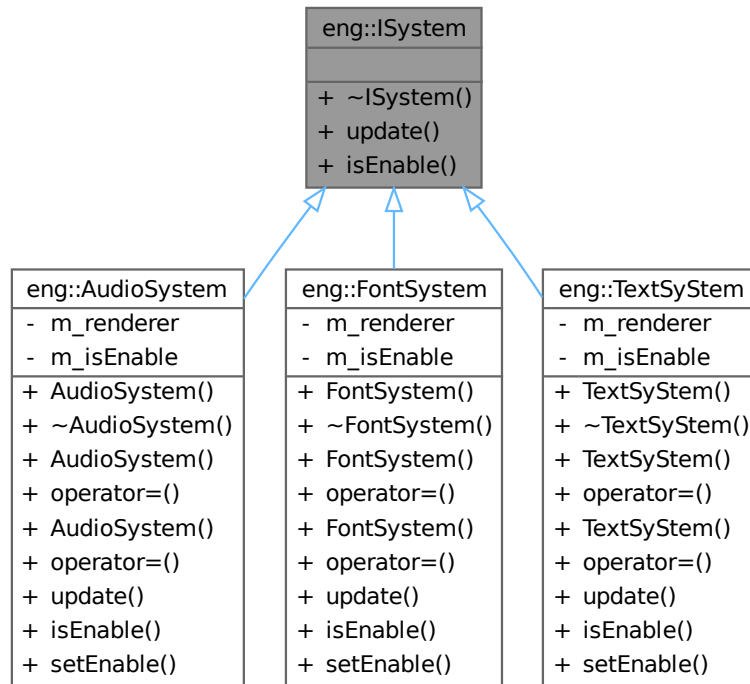
The documentation for this class was generated from the following file:

- `/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IRenderer.hpp`

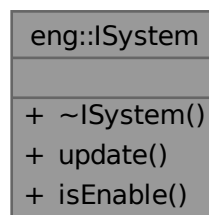
7.32 eng::ISystem Class Reference

```
#include <Systems.hpp>
```

Inheritance diagram for eng::ISystem:



Collaboration diagram for eng::ISystem:



Public Member Functions

- virtual `~ISystem()`=default
- virtual void `update` (`ecs::Registry` ®istry, float dt)=0
- virtual bool `isEnabled` ()=0

7.32.1 Detailed Description

Definition at line 16 of file [Systems.hpp](#).

7.32.2 Constructor & Destructor Documentation

7.32.2.1 ~ISystem()

virtual eng::ISystem::~ISystem () [virtual], [default]

7.32.3 Member Function Documentation

7.32.3.1 isEnabled()

virtual bool eng::ISystem::isEnabled () [pure virtual]

Implemented in [eng::AudioSystem](#), [eng::FontSystem](#), and [eng::TextSyStem](#).

7.32.3.2 update()

virtual void eng::ISystem::update (
 [ecs::Registry](#) & registry,
 float dt) [pure virtual]

Implemented in [eng::AudioSystem](#), [eng::FontSystem](#), and [eng::TextSyStem](#).

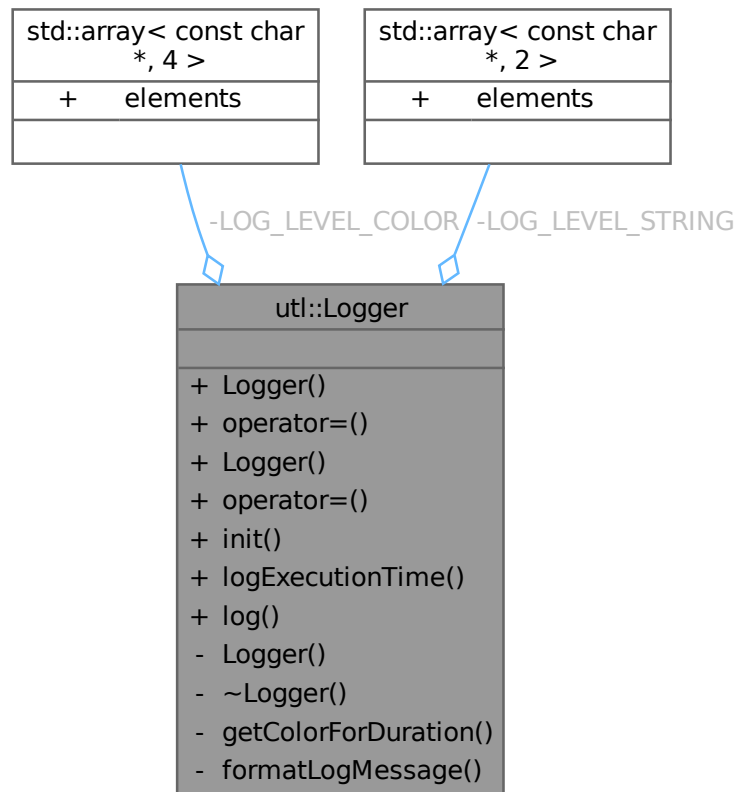
The documentation for this class was generated from the following file:

- [/home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/Systems.hpp](#)

7.33 utl::Logger Class Reference

```
#include <Logger.hpp>
```

Collaboration diagram for `utl::Logger`:



Public Member Functions

- `Logger` (`const Logger &`)=delete
- `Logger & operator=` (`const Logger &`)=delete
- `Logger` (`Logger &&`)=delete
- `Logger & operator=` (`Logger &&`)=delete

Static Public Member Functions

- static void `init` ()
- template<typename Func >
static void `logExecutionTime` (`const std::string &message`, `Func &&func`)
- static void `log` (`const std::string &message`, `const LogLevel &logLevel`)

Private Types

- enum `ColorIndex` : `uint8_t` { `COLOR_ERROR` , `COLOR_INFO` , `COLOR_WARNING` , `COLOR_RESET` }

Private Member Functions

- [Logger](#) ()=default
- [~Logger](#) ()=default

Static Private Member Functions

- static const char * [getColorForDuration](#) (const float duration)
- static std::string [formatLogMessage](#) ([LogLevel](#) level, const std::string &message)

Static Private Attributes

- static constexpr std::array< const char *, 4 > [LOG_LEVEL_COLOR](#)
- static constexpr std::array< const char *, 2 > [LOG_LEVEL_STRING](#) = {"INFO", "WARNING"}

7.33.1 Detailed Description

Definition at line 17 of file [Logger.hpp](#).

7.33.2 Member Enumeration Documentation

7.33.2.1 ColorIndex

```
enum utl::Logger::ColorIndex : uint8_t [private]
```

Enumerator

COLOR_ERROR	
COLOR_INFO	
COLOR_WARNING	
COLOR_RESET	

Definition at line 47 of file [Logger.hpp](#).

7.33.3 Constructor & Destructor Documentation

7.33.3.1 [Logger](#)() [1/3]

```
utl::Logger::Logger (
    const Logger & ) [delete]
```

7.33.3.2 [Logger](#)() [2/3]

```
utl::Logger::Logger (
    Logger && ) [delete]
```

7.33.3.3 Logger() [3/3]

utl::Logger::Logger () [private], [default]

7.33.3.4 ~Logger()

utl::Logger::~~Logger () [private], [default]

7.33.4 Member Function Documentation

7.33.4.1 formatLogMessage()

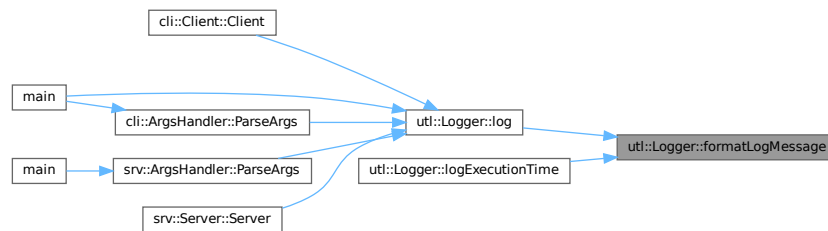
static std::string utl::Logger::formatLogMessage (
[LogLevel](#) level,
 const std::string & message) [inline], [static], [nodiscard], [private]

Definition at line 74 of file [Logger.hpp](#).

References [LOG_LEVEL_STRING](#).

Referenced by [log\(\)](#), and [logExecutionTime\(\)](#).

Here is the caller graph for this function:



7.33.4.2 getColorForDuration()

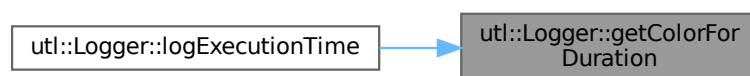
static const char * utl::Logger::getColorForDuration (
 const float duration) [inline], [static], [nodiscard], [private]

Definition at line 67 of file [Logger.hpp](#).

References [COLOR_ERROR](#), [COLOR_INFO](#), [COLOR_WARNING](#), and [LOG_LEVEL_COLOR](#).

Referenced by [logExecutionTime\(\)](#).

Here is the caller graph for this function:



7.33.4.3 init()

void utl::Logger::init () [static]

Definition at line 7 of file [logger.cpp](#).

Referenced by [main\(\)](#).

Here is the caller graph for this function:



7.33.4.4 log()

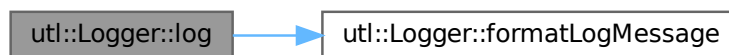
```
static void utl::Logger::log (  
    const std::string & message,  
    const LogLevel & logLevel) [inline], [static]
```

Definition at line 40 of file [Logger.hpp](#).

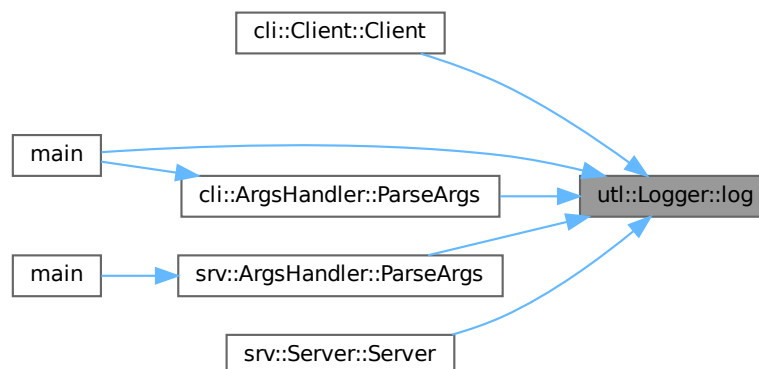
References [COLOR_INFO](#), [COLOR_RESET](#), [COLOR_WARNING](#), [formatLogMessage\(\)](#), [utl::INFO](#), and [LOG_LEVEL_COLOR](#).

Referenced by [cli::Client::Client\(\)](#), [main\(\)](#), [cli::ArgsHandler::ParseArgs\(\)](#), [srv::ArgsHandler::ParseArgs\(\)](#), and [srv::Server::Server\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.33.4.5 logExecutionTime()

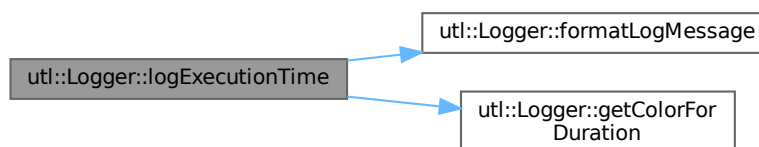
```

template<typename Func >
static void utl::Logger::logExecutionTime (
    const std::string & message,
    Func && func) [inline], [static]
  
```

Definition at line 28 of file [Logger.hpp](#).

References [COLOR_RESET](#), [formatLogMessage\(\)](#), [getColorForDuration\(\)](#), [utl::INFO](#), and [LOG_LEVEL_COLOR](#).

Here is the call graph for this function:



7.33.4.6 operator=() [1/2]

```

Logger & utl::Logger::operator= (
    const Logger & ) [delete]
  
```


7.33.4.7 operator=() [2/2]

[Logger](#) & utl::Logger::operator= (
 [Logger](#) &&) [delete]

7.33.5 Member Data Documentation

7.33.5.1 LOG_LEVEL_COLOR

std::array<const char *, 4> utl::Logger::LOG_LEVEL_COLOR [static], [constexpr], [private]

Initial value:

```
= {  
    "\033[31m",  
    "\033[32m",  
    "\033[33m",  
    "\033[0m\n"  
}
```

Definition at line 55 of file [Logger.hpp](#).

Referenced by [getColorForDuration\(\)](#), [log\(\)](#), and [logExecutionTime\(\)](#).

7.33.5.2 LOG_LEVEL_STRING

std::array<const char *, 2> utl::Logger::LOG_LEVEL_STRING = {"INFO", "WARNING"} [static], [constexpr], [private]

Definition at line 62 of file [Logger.hpp](#).

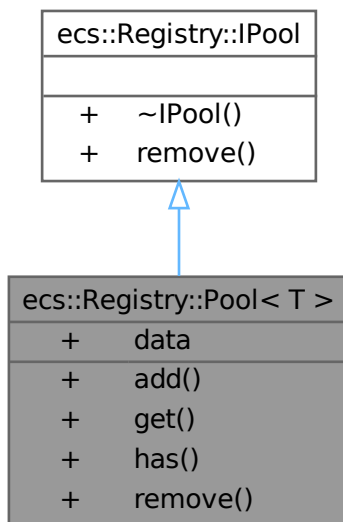
Referenced by [formatLogMessage\(\)](#).

The documentation for this class was generated from the following files:

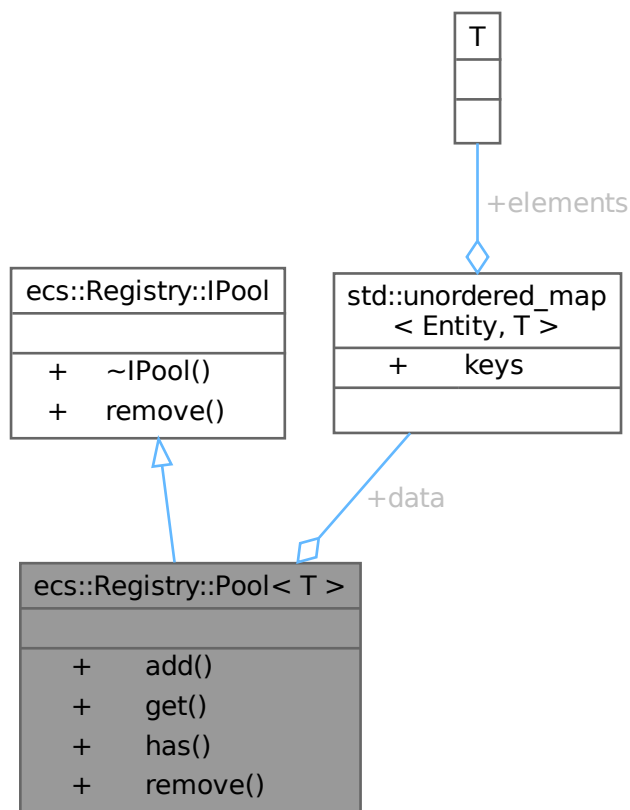
- /home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/[Logger.hpp](#)
- /home/masina/Projects/Epitech/rtype/modules/Utils/src/[logger.cpp](#)

7.34 ecs::Registry::Pool< T > Class Template Reference

Inheritance diagram for ecs::Registry::Pool< T >:



Collaboration diagram for `ecs::Registry::Pool< T >`:



Public Member Functions

- `template<typename... Args>`
`T & add (Entity e, Args &&...args)`
- `T * get (Entity e)`
- `bool has (Entity e)`
- `void remove (Entity e)` override

Public Member Functions inherited from `ecs::Registry::IPool`

- `virtual ~IPool ()`=default

Public Attributes

- `std::unordered_map< Entity, T > data`

7.34.1 Detailed Description

```
template<typename T>
class ecs::Registry::Pool< T >
```

Definition at line 90 of file [Registry.hpp](#).

7.34.2 Member Function Documentation

7.34.2.1 add()

```
template<typename T >
template<typename... Args>
T & ecs::Registry::Pool< T >::add (
    Entity e,
    Args &&... args) [inline]
```

Definition at line 95 of file [Registry.hpp](#).

References [ecs::Registry::Pool< T >::data](#).

7.34.2.2 get()

```
template<typename T >
T * ecs::Registry::Pool< T >::get (
    Entity e) [inline]
```

Definition at line 100 of file [Registry.hpp](#).

References [ecs::Registry::Pool< T >::data](#).

7.34.2.3 has()

```
template<typename T >
bool ecs::Registry::Pool< T >::has (
    Entity e) [inline]
```

Definition at line 108 of file [Registry.hpp](#).

References [ecs::Registry::Pool< T >::data](#).

7.34.2.4 remove()

```
template<typename T >
void ecs::Registry::Pool< T >::remove (
    Entity e) [inline], [override], [virtual]
```

Implements [ecs::Registry::IPool](#).

Definition at line 110 of file [Registry.hpp](#).

References [ecs::Registry::Pool< T >::data](#).

7.34.3 Member Data Documentation

7.34.3.1 data

```
template<typename T >
std::unordered_map<Entity, T> ecs::Registry::Pool< T >::data
```

Definition at line 93 of file [Registry.hpp](#).

Referenced by [ecs::Registry::Pool< T >::add\(\)](#), [ecs::Registry::Pool< T >::get\(\)](#), [ecs::Registry::Pool< T >::has\(\)](#), and [ecs::Registry::Pool< T >::remove\(\)](#).

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

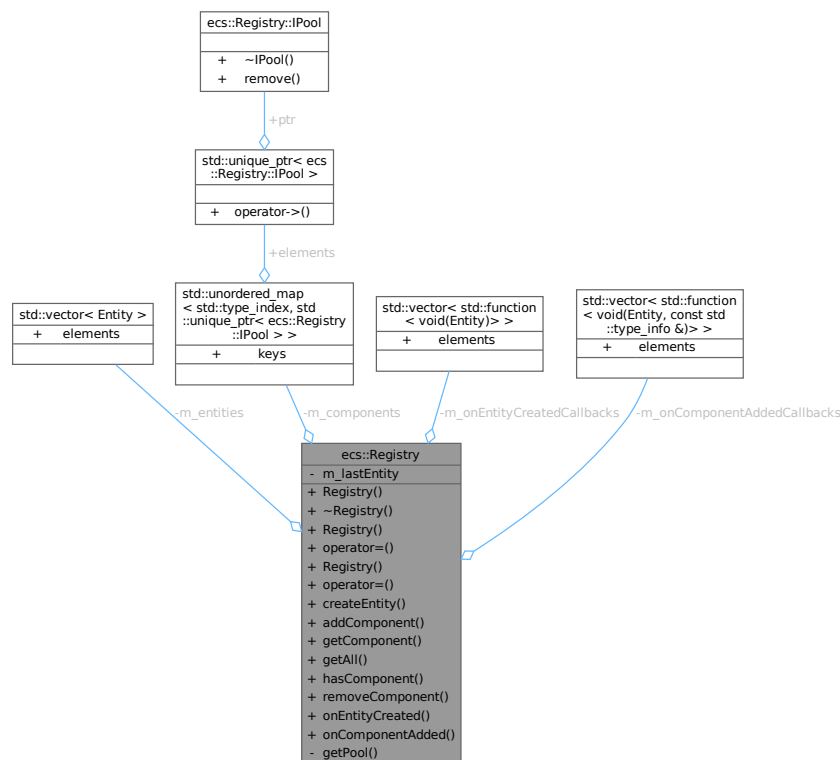
- [/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Registry.hpp](#)

7.35 ecs::Registry Class Reference

Class for managing entities and their components.

```
#include <Registry.hpp>
```

Collaboration diagram for `ecs::Registry`:



Classes

- class [IPool](#)
- class [Pool](#)

Public Member Functions

- [Registry](#) ()=default
- [~Registry](#) ()=default
- [Registry](#) (const [Registry](#) &)=delete
- [Registry](#) & [operator=](#) (const [Registry](#) &)=delete
- [Registry](#) ([Registry](#) &&)=delete
- [Registry](#) & [operator=](#) ([Registry](#) &&)=delete
- [Entity](#) [createEntity](#) ()
- template<typename T , typename... Args>
T & [addComponent](#) ([Entity](#) e, Args &&...args)
- template<typename T >
T * [getComponent](#) ([Entity](#) e)
- template<typename T >
std::unordered_map< [Entity](#), T > & [getAll](#) ()
- template<typename T >
bool [hasComponent](#) ([Entity](#) e)
- template<typename T >
void [removeComponent](#) ([Entity](#) e)
- void [onEntityCreated](#) (std::function< void([Entity](#))> cb)
- void [onComponentAdded](#) (std::function< void([Entity](#), const std::type_info &)> cb)

Private Member Functions

- template<typename T >
[Pool](#)< T > & [getPool](#) ()

Private Attributes

- [Entity](#) [m_lastEntity](#) = [INVALID_ENTITY](#)
- std::vector< [Entity](#) > [m_entities](#)
- std::unordered_map< std::type_index, std::unique_ptr< [IPool](#) > > [m_components](#)
- std::vector< std::function< void([Entity](#))> > > [m_onEntityCreatedCallbacks](#)
- std::vector< std::function< void([Entity](#), const std::type_info &)> > > [m_onComponentAddedCallbacks](#)

7.35.1 Detailed Description

Class for managing entities and their components.

Definition at line 24 of file [Registry.hpp](#).

7.35.2 Constructor & Destructor Documentation

7.35.2.1 [Registry](#)() [1/3]

`ecs::Registry::Registry ()` [default]

7.35.2.2 ~Registry()

ecs::Registry::~~Registry () [default]

7.35.2.3 Registry() [2/3]

ecs::Registry::Registry (
const [Registry](#) &) [delete]

7.35.2.4 Registry() [3/3]

ecs::Registry::Registry (
[Registry](#) &&) [delete]

7.35.3 Member Function Documentation

7.35.3.1 addComponent()

```
template<typename T , typename... Args>
T & ecs::Registry::addComponent (
    Entity e,
    Args &&... args) [inline]
```

Definition at line 44 of file [Registry.hpp](#).

References [getPool\(\)](#), and [m_onComponentAddedCallbacks](#).

Referenced by [cli::Client::Client\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.35.3.2 createEntity()

[Entity](#) ecs::Registry::createEntity () [inline]

Definition at line 35 of file [Registry.hpp](#).

References [m_entities](#), [m_lastEntity](#), and [m_onEntityCreatedCallbacks](#).

Referenced by [cli::Client::Client\(\)](#).

Here is the caller graph for this function:



7.35.3.3 getAll()

```
template<typename T >
std::unordered_map< Entity, T > & ecs::Registry::getAll () [inline]
```

Definition at line 59 of file [Registry.hpp](#).

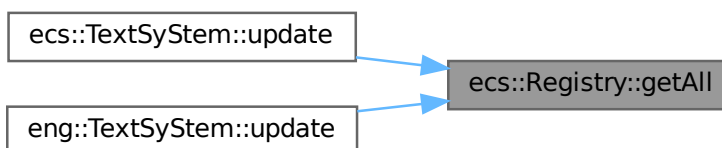
References [getPool\(\)](#).

Referenced by [ecs::TextSyStem::update\(\)](#), and [eng::TextSyStem::update\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.35.3.4 GetComponent()

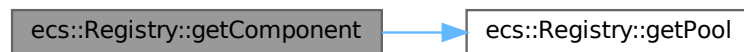
```
template<typename T >  
T * ecs::Registry::GetComponent (   
    Entity e) [inline]
```

Definition at line 53 of file [Registry.hpp](#).

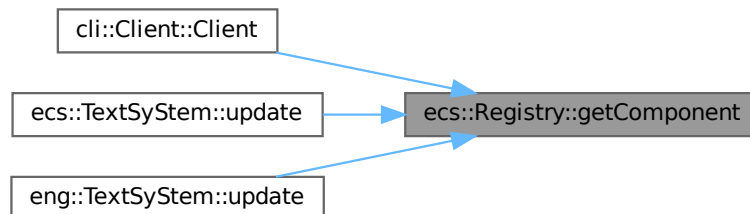
References [getPool\(\)](#).

Referenced by [cli::Client::Client\(\)](#), [ecs::TextSyStem::update\(\)](#), and [eng::TextSyStem::update\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.35.3.5 getPool()

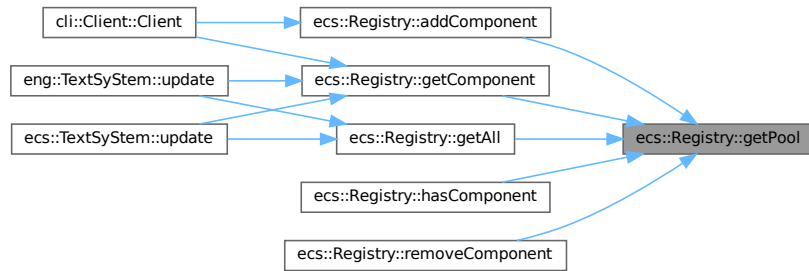
```
template<typename T >  
Pool< T > & ecs::Registry::getPool () [inline], [private]
```

Definition at line 113 of file [Registry.hpp](#).

References [m_components](#).

Referenced by [addComponent\(\)](#), [getAll\(\)](#), [GetComponent\(\)](#), [hasComponent\(\)](#), and [removeComponent\(\)](#).

Here is the caller graph for this function:



7.35.3.6 hasComponent()

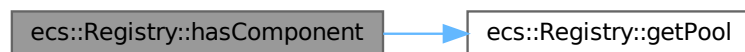
```

template<typename T >
bool ecs::Registry::hasComponent (
    Entity e) [inline]
  
```

Definition at line 61 of file [Registry.hpp](#).

References [getPool\(\)](#).

Here is the call graph for this function:



7.35.3.7 onComponentAdded()

```

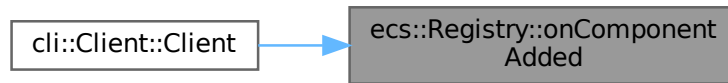
void ecs::Registry::onComponentAdded (
    std::function< void(Entity, const std::type_info &)> cb) [inline]
  
```

Definition at line 77 of file [Registry.hpp](#).

References [m_onComponentAddedCallbacks](#).

Referenced by [cli::Client::Client\(\)](#).

Here is the caller graph for this function:



7.35.3.8 onEntityCreated()

```
void ecs::Registry::onEntityCreated (
    std::function< void(Entity)> cb) [inline]
```

Definition at line 72 of file [Registry.hpp](#).

References [m_onEntityCreatedCallbacks](#).

7.35.3.9 operator=() [1/2]

```
Registry & ecs::Registry::operator= (
    const Registry & ) [delete]
```

7.35.3.10 operator=() [2/2]

```
Registry & ecs::Registry::operator= (
    Registry && ) [delete]
```

7.35.3.11 removeComponent()

```
template<typename T >
void ecs::Registry::removeComponent (
    Entity e) [inline]
```

Definition at line 67 of file [Registry.hpp](#).

References [getPool\(\)](#).

Here is the call graph for this function:



7.35.4 Member Data Documentation

7.35.4.1 m_components

`std::unordered_map<std::type_index, std::unique_ptr<IPool> > ecs::Registry::m_components` [private]

Definition at line 124 of file [Registry.hpp](#).

Referenced by [getPool\(\)](#).

7.35.4.2 m_entities

`std::vector<Entity> ecs::Registry::m_entities` [private]

Definition at line 123 of file [Registry.hpp](#).

Referenced by [createEntity\(\)](#).

7.35.4.3 m_lastEntity

`Entity ecs::Registry::m_lastEntity = INVALID_ENTITY` [private]

Definition at line 122 of file [Registry.hpp](#).

Referenced by [createEntity\(\)](#).

7.35.4.4 m_onComponentAddedCallbacks

`std::vector<std::function<void(Entity, const std::type_info &)> > ecs::Registry::m_onComponentAddedCallbacks` [private]

Definition at line 126 of file [Registry.hpp](#).

Referenced by [addComponent\(\)](#), and [onComponentAdded\(\)](#).

7.35.4.5 m_onEntityCreatedCallbacks

`std::vector<std::function<void(Entity)> > ecs::Registry::m_onEntityCreatedCallbacks` [private]

Definition at line 125 of file [Registry.hpp](#).

Referenced by [createEntity\(\)](#), and [onEntityCreated\(\)](#).

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

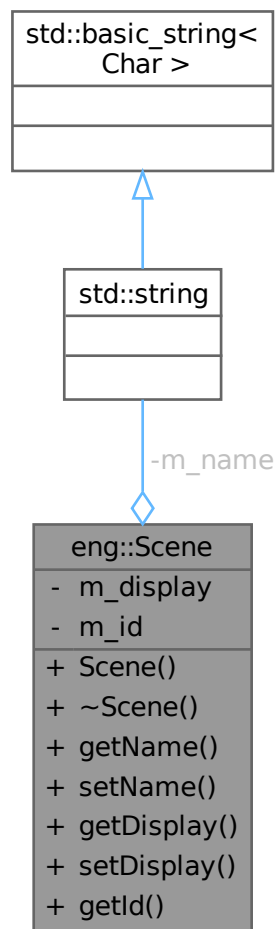
- [/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Registry.hpp](#)

7.36 eng::Scene Class Reference

class for scene and manage entities

```
#include <Scene.hpp>
```

Collaboration diagram for eng::Scene:



Public Member Functions

- [Scene](#) ()=default
- [~Scene](#) ()=default
- `std::string & getName ()`
- `void setName (const std::string &newName)`
- `bool getDisplay () const`
- `void setDisplay (const bool display)`
- `scene_id_t getId () const`

Private Attributes

- `std::string m_name` = "default_name"
- `bool m_display` = false
- `scene_id_t m_id` = 0

7.36.1 Detailed Description

class for scene and manage entities

Definition at line 21 of file [Scene.hpp](#).

7.36.2 Constructor & Destructor Documentation

7.36.2.1 Scene()

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

7.36.2.2 ~Scene()

`eng::Scene::~Scene ()` [default]

7.36.3 Member Function Documentation

7.36.3.1 getDisplay()

`bool eng::Scene::getDisplay () const` [inline], [nodiscard]

Definition at line 30 of file [Scene.hpp](#).

References [m_display](#).

7.36.3.2 getId()

`scene_id_t eng::Scene::getId () const` [inline], [nodiscard]

Definition at line 33 of file [Scene.hpp](#).

References [m_id](#).

7.36.3.3 getName()

`std::string & eng::Scene::getName ()` [inline], [nodiscard]

Definition at line 27 of file [Scene.hpp](#).

References [m_name](#).

7.36.3.4 setDisplay()

```
void eng::Scene::setDisplay (
    const bool display) [inline]
```

Definition at line 31 of file [Scene.hpp](#).

References [m_display](#).

7.36.3.5 setName()

```
void eng::Scene::setName (
    const std::string & newName) [inline]
```

Definition at line 28 of file [Scene.hpp](#).

References [m_name](#).

7.36.4 Member Data Documentation

7.36.4.1 m_display

```
bool eng::Scene::m_display = false [private]
```

Definition at line 37 of file [Scene.hpp](#).

Referenced by [getDisplay\(\)](#), and [setDisplay\(\)](#).

7.36.4.2 m_id

```
scene\_id\_t eng::Scene::m_id = 0 [private]
```

Definition at line 38 of file [Scene.hpp](#).

Referenced by [getId\(\)](#).

7.36.4.3 m_name

```
std::string eng::Scene::m_name = "default_name" [private]
```

Definition at line 36 of file [Scene.hpp](#).

Referenced by [getName\(\)](#), and [setName\(\)](#).

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

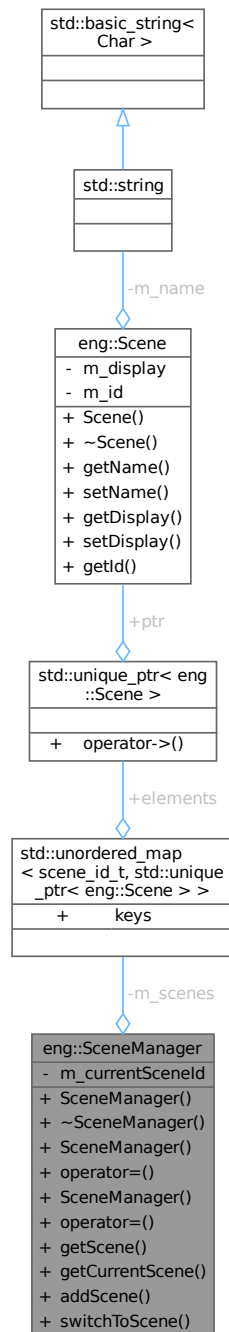
- [/home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/Scene.hpp](#)

7.37 eng::SceneManager Class Reference

Class for managing scenes.

```
#include <SceneManager.hpp>
```

Collaboration diagram for eng::SceneManager:



Public Member Functions

- [SceneManager](#) ()=default
- [~SceneManager](#) ()=default
- [SceneManager](#) (const [SceneManager](#) &)=delete
- [SceneManager](#) & operator= (const [SceneManager](#) &)=delete
- [SceneManager](#) ([SceneManager](#) &&)=delete
- [SceneManager](#) & operator= ([SceneManager](#) &&)=delete
- std::unique_ptr< [Scene](#) > & [getScene](#) ([scene_id_t](#) sceneId)
- std::unique_ptr< [Scene](#) > & [getCurrentScene](#) ()
- [scene_id_t](#) [addScene](#) (std::unique_ptr< [Scene](#) > scene)
- void [switchToScene](#) ([scene_id_t](#) sceneId)

Private Attributes

- std::unordered_map< [scene_id_t](#), std::unique_ptr< [Scene](#) > > [m_scenes](#)
- [scene_id_t](#) [m_currentSceneId](#) = 0

7.37.1 Detailed Description

Class for managing scenes.

Definition at line 22 of file [SceneManager.hpp](#).

7.37.2 Constructor & Destructor Documentation

7.37.2.1 SceneManager() [1/3]

eng::SceneManager::SceneManager () [default]

7.37.2.2 ~SceneManager()

eng::SceneManager::~SceneManager () [default]

7.37.2.3 SceneManager() [2/3]

eng::SceneManager::SceneManager (
const [SceneManager](#) &) [delete]

7.37.2.4 SceneManager() [3/3]

eng::SceneManager::SceneManager (
[SceneManager](#) &&) [delete]

7.37.3 Member Function Documentation

7.37.3.1 addScene()

```
scene_id_t eng::SceneManager::addScene (
    std::unique_ptr< Scene > scene)
```

7.37.3.2 getCurrentScene()

```
std::unique_ptr< Scene > & eng::SceneManager::getCurrentScene ()
```

7.37.3.3 getScene()

```
std::unique_ptr< Scene > & eng::SceneManager::getScene (
    scene_id_t sceneId)
```

7.37.3.4 operator=() [1/2]

```
SceneManager & eng::SceneManager::operator= (
    const SceneManager & ) [delete]
```

7.37.3.5 operator=() [2/2]

```
SceneManager & eng::SceneManager::operator= (
    SceneManager && ) [delete]
```

7.37.3.6 switchToScene()

```
void eng::SceneManager::switchToScene (
    scene_id_t sceneId)
```

7.37.4 Member Data Documentation

7.37.4.1 m_currentSceneId

```
scene_id_t eng::SceneManager::m_currentSceneId = 0 [private]
```

Definition at line 41 of file [SceneManager.hpp](#).

7.37.4.2 m_scenes

```
std::unordered_map<scene_id_t, std::unique_ptr<Scene> > eng::SceneManager::m_scenes [private]
```

Definition at line 40 of file [SceneManager.hpp](#).

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

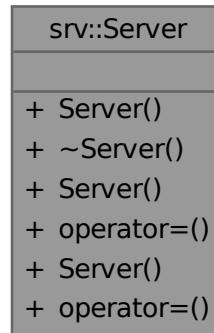
- [/home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/SceneManager.hpp](#)

7.38 srv::Server Class Reference

Class for the server.

#include <Server.hpp>

Collaboration diagram for srv::Server:



Public Member Functions

- [Server](#) (const [ArgsConfig](#) &config)
- [~Server](#) ()=default
- [Server](#) (const [Server](#) &)=delete
- [Server](#) & [operator=](#) (const [Server](#) &)=delete
- [Server](#) ([Server](#) &&)=delete
- [Server](#) & [operator=](#) ([Server](#) &&)=delete

7.38.1 Detailed Description

Class for the server.

Definition at line 19 of file [Server.hpp](#).

7.38.2 Constructor & Destructor Documentation

7.38.2.1 Server() [1/3]

```

srv::Server::Server (
    const ArgsConfig & config) [explicit]
  
```

Definition at line 5 of file [server.cpp](#).

References [BUILD_TYPE](#), [GIT_COMMIT_HASH](#), [GIT_TAG](#), [utl::INFO](#), [utl::Logger::log\(\)](#), [PROJECT_NAME](#), and [PROJECT_VERSION](#).

Here is the call graph for this function:



7.38.2.2 ~Server()

srv::Server::~~Server () [default]

7.38.2.3 Server() [2/3]

srv::Server::Server (
 const [Server](#) &) [delete]

7.38.2.4 Server() [3/3]

srv::Server::Server (
 [Server](#) &&) [delete]

7.38.3 Member Function Documentation

7.38.3.1 operator=() [1/2]

[Server](#) & srv::Server::operator= (
 const [Server](#) &) [delete]

7.38.3.2 operator=() [2/2]

[Server](#) & srv::Server::operator= (
 [Server](#) &&) [delete]

The documentation for this class was generated from the following files:

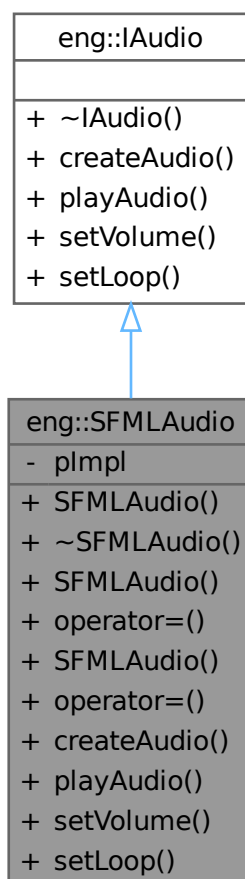
- [/home/masina/Projects/Epitech/rtype/server/include/Server/Server.hpp](#)
- [/home/masina/Projects/Epitech/rtype/server/src/server.cpp](#)

7.39 eng::SFMLAudio Class Reference

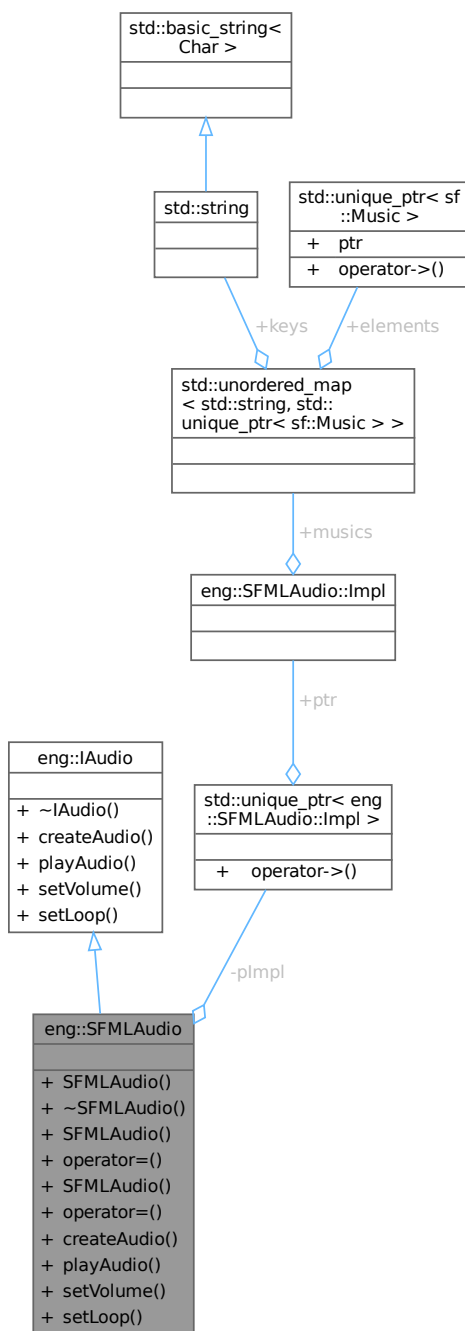
Class for audio management.

```
#include <SFMLAudio.hpp>
```

Inheritance diagram for eng::SFMLAudio:



Collaboration diagram for `eng::SFMLAudio`:



Classes

- struct [Impl](#)

Public Member Functions

- [SFMLAudio](#) ()

- [~SFMLAudio](#) () override
- [SFMLAudio](#) (const [SFMLAudio](#) &)=delete
- [SFMLAudio](#) & [operator=](#) (const [SFMLAudio](#) &)=delete
- [SFMLAudio](#) ([SFMLAudio](#) &&)=delete
- [SFMLAudio](#) & [operator=](#) ([SFMLAudio](#) &&)=delete
- void [createAudio](#) (const std::string &path, float volume, bool loop, const std::string &name) override
- void [playAudio](#) (const std::string &name) override
- void [setVolume](#) (const std::string &name, float volume) override
- void [setLoop](#) (const std::string &name, bool loop) override

Public Member Functions inherited from [eng::IAudio](#)

- virtual [~IAudio](#) ()=default

Private Attributes

- std::unique_ptr< [Impl](#) > [pImpl](#)

7.39.1 Detailed Description

Class for audio management.

Definition at line 22 of file [SFMLAudio.hpp](#).

7.39.2 Constructor & Destructor Documentation

7.39.2.1 SFMLAudio() [1/3]

`eng::SFMLAudio::SFMLAudio ()`

Definition at line 14 of file [SFMLAudio.cpp](#).

7.39.2.2 ~SFMLAudio()

`eng::SFMLAudio::~~SFMLAudio ()` [override], [default]

7.39.2.3 SFMLAudio() [2/3]

`eng::SFMLAudio::SFMLAudio (`
 const [SFMLAudio](#) &) [delete]

7.39.2.4 SFMLAudio() [3/3]

`eng::SFMLAudio::SFMLAudio (`
 [SFMLAudio](#) &&) [delete]

7.39.3 Member Function Documentation

7.39.3.1 createAudio()

```
void eng::SFMLAudio::createAudio (
    const std::string & path,
    float volume,
    bool loop,
    const std::string & name) [override], [virtual]
```

Implements [eng::IAudio](#).

Definition at line 17 of file [SFMLAudio.cpp](#).

References [pImpl](#).

7.39.3.2 operator=() [1/2]

```
SFMLAudio & eng::SFMLAudio::operator= (
    const SFMLAudio & ) [delete]
```

7.39.3.3 operator=() [2/2]

```
SFMLAudio & eng::SFMLAudio::operator= (
    SFMLAudio && ) [delete]
```

7.39.3.4 playAudio()

```
void eng::SFMLAudio::playAudio (
    const std::string & name) [override], [virtual]
```

Implements [eng::IAudio](#).

Definition at line 30 of file [SFMLAudio.cpp](#).

References [pImpl](#).

7.39.3.5 setLoop()

```
void eng::SFMLAudio::setLoop (
    const std::string & name,
    bool loop) [override], [virtual]
```

Implements [eng::IAudio](#).

Definition at line 44 of file [SFMLAudio.cpp](#).

References [pImpl](#).

7.39.3.6 setVolume()

```
void eng::SFMLAudio::setVolume (
    const std::string & name,
    float volume) [override], [virtual]
```

Implements [eng::IAudio](#).

Definition at line 37 of file [SFMLAudio.cpp](#).

References [pImpl](#).

7.39.4 Member Data Documentation

7.39.4.1 pImpl

```
std::unique_ptr<Impl> eng::SFMLAudio::pImpl [private]
```

Definition at line 40 of file [SFMLAudio.hpp](#).

Referenced by [createAudio\(\)](#), [playAudio\(\)](#), [setLoop\(\)](#), and [setVolume\(\)](#).

The documentation for this class was generated from the following files:

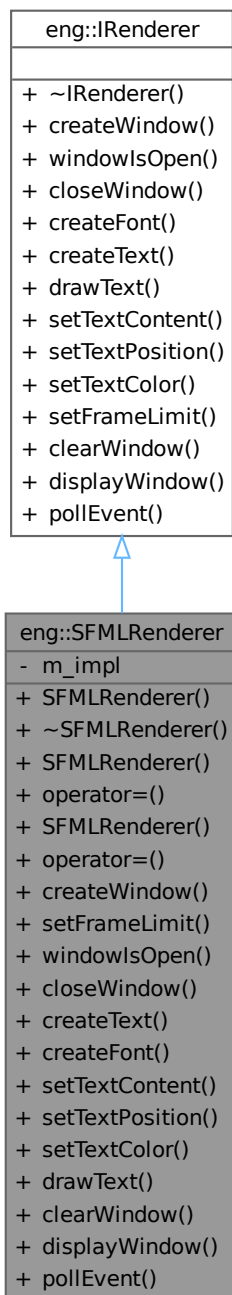
- [/home/masina/Projects/Epitech/rtype/modules/Audio/SFMLAudio/include/SFMLAudio/SFMLAudio.hpp](#)
- [/home/masina/Projects/Epitech/rtype/modules/Audio/SFMLAudio/src/SFMLAudio.cpp](#)

7.40 eng::SFMLRenderer Class Reference

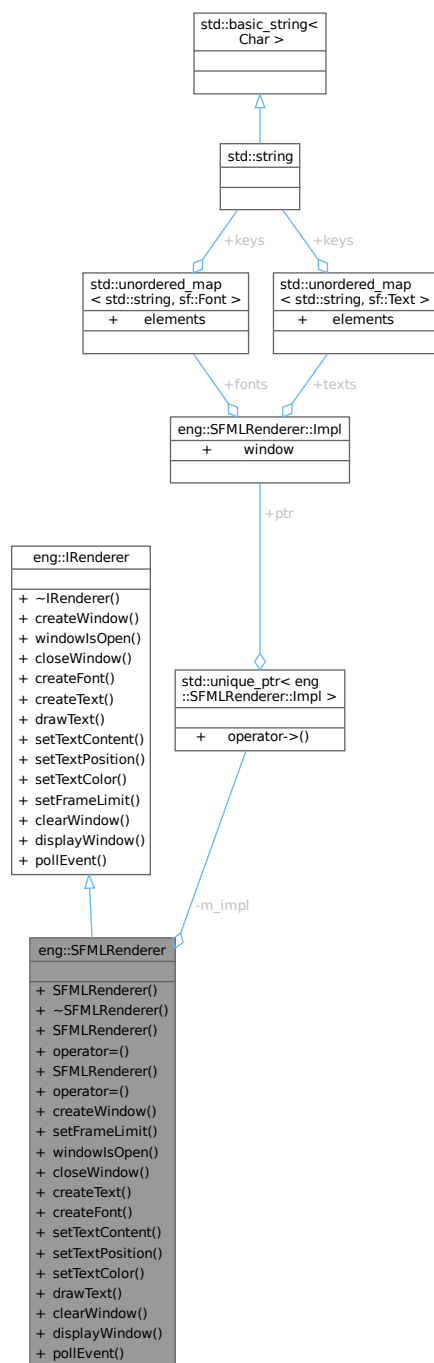
Class for the R-Type game.

```
#include <SFMLRenderer.hpp>
```

Inheritance diagram for eng::SFMLRenderer:



Collaboration diagram for eng::SFMLRenderer:



Classes

- struct [Impl](#)

Public Member Functions

- [SFMLRenderer](#) ()

- [~SFMLRenderer](#) () override
- [SFMLRenderer](#) (const [SFMLRenderer](#) &)=delete
- [SFMLRenderer](#) & operator= (const [SFMLRenderer](#) &)=delete
- [SFMLRenderer](#) ([SFMLRenderer](#) &&)=delete
- [SFMLRenderer](#) & operator= ([SFMLRenderer](#) &&)=delete
- void [createWindow](#) (const std::string &title, unsigned int height, unsigned int width, unsigned int frameLimit, bool fullscreen) override
- void [setFrameLimit](#) (unsigned int frameLimit) override
- bool [windowIsOpen](#) () const override
- void [closeWindow](#) () override
- void [createText](#) ([Text](#) text) override
- void [createFont](#) ([Font](#) font) override
- void [setTextContent](#) (const std::string &name, const std::string &content) override
- void [setTextPosition](#) (const std::string &name, int x, int y) override
- void [setTextColor](#) (const std::string &name, [Color](#) color) override
- void [drawText](#) (const std::string &name) override
- void [clearWindow](#) ([Color](#) color) override
- void [displayWindow](#) () override
- bool [pollEvent](#) ([Event](#) &event) override

Public Member Functions inherited from [eng::IRenderer](#)

- virtual [~IRenderer](#) ()=default

Private Attributes

- std::unique_ptr< [Impl](#) > [m_impl](#)

7.40.1 Detailed Description

Class for the R-Type game.

Definition at line 22 of file [SFMLRenderer.hpp](#).

7.40.2 Constructor & Destructor Documentation

7.40.2.1 SFMLRenderer() [1/3]

[eng::SFMLRenderer::SFMLRenderer](#) ()

Definition at line 18 of file [SFMLRenderer.cpp](#).

7.40.2.2 ~SFMLRenderer()

[eng::SFMLRenderer::~~SFMLRenderer](#) () [override], [default]

7.40.2.3 SFMLRenderer() [2/3]

eng::SFMLRenderer::SFMLRenderer (
 const [SFMLRenderer](#) &) [delete]

7.40.2.4 SFMLRenderer() [3/3]

eng::SFMLRenderer::SFMLRenderer (
 [SFMLRenderer](#) &&) [delete]

7.40.3 Member Function Documentation

7.40.3.1 clearWindow()

void eng::SFMLRenderer::clearWindow (
 [Color](#) color) [override], [virtual]

Implements [eng::IRenderer](#).

Definition at line 105 of file [SFMLRenderer.cpp](#).

References [eng::Color::a](#), [eng::Color::b](#), [eng::Color::g](#), and [eng::Color::r](#).

7.40.3.2 closeWindow()

void eng::SFMLRenderer::closeWindow () [override], [virtual]

Implements [eng::IRenderer](#).

Definition at line 32 of file [SFMLRenderer.cpp](#).

7.40.3.3 createFont()

void eng::SFMLRenderer::createFont (
 [Font](#) font) [override], [virtual]

Implements [eng::IRenderer](#).

Definition at line 36 of file [SFMLRenderer.cpp](#).

References [eng::Font::name](#), and [eng::Font::path](#).

7.40.3.4 createText()

void eng::SFMLRenderer::createText (
 [Text](#) text) [override], [virtual]

Implements [eng::IRenderer](#).

Definition at line 46 of file [SFMLRenderer.cpp](#).

References [eng::Color::a](#), [eng::Color::b](#), [eng::Text::color](#), [eng::Text::content](#), [eng::Text::fontName](#), [eng::Color::g](#), [eng::Text::name](#), [eng::Color::r](#), [eng::Text::size](#), [eng::Text::x](#), and [eng::Text::y](#).

7.40.3.5 createWindow()

```
void eng::SFMLRenderer::createWindow (
    const std::string & title,
    unsigned int height,
    unsigned int width,
    unsigned int frameLimit,
    bool fullscreen) [override], [virtual]
```

Implements [eng::IRenderer](#).

Definition at line 22 of file [SFMLRenderer.cpp](#).

7.40.3.6 displayWindow()

```
void eng::SFMLRenderer::displayWindow () [override], [virtual]
```

Implements [eng::IRenderer](#).

Definition at line 110 of file [SFMLRenderer.cpp](#).

7.40.3.7 drawText()

```
void eng::SFMLRenderer::drawText (
    const std::string & name) [override], [virtual]
```

Implements [eng::IRenderer](#).

Definition at line 93 of file [SFMLRenderer.cpp](#).

7.40.3.8 operator=() [1/2]

```
SFMLRenderer & eng::SFMLRenderer::operator= (
    const SFMLRenderer & ) [delete]
```

7.40.3.9 operator=() [2/2]

```
SFMLRenderer & eng::SFMLRenderer::operator= (
    SFMLRenderer && ) [delete]
```

7.40.3.10 pollEvent()

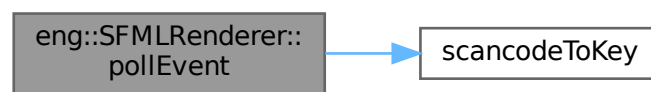
```
bool eng::SFMLRenderer::pollEvent (  
    Event & event) [override], [virtual]
```

Implements [eng::IRenderer](#).

Definition at line 206 of file [SFMLRenderer.cpp](#).

References [eng::Closed](#), [eng::KeyPressed](#), [eng::KeyReleased](#), [eng::None](#), and [scancodeToKey\(\)](#).

Here is the call graph for this function:



7.40.3.11 setFrameLimit()

```
void eng::SFMLRenderer::setFrameLimit (  
    unsigned int frameLimit) [override], [virtual]
```

Implements [eng::IRenderer](#).

Definition at line 34 of file [SFMLRenderer.cpp](#).

7.40.3.12 setTextColor()

```
void eng::SFMLRenderer::setTextColor (  
    const std::string & name,  
    Color color) [override], [virtual]
```

Implements [eng::IRenderer](#).

Definition at line 81 of file [SFMLRenderer.cpp](#).

References [eng::Color::a](#), [eng::Color::b](#), [eng::Color::g](#), and [eng::Color::r](#).

7.40.3.13 setTextContent()

```
void eng::SFMLRenderer::setTextContent (  
    const std::string & name,  
    const std::string & content) [override], [virtual]
```

Implements [eng::IRenderer](#).

Definition at line 57 of file [SFMLRenderer.cpp](#).

7.40.3.14 setTexturePosition()

```
void eng::SFMLRenderer::setTexturePosition (
    const std::string & name,
    int x,
    int y)  [override], [virtual]
```

Implements [eng::IRenderer](#).

Definition at line 69 of file [SFMLRenderer.cpp](#).

7.40.3.15 windowIsOpen()

```
bool eng::SFMLRenderer::windowIsOpen () const  [override], [virtual]
```

Implements [eng::IRenderer](#).

Definition at line 30 of file [SFMLRenderer.cpp](#).

7.40.4 Member Data Documentation

7.40.4.1 m_impl

```
std::unique_ptr<Impl> eng::SFMLRenderer::m_impl  [private]
```

Definition at line 53 of file [SFMLRenderer.hpp](#).

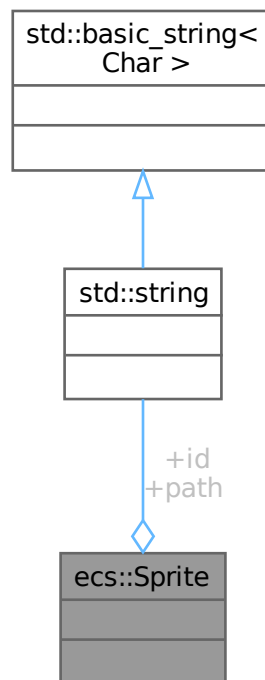
The documentation for this class was generated from the following files:

- [/home/masina/Projects/Epitech/rtype/modules/Renderer/SFMLRenderer/include/SFMLRenderer/SFMLRenderer.hpp](#)
- [/home/masina/Projects/Epitech/rtype/modules/Renderer/SFMLRenderer/src/SFMLRenderer.cpp](#)

7.41 ecs::Sprite Struct Reference

```
#include <Component.hpp>
```


Collaboration diagram for ecs::Sprite:



Public Attributes

- `std::string` [id](#)
- `std::string` [path](#)

7.41.1 Detailed Description

Definition at line 33 of file [Component.hpp](#).

7.41.2 Member Data Documentation

7.41.2.1 id

`std::string` `ecs::Sprite::id`

Definition at line 35 of file [Component.hpp](#).

7.41.2.2 path

std::string ecs::Sprite::path

Definition at line 36 of file [Component.hpp](#).

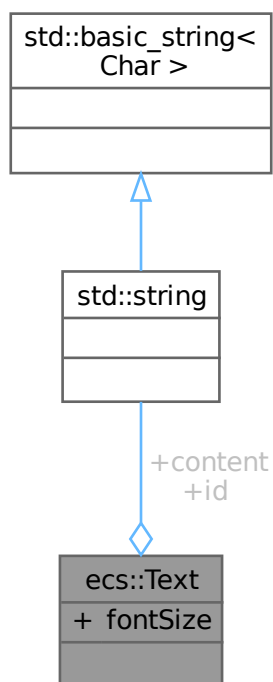
The documentation for this struct was generated from the following file:

- [/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.hpp](#)

7.42 ecs::Text Struct Reference

```
#include <Component.hpp>
```

Collaboration diagram for ecs::Text:



Public Attributes

- std::string [id](#)
- std::string [content](#)
- int [fontSize](#)

7.42.1 Detailed Description

Definition at line 38 of file [Component.hpp](#).

7.42.2 Member Data Documentation

7.42.2.1 content

```
std::string ecs::Text::content
```

Definition at line 41 of file [Component.hpp](#).

7.42.2.2 fontSize

```
int ecs::Text::fontSize
```

Definition at line 42 of file [Component.hpp](#).

7.42.2.3 id

```
std::string ecs::Text::id
```

Definition at line 40 of file [Component.hpp](#).

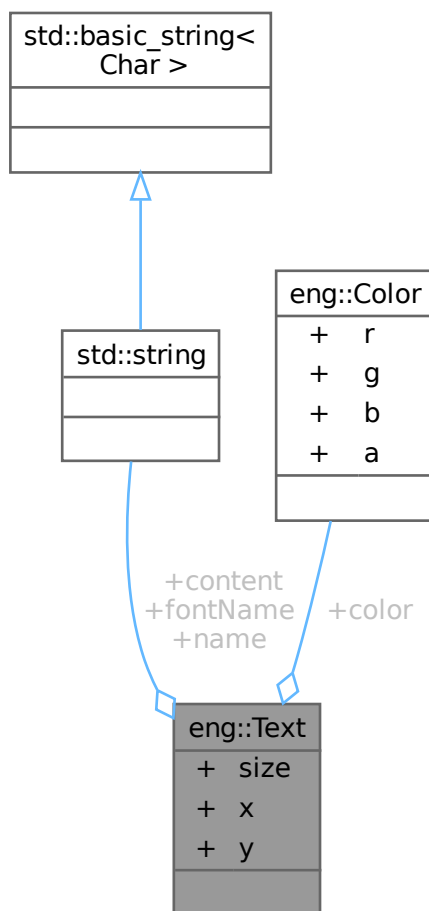
The documentation for this struct was generated from the following file:

- [/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.hpp](#)

7.43 eng::Text Struct Reference

```
#include <IRenderer.hpp>
```

Collaboration diagram for `eng::Text`:



Public Attributes

- `std::string` `fontName`
- `Color` `color`
- `std::string` `content`
- `int` `size`
- `int` `x`
- `int` `y`
- `std::string` `name`

7.43.1 Detailed Description

Definition at line 32 of file [IRenderer.hpp](#).

7.43.2 Member Data Documentation

7.43.2.1 color

[Color](#) eng::Text::color

Definition at line 35 of file [IRenderer.hpp](#).

Referenced by [eng::SFMLRenderer::createText\(\)](#).

7.43.2.2 content

std::string eng::Text::content

Definition at line 36 of file [IRenderer.hpp](#).

Referenced by [eng::SFMLRenderer::createText\(\)](#).

7.43.2.3 fontName

std::string eng::Text::fontName

Definition at line 34 of file [IRenderer.hpp](#).

Referenced by [cli::Client::Client\(\)](#), and [eng::SFMLRenderer::createText\(\)](#).

7.43.2.4 name

std::string eng::Text::name

Definition at line 40 of file [IRenderer.hpp](#).

Referenced by [eng::SFMLRenderer::createText\(\)](#).

7.43.2.5 size

int eng::Text::size

Definition at line 37 of file [IRenderer.hpp](#).

Referenced by [eng::SFMLRenderer::createText\(\)](#).

7.43.2.6 x

int eng::Text::x

Definition at line 38 of file [IRenderer.hpp](#).

Referenced by [eng::SFMLRenderer::createText\(\)](#).

7.43.2.7 y

int eng::Text::y

Definition at line 39 of file [IRenderer.hpp](#).

Referenced by [eng::SFMLRenderer::createText\(\)](#).

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

- [/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IRenderer.hpp](#)

7.44 ecs::TextSyStem Class Reference

Class for managing entities and their components.

#include <Systems.hpp>

Collaboration diagram for ecs::TextSyStem:

ecs::TextSyStem
- m_drawCallback
+ TextSyStem()
+ ~TextSyStem()
+ TextSyStem()
+ operator=()
+ TextSyStem()
+ operator=()
+ setDrawCallback()
+ update()

Public Types

- using [DrawCallback](#) = std::function<void(const [Text](#) &, const [Transform](#) &, const [Color](#) &)>

Public Member Functions

- [TextSyStem](#) ()=default
- [~TextSyStem](#) ()=default
- [TextSyStem](#) (const [TextSyStem](#) &)=delete
- [TextSyStem](#) & operator= (const [TextSyStem](#) &)=delete
- [TextSyStem](#) ([TextSyStem](#) &&)=delete
- [TextSyStem](#) & operator= ([TextSyStem](#) &&)=delete
- void [setDrawCallback](#) ([DrawCallback](#) cb)
- void [update](#) ([Registry](#) ®istry) const

Private Attributes

- [DrawCallback m_drawCallback](#)

7.44.1 Detailed Description

Class for managing entities and their components.

Definition at line 20 of file [Systems.hpp](#).

7.44.2 Member Typedef Documentation

7.44.2.1 DrawCallback

using [ecs::TextSyStem::DrawCallback](#) = std::function<void(const [Text](#) &, const [Transform](#) &, const [Color](#) &)>

Definition at line 31 of file [Systems.hpp](#).

7.44.3 Constructor & Destructor Documentation

7.44.3.1 TextSyStem() [1/3]

ecs::TextSyStem::TextSyStem () [default]

7.44.3.2 ~TextSyStem()

ecs::TextSyStem::~~TextSyStem () [default]

7.44.3.3 TextSyStem() [2/3]

ecs::TextSyStem::TextSyStem (
 const [TextSyStem](#) &) [delete]

7.44.3.4 TextSyStem() [3/3]

ecs::TextSyStem::TextSyStem (
 [TextSyStem](#) &&) [delete]

7.44.4 Member Function Documentation

7.44.4.1 operator=() [1/2]

[TextSyStem](#) & ecs::TextSyStem::operator= (
 const [TextSyStem](#) &) [delete]

7.44.4.2 operator=() [2/2]

[TextSyStem](#) & ecs::TextSyStem::operator= (
[TextSyStem](#) &&) [delete]

7.44.4.3 setDrawCallback()

void ecs::TextSyStem::setDrawCallback (
[DrawCallback](#) cb) [inline]

Definition at line 33 of file [Systems.hpp](#).

References [m_drawCallback](#).

7.44.4.4 update()

void ecs::TextSyStem::update (
[Registry](#) & registry) const [inline]

Definition at line 35 of file [Systems.hpp](#).

References [ecs::Registry::getAll\(\)](#), [ecs::Registry::getComponent\(\)](#), and [m_drawCallback](#).

Here is the call graph for this function:



7.44.5 Member Data Documentation

7.44.5.1 m_drawCallback

[DrawCallback](#) ecs::TextSyStem::m_drawCallback [private]

Definition at line 50 of file [Systems.hpp](#).

Referenced by [setDrawCallback\(\)](#), and [update\(\)](#).

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

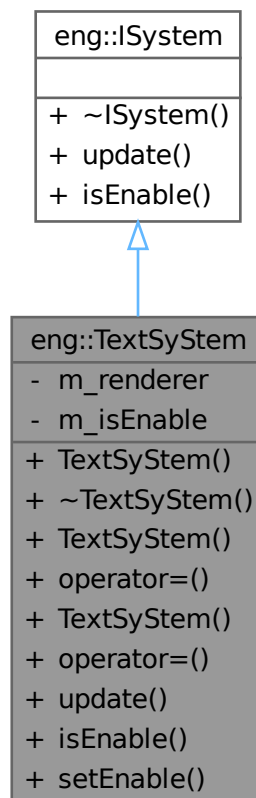
- [/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Systems.hpp](#)

7.45 eng::TextSyStem Class Reference

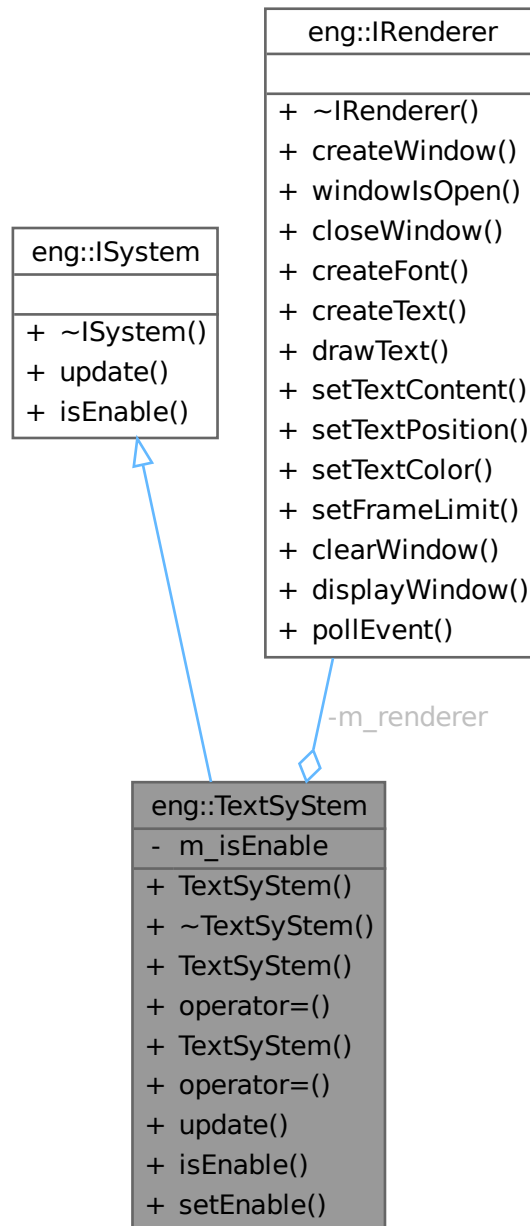
Class for managing entities and their components.

```
#include <Systems.hpp>
```

Inheritance diagram for eng::TextSyStem:



Collaboration diagram for eng::TextSyStem:



Public Member Functions

- `TextSyStem (IRenderer &renderer)`
- `~TextSyStem ()` override=default
- `TextSyStem (const TextSyStem &)=delete`
- `TextSyStem & operator= (const TextSyStem &)=delete`
- `TextSyStem (TextSyStem &&)=delete`
- `TextSyStem & operator= (TextSyStem &&)=delete`

- void [update](#) ([ecs::Registry](#) ®istry, float dt) override
- bool [isEnabled](#) () override
- void [setEnabled](#) (const bool enable)

Public Member Functions inherited from [eng::ISystem](#)

- virtual [~ISystem](#) ()=default

Private Attributes

- [IRenderer](#) & [m_renderer](#)
- bool [m_isEnable](#) = true

7.45.1 Detailed Description

Class for managing entities and their components.

Definition at line 29 of file [Systems.hpp](#).

7.45.2 Constructor & Destructor Documentation

7.45.2.1 TextSyStem() [1/3]

```
eng::TextSyStem::TextSyStem (
    IRenderer & renderer)    [inline], [explicit]
```

Definition at line 32 of file [Systems.hpp](#).

7.45.2.2 ~TextSyStem()

```
eng::TextSyStem::~TextSyStem ()    [override], [default]
```

7.45.2.3 TextSyStem() [2/3]

```
eng::TextSyStem::TextSyStem (
    const TextSyStem & )    [delete]
```

7.45.2.4 TextSyStem() [3/3]

```
eng::TextSyStem::TextSyStem (
    TextSyStem && )    [delete]
```

7.45.3 Member Function Documentation

7.45.3.1 isEnabled()

bool eng::TextSyStem::isEnabled () [inline], [override], [virtual]

Implements [eng::ISystem](#).

Definition at line 58 of file [Systems.hpp](#).

7.45.3.2 operator=() [1/2]

[TextSyStem](#) & eng::TextSyStem::operator= (
 const [TextSyStem](#) &) [delete]

7.45.3.3 operator=() [2/2]

[TextSyStem](#) & eng::TextSyStem::operator= (
 [TextSyStem](#) &&) [delete]

7.45.3.4 setEnable()

void eng::TextSyStem::setEnable (
 const bool enable) [inline]

Definition at line 59 of file [Systems.hpp](#).

7.45.3.5 update()

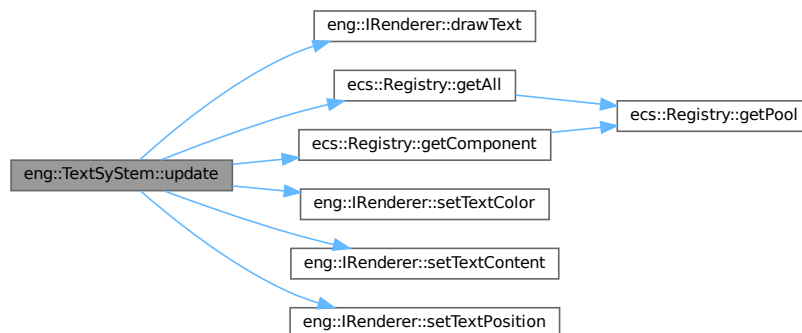
void eng::TextSyStem::update (
 [ecs::Registry](#) & registry,
 float dt) [inline], [override], [virtual]

Implements [eng::ISystem](#).

Definition at line 40 of file [Systems.hpp](#).

References [eng::IRenderer::drawText\(\)](#), [ecs::Registry::getAll\(\)](#), [ecs::Registry::getComponent\(\)](#), [m_renderer](#), [eng::IRenderer::setTextColor\(\)](#), [eng::IRenderer::setTextContent\(\)](#), and [eng::IRenderer::setTextPosition\(\)](#).

Here is the call graph for this function:



7.45.4 Member Data Documentation

7.45.4.1 m_isEnable

bool eng::TextSyStem::m_isEnable = true [private]

Definition at line 63 of file [Systems.hpp](#).

7.45.4.2 m_renderer

[IRenderer&](#) eng::TextSyStem::m_renderer [private]

Definition at line 62 of file [Systems.hpp](#).

Referenced by [update\(\)](#).

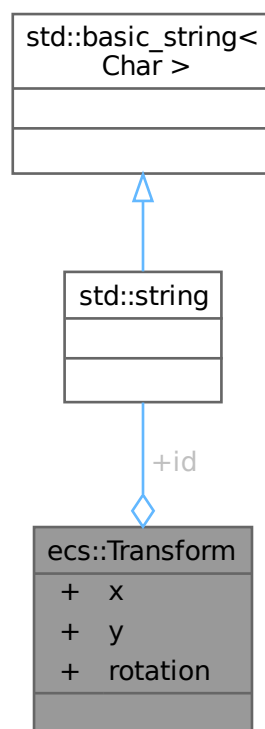
The documentation for this class was generated from the following file:

- [/home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/Systems.hpp](#)

7.46 ecs::Transform Struct Reference

```
#include <Component.hpp>
```

Collaboration diagram for ecs::Transform:



Public Attributes

- `std::string` [id](#)
- `float` [x](#)
- `float` [y](#)
- `float` [rotation](#)

7.46.1 Detailed Description

Definition at line [44](#) of file [Component.hpp](#).

7.46.2 Member Data Documentation

7.46.2.1 id

`std::string` `ecs::Transform::id`

Definition at line [46](#) of file [Component.hpp](#).

7.46.2.2 rotation

`float` `ecs::Transform::rotation`

Definition at line [48](#) of file [Component.hpp](#).

7.46.2.3 x

`float` `ecs::Transform::x`

Definition at line [47](#) of file [Component.hpp](#).

7.46.2.4 y

`float` `ecs::Transform::y`

Definition at line [47](#) of file [Component.hpp](#).

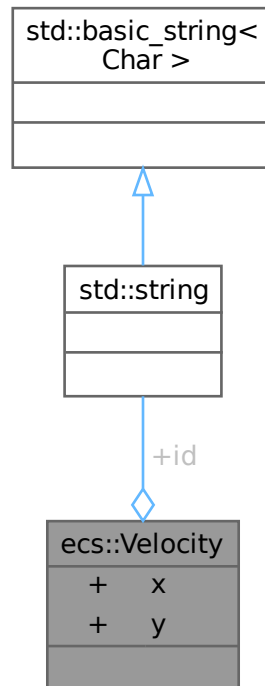
The documentation for this struct was generated from the following file:

- [/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.hpp](#)

7.47 ecs::Velocity Struct Reference

```
#include <Component.hpp>
```

Collaboration diagram for `ecs::Velocity`:



Public Attributes

- `std::string` [id](#)
- `float` [x](#)
- `float` [y](#)

7.47.1 Detailed Description

Definition at line [50](#) of file [Component.hpp](#).

7.47.2 Member Data Documentation

7.47.2.1 `id`

`std::string` `ecs::Velocity::id`

Definition at line [52](#) of file [Component.hpp](#).

7.47.2.2 x

float ecs::Velocity::x

Definition at line 53 of file [Component.hpp](#).

7.47.2.3 y

float ecs::Velocity::y

Definition at line 53 of file [Component.hpp](#).

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

- [/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.hpp](#)

Chapter 8

File Documentation

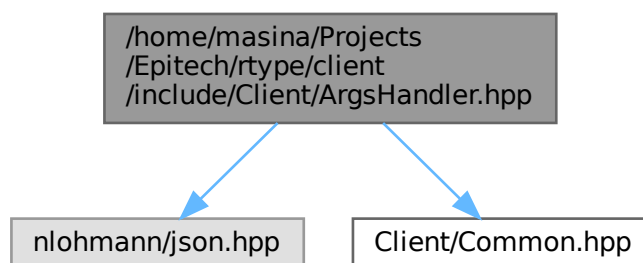
8.1 /home/masina/Projects/Epitech/rtype/client/include/Client/ArgsHandler.hpp File Reference

This file contains the ArgsHandler class declaration.

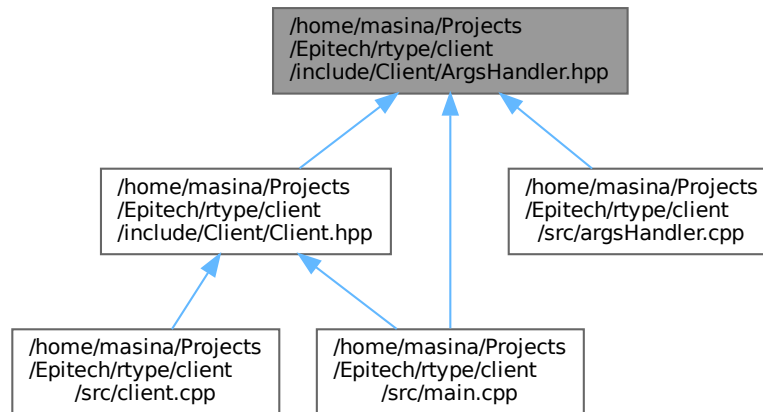
```
#include <nlohmann/json.hpp>
```

```
#include "Client/Common.hpp"
```

Include dependency graph for ArgsHandler.hpp:



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



Classes

- struct [cli::ArgsConfig](#)
- struct [cli::EnvConfig](#)
- class [cli::ArgsHandler](#)

Class to handle command line arguments.

Namespaces

- namespace [cli](#)

Typedefs

- using [cli::json](#) = [nlohmann::json](#)

8.1.1 Detailed Description

This file contains the ArgsHandler class declaration.

Definition in file [ArgsHandler.hpp](#).

8.2 ArgsHandler.hpp

[Go to the documentation of this file.](#)

```

00001 ///  

00002 ///  

00003 ///  

00004 ///  

00005 ///  

00006 ///  

00007 #pragma once  

00008 ///  

00009 #include <nlohmann/json.hpp>  

00010 ///  

00011 #include "Client/Common.hpp"  

00012 ///  

00013 namespace cli  

00014 {  

00015     ///  

00016     using json = nlohmann::json;  

00017     ///  

00018     struct ArgsConfig  

00019     {  

00020         bool exit = false;  

00021         unsigned int width = Config::Window::DEFAULT_WINDOW_WIDTH;  

00022         unsigned int height = Config::Window::DEFAULT_WINDOW_HEIGHT;  

00023         unsigned int frameLimit = Config::Window::DEFAULT_WINDOW_FRAME_LIMIT;  

00024         bool fullscreen = Config::Window::DEFAULT_WINDOW_FULLSCREEN;  

00025         static ArgsConfig fromFile(const std::string &path);  

00026     }; // struct Config  

00027     struct EnvConfig  

00028     {  

00029     };  

00030     ///  

00031     ///  

00032     ///  

00033     ///  

00034     ///  

00035     ///  

00036     class ArgsHandler  

00037     {  

00038     public:  

00039         ArgsHandler() = default;  

00040         ~ArgsHandler() = default;  

00041         ///  

00042         ArgsHandler(const ArgsHandler &) = delete;  

00043         ArgsHandler &operator=(const ArgsHandler &) = delete;  

00044         ArgsHandler(ArgsHandler &&) = delete;  

00045         ArgsHandler &operator=(ArgsHandler &&) = delete;  

00046         static ArgsConfig ParseArgs(int argc, const char *const argv[]);  

00047         static EnvConfig ParseEnv(const char *const env[]);  

00048     private:  

00049     }; // class ArgsHandler  

00050 } // namespace cli

```

8.3 /home/masina/Projects/Epitech/rtype/server/include/Server/↵ ArgsHandler.hpp File Reference

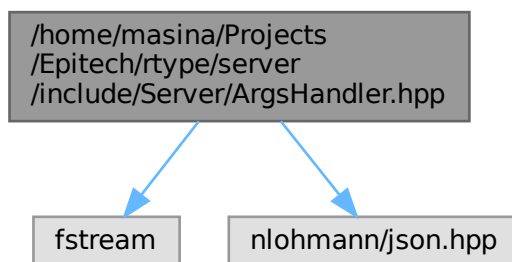
This file contains the ArgsHandler class declaration.

```

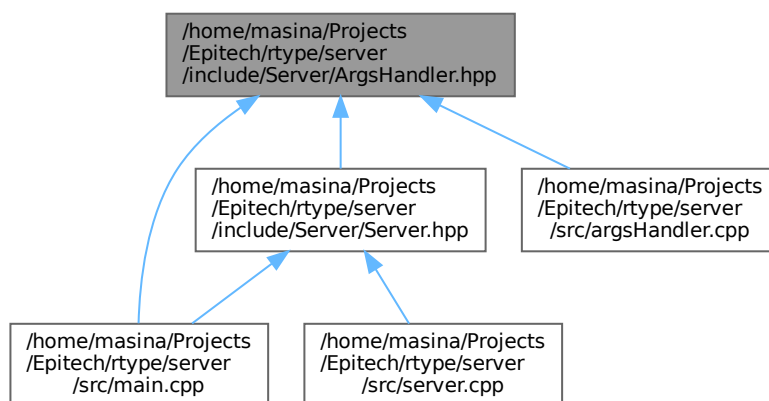
#include <fstream>
#include "nlohmann/json.hpp"

```

Include dependency graph for ArgsHandler.hpp:



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



Classes

- struct [srv::ArgsConfig](#)
- struct [srv::EnvConfig](#)
- class [srv::ArgsHandler](#)

Class to handle command line arguments.

Namespaces

- namespace [srv](#)

Typedefs

- using [srv::json](#) = [nlohmann::json](#)

8.3.1 Detailed Description

This file contains the ArgsHandler class declaration.

Definition in file [ArgsHandler.hpp](#).

8.4 ArgsHandler.hpp

[Go to the documentation of this file.](#)

```

00001 ///  

00002 ///  

00003 ///  

00004 ///  

00005 ///  

00006 ///  

00007 #pragma once  

00008  

00009 #include <fstream>  

00010  

00011 #include "nlohmann/json.hpp"  

00012  

00013 namespace srv  

00014 {  

00015  

00016     using json = nlohmann::json;  

00017  

00018     struct ArgsConfig  

00019     {  

00020         bool exit = false;  

00021         std::string host = "0.0.0.0";  

00022         unsigned int port = 2560;  

00023  

00024         static ArgsConfig fromFile(const std::string &path);  

00025     }; // struct Config  

00026     struct EnvConfig  

00027     {  

00028     };  

00029  

00030     ///  

00031     ///  

00032     ///  

00033     ///  

00034     ///  

00035     class ArgsHandler  

00036     {  

00037  

00038     public:  

00039         ArgsHandler() = default;  

00040         ~ArgsHandler() = default;  

00041  

00042         ArgsHandler(const ArgsHandler &) = delete;  

00043         ArgsHandler &operator=(const ArgsHandler &) = delete;  

00044         ArgsHandler(ArgsHandler &&) = delete;  

00045         ArgsHandler &operator=(ArgsHandler &&) = delete;  

00046  

00047         static ArgsConfig ParseArgs(int argc, const char *const argv[]);  

00048         static EnvConfig ParseEnv(const char *const env[]);  

00049  

00050     private:  

00051     }; // class ArgsHandler  

00052  

00053 } // namespace srv

```

8.5 /home/masina/Projects/Epitech/rtype/client/include/Client/↵ Client.hpp File Reference

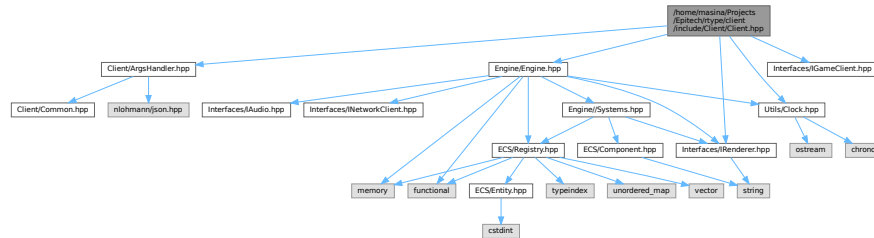
This file contains the Client class declaration.

```

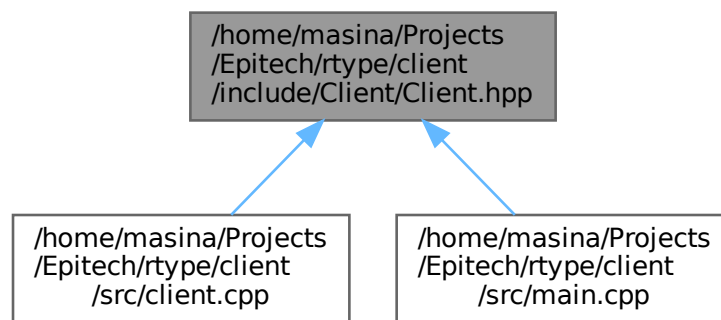
#include "Client/ArgsHandler.hpp"
#include "Engine/Engine.hpp"

```

```
#include "Interfaces/IGameClient.hpp"
#include "Interfaces/IRenderer.hpp"
#include "Utils/Clock.hpp"
Include dependency graph for Client.hpp:
```



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



Classes

- class `cli::Client`
Class for the client.

Namespaces

- namespace `cli`

8.5.1 Detailed Description

This file contains the Client class declaration.

Definition in file `Client.hpp`.

8.6 Client.hpp

Go to the documentation of this file.

```

00001 ///  

00002 ///  

00003 ///  

00004 ///  

00005 ///  

00006 ///  

00007 #pragma once  

00008  

00009 #include "Client/ArgsHandler.hpp"  

00010 #include "Engine/Engine.hpp"  

00011 #include "Interfaces/IGameClient.hpp"  

00012 #include "Interfaces/IRenderer.hpp"  

00013 #include "Utils/Clock.hpp"  

00014  

00015 namespace cli  

00016 {  

00017  

00018     ///  

00019     ///  

00020     ///  

00021     ///  

00022     ///  

00023     class Client  

00024     {  

00025  

00026     public:  

00027         explicit Client(const ArgsConfig &cfg);  

00028         ~Client() = default;  

00029  

00030         Client(const Client &) = delete;  

00031         Client &operator=(const Client &) = delete;  

00032         Client(Client &&) = delete;  

00033         Client &operator=(Client &&) = delete;  

00034  

00035     private:  

00036         std::unique_ptr<IGameClient> m_game;  

00037         std::unique_ptr<eng::Engine> m_engine;  

00038     }; // class Client  

00039  

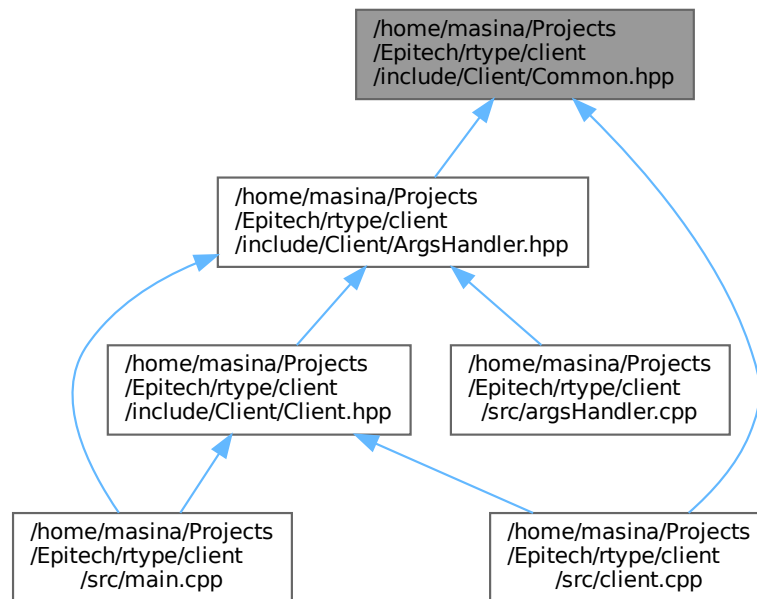
00040 } // namespace cli

```

8.7 /home/masina/Projects/Epitech/rtype/client/include/Client/↵ Common.hpp File Reference

This file contains common definitions and constants.

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



Namespaces

- namespace `cli`
- namespace `cli::Paths`
- namespace `cli::Paths::Audio`
- namespace `cli::Paths::Fonts`
- namespace `cli::Config`
- namespace `cli::Config::Window`
- namespace `cli::Config::Audio`

Variables

- constexpr auto `cli::Paths::Audio::AUDIO_TITLE` = "assets/audio/title.mp3"
- constexpr auto `cli::Paths::Audio::AUDIO_COIN` = "assets/audio/coin.mp3"
- constexpr auto `cli::Paths::Audio::AUDIO_BATTLE_THEME` = "assets/audio/battle_theme.mp3"
- constexpr auto `cli::Paths::Fonts::FONTS_RTYPE` = "assets/fonts/r-type.otf"
- constexpr auto `cli::Config::Window::DEFAULT_WINDOW_WIDTH` = 960
- constexpr auto `cli::Config::Window::DEFAULT_WINDOW_HEIGHT` = 540
- constexpr auto `cli::Config::Window::DEFAULT_WINDOW_FRAME_LIMIT` = 240
- constexpr auto `cli::Config::Window::DEFAULT_WINDOW_FULLSCREEN` = false
- constexpr auto `cli::Config::Audio::DEFAULT_AUDIO_VOLUME` = 50
- constexpr auto `cli::Config::Audio::DEFAULT_AUDIO_MUTED` = false

8.7.1 Detailed Description

This file contains common definitions and constants.

Definition in file [Common.hpp](#).

8.8 Common.hpp

[Go to the documentation of this file.](#)

```

00001 ///  

00002 ///  

00003 ///  

00004 ///  

00005 ///  

00006 ///  

00007 #pragma once  

00008  

00009 namespace cli  

00010 {  

00011     namespace Paths  

00012     {  

00013         namespace Audio  

00014         {  

00015             inline constexpr auto AUDIO_TITLE = "assets/audio/title.mp3";  

00016             inline constexpr auto AUDIO_COIN = "assets/audio/coin.mp3";  

00017             inline constexpr auto AUDIO_BATTLE_THEME = "assets/audio/battle_theme.mp3";  

00018         } // namespace Audio  

00019         namespace Fonts  

00020         {  

00021             inline constexpr auto FONTS_RTYPE = "assets/fonts/r-type.otf";  

00022         } // namespace Fonts  

00023     } // namespace Paths  

00024     namespace Config  

00025     {  

00026         namespace Window  

00027         {  

00028             inline constexpr auto DEFAULT_WINDOW_WIDTH = 960;  

00029             inline constexpr auto DEFAULT_WINDOW_HEIGHT = 540;  

00030             inline constexpr auto DEFAULT_WINDOW_FRAME_LIMIT = 240;  

00031             inline constexpr auto DEFAULT_WINDOW_FULLSCREEN = false;  

00032         } // namespace Window  

00033         namespace Audio  

00034         {  

00035             inline constexpr auto DEFAULT_AUDIO_VOLUME = 50; // unused  

00036             inline constexpr auto DEFAULT_AUDIO_MUTED = false; // unused  

00037         } // namespace Audio  

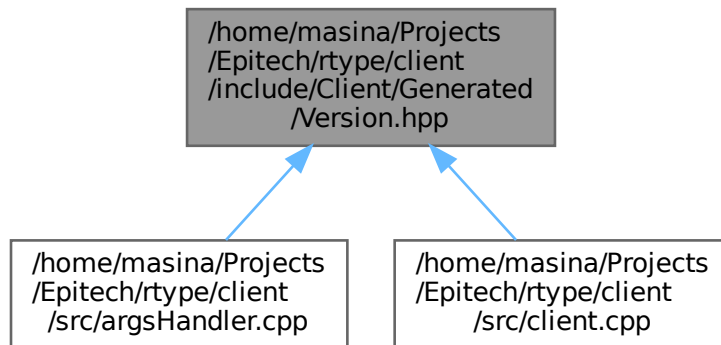
00038     } // namespace Config  

00039 } // namespace cli

```

8.9 /home/masina/Projects/Epitech/rtype/client/include/Client/Generated/Version.hpp File Reference

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



Macros

- `#define PROJECT_NAME "r-type_client"`
- `#define PROJECT_VERSION "0.0.0"`
- `#define PROJECT_VERSION_MAJOR "0"`
- `#define PROJECT_VERSION_MINOR "0"`
- `#define PROJECT_VERSION_PATCH "0"`
- `#define GIT_COMMIT_HASH "7e484bf"`
- `#define GIT_TAG "7e484bf"`
- `#define BUILD_TYPE "Release"`

8.9.1 Macro Definition Documentation

8.9.1.1 BUILD_TYPE

```
#define BUILD_TYPE "Release"
```

Definition at line 15 of file [Version.hpp](#).

Referenced by [cli::Client::Client\(\)](#), and [srv::Server::Server\(\)](#).

8.9.1.2 GIT_COMMIT_HASH

```
#define GIT_COMMIT_HASH "7e484bf"
```

Definition at line 13 of file [Version.hpp](#).

Referenced by [cli::Client::Client\(\)](#), and [srv::Server::Server\(\)](#).

8.9.1.3 GIT_TAG

```
#define GIT_TAG "7e484bf"
```

Definition at line 14 of file [Version.hpp](#).

Referenced by [cli::Client::Client\(\)](#), and [srv::Server::Server\(\)](#).

8.9.1.4 PROJECT_NAME

```
#define PROJECT_NAME "r-type_client"
```

Definition at line 7 of file [Version.hpp](#).

Referenced by [cli::Client::Client\(\)](#), and [srv::Server::Server\(\)](#).

8.9.1.5 PROJECT_VERSION

```
#define PROJECT_VERSION "0.0.0"
```

Definition at line 8 of file [Version.hpp](#).

Referenced by [cli::Client::Client\(\)](#), and [srv::Server::Server\(\)](#).

8.9.1.6 PROJECT_VERSION_MAJOR

```
#define PROJECT_VERSION_MAJOR "0"
```

Definition at line 9 of file [Version.hpp](#).

8.9.1.7 PROJECT_VERSION_MINOR

```
#define PROJECT_VERSION_MINOR "0"
```

Definition at line 10 of file [Version.hpp](#).

8.9.1.8 PROJECT_VERSION_PATCH

```
#define PROJECT_VERSION_PATCH "0"
```

Definition at line 11 of file [Version.hpp](#).

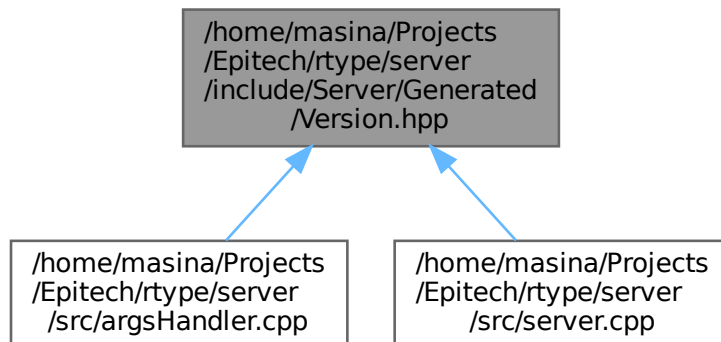
8.10 Version.hpp

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 //
=====
00004 // DO NOT EDIT THIS FILE MANUALLY. IT IS GENERATED BY CMAKE DURING THE BUILD PROCESS.
00005 //
=====
00006
00007 #define PROJECT_NAME "r-type_client"
00008 #define PROJECT_VERSION "0.0.0"
00009 #define PROJECT_VERSION_MAJOR "0"
00010 #define PROJECT_VERSION_MINOR "0"
00011 #define PROJECT_VERSION_PATCH "0"
00012
00013 #define GIT_COMMIT_HASH "7e484bf"
00014 #define GIT_TAG "7e484bf"
00015 #define BUILD_TYPE "Release"
```

8.11 /home/masina/Projects/Epitech/rtype/server/include/Server/↵ Generated/Version.hpp File Reference

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



Macros

- #define `PROJECT_NAME` "r-type_server"
- #define `PROJECT_VERSION` "0.0.0"
- #define `PROJECT_VERSION_MAJOR` "0"
- #define `PROJECT_VERSION_MINOR` "0"
- #define `PROJECT_VERSION_PATCH` "0"
- #define `GIT_COMMIT_HASH` "7e484bf"
- #define `GIT_TAG` "7e484bf"
- #define `BUILD_TYPE` "Release"

8.11.1 Macro Definition Documentation

8.11.1.1 BUILD_TYPE

```
#define BUILD_TYPE "Release"
```

Definition at line 15 of file [Version.hpp](#).

8.11.1.2 GIT_COMMIT_HASH

```
#define GIT_COMMIT_HASH "7e484bf"
```

Definition at line 13 of file [Version.hpp](#).

8.11.1.3 GIT_TAG

```
#define GIT_TAG "7e484bf"
```

Definition at line 14 of file [Version.hpp](#).

8.11.1.4 PROJECT_NAME

```
#define PROJECT_NAME "r-type_server"
```

Definition at line 7 of file [Version.hpp](#).

8.11.1.5 PROJECT_VERSION

```
#define PROJECT_VERSION "0.0.0"
```

Definition at line 8 of file [Version.hpp](#).

8.11.1.6 PROJECT_VERSION_MAJOR

```
#define PROJECT_VERSION_MAJOR "0"
```

Definition at line 9 of file [Version.hpp](#).

8.11.1.7 PROJECT_VERSION_MINOR

```
#define PROJECT_VERSION_MINOR "0"
```

Definition at line 10 of file [Version.hpp](#).

8.11.1.8 PROJECT_VERSION_PATCH

```
#define PROJECT_VERSION_PATCH "0"
```

Definition at line 11 of file [Version.hpp](#).

8.12 Version.hpp

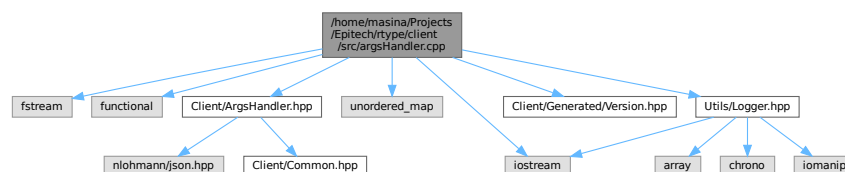
[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 //
=====
00004 // DO NOT EDIT THIS FILE MANUALLY. IT IS GENERATED BY CMAKE DURING THE BUILD PROCESS.
00005 //
=====
00006
00007 #define PROJECT_NAME "r-type_server"
00008 #define PROJECT_VERSION "0.0.0"
00009 #define PROJECT_VERSION_MAJOR "0"
00010 #define PROJECT_VERSION_MINOR "0"
00011 #define PROJECT_VERSION_PATCH "0"
00012
00013 #define GIT_COMMIT_HASH "7e484bf"
00014 #define GIT_TAG "7e484bf"
00015 #define BUILD_TYPE "Release"
```

8.13 /home/masina/Projects/Epitech/rtype/client/src/argsHandler.cpp

File Reference

```
#include <fstream>
#include <functional>
#include <iostream>
#include <unordered_map>
#include "Client/ArgsHandler.hpp"
#include "Client/Generated/Version.hpp"
#include "Utils/Logger.hpp"
Include dependency graph for argsHandler.cpp:
```



Macros

- `#define APP_EXTENSION ""`

Variables

- static constexpr std::string_view [HELP_MESSAGE](#)
- static constexpr std::string_view [VERSION_MESSAGE](#)

8.13.1 Macro Definition Documentation

8.13.1.1 APP_EXTENSION

```
#define APP_EXTENSION ""
```

Definition at line 9 of file [argsHandler.cpp](#).

8.13.2 Variable Documentation

8.13.2.1 HELP_MESSAGE

```
std::string_view HELP_MESSAGE [static], [constexpr]
```

Initial value:

```
= "Usage: " PROJECT_NAME APP_EXTENSION " [options]\n\n"
    "Options:\n"
    "\t--help, -h      Show this help message\n"
    "\t--version, -v     Show version information\n"
    "\t--config, -c      Specify path to config file\n"
```

Definition at line 16 of file [argsHandler.cpp](#).

Referenced by [cli::ArgsHandler::ParseArgs\(\)](#), and [srv::ArgsHandler::ParseArgs\(\)](#).

8.13.2.2 VERSION_MESSAGE

```
std::string_view VERSION_MESSAGE [static], [constexpr]
```

Initial value:

```
= PROJECT_NAME " version " PROJECT_VERSION "\n"
    "Build type: " BUILD_TYPE "\n"
    "Git tag: " GIT_TAG "\n"
    "Git commit hash: " GIT_COMMIT_HASH "\n"
```

Definition at line 21 of file [argsHandler.cpp](#).

Referenced by [cli::ArgsHandler::ParseArgs\(\)](#), and [srv::ArgsHandler::ParseArgs\(\)](#).

8.14 argsHandler.cpp

[Go to the documentation of this file.](#)

```
00001 #include <fstream>
00002 #include <functional>
00003 #include <iostream>
00004 #include <unordered_map>
00005
00006 #ifdef _WIN32
00007 #define APP_EXTENSION ".exe"
00008 #else
00009 #define APP_EXTENSION ""
00010 #endif
00011
00012 #include "Client/ArgsHandler.hpp"
00013 #include "Client/Generated/Version.hpp"
00014 #include "Utils/Logger.hpp"
00015
00016 static constexpr std::string_view HELP_MESSAGE = "Usage: " PROJECT_NAME APP_EXTENSION " [options]\n\n"
00017     "Options:\n"
00018     "\t--help, -h      Show this help message\n"
00019     "\t--version, -v     Show version information\n"
```

```

00020                                     "\t--config, -c    Specify path to config file\n";
00021 static constexpr std::string_view VERSION_MESSAGE = PROJECT_NAME " version " PROJECT_VERSION "\n"
00022                                     "Build type: " BUILD_TYPE "\n"
00023                                     "Git tag: " GIT_TAG "\n"
00024                                     "Git commit hash: " GIT_COMMIT_HASH "\n";
00025
00026 cli::ArgsConfig cli::ArgsConfig::fromFile(const std::string &path)
00027 {
00028     ArgsConfig cfg;
00029     std::ifstream file(path);
00030     if (!file.is_open())
00031     {
00032         throw std::runtime_error("Cannot open config file: " + path);
00033     }
00034
00035     json j;
00036     file » j;
00037
00038     if (j.contains("window"))
00039     {
00040         const auto &w = j["window"];
00041         if (w.contains("width"))
00042         {
00043             cfg.width = w["width"];
00044         }
00045         if (w.contains("height"))
00046         {
00047             cfg.height = w["height"];
00048         }
00049         if (w.contains("frame_limit"))
00050         {
00051             cfg.frameLimit = w["frame_limit"];
00052         }
00053         if (w.contains("fullscreen"))
00054         {
00055             cfg.fullscreen = w["fullscreen"];
00056         }
00057     }
00058     return cfg;
00059 }
00060
00061 cli::ArgsConfig cli::ArgsHandler::ParseArgs(const int argc, const char *const argv[])
00062 {
00063     if (argc <= 1)
00064         return {};
00065
00066     using ArgHandler = std::function<void(const char *arg)>;
00067     std::unordered_map<std::string_view, ArgHandler> handlers;
00068     ArgsConfig config{};
00069     for (const auto *const opt : {"-h", "--help"})
00070     {
00071         handlers[opt] = [&config](const char *)
00072         {
00073             std::cout « HELP_MESSAGE;
00074             config.exit = true;
00075         };
00076     }
00077     for (const auto *const opt : {"-v", "--version"})
00078     {
00079         handlers[opt] = [&config](const char *)
00080         {
00081             std::cout « VERSION_MESSAGE;
00082             config.exit = true;
00083         };
00084     }
00085     for (const auto *const opt : {"-c", "--config"})
00086     {
00087         handlers[opt] = [&config](const char *arg)
00088         {
00089             if (!arg)
00090             {
00091                 throw std::runtime_error("Missing config file argument");
00092             }
00093             config = ArgsConfig::fromFile(arg);
00094             utl::Logger::log("Loaded config from file: " + std::string(arg), utl::LogLevel::INFO);
00095             std::cout « "\tWidth: " « config.width « '\n'
00096                     « "\tHeight: " « config.height « '\n'
00097                     « "\tFrameLimit: " « config.frameLimit « "\n"
00098                     « "\tFullscreen: " « (config.fullscreen ? "true" : "false") « '\n';
00099         };
00100     }
00101 }
00102
00103 const std::string_view key = argv[1];
00104 const char *argValue = (argc > 2) ? argv[2] : nullptr;
00105
00106 if (const auto it = handlers.find(key); it != handlers.end())

```



```

00107 {
00108     it->second(argValue);
00109     return config;
00110 }
00111
00112 throw std::runtime_error("Unknown argument: " + std::string(key));
00113 }
00114
00115 cli::EnvConfig cli::ArgsHandler::ParseEnv(const char *const env[])
00116 {
00117     (void)env; // Currently unused
00118     return {};
00119 }

```

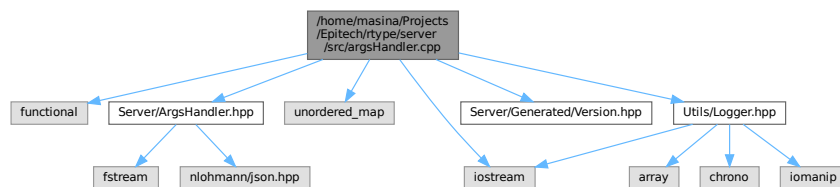
8.15 /home/masina/Projects/Epitech/rtype/server/src/argsHandler.cpp File Reference

```

#include <functional>
#include <iostream>
#include <unordered_map>
#include "Server/ArgsHandler.hpp"
#include "Server/Generated/Version.hpp"
#include "Utils/Logger.hpp"

```

Include dependency graph for argsHandler.cpp:



Macros

- #define `APP_EXTENSION` ""

Variables

- static constexpr std::string_view `HELP_MESSAGE`
- static constexpr std::string_view `VERSION_MESSAGE`

8.15.1 Macro Definition Documentation

8.15.1.1 APP_EXTENSION

```
#define APP_EXTENSION ""
```

Definition at line 9 of file `argsHandler.cpp`.

8.15.2 Variable Documentation

8.15.2.1 HELP_MESSAGE

std::string_view HELP_MESSAGE [static], [constexpr]

Initial value:

```
= "Usage: " PROJECT_NAME APP_EXTENSION " [options]\n\n"
    "Options:\n"
    "\t--help, -h      Show this help message\n"
    "\t--version, -v     Show version information\n"
    "\t--config, -c      Specify path to config file\n"
```

Definition at line 16 of file [argsHandler.cpp](#).

8.15.2.2 VERSION_MESSAGE

std::string_view VERSION_MESSAGE [static], [constexpr]

Initial value:

```
= PROJECT_NAME " version " PROJECT_VERSION "\n"
    "Build type: " BUILD_TYPE "\n"
    "Git tag: " GIT_TAG "\n"
    "Git commit hash: " GIT_COMMIT_HASH "\n"
```

Definition at line 21 of file [argsHandler.cpp](#).

8.16 argsHandler.cpp

[Go to the documentation of this file.](#)

```
00001 #include <functional>
00002 #include <iostream>
00003 #include <unordered_map>
00004
00005 #ifdef _WIN32
00006 #include <windows.h>
00007 #define APP_EXTENSION ".exe"
00008 #else
00009 #define APP_EXTENSION ""
00010 #endif
00011
00012 #include "Server/ArgsHandler.hpp"
00013 #include "Server/Generated/Version.hpp"
00014 #include "Utils/Logger.hpp"
00015
00016 static constexpr std::string_view HELP_MESSAGE = "Usage: " PROJECT_NAME APP_EXTENSION " [options]\n\n"
00017     "Options:\n"
00018     "\t--help, -h      Show this help message\n"
00019     "\t--version, -v     Show version information\n"
00020     "\t--config, -c      Specify path to config file\n";
00021 static constexpr std::string_view VERSION_MESSAGE = PROJECT_NAME " version " PROJECT_VERSION "\n"
00022     "Build type: " BUILD_TYPE "\n"
00023     "Git tag: " GIT_TAG "\n"
00024     "Git commit hash: " GIT_COMMIT_HASH "\n";
00025
00026 srv::ArgsConfig srv::ArgsConfig::fromFile(const std::string &path)
00027 {
00028     ArgsConfig cfg;
00029     std::ifstream file(path);
00030     if (!file.is_open())
00031     {
00032         throw std::runtime_error("Cannot open config file: " + path);
00033     }
00034
00035     json j;
00036     file » j;
00037
00038     if (j.contains("host"))
00039     {
00040         cfg.host = j["host"];
```

```

00041     }
00042     if (j.contains("port"))
00043     {
00044         cfg.port = j["port"];
00045     }
00046     return cfg;
00047 }
00048
00049 srv::ArgsConfig srv::ArgsHandler::ParseArgs(const int argc, const char *const argv[])
00050 {
00051     if (argc <= 1)
00052     {
00053         return {};
00054     }
00055
00056     using ArgHandler = std::function<void(const char *arg)>;
00057     std::unordered_map<std::string_view, ArgHandler> handlers;
00058     ArgsConfig config{};
00059     for (const auto *const opt : {"-h", "--help"})
00060     {
00061         handlers[opt] = [&config](const char *)
00062         {
00063             std::cout << HELP_MESSAGE;
00064             config.exit = true;
00065         };
00066     }
00067     for (const auto *const opt : {"-v", "--version"})
00068     {
00069         handlers[opt] = [&config](const char *)
00070         {
00071             std::cout << VERSION_MESSAGE;
00072             config.exit = true;
00073         };
00074     }
00075     for (const auto *const opt : {"-c", "--config"})
00076     {
00077         handlers[opt] = [&config](const char *arg)
00078         {
00079             if (!arg)
00080             {
00081                 throw std::runtime_error("Missing config file argument");
00082             }
00083             config = ArgsConfig::fromFile(arg);
00084             utl::Logger::log("Loaded config from file: " + std::string(arg), utl::LogLevel::INFO);
00085             std::cout << "\tHost: " << config.host << '\n' << "\tPort: " << config.port << '\n';
00086             std::cout << "\tHost: " << config.host << '\n' << "\tPort: " << config.port << '\n';
00087         };
00088     }
00089
00090     const std::string_view key = argv[1];
00091     const char *argValue = (argc > 2) ? argv[2] : nullptr;
00092
00093     if (const auto it = handlers.find(key); it != handlers.end())
00094     {
00095         it->second(argValue);
00096         return config;
00097     }
00098
00099     throw std::runtime_error("Unknown argument: " + std::string(key));
00100 }
00101
00102 srv::EnvConfig srv::ArgsHandler::ParseEnv(const char *const env[]) { return {}; }

```

8.17 /home/masina/Projects/Epitech/rtype/client/src/client.cpp File Reference

```

#include "Client/Client.hpp"
#include "Client/Common.hpp"
#include "Client/Generated/Version.hpp"
#include "ECS/Component.hpp"
#include "SFMLAudio/SFMLAudio.hpp"
#include "SFMLRenderer/SFMLRenderer.hpp"
#include "Utils/Clock.hpp"
#include "Utils/Logger.hpp"

```



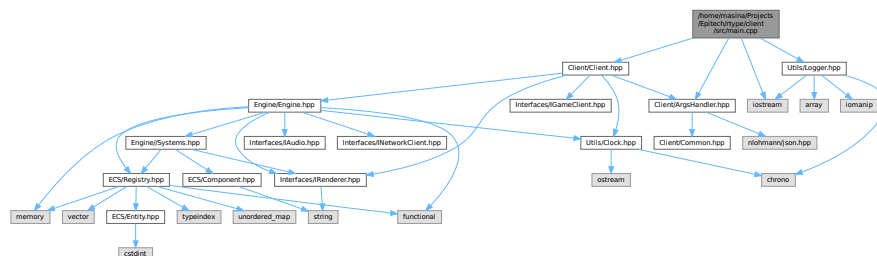
```

00062     }
00063     });
00064
00065     const auto titleEntity = m_engine->getRegistry()->createEntity();
00066     m_engine->getRegistry()->addComponent<ecs::Transform>(titleEntity, "entity_" + std::to_string(titleEntity), 10.F,
00067     10.F, 0.F);
00068     m_engine->getRegistry()->addComponent<ecs::Color>(titleEntity, "entity_" + std::to_string(titleEntity), 255, 255,
00069     255, 255);
00070     m_engine->getRegistry()->addComponent<ecs::Text>(titleEntity, "entity_" + std::to_string(titleEntity),
00071     std::string("RType Client"), 50);
00072
00073     const auto fpsEntity = m_engine->getRegistry()->createEntity();
00074     m_engine->getRegistry()->addComponent<ecs::Transform>(fpsEntity, "entity_" + std::to_string(fpsEntity), 10.F,
70.F,
00075     0.F);
00076     m_engine->getRegistry()->addComponent<ecs::Color>(fpsEntity, "entity_" + std::to_string(fpsEntity), 255, 255, 255,
00077     255);
00078     m_engine->getRegistry()->addComponent<ecs::Text>(fpsEntity, "entity_" + std::to_string(fpsEntity),
00079     std::string("FPS 0"), 20);
00080
00081     eng::Event event;
00082
00083     while (m_engine->getRenderrer()->windowIsOpen())
00084     {
00085         const float dt = m_engine->getClock()->getDeltaSeconds();
00086         m_engine->getClock()->restart();
00087
00088         while (m_engine->getRenderrer()->pollEvent(event))
00089         {
00090             if (event.type == eng::EventType::Closed ||
00091                 (event.type == eng::EventType::KeyPressed && event.key == eng::Key::Escape))
00092             {
00093                 m_engine->getRenderrer()->closeWindow();
00094             }
00095         }
00096         if (auto *fpsText = m_engine->getRegistry()->getComponent<ecs::Text>(fpsEntity))
00097         {
00098             fpsText->content = "FPS " + std::to_string(static_cast<int>(1.F / dt));
00099         }
00100
00101         m_engine->getRenderrer()->clearWindow({.r = 0, .g = 0, .b = 0, .a = 255});
00102         m_engine->updateSystems(dt);
00103         m_engine->getRenderrer()->displayWindow();
00104     }
00105 }

```

8.19 /home/masina/Projects/Epitech/rtype/client/src/main.cpp File Reference

```
#include <iostream>
#include "Client/ArgsHandler.hpp"
#include "Client/Client.hpp"
#include "Utils/Logger.hpp"
Include dependency graph for main.cpp:
```



Functions

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

8.19.1 Function Documentation

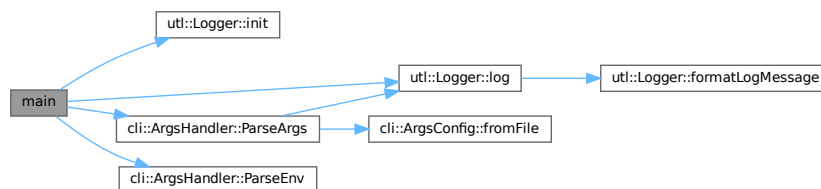
8.19.1.1 main()

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

Definition at line 7 of file [main.cpp](#).

References [cli::ArgsConfig::exit](#), [utl::Logger::init\(\)](#), [utl::Logger::log\(\)](#), [cli::ArgsHandler::ParseArgs\(\)](#), [cli::ArgsHandler::ParseEnv\(\)](#), and [utl::WARNING](#).

Here is the call graph for this function:



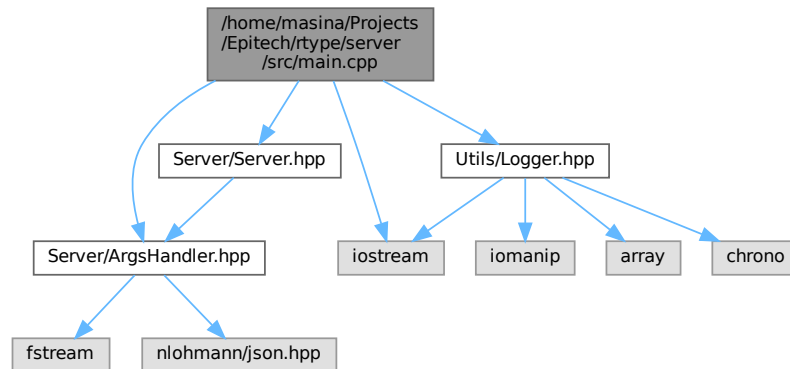
8.20 main.cpp

[Go to the documentation of this file.](#)

```
00001 #include <iostream>
00002
00003 #include "Client/ArgsHandler.hpp"
00004 #include "Client/Client.hpp"
00005 #include "Utils/Logger.hpp"
00006
00007 int main(const int argc, const char *const argv[], const char *const env[])
00008 {
00009     utl::Logger::init();
00010
00011     try
00012     {
00013         const cli::ArgsConfig argsConf = cli::ArgsHandler::ParseArgs(argc, argv);
00014         const cli::EnvConfig envConf = cli::ArgsHandler::ParseEnv(env);
00015         if (argsConf.exit)
00016         {
00017             return EXIT_SUCCESS;
00018         }
00019         cli::Client client(argsConf);
00020     }
00021     catch (const std::exception &e)
00022     {
00023         utl::Logger::log(std::string("Exception: ") + e.what(), utl::LogLevel::WARNING);
00024         return EXIT_FAILURE;
00025     }
00026     catch (...)
00027     {
00028         utl::Logger::log("Unknown exception", utl::LogLevel::WARNING);
00029         return EXIT_FAILURE;
00030     }
00031     return EXIT_SUCCESS;
00032 }
```

8.21 /home/masina/Projects/Epitech/rtype/server/src/main.cpp File Reference

```
#include <iostream>
#include "Server/ArgsHandler.hpp"
#include "Server/Server.hpp"
#include "Utils/Logger.hpp"
Include dependency graph for main.cpp:
```



Functions

- int [main](#) (const int argc, const char *const argv[], const char *const env[])

8.21.1 Function Documentation

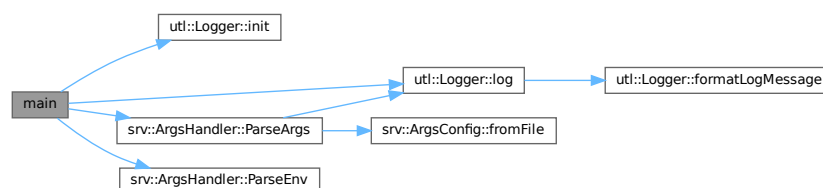
8.21.1.1 main()

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

Definition at line 7 of file [main.cpp](#).

References [srv::ArgsConfig::exit](#), [utl::Logger::init\(\)](#), [utl::Logger::log\(\)](#), [srv::ArgsHandler::ParseArgs\(\)](#), [srv::ArgsHandler::ParseEnv\(\)](#), and [utl::WARNING](#).

Here is the call graph for this function:



8.22 main.cpp

[Go to the documentation of this file.](#)

```

00001 #include <iostream>
00002
00003 #include "Server/ArgsHandler.hpp"
00004 #include "Server/Server.hpp"
00005 #include "Utils/Logger.hpp"
00006
00007 int main(const int argc, const char *const argv[], const char *const env[])
00008 {
00009     utl::Logger::init();
00010     try
00011     {
00012         const srv::ArgsConfig argsConf = srv::ArgsHandler::ParseArgs(argc, argv);
00013         const srv::EnvConfig envConf = srv::ArgsHandler::ParseEnv(env);
00014         if (argsConf.exit)
00015         {
00016             return EXIT_SUCCESS;
00017         }
00018         srv::Server server(argsConf);
00019     }
00020     catch (const std::exception &e)
00021     {
00022         utl::Logger::log(std::string("Exception: ") + e.what(), utl::LogLevel::WARNING);
00023         return EXIT_FAILURE;
00024     }
00025     catch (...)
00026     {
00027         utl::Logger::log("Unknown exception", utl::LogLevel::WARNING);
00028         return EXIT_FAILURE;
00029     }
00030     return EXIT_SUCCESS;
00031 }

```

8.23 /home/masina/Projects/Epitech/rtype/modules/Audio/SFMLAudio/include/SFMLAudio/SFMLAudio.hpp File Reference

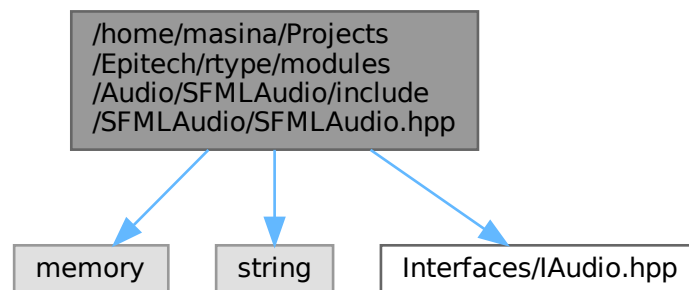
SFMLAudio class declaration.

```

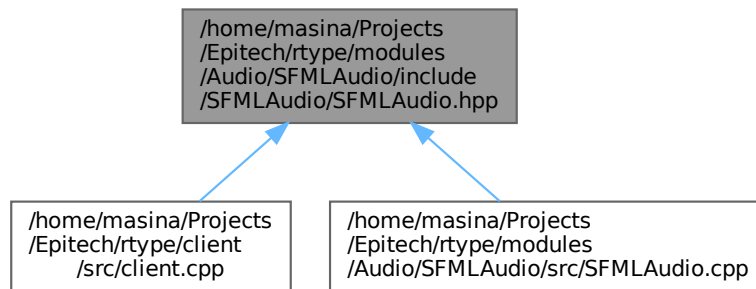
#include <memory>
#include <string>
#include "Interfaces/IAudio.hpp"

```

Include dependency graph for SFMLAudio.hpp:



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



Classes

- class [eng::SFMLAudio](#)
Class for audio management.

Namespaces

- namespace [eng](#)

8.23.1 Detailed Description

SFMLAudio class declaration.

Definition in file [SFMLAudio.hpp](#).

8.24 SFMLAudio.hpp

[Go to the documentation of this file.](#)

```

00001 ///  

00002 ///  

00003 ///  

00004 ///  

00005 ///  

00006 ///  

00007 #pragma once  

00008  

00009 #include <memory>  

00010 #include <string>  

00011  

00012 #include "Interfaces/IAudio.hpp"  

00013  

00014 namespace eng  

00015 {  

00016  

00017     ///  

00018     ///  

00019     ///  

00020     ///  

00021     ///  

00022     class SFMLAudio final : public IAudio  

00023     {
  
```

```

00024     public:
00025         SFMLAudio();
00026         ~SFMLAudio() override;
00027
00028         SFMLAudio(const SFMLAudio &) = delete;
00029         SFMLAudio &operator=(const SFMLAudio &) = delete;
00030         SFMLAudio(SFMLAudio &&) = delete;
00031         SFMLAudio &operator=(SFMLAudio &&) = delete;
00032
00033         void createAudio(const std::string &path, float volume, bool loop, const std::string &name) override;
00034         void playAudio(const std::string &name) override;
00035         void setVolume(const std::string &name, float volume) override;
00036         void setLoop(const std::string &name, bool loop) override;
00037
00038     private:
00039         struct Impl;
00040         std::unique_ptr<Impl> pImpl;
00041     }; // class SFMLAudio
00042
00043 } // namespace eng

```

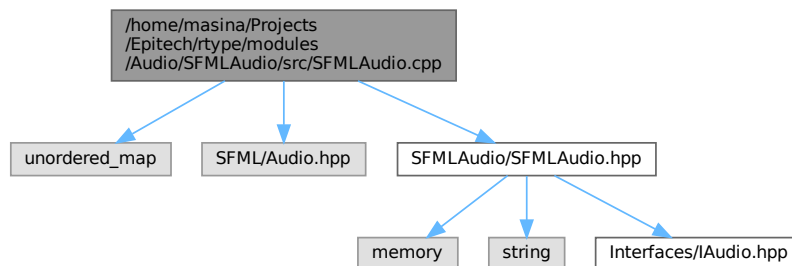
8.25 /home/masina/Projects/Epitech/rtype/modules/Audio/↵ SFMLAudio/src/SFMLAudio.cpp File Reference

```

#include <unordered_map>
#include <SFML/Audio.hpp>
#include "SFMLAudio/SFMLAudio.hpp"

```

Include dependency graph for SFMLAudio.cpp:



Classes

- struct `eng::SFMLAudio::Impl`

Namespaces

- namespace `eng`

8.26 SFMLAudio.cpp

[Go to the documentation of this file.](#)

```

00001 #include <unordered_map>
00002
00003 #include <SFML/Audio.hpp>
00004

```

```

00005 #include "SFMLAudio/SFMLAudio.hpp"
00006
00007 namespace eng
00008 {
00009     struct SFMLAudio::Impl
00010     {
00011         std::unordered_map<std::string, std::unique_ptr<sf::Music>» musics;
00012     };
00013
00014     SFMLAudio::SFMLAudio() : pImpl(std::make_unique<Impl>()) {}
00015     SFMLAudio::~SFMLAudio() = default;
00016
00017     void SFMLAudio::createAudio(const std::string &path, float volume, bool loop, const std::string &name)
00018     {
00019         auto music = std::make_unique<sf::Music>();
00020         if (!music->openFromFile(path))
00021         {
00022             return;
00023         }
00024
00025         music->setVolume(volume);
00026         music->setLooping(loop);
00027         pImpl->musics[name] = std::move(music);
00028     }
00029
00030     void SFMLAudio::playAudio(const std::string &name)
00031     {
00032         auto it = pImpl->musics.find(name);
00033         if (it != pImpl->musics.end())
00034             it->second->play();
00035     }
00036
00037     void SFMLAudio::setVolume(const std::string &name, float volume)
00038     {
00039         auto it = pImpl->musics.find(name);
00040         if (it != pImpl->musics.end())
00041             it->second->setVolume(volume);
00042     }
00043
00044     void SFMLAudio::setLoop(const std::string &name, bool loop)
00045     {
00046         auto it = pImpl->musics.find(name);
00047         if (it != pImpl->musics.end())
00048             it->second->setLooping(loop);
00049     }
00050 } // namespace eng

```

8.27 /home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.hpp File Reference ↩

This file contains the component definitions.

#include <string>

Include dependency graph for Component.hpp:

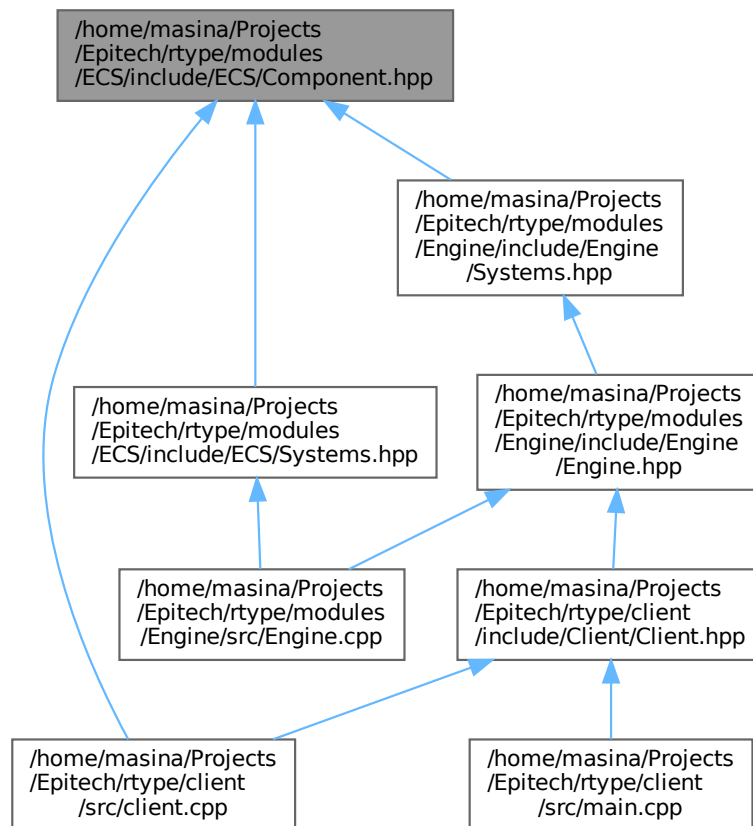
```

/home/masina/Projects
/Epitech/rtype/modules
/ECS/include/ECS/Component.hpp

```

string

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



Classes

- struct [ecs::Audio](#)
- struct [ecs::Color](#)
- struct [ecs::Font](#)
- struct [ecs::Sprite](#)
- struct [ecs::Text](#)
- struct [ecs::Transform](#)
- struct [ecs::Velocity](#)

Namespaces

- namespace [ecs](#)

8.27.1 Detailed Description

This file contains the component definitions.

Definition in file [Component.hpp](#).

8.28 Component.hpp

[Go to the documentation of this file.](#)

```

00001 ///  

00002 ///  

00003 ///  

00004 ///  

00005 ///  

00006 ///  

00007 #pragma once  

00008  

00009 #include <string>  

00010  

00011 namespace ecs  

00012 {  

00013     struct Audio  

00014     {  

00015         std::string id;  

00016         std::string path;  

00017         float volume;  

00018         bool loop;  

00019     };  

00020     struct Color  

00021     {  

00022         std::string id;  

00023         int r;  

00024         int g;  

00025         int b;  

00026         int a;  

00027     };  

00028     struct Font  

00029     {  

00030         std::string id;  

00031         std::string path;  

00032     };  

00033     struct Sprite  

00034     {  

00035         std::string id;  

00036         std::string path;  

00037     };  

00038     struct Text  

00039     {  

00040         std::string id;  

00041         std::string content;  

00042         int fontSize;  

00043     };  

00044     struct Transform  

00045     {  

00046         std::string id;  

00047         float x, y;  

00048         float rotation;  

00049     };  

00050     struct Velocity  

00051     {  

00052         std::string id;  

00053         float x, y;  

00054     };  

00055 } // namespace ecs

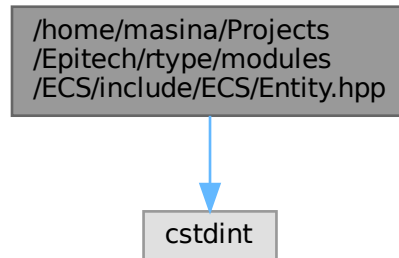
```

8.29 /home/masina/Projects/Epitech/rtype/modules/ECS/include/↵ ECS/Entity.hpp File Reference

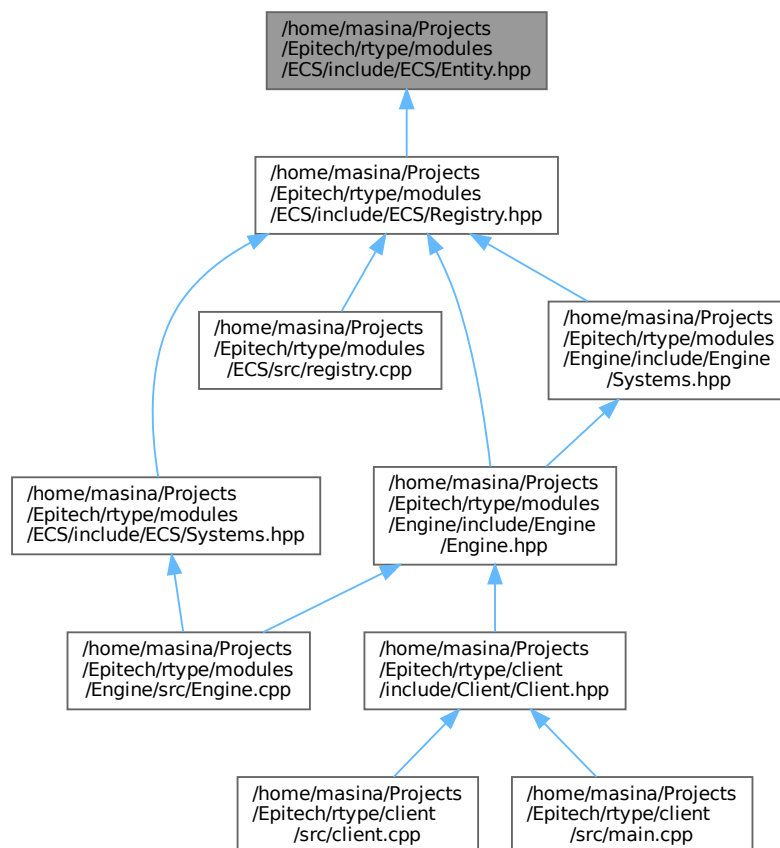
This file contains the entity definitions.

```
#include <cstdint>
```

Include dependency graph for Entity.hpp:



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



Namespaces

- namespace [ecs](#)

Typedefs

- using `ecs::Entity` = `std::uint32_t`

Variables

- constexpr `Entity` `ecs::INVALID_ENTITY` = 0

8.29.1 Detailed Description

This file contains the entity definitions.

Definition in file [Entity.hpp](#).

8.30 Entity.hpp

[Go to the documentation of this file.](#)

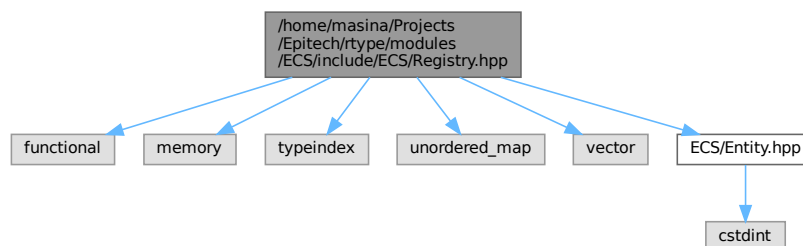
```
00001 ///  
00002 ///  
00003 ///  
00004 ///  
00005 ///  
00006 ///  
00007 #pragma once  
00008  
00009 #include <cstdint>  
00010  
00011 namespace ecs  
00012 {  
00013     using Entity = std::uint32_t;  
00014     constexpr Entity INVALID_ENTITY = 0;  
00015 } // namespace ecs
```

8.31 /home/masina/Projects/Epitech/rtype/modules/ECS/include/↵ ECS/Registry.hpp File Reference

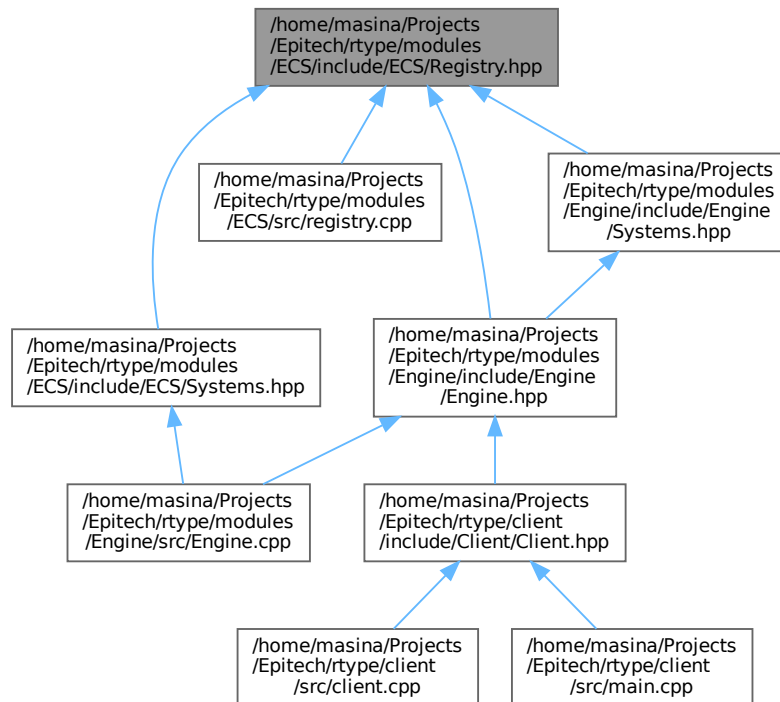
This file contains the Registry class declaration.

```
#include <functional>  
#include <memory>  
#include <typeindex>  
#include <unordered_map>  
#include <vector>  
#include "ECS/Entity.hpp"
```

Include dependency graph for Registry.hpp:



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



Classes

- class `ecs::Registry`
Class for managing entities and their components.
- class `ecs::Registry::IPool`
- class `ecs::Registry::Pool< T >`

Namespaces

- namespace `ecs`

8.31.1 Detailed Description

This file contains the Registry class declaration.

Definition in file [Registry.hpp](#).

8.32 Registry.hpp

[Go to the documentation of this file.](#)

```

00001 ///  

00002 ///  

00003 ///  

00004 ///  

00005 ///  

00006 ///  

00007 #pragma once  

00008  

00009 #include <functional>  

00010 #include <memory>  

00011 #include <typeindex>  

00012 #include <unordered_map>  

00013 #include <vector>  

00014  

00015 #include "ECS/Entity.hpp"  

00016  

00017 namespace ecs  

00018 {  

00019     ///  

00020     ///  

00021     ///  

00022     ///  

00023     ///  

00024     class Registry  

00025     {  

00026     public:  

00027         Registry() = default;  

00028         ~Registry() = default;  

00029  

00030         Registry(const Registry &) = delete;  

00031         Registry &operator=(const Registry &) = delete;  

00032         Registry(Registry &&) = delete;  

00033         Registry &operator=(Registry &&) = delete;  

00034  

00035         Entity createEntity()  

00036         {  

00037             const Entity entity = ++m_lastEntity;  

00038             m_entities.push_back(entity);  

00039             for (auto &cb : m_onEntityCreatedCallbacks)  

00040                 cb(entity);  

00041             return entity;  

00042         }  

00043  

00044         template <typename T, typename... Args> T &addComponent(Entity e, Args &&...args)  

00045         {  

00046             auto &pool = getPool<T>();  

00047             T &comp = pool.add(e, std::forward<Args>(args)...);  

00048             for (auto &cb : m_onComponentAddedCallbacks)  

00049                 cb(e, typeid(T));  

00050             return comp;  

00051         }  

00052  

00053         template <typename T> T *getComponent(Entity e)  

00054         {  

00055             auto &pool = getPool<T>();  

00056             return pool.get(e);  

00057         }  

00058  

00059         template <typename T> std::unordered_map<Entity, T> &getAll() { return getPool<T>().data; }  

00060  

00061         template <typename T> bool hasComponent(Entity e)  

00062         {  

00063             auto &pool = getPool<T>();  

00064             return pool.has(e);  

00065         }  

00066  

00067         template <typename T> void removeComponent(Entity e)  

00068         {  

00069             auto &pool = getPool<T>();  

00070             pool.remove(e);  

00071         }  

00072         void onEntityCreated(std::function<void(Entity)> cb)  

00073         {  

00074             m_onEntityCreatedCallbacks.push_back(std::move(cb));  

00075         }  

00076  

00077         void onComponentAdded(std::function<void(Entity, const std::type_info &)> cb)  

00078         {  

00079             m_onComponentAddedCallbacks.push_back(std::move(cb));  

00080         }  

00081  

00082     private:
```

```

00083     class IPool
00084     {
00085     public:
00086         virtual ~IPool() = default;
00087         virtual void remove(Entity e) = 0;
00088     };
00089
00090     template <typename T> class Pool final : public IPool
00091     {
00092     public:
00093         std::unordered_map<Entity, T> data;
00094
00095         template <typename... Args> T &add(Entity e, Args &&...args)
00096         {
00097             return data.emplace(e, T{std::forward<Args>(args)...}).first->second;
00098         }
00099
00100         T *get(Entity e)
00101         {
00102             auto it = data.find(e);
00103             if (it != data.end())
00104                 return &it->second;
00105             return nullptr;
00106         }
00107
00108         bool has(Entity e) { return data.find(e) != data.end(); }
00109
00110         void remove(Entity e) override { data.erase(e); }
00111     };
00112
00113     template <typename T> Pool<T> &getPool()
00114     {
00115         std::type_index ti(typeid(T));
00116         if (!m_components.contains(ti))
00117         {
00118             m_components[ti] = std::make_unique<Pool<T>>();
00119         }
00120         return *static_cast<Pool<T>*>(m_components[ti].get());
00121     }
00122     Entity m_lastEntity = INVALID_ENTITY;
00123     std::vector<Entity> m_entities;
00124     std::unordered_map<std::type_index, std::unique_ptr<IPool>» m_components;
00125     std::vector<std::function<void(Entity)>» m_onEntityCreatedCallbacks;
00126     std::vector<std::function<void(Entity, const std::type_info &)>» m_onComponentAddedCallbacks;
00127
00128 }; // class Registry
00129
00130 } // namespace ecs

```

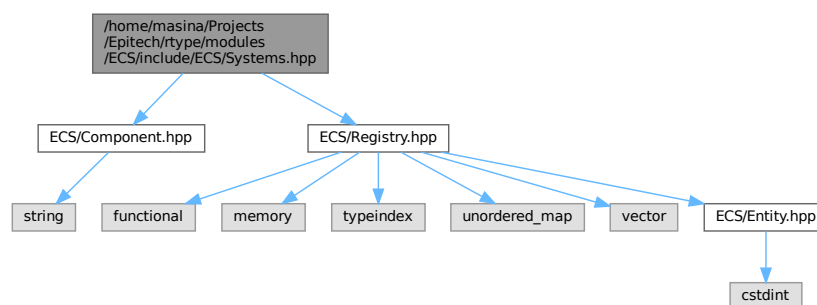
8.33 /home/masina/Projects/Epitech/rtype/modules/ECS/include/↵ ECS/Systems.hpp File Reference

This file contains the system definitions.

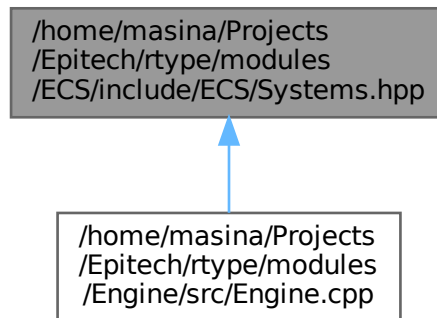
```
#include "ECS/Component.hpp"
```

```
#include "ECS/Registry.hpp"
```

Include dependency graph for Systems.hpp:



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



Classes

- class [ecs::TextSyStem](#)
Class for managing entities and their components.
- class [ecs::FontSystem](#)
Class for managing entities and their components.
- class [ecs::AudioSystem](#)
Class for managing entities and their components.

Namespaces

- namespace [ecs](#)

8.33.1 Detailed Description

This file contains the system definitions.

Definition in file [Systems.hpp](#).

8.34 Systems.hpp

[Go to the documentation of this file.](#)

```

00001 ///
00002 /// @file Systems.hpp
00003 /// @brief This file contains the system definitions
00004 /// @namespace ecs
00005 ///
00006
00007 #pragma once
00008
00009 #include "ECS/Component.hpp"
00010 #include "ECS/Registry.hpp"
00011
00012 namespace ecs
00013 {

```

```

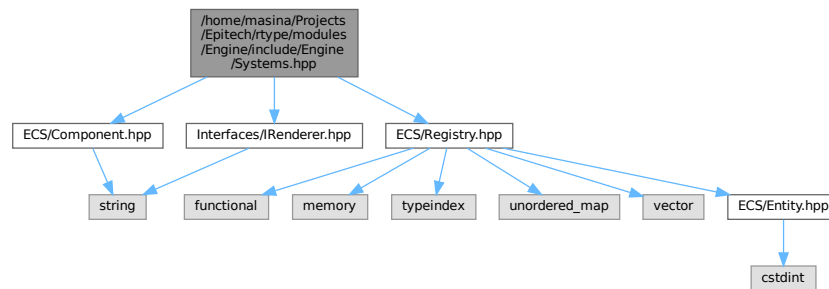
00014
00015 ///
00016 /// @class TextSyStem
00017 /// @brief Class for managing entities and their components
00018 /// @namespace ecs
00019 ///
00020 class TextSyStem
00021 {
00022     public:
00023         TextSyStem() = default;
00024         ~TextSyStem() = default;
00025
00026         TextSyStem(const TextSyStem &) = delete;
00027         TextSyStem &operator=(const TextSyStem &) = delete;
00028         TextSyStem(TextSyStem &&) = delete;
00029         TextSyStem &operator=(TextSyStem &&) = delete;
00030
00031         using DrawCallback = std::function<void(const Text &, const Transform &, const Color &)>;
00032
00033         void setDrawCallback(DrawCallback cb) { m_drawCallback = std::move(cb); }
00034
00035         void update(Registry &registry) const
00036         {
00037             for (auto &[entity, text] : registry.getAll<Text>())
00038             {
00039                 const auto *transform = registry.getComponent<Transform>(entity);
00040                 const auto *color = registry.getComponent<Color>(entity);
00041                 if (!transform)
00042                     continue;
00043
00044                 if (m_drawCallback)
00045                     m_drawCallback(text, *transform, *color);
00046             }
00047         }
00048
00049     private:
00050         DrawCallback m_drawCallback;
00051 }; // class TextRenderSystem
00052
00053 ///
00054 /// @class FontSystem
00055 /// @brief Class for managing entities and their components
00056 /// @namespace ecs
00057 ///
00058 class FontSystem
00059 {
00060     public:
00061         FontSystem() = default;
00062         ~FontSystem() = default;
00063
00064         FontSystem(const FontSystem &) = delete;
00065         FontSystem &operator=(const FontSystem &) = delete;
00066         FontSystem(FontSystem &&) = delete;
00067         FontSystem &operator=(FontSystem &&) = delete;
00068
00069         void update(Registry &registry) {}
00070
00071     private:
00072 }; // class FontSystem
00073
00074 ///
00075 /// @class AudioSystem
00076 /// @brief Class for managing entities and their components
00077 /// @namespace ecs
00078 ///
00079 class AudioSystem
00080 {
00081     public:
00082         AudioSystem() = default;
00083         ~AudioSystem() = default;
00084
00085         AudioSystem(const AudioSystem &) = delete;
00086         AudioSystem &operator=(const AudioSystem &) = delete;
00087         AudioSystem(AudioSystem &&) = delete;
00088         AudioSystem &operator=(AudioSystem &&) = delete;
00089
00090         void update(Registry &registry, float dt) {}
00091
00092     private:
00093 }; // class AudioSystem
00094
00095 } // namespace ecs

```

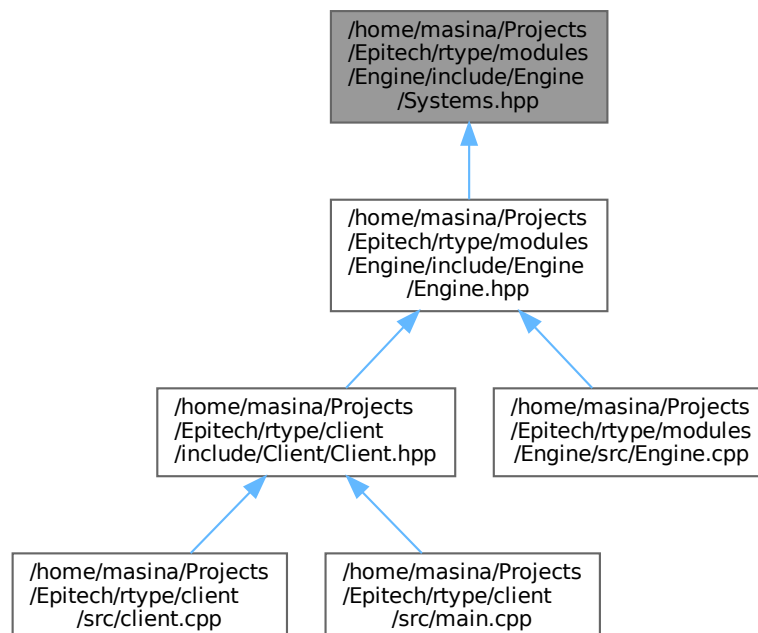
8.35 /home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/Systems.hpp File Reference

This file contains the system definitions.

```
#include "ECS/Component.hpp"
#include "ECS/Registry.hpp"
#include "Interfaces/IRenderer.hpp"
Include dependency graph for Systems.hpp:
```



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



Classes

- class [eng::ISystem](#)

- class [eng::TextSyStem](#)
Class for managing entities and their components.
- class [eng::FontSystem](#)
Class for managing entities and their components.
- class [eng::AudioSystem](#)
Class for managing entities and their components.

Namespaces

- namespace [eng](#)
- namespace [ecs](#)

8.35.1 Detailed Description

This file contains the system definitions.

Definition in file [Systems.hpp](#).

8.36 Systems.hpp

[Go to the documentation of this file.](#)

```
00001 ///  
00002 ///  
00003 ///  
00004 ///  
00005 ///  
00006 ///  
00007 #pragma once  
00008  
00009 #include "ECS/Component.hpp"  
00010 #include "ECS/Registry.hpp"  
00011 #include "Interfaces/IRenderer.hpp"  
00012  
00013 namespace eng  
00014 {  
00015  
00016     class ISystem  
00017     {  
00018     public:  
00019         virtual ~ISystem() = default;  
00020         virtual void update(ecs::Registry &registry, float dt) = 0;  
00021         virtual bool isEnabled() = 0;  
00022     };  
00023  
00024     ///  
00025     ///  
00026     ///  
00027     ///  
00028     ///  
00029     class TextSyStem final : public ISystem  
00030     {  
00031     public:  
00032         explicit TextSyStem(IRenderer &renderer) : m_renderer(renderer) {}  
00033         ~TextSyStem() override = default;  
00034  
00035         TextSyStem(const TextSyStem &) = delete;  
00036         TextSyStem &operator=(const TextSyStem &) = delete;  
00037         TextSyStem(TextSyStem &&) = delete;  
00038         TextSyStem &operator=(TextSyStem &&) = delete;  
00039  
00040         void update(ecs::Registry &registry, float dt) override  
00041         {  
00042             for (auto &[entity, text] : registry.getAll<ecs::Text>())  
00043             {  
00044                 const auto *transform = registry.getComponent<ecs::Transform>(entity);  
00045                 const auto *color = registry.getComponent<ecs::Color>(entity);  
00046                 if (!transform || !color)  
00047                     continue;
```

```

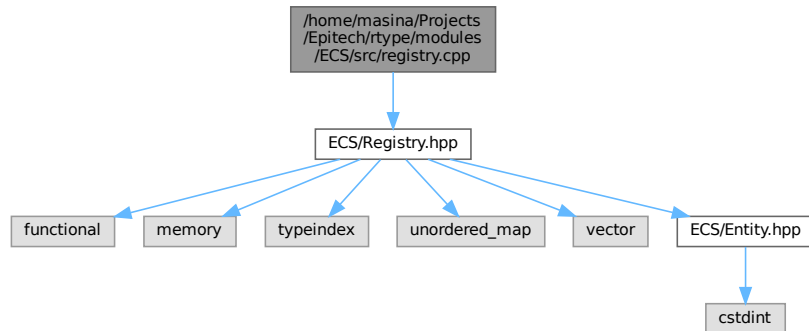
00048         m_renderer.setTextContent(text.id, text.content);
00049         m_renderer.setTextPosition(text.id, static_cast<int>(transform->x), static_cast<int>(transform->y));
00050         m_renderer.setTextColor(text.id, {r = static_cast<std::uint8_t>(color->r),
00051         .g = static_cast<std::uint8_t>(color->g),
00052         .b = static_cast<std::uint8_t>(color->b),
00053         .a = static_cast<std::uint8_t>(color->a)});
00054         m_renderer.drawText(text.id);
00055     }
00056 }
00057
00058     bool isEnabled() override { return m_isEnable; }
00059     void setEnable(const bool enable) { m_isEnable = enable; }
00060
00061     private:
00062         IRenderer &m_renderer;
00063         bool m_isEnable = true;
00064 }; // class TextRenderSystem
00065
00066 ///
00067 /// @class FontSystem
00068 /// @brief Class for managing entities and their components
00069 /// @namespace ecs
00070 ///
00071 class FontSystem final : public ISystem
00072 {
00073     public:
00074         explicit FontSystem(IRenderer &renderer) : m_renderer(renderer) {}
00075         ~FontSystem() override = default;
00076
00077         FontSystem(const FontSystem &) = delete;
00078         FontSystem &operator=(const FontSystem &) = delete;
00079         FontSystem(FontSystem &&) = delete;
00080         FontSystem &operator=(FontSystem &&) = delete;
00081
00082         void update(ecs::Registry &registry, const float dt) override {}
00083         bool isEnabled() override { return m_isEnable; }
00084         void setEnable(const bool enable) { m_isEnable = enable; }
00085
00086     private:
00087         IRenderer &m_renderer;
00088         bool m_isEnable = true;
00089 }; // class FontSystem
00090
00091 ///
00092 /// @class AudioSystem
00093 /// @brief Class for managing entities and their components
00094 /// @namespace ecs
00095 ///
00096 class AudioSystem final : public ISystem
00097 {
00098     public:
00099         explicit AudioSystem(IRenderer &renderer) : m_renderer(renderer) {}
00100         ~AudioSystem() override = default;
00101
00102         AudioSystem(const AudioSystem &) = delete;
00103         AudioSystem &operator=(const AudioSystem &) = delete;
00104         AudioSystem(AudioSystem &&) = delete;
00105         AudioSystem &operator=(AudioSystem &&) = delete;
00106
00107         void update(ecs::Registry &registry, float dt) override {}
00108         bool isEnabled() override { return m_isEnable; }
00109         void setEnable(const bool enable) { m_isEnable = enable; }
00110
00111     private:
00112         IRenderer &m_renderer;
00113         bool m_isEnable = true;
00114 }; // class AudioSystem
00115
00116 } // namespace eng

```

8.37 /home/masina/Projects/Epitech/rtype/modules/↵ ECS/src/registry.cpp File Reference

```
#include "ECS/Registry.hpp"
```

Include dependency graph for registry.cpp:



8.38 registry.cpp

[Go to the documentation of this file.](#)

```
00001 #include "ECS/Registry.hpp"
```

8.39 /home/masina/Projects/Epitech/rtype/modules/Engine/include/↵ Engine/Engine.hpp File Reference

This file contains the Engine class declaration.

```
#include <functional>
```

```
#include <memory>
```

```
#include "ECS/Registry.hpp"
```

```
#include "Engine//Systems.hpp"
```

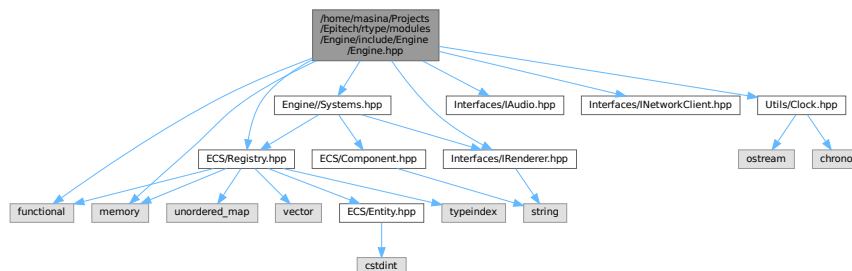
```
#include "Interfaces/IAudio.hpp"
```

```
#include "Interfaces/INetworkClient.hpp"
```

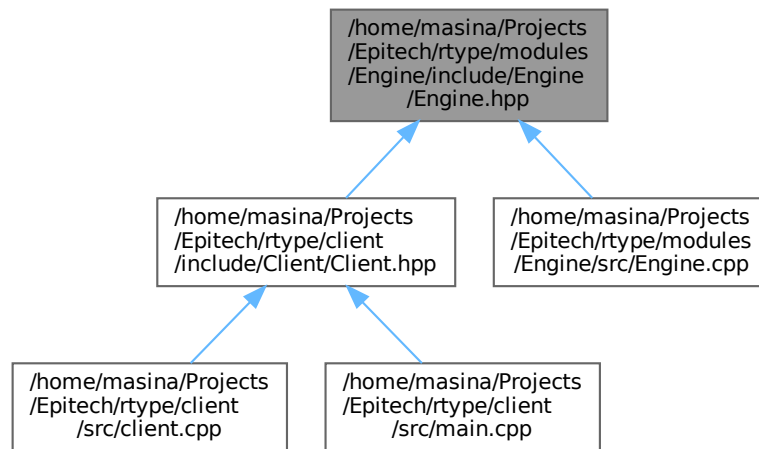
```
#include "Interfaces/IRenderer.hpp"
```

```
#include "Utils/Clock.hpp"
```

Include dependency graph for Engine.hpp:



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



Classes

- class `eng::Engine`
Class for the game engine.

Namespaces

- namespace `eng`

8.39.1 Detailed Description

This file contains the Engine class declaration.

Definition in file [Engine.hpp](#).

8.40 Engine.hpp

[Go to the documentation of this file.](#)

```

00001 ///
00002 /// @file Engine.hpp
00003 /// @brief This file contains the Engine class declaration
00004 /// @namespace eng
00005 ///
00006
00007 #pragma once
00008
00009 #include <functional>
00010 #include <memory>
00011
00012 #include "ECS/Registry.hpp"
00013 #include "Engine//Systems.hpp"
00014 #include "Interfaces/IAudio.hpp"
00015 #include "Interfaces/INetworkClient.hpp"
00016 #include "Interfaces/IRenderer.hpp"

```

```

00017 #include "Utils/Clock.hpp"
00018
00019 namespace eng
00020 {
00021
00022     ///
00023     /// @class Engine
00024     /// @brief Class for the game engine
00025     /// @namespace eng
00026     ///
00027     class Engine
00028     {
00029
00030     public:
00031         Engine(const std::function<std::unique_ptr<IAudio>()> &audioFactory,
00032               const std::function<std::unique_ptr<INetworkClient>()> &networkFactory,
00033               const std::function<std::unique_ptr<IRenderer>()> &rendererFactory);
00034         ~Engine() = default;
00035
00036         Engine(const Engine &) = delete;
00037         Engine &operator=(const Engine &) = delete;
00038         Engine(Engine &&) = delete;
00039         Engine &operator=(Engine &&) = delete;
00040
00041         std::unique_ptr<IAudio> &getAudio() { return m_audio; }
00042         std::unique_ptr<INetworkClient> &getNetworkClient() { return m_networkClient; }
00043         std::unique_ptr<IRenderer> &getRenderer() { return m_renderer; }
00044         std::unique_ptr<utl::Clock> &getClock() { return m_clock; }
00045         std::unique_ptr<ecs::Registry> &getRegistry() { return m_registry; } // to remove
00046
00047         void addSystem(std::unique_ptr<ISystem> system) { m_systems.emplace_back(std::move(system)); }
00048         void updateSystems(const float dt) const
00049         {
00050             for (auto &system : m_systems)
00051             {
00052                 system->update(*m_registry, dt);
00053             }
00054         }
00055
00056     private:
00057         std::unique_ptr<IAudio> m_audio;
00058         std::unique_ptr<INetworkClient> m_networkClient;
00059         std::unique_ptr<IRenderer> m_renderer;
00060
00061         std::unique_ptr<utl::Clock> m_clock;
00062         std::unique_ptr<ecs::Registry> m_registry; // to remove, maybe one registry per scene
00063         std::vector<std::unique_ptr<ISystem> > m_systems;
00064     }; // class Engine
00065
00066 } // namespace eng

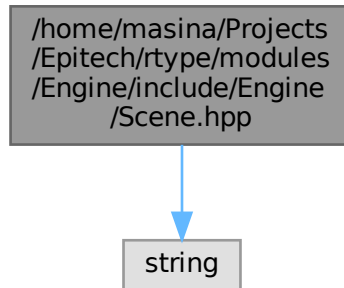
```

8.41 /home/masina/Projects/Epitech/rtype/modules/Engine/include/↵ Engine/Scene.hpp File Reference

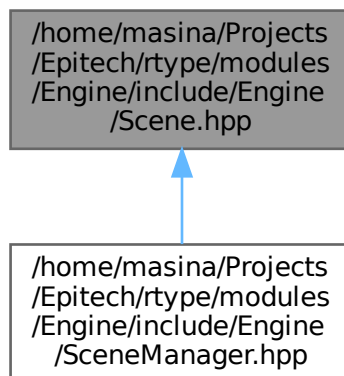
This file contains the Scene class.

```
#include <string>
```

Include dependency graph for Scene.hpp:



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



Classes

- class [eng::Scene](#)
class for scene and manage entities

Namespaces

- namespace [eng](#)

Typedefs

- using [eng::scene_id_t](#) = unsigned int

8.41.1 Detailed Description

This file contains the Scene class.

Definition in file [Scene.hpp](#).

8.42 Scene.hpp

[Go to the documentation of this file.](#)

```

00001 ///  

00002 ///  

00003 ///  

00004 ///  

00005 ///  

00006 ///  

00007 #pragma once  

00008  

00009 #include <string>  

00010  

00011 namespace eng  

00012 {  

00013  

00014     using scene_id_t = unsigned int;  

00015  

00016     ///  

00017     ///  

00018     ///  

00019     ///  

00020     ///  

00021     class Scene  

00022     {  

00023     public:  

00024         Scene() = default;  

00025         ~Scene() = default;  

00026  

00027         [[nodiscard]] std::string &getName() { return m_name; }  

00028         void setName(const std::string &newName) { m_name = newName; }  

00029  

00030         [[nodiscard]] bool getDisplay() const { return m_display; }  

00031         void setDisplay(const bool display) { m_display = display; }  

00032  

00033         [[nodiscard]] scene_id_t getId() const { return m_id; }  

00034  

00035     private:  

00036         std::string m_name = "default_name";  

00037         bool m_display = false;  

00038         scene_id_t m_id = 0;  

00039     }; // class Scene  

00040  

00041 } // namespace eng

```

8.43 /home/masina/Projects/Epitech/rtype/modules/Engine/include/↵ Engine/SceneManager.hpp File Reference

This file contains the SceneManager class declaration.

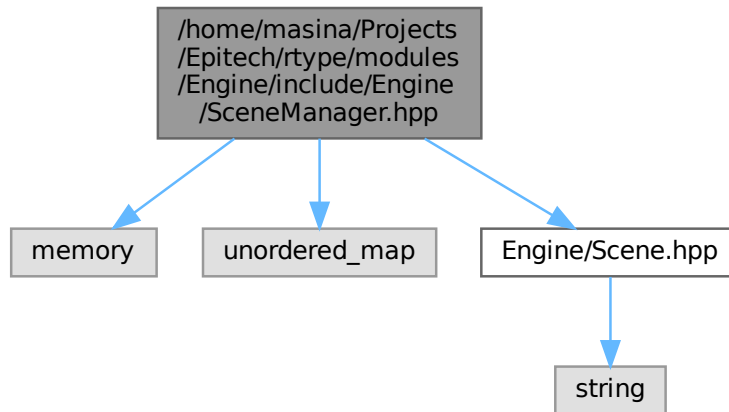
```

#include <memory>
#include <unordered_map>

```

```
#include "Engine/Scene.hpp"
```

Include dependency graph for SceneManager.hpp:



Classes

- class [eng::SceneManager](#)
Class for managing scenes.

Namespaces

- namespace [eng](#)

8.43.1 Detailed Description

This file contains the SceneManager class declaration.

Definition in file [SceneManager.hpp](#).

8.44 SceneManager.hpp

[Go to the documentation of this file.](#)

```

00001 ///  

00002 ///  

00003 ///  

00004 ///  

00005 ///  

00006 ///  

00007 #pragma once  

00008  

00009 #include <memory>  

00010 #include <unordered_map>  

00011  

00012 #include "Engine/Scene.hpp"  

00013  

00014 namespace eng  

00015 {

```

```

00016
00017 ///
00018 /// @class SceneManager
00019 /// @brief Class for managing scenes
00020 /// @namespace eng
00021 ///
00022 class SceneManager
00023 {
00024
00025     public:
00026         SceneManager() = default;
00027         ~SceneManager() = default;
00028
00029         SceneManager(const SceneManager &) = delete;
00030         SceneManager &operator=(const SceneManager &) = delete;
00031         SceneManager(SceneManager &&) = delete;
00032         SceneManager &operator=(SceneManager &&) = delete;
00033
00034         std::unique_ptr<Scene> &getScene(scene_id_t sceneId);
00035         std::unique_ptr<Scene> &getCurrentScene();
00036         scene_id_t addScene(std::unique_ptr<Scene> scene);
00037         void switchToScene(scene_id_t sceneId);
00038
00039     private:
00040         std::unordered_map<scene_id_t, std::unique_ptr<Scene>» m_scenes;
00041         scene_id_t m_currentSceneId = 0;
00042
00043 }; // class SceneManager
00044
00045 } // namespace eng

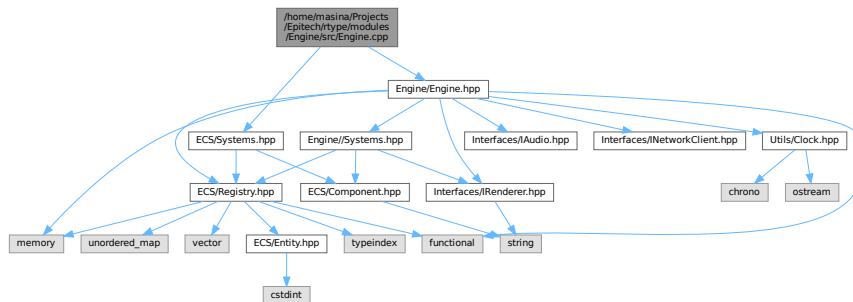
```

8.45 /home/masina/Projects/Epitech/rtype/modules/Engine/src/↵ Engine.cpp File Reference

#include "Engine/Engine.hpp"

#include "ECS/Systems.hpp"

Include dependency graph for Engine.cpp:



8.46 Engine.cpp

[Go to the documentation of this file.](#)

```

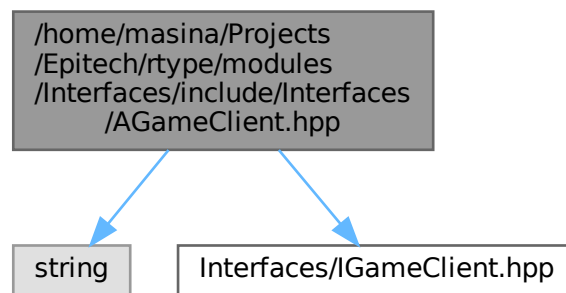
00001 #include "Engine/Engine.hpp"
00002 #include "ECS/Systems.hpp"
00003
00004 eng::Engine::Engine(const std::function<std::unique_ptr<IAudio>()> &audioFactory,
00005                     const std::function<std::unique_ptr<INetworkClient>()> &networkFactory,
00006                     const std::function<std::unique_ptr<IRenderer>()> &rendererFactory)
00007 : m_audio(audioFactory()), m_networkClient(networkFactory()), m_renderer(rendererFactory()),
00008   m_clock(std::make_unique<util::Clock>()), m_registry(std::make_unique<ecs::Registry>())
00009 {
00010 }

```

8.47 /home/masina/Projects/Epitech/rtype/modules/ Interfaces/include/Interfaces/AGameClient.hpp File Reference

This file contains the game abstract class.

```
#include <string>
#include "Interfaces/IGameClient.hpp"
Include dependency graph for AGameClient.hpp:
```



Classes

- class [cli::AGameClient](#)
Abstraction for the games.

Namespaces

- namespace [cli](#)

8.47.1 Detailed Description

This file contains the game abstract class.

Definition in file [AGameClient.hpp](#).

8.48 AGameClient.hpp

[Go to the documentation of this file.](#)

```

00001 ///  

00002 ///  

00003 ///  

00004 ///  

00005 ///  

00006 ///  

00007 #pragma once  

00008 ///  

00009 #include <string>  

00010 ///  

00011 #include "Interfaces/IGameClient.hpp"  

00012 ///  

00013 namespace cli  

00014 {  

00015     ///  

00016     ///  

00017     ///  

00018     ///  

00019     ///  

00020     ///  

00021     class AGameClient : public IGameClient  

00022     {  

00023     public:  

00024         ~AGameClient() override = default;  

00025         ///  

00026         [[nodiscard]] std::string &getName() override { return m_name; }  

00027         void setName(const std::string &newName) override { m_name = newName; }  

00028         ///  

00029     private:  

00030         std::string m_name = "default_name";  

00031     }; // class AGameClient  

00032 }  

00033 } // namespace cli

```

8.49 /home/masina/Projects/Epitech/rtype/modules/ Interfaces/include/Interfaces/AGameServer.hpp File Reference

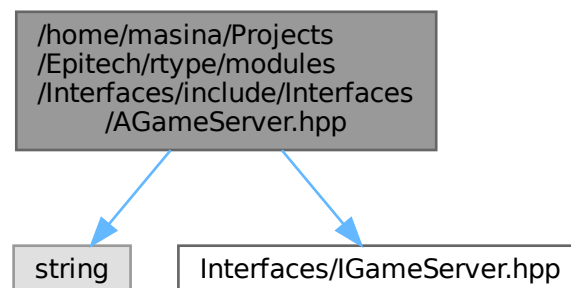
This file contains the game abstract class.

```

#include <string>
#include "Interfaces/IGameServer.hpp"

```

Include dependency graph for AGameServer.hpp:



Classes

- class [srv::AGameServer](#)
Abstraction for the games.

Namespaces

- namespace [srv](#)

8.49.1 Detailed Description

This file contains the game abstract class.

Definition in file [AGameServer.hpp](#).

8.50 AGameServer.hpp

[Go to the documentation of this file.](#)

```

00001 ///  

00002 ///  

00003 ///  

00004 ///  

00005 ///  

00006 ///  

00007 #pragma once  

00008  

00009 #include <string>  

00010  

00011 #include "Interfaces/IGameServer.hpp"  

00012  

00013 namespace srv  

00014 {  

00015  

00016     ///  

00017     ///  

00018     ///  

00019     ///  

00020     ///  

00021     class AGameServer : public IGameServer  

00022     {  

00023     public:  

00024         ~AGameServer() override = default;  

00025  

00026         [[nodiscard]] std::string &getName() override { return m_name; }  

00027         void setName(const std::string &newName) override { m_name = newName; }  

00028  

00029     private:  

00030         std::string m_name = "default_name";  

00031     }; // class AGameServer  

00032  

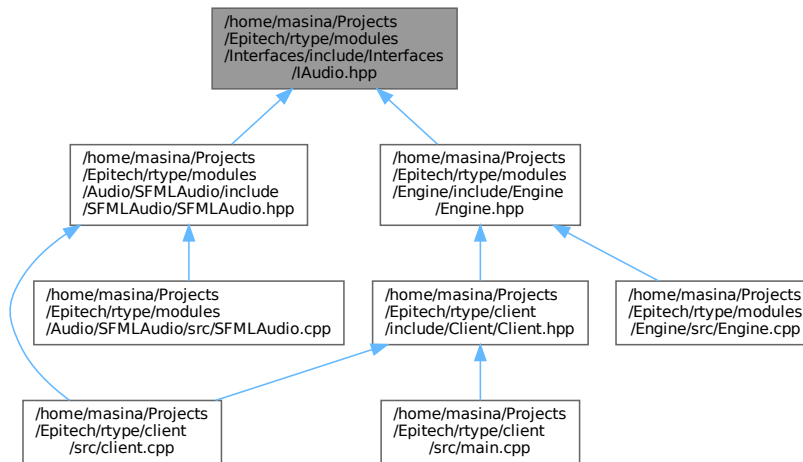
00033 } // namespace srv

```

8.51 /home/masina/Projects/Epitech/rtype/modules/ Interfaces/include/Interfaces/IAudio.hpp File Reference

This file contains the Audio interface.

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



Classes

- class [eng::IAudio](#)
Interface for the audio.

Namespaces

- namespace [eng](#)

8.51.1 Detailed Description

This file contains the Audio interface.

Definition in file [IAudio.hpp](#).

8.52 IAudio.hpp

[Go to the documentation of this file.](#)

```

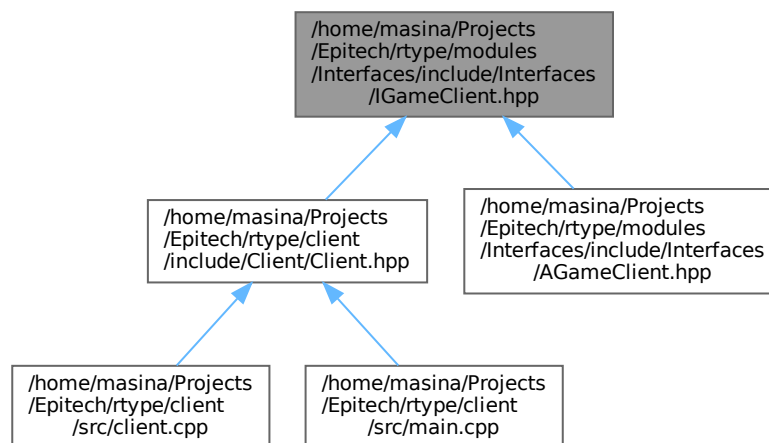
00001 ///
00002 /// @file IAudio.hpp
00003 /// @brief This file contains the Audio interface
00004 /// @namespace eng
00005 ///
00006
00007 #pragma once
00008
00009 namespace eng
00010 {
00011
00012     ///
00013     /// @class IAudio
00014     /// @brief Interface for the audio
00015     /// @namespace eng
00016     ///
00017     class IAudio
  
```

```
00018 {  
00019     public:  
00020         virtual ~IAudio() = default;  
00021  
00022         virtual void createAudio(const std::string &path, float volume, bool loop, const std::string &name) = 0;  
00023         virtual void playAudio(const std::string &name) = 0;  
00024  
00025         virtual void setVolume(const std::string &name, float volume) = 0;  
00026         virtual void setLoop(const std::string &name, bool loop) = 0;  
00027  
00028     private:  
00029 }; // class IAudio  
00030  
00031 } // namespace eng
```

8.53 /home/masina/Projects/Epitech/rtype/modules/ Interfaces/include/Interfaces/IGameClient.hpp File Reference

This file contains the Game interface.

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



Classes

- class `cli::IGameClient`
Interface for the games.

Namespaces

- namespace `cli`

8.53.1 Detailed Description

This file contains the Game interface.

Definition in file [IGameClient.hpp](#).

8.54 IGameClient.hpp

[Go to the documentation of this file.](#)

```

00001 ///  

00002 ///  

00003 ///  

00004 ///  

00005 ///  

00006 ///  

00007 #pragma once  

00008  

00009 namespace cli  

00010 {  

00011  

00012     ///  

00013     ///  

00014     ///  

00015     ///  

00016     ///  

00017     class IGameClient  

00018     {  

00019     public:  

00020         virtual ~IGameClient() = default;  

00021  

00022         [[nodiscard]] virtual std::string &getName();  

00023         virtual void setName(const std::string &newName);  

00024  

00025     private:  

00026     }; // class IGameClient  

00027  

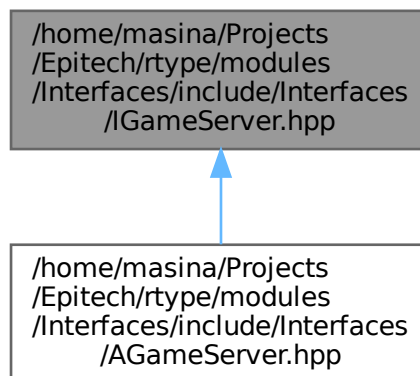
00028 } // namespace cli

```

8.55 /home/masina/Projects/Epitech/rtype/modules/ Interfaces/include/Interfaces/IGameServer.hpp File Reference

This file contains the Game interface.

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



Classes

- class [srv::IGameServer](#)
Interface for the games.

Namespaces

- namespace [srv](#)

8.55.1 Detailed Description

This file contains the Game interface.

Definition in file [IGameServer.hpp](#).

8.56 IGameServer.hpp

[Go to the documentation of this file.](#)

```

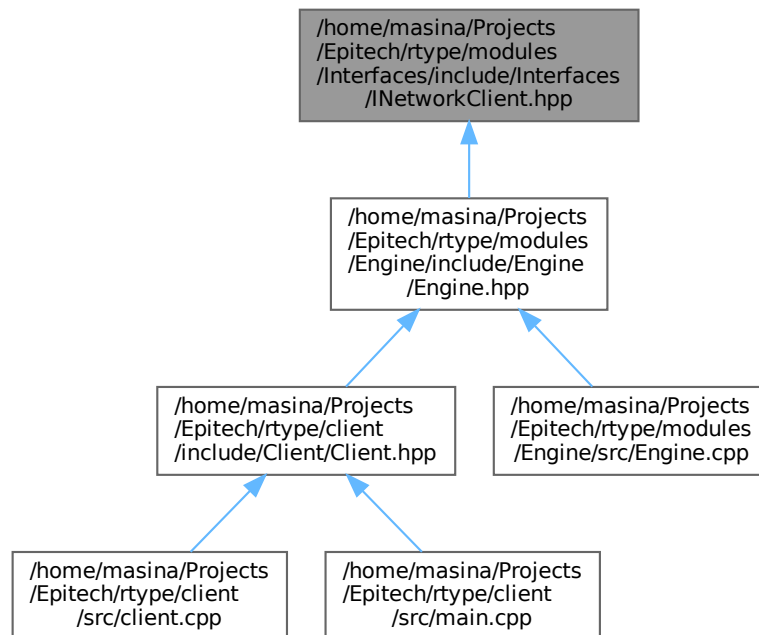
00001 ///
00002 /// @file IGameServer.hpp
00003 /// @brief This file contains the Game interface
00004 /// @namespace srv
00005 ///
00006
00007 #pragma once
00008
00009 namespace srv
00010 {
00011
00012     ///
00013     /// @class IGameServer
00014     /// @brief Interface for the games
00015     /// @namespace srv
00016     ///
00017     class IGameServer
00018     {
00019     public:
00020         virtual ~IGameServer() = default;
00021
00022         [[nodiscard]] virtual std::string &getName();
00023         virtual void setName(const std::string &newName);
00024
00025     private:
00026     }; // class IGameServer
00027
00028 } // namespace srv

```

8.57 /home/masina/Projects/Epitech/rtype/modules/↵ Interfaces/include/Interfaces/INetworkClient.hpp File Reference

This file contains the client network interface.

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



Classes

- class [eng::INetworkClient](#)
Interface for the client network.

Namespaces

- namespace [eng](#)

8.57.1 Detailed Description

This file contains the client network interface.

Definition in file [INetworkClient.hpp](#).

8.58 INetworkClient.hpp

[Go to the documentation of this file.](#)

```

00001 ///
00002 /// @file INetworkClient.hpp
00003 /// @brief This file contains the client network interface
00004 /// @namespace eng
00005 ///
00006
00007 #pragma once
  
```

```
00008
00009 namespace eng
00010 {
00011
00012     ///
00013     /// @class INetworkClient
00014     /// @brief Interface for the client network
00015     /// @namespace eng
00016     ///
00017     class INetworkClient
00018     {
00019     public:
00020         virtual ~INetworkClient() = default;
00021
00022     private:
00023     }; // class INetworkClient
00024
00025 } // namespace eng
```

8.59 /home/masina/Projects/Epitech/rtype/modules/ Interfaces/include/Interfaces/INetworkServer.hpp File Reference

This file contains the server network interface.

Classes

- class [srv::INetworkServer](#)
Interface for the server network.

Namespaces

- namespace [srv](#)

8.59.1 Detailed Description

This file contains the server network interface.

Definition in file [INetworkServer.hpp](#).

8.60 INetworkServer.hpp

[Go to the documentation of this file.](#)

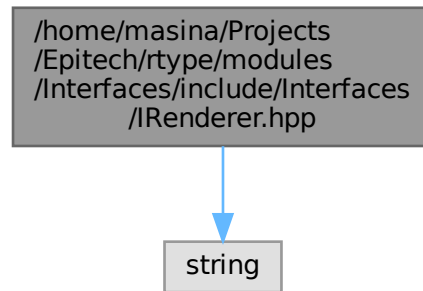
```
00001 ///
00002 /// @file INetworkServer.hpp
00003 /// @brief This file contains the server network interface
00004 /// @namespace srv
00005 ///
00006
00007 #pragma once
00008
00009 namespace srv
00010 {
00011
00012     ///
00013     /// @class INetworkServer
00014     /// @brief Interface for the server network
00015     /// @namespace srv
00016     ///
00017     class INetworkServer
00018     {
00019     public:
00020         virtual ~INetworkServer() = default;
00021
00022     private:
00023     }; // class INetworkServer
00024
00025 } // namespace srv
```

8.61 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IRenderer.hpp File Reference

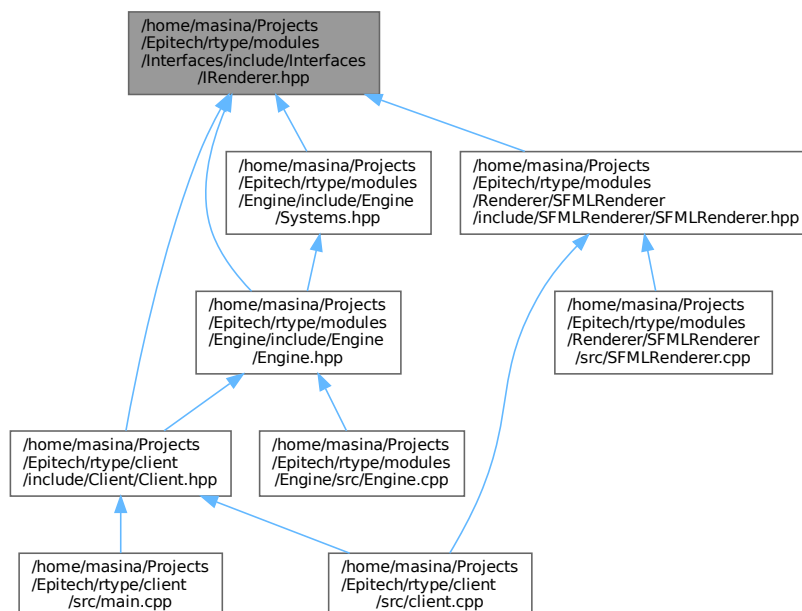
This file contains the IRenderer class declaration.

```
#include <string>
```

Include dependency graph for IRenderer.hpp:



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



Classes

- struct [eng::Audio](#)

- struct `eng::Color`
- struct `eng::Font`
- struct `eng::Text`
- struct `eng::Event`
- class `eng::IRenderer`

Interface for the renderer.

Namespaces

- namespace `eng`
- namespace `cli`

Enumerations

- enum class `eng::Key` {
`eng::Unknown` , `eng::Escape` , `eng::Space` , `eng::Up` ,
`eng::Down` , `eng::Left` , `eng::Right` , `eng::A` ,
`eng::B` , `eng::C` , `eng::D` , `eng::E` ,
`eng::F` , `eng::G` , `eng::H` , `eng::I` ,
`eng::J` , `eng::K` , `eng::L` , `eng::M` ,
`eng::N` , `eng::O` , `eng::P` , `eng::Q` ,
`eng::R` , `eng::S` , `eng::T` , `eng::U` ,
`eng::V` , `eng::W` , `eng::X` , `eng::Y` ,
`eng::Z` , `eng::Num0` , `eng::Num1` , `eng::Num2` ,
`eng::Num3` , `eng::Num4` , `eng::Num5` , `eng::Num6` ,
`eng::Num7` , `eng::Num8` , `eng::Num9` }
- enum class `eng::EventType` { `eng::Closed` , `eng::KeyPressed` , `eng::KeyReleased` , `eng::None` }

8.61.1 Detailed Description

This file contains the IRenderer class declaration.

Definition in file [IRenderer.hpp](#).

8.62 IRenderer.hpp

[Go to the documentation of this file.](#)

```
00001 ///  
00002 ///  
00003 ///  
00004 ///  
00005 ///  
00006  
00007 #pragma once  
00008  
00009 #include <string>  
00010  
00011 namespace eng  
00012 {  
00013     struct Audio  
00014     {  
00015         std::string path;  
00016         float volume;  
00017         bool loop;  
00018         std::string name;  
00019     };  
00020     struct Color  
00021     {  
00022         uint8_t r;
```

```

00023         uint8_t g;
00024         uint8_t b;
00025         uint8_t a;
00026     };
00027     struct Font
00028     {
00029         std::string path;
00030         std::string name;
00031     };
00032     struct Text
00033     {
00034         std::string fontName;
00035         Color color;
00036         std::string content;
00037         int size;
00038         int x;
00039         int y;
00040         std::string name;
00041     };
00042
00043     enum class Key
00044     {
00045         Unknown,
00046         Escape,
00047         Space,
00048         Up,
00049         Down,
00050         Left,
00051         Right,
00052         A,
00053         B,
00054         C,
00055         D,
00056         E,
00057         F,
00058         G,
00059         H,
00060         I,
00061         J,
00062         K,
00063         L,
00064         M,
00065         N,
00066         O,
00067         P,
00068         Q,
00069         R,
00070         S,
00071         T,
00072         U,
00073         V,
00074         W,
00075         X,
00076         Y,
00077         Z,
00078         Num0,
00079         Num1,
00080         Num2,
00081         Num3,
00082         Num4,
00083         Num5,
00084         Num6,
00085         Num7,
00086         Num8,
00087         Num9
00088     };
00089     enum class EventType
00090     {
00091         Closed,
00092         KeyPressed,
00093         KeyReleased,
00094         None
00095     };
00096
00097     struct Event
00098     {
00099         EventType type = EventType::None;
00100         Key key = Key::Unknown;
00101     };
00102
00103     ///
00104     /// @class IRenderer
00105     /// @brief Interface for the renderer
00106     /// @namespace cli
00107     ///
00108     class IRenderer
00109     {

```

```

00110
00111     public:
00112         virtual ~IRenderer() = default;
00113
00114         virtual void createWindow(const std::string &title, unsigned int height, unsigned int width,
00115                                   unsigned int frameLimit, bool fullscreen) = 0;
00116         [[nodiscard]] virtual bool windowIsOpen() const = 0;
00117         virtual void closeWindow() = 0;
00118
00119         virtual void createFont(Font font) = 0;
00120         virtual void createText(Text text) = 0;
00121         virtual void drawText(const std::string &name) = 0;
00122         virtual void setTextContent(const std::string &name, const std::string &content) = 0;
00123         virtual void setTextPosition(const std::string &name, int x, int y) = 0;
00124         virtual void setTextColor(const std::string &name, Color color) = 0;
00125         virtual void setFrameLimit(unsigned int frameLimit) = 0;
00126
00127         virtual void clearWindow(Color color) = 0;
00128         virtual void displayWindow() = 0;
00129
00130         [[nodiscard]] virtual bool pollEvent(Event &event) = 0;
00131
00132     private:
00133 }; // class IRenderer
00134
00135 } // namespace eng

```

8.63 /home/masina/Projects/Epitech/rtype/modules/Renderer/SFMLRenderer/include/SFMLRenderer/SFMLRenderer.hpp File Reference

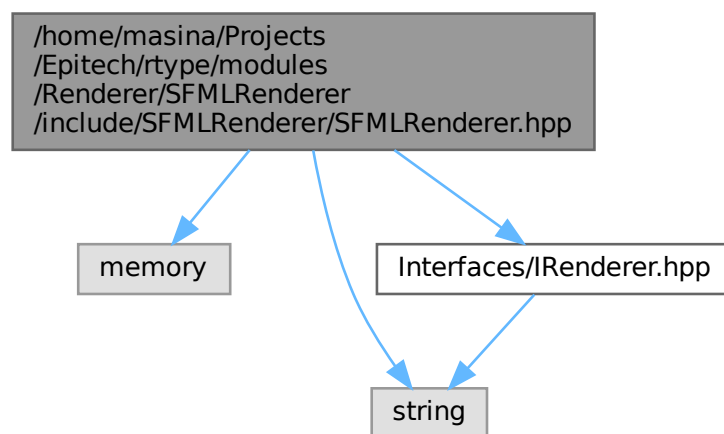
SFMLRenderer class declaration with PImpl.

```
#include <memory>
```

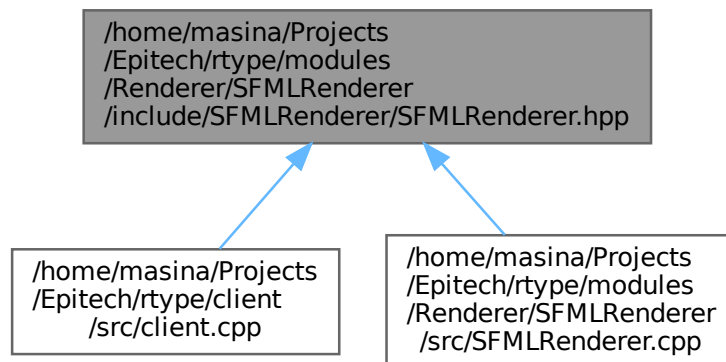
```
#include <string>
```

```
#include "Interfaces/IRenderer.hpp"
```

Include dependency graph for SFMLRenderer.hpp:



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



Classes

- class [eng::SFMLRenderer](#)
Class for the R-Type game.

Namespaces

- namespace [eng](#)

8.63.1 Detailed Description

SFMLRenderer class declaration with PImpl.

Definition in file [SFMLRenderer.hpp](#).

8.64 SFMLRenderer.hpp

[Go to the documentation of this file.](#)

```

00001 ///  

00002 ///  

00003 ///  

00004 ///  

00005 ///  

00006 ///  

00007 #pragma once  

00008  

00009 #include <memory>  

00010 #include <string>  

00011  

00012 #include "Interfaces/IRenderer.hpp"  

00013  

00014 namespace eng  

00015 {  

00016  

00017     ///  

00018     ///  

    
```

File Reference

```

00019  /// @brief Class for the R-Type game
00020  /// @namespace eng
00021  ///
00022  class SFMLRenderer final : public IRenderer
00023  {
00024  public:
00025      SFMLRenderer();
00026      ~SFMLRenderer() override;
00027
00028      SFMLRenderer(const SFMLRenderer &) = delete;
00029      SFMLRenderer &operator=(const SFMLRenderer &) = delete;
00030      SFMLRenderer(SFMLRenderer &&) = delete;
00031      SFMLRenderer &operator=(SFMLRenderer &&) = delete;
00032
00033      void createWindow(const std::string &title, unsigned int height, unsigned int width,
00034                      unsigned int frameLimit, bool fullscreen) override;
00035      void setFrameLimit(unsigned int frameLimit) override;
00036      bool windowIsOpen() const override;
00037      void closeWindow() override;
00038
00039      void createText(Text text) override;
00040      void createFont(Font font) override;
00041      void setTextContent(const std::string &name, const std::string &content) override;
00042      void setPosition(const std::string &name, int x, int y) override;
00043      void setColor(const std::string &name, Color color) override;
00044      void drawText(const std::string &name) override;
00045
00046      void clearWindow(Color color) override;
00047      void displayWindow() override;
00048
00049      bool pollEvent(Event &event) override;
00050
00051  private:
00052      struct Impl;
00053      std::unique_ptr<Impl> m_impl;
00054  }; // class SFMLRenderer
00055
00056 } // namespace eng

```

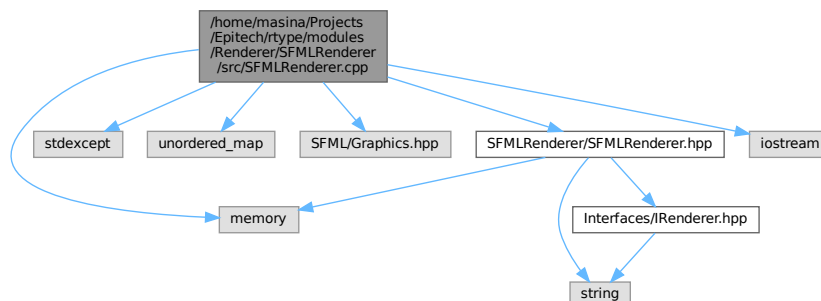
8.65 /home/masina/Projects/Epitech/rtype/modules/Renderer/SFMLRenderer/src/SFMLRenderer.cpp File Reference

```

#include <memory>
#include <stdexcept>
#include <unordered_map>
#include <SFML/Graphics.hpp>
#include "SFMLRenderer/SFMLRenderer.hpp"
#include <iostream>

```

Include dependency graph for SFMLRenderer.cpp:



Classes

- struct `eng::SFMLRenderer::Impl`

Functions

- static [eng::Key](#) [scancodeToKey](#) (const sf::Keyboard::Scancode sc)

8.65.1 Function Documentation

8.65.1.1 scancodeToKey()

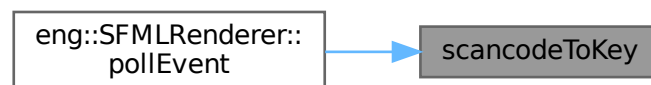
static [eng::Key](#) scancodeToKey (
 const sf::Keyboard::Scancode sc) [static]

Definition at line 112 of file [SFMLRenderer.cpp](#).

References [eng::A](#), [eng::B](#), [eng::C](#), [eng::D](#), [eng::Down](#), [eng::E](#), [eng::Escape](#), [eng::F](#), [eng::G](#), [eng::H](#), [eng::I](#), [eng::J](#), [eng::K](#), [eng::L](#), [eng::Left](#), [eng::M](#), [eng::N](#), [eng::Num0](#), [eng::Num1](#), [eng::Num2](#), [eng::Num3](#), [eng::Num4](#), [eng::Num5](#), [eng::Num6](#), [eng::Num7](#), [eng::Num8](#), [eng::Num9](#), [eng::O](#), [eng::P](#), [eng::Q](#), [eng::R](#), [eng::Right](#), [eng::S](#), [eng::Space](#), [eng::T](#), [eng::U](#), [eng::Unknown](#), [eng::Up](#), [eng::V](#), [eng::W](#), [eng::X](#), [eng::Y](#), and [eng::Z](#).

Referenced by [eng::SFMLRenderer::pollEvent\(\)](#).

Here is the caller graph for this function:



8.66 SFMLRenderer.cpp

[Go to the documentation of this file.](#)

```

00001 #include <memory>
00002 #include <stdexcept>
00003 #include <unordered_map>
00004
00005 #include <SFML/Graphics.hpp>
00006
00007 #include "SFMLRenderer/SFMLRenderer.hpp"
00008
00009 #include <iostream>
00010
00011 struct eng::SFMLRenderer::Impl
00012 {
00013     sf::RenderWindow window;
00014     std::unordered_map<std::string, sf::Font> fonts;
00015     std::unordered_map<std::string, sf::Text> texts;
00016 };
00017
00018 eng::SFMLRenderer::SFMLRenderer() : m_impl(std::make_unique<Impl>()) {}
00019
00020 eng::SFMLRenderer::~SFMLRenderer() = default;
00021
00022 void eng::SFMLRenderer::createWindow(const std::string &title, unsigned int height, unsigned int width,
00023                                     const unsigned int frameLimit, const bool fullscreen)
00024 {
00025     const sf::VideoMode mode = fullscreen ? sf::VideoMode::getDesktopMode() : sf::VideoMode({width, height});
  
```

```

00026     m_impl->window.create(mode, title, fullscreen ? sf::State::Fullscreen : sf::State::Windowed);
00027     m_impl->window.setFramerateLimit(frameLimit);
00028 }
00029
00030 bool eng::SFMLRenderrer::windowIsOpen() const { return m_impl->window.isOpen(); }
00031
00032 void eng::SFMLRenderrer::closeWindow() { m_impl->window.close(); }
00033
00034 void eng::SFMLRenderrer::setFrameLimit(unsigned int frameLimit) { m_impl->window.setFramerateLimit(frameLimit); }
00035
00036 void eng::SFMLRenderrer::createFont(Font font)
00037 {
00038     sf::Font sfFont;
00039     if (!sfFont.openFromFile(font.path))
00040     {
00041         throw std::runtime_error("Failed to load font: " + font.path);
00042     }
00043     m_impl->fonts.emplace(font.name, std::move(sfFont));
00044 }
00045
00046 void eng::SFMLRenderrer::createText(Text text)
00047 {
00048     const auto &font = m_impl->fonts.at(text.fontName);
00049     sf::Text sfText(font);
00050     sfText.setString(text.content);
00051     sfText.setCharacterSize(text.size);
00052     sfText.setPosition({static_cast<float>(text.x), static_cast<float>(text.y)});
00053     sfText.setFillColor(sf::Color(text.color.r, text.color.g, text.color.b, text.color.a));
00054     m_impl->texts.emplace(text.name, std::move(sfText));
00055 }
00056
00057 void eng::SFMLRenderrer::setTextContent(const std::string &name, const std::string &content)
00058 {
00059     if (const auto it = m_impl->texts.find(name); it != m_impl->texts.end())
00060     {
00061         it->second.setString(content);
00062     }
00063     else
00064     {
00065         throw std::runtime_error("Text not found: " + name);
00066     }
00067 }
00068
00069 void eng::SFMLRenderrer::setTextPosition(const std::string &name, int x, int y)
00070 {
00071     if (const auto it = m_impl->texts.find(name); it != m_impl->texts.end())
00072     {
00073         it->second.setPosition({static_cast<float>(x), static_cast<float>(y)});
00074     }
00075     else
00076     {
00077         throw std::runtime_error("Text not found: " + name);
00078     }
00079 }
00080
00081 void eng::SFMLRenderrer::setTextColor(const std::string &name, const Color color)
00082 {
00083     if (const auto it = m_impl->texts.find(name); it != m_impl->texts.end())
00084     {
00085         it->second.setFillColor(sf::Color(color.r, color.g, color.b, color.a));
00086     }
00087     else
00088     {
00089         throw std::runtime_error("Text not found: " + name);
00090     }
00091 }
00092
00093 void eng::SFMLRenderrer::drawText(const std::string &name)
00094 {
00095     if (const auto it = m_impl->texts.find(name); it != m_impl->texts.end())
00096     {
00097         m_impl->window.draw(it->second);
00098     }
00099     else
00100     {
00101         throw std::runtime_error("Text not found: " + name);
00102     }
00103 }
00104
00105 void eng::SFMLRenderrer::clearWindow(const Color color)
00106 {
00107     m_impl->window.clear(sf::Color(color.r, color.g, color.b, color.a));
00108 }
00109
00110 void eng::SFMLRenderrer::displayWindow() { m_impl->window.display(); }
00111
00112 static eng::Key scanCodeToKey(const sf::Keyboard::ScanCode sc)

```

```

00113 {
00114     using S = sf::Keyboard::Scancode;
00115     switch (sc)
00116     {
00117         case S::Escape:
00118             return eng::Key::Escape;
00119         case S::Space:
00120             return eng::Key::Space;
00121         case S::Up:
00122             return eng::Key::Up;
00123         case S::Down:
00124             return eng::Key::Down;
00125         case S::Left:
00126             return eng::Key::Left;
00127         case S::Right:
00128             return eng::Key::Right;
00129         case S::A:
00130             return eng::Key::A;
00131         case S::B:
00132             return eng::Key::B;
00133         case S::C:
00134             return eng::Key::C;
00135         case S::D:
00136             return eng::Key::D;
00137         case S::E:
00138             return eng::Key::E;
00139         case S::F:
00140             return eng::Key::F;
00141         case S::G:
00142             return eng::Key::G;
00143         case S::H:
00144             return eng::Key::H;
00145         case S::I:
00146             return eng::Key::I;
00147         case S::J:
00148             return eng::Key::J;
00149         case S::K:
00150             return eng::Key::K;
00151         case S::L:
00152             return eng::Key::L;
00153         case S::M:
00154             return eng::Key::M;
00155         case S::N:
00156             return eng::Key::N;
00157         case S::O:
00158             return eng::Key::O;
00159         case S::P:
00160             return eng::Key::P;
00161         case S::Q:
00162             return eng::Key::Q;
00163         case S::R:
00164             return eng::Key::R;
00165         case S::S:
00166             return eng::Key::S;
00167         case S::T:
00168             return eng::Key::T;
00169         case S::U:
00170             return eng::Key::U;
00171         case S::V:
00172             return eng::Key::V;
00173         case S::W:
00174             return eng::Key::W;
00175         case S::X:
00176             return eng::Key::X;
00177         case S::Y:
00178             return eng::Key::Y;
00179         case S::Z:
00180             return eng::Key::Z;
00181         case S::Num0:
00182             return eng::Key::Num0;
00183         case S::Num1:
00184             return eng::Key::Num1;
00185         case S::Num2:
00186             return eng::Key::Num2;
00187         case S::Num3:
00188             return eng::Key::Num3;
00189         case S::Num4:
00190             return eng::Key::Num4;
00191         case S::Num5:
00192             return eng::Key::Num5;
00193         case S::Num6:
00194             return eng::Key::Num6;
00195         case S::Num7:
00196             return eng::Key::Num7;
00197         case S::Num8:
00198             return eng::Key::Num8;
00199         case S::Num9:

```



```

00200         return eng::Key::Num9;
00201     default:
00202         return eng::Key::Unknown;
00203     }
00204 }
00205
00206 bool eng::SFMLRenderer::pollEvent(Event &event)
00207 {
00208     if (const auto eventOpt = m_impl->window.pollEvent())
00209     {
00210         const auto &e = *eventOpt;
00211
00212         if (e.is<sf::Event::Closed>())
00213         {
00214             event.type = EventType::Closed;
00215             return true;
00216         }
00217
00218         if (const auto *const key = e.getIf<sf::Event::KeyPressed>())
00219         {
00220             event.type = EventType::KeyPressed;
00221             std::cout << "Key pressed: " << std::to_string(static_cast<int>(key->scancode)) << '\n';
00222             event.key = scancodeToKey(key->scancode);
00223             return true;
00224         }
00225
00226         if (const auto *const key = e.getIf<sf::Event::KeyReleased>())
00227         {
00228             event.type = EventType::KeyReleased;
00229             std::cout << "Key released: " << std::to_string(static_cast<int>(key->scancode)) << '\n';
00230             event.key = scancodeToKey(key->scancode);
00231             return true;
00232         }
00233
00234         event.type = EventType::None;
00235         return true;
00236     }
00237     return false;
00238 }

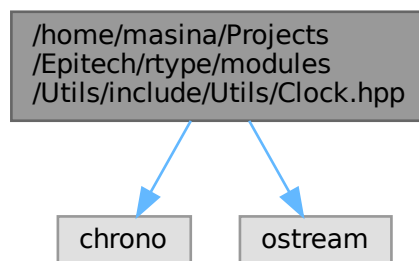
```

8.67 /home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/Clock.hpp File Reference ↩

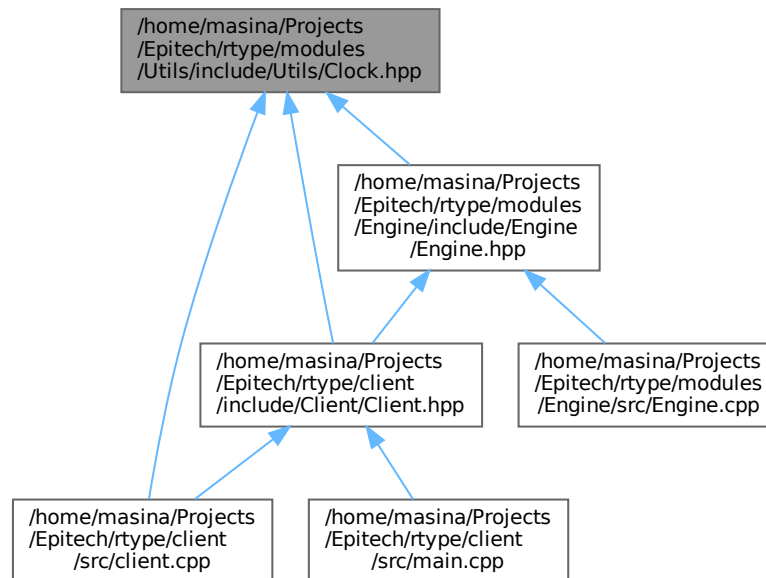
This file contains the Clock class.

```
#include <chrono>
#include <ostream>
```

Include dependency graph for Clock.hpp:



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



Classes

- class [utl::Clock](#)
Class for clock.

Namespaces

- namespace [utl](#)

8.67.1 Detailed Description

This file contains the Clock class.

Definition in file [Clock.hpp](#).

8.68 Clock.hpp

[Go to the documentation of this file.](#)

```

00001 ///
00002 /// @file Clock.hpp
00003 /// @brief This file contains the Clock class
00004 /// @namespace utl
00005 ///
00006
00007 #pragma once
00008
00009 #include <chrono>
00010 #include <ostream>

```

```

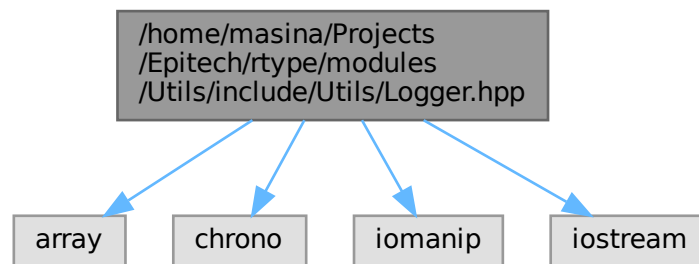
00011
00012 namespace utl
00013 {
00014
00015     ///
00016     /// @class Clock
00017     /// @brief Class for clock
00018     /// @namespace utl
00019     ///
00020     class Clock
00021     {
00022
00023     public:
00024         using TimePoint = std::chrono::time_point<std::chrono::high_resolution_clock>;
00025
00026         explicit Clock(const bool startNow = true) : m_start{startNow ? now() : TimePoint()}, m_pausedDuration{0} {}
00027         ~Clock() = default;
00028
00029         Clock(const Clock &) = delete;
00030         Clock &operator=(const Clock &) = delete;
00031         Clock(Clock &&) = delete;
00032         Clock &operator=(Clock &&) = delete;
00033
00034         friend std::ostream &operator<<(std::ostream &os, const Clock &clock)
00035         {
00036             os << "Elapsed time: " << clock.getDeltaSeconds() << " seconds";
00037             return os;
00038         }
00039
00040         static TimePoint now() { return std::chrono::high_resolution_clock::now(); }
00041         void restart()
00042         {
00043             m_start = now();
00044             m_pausedDuration = Duration(0);
00045             m_isPaused = false;
00046         }
00047         void pause()
00048         {
00049             if (!m_isPaused)
00050             {
00051                 m_pausedTime = now();
00052                 m_isPaused = true;
00053             }
00054         }
00055         void resume()
00056         {
00057             if (m_isPaused)
00058             {
00059                 m_pausedDuration += now() - m_pausedTime;
00060                 m_isPaused = false;
00061             }
00062         }
00063         [[nodiscard]] float getDeltaSeconds() const
00064         {
00065             if (m_isPaused)
00066             {
00067                 return std::chrono::duration<float>(m_pausedTime - m_start - m_pausedDuration).count();
00068             }
00069             return std::chrono::duration<float>(now() - m_start - m_pausedDuration).count();
00070         }
00071
00072         template <typename Duration = std::chrono::seconds> [[nodiscard]] auto getElapsed() const
00073         {
00074             return std::chrono::duration_cast<Duration>(now() - m_start - m_pausedDuration);
00075         }
00076
00077     private:
00078         using Duration = std::chrono::high_resolution_clock::duration;
00079
00080         TimePoint m_start;
00081         TimePoint m_pausedTime;
00082         Duration m_pausedDuration;
00083         bool m_isPaused{false};
00084
00085     }; // class Clock
00086
00087 } // namespace utl

```

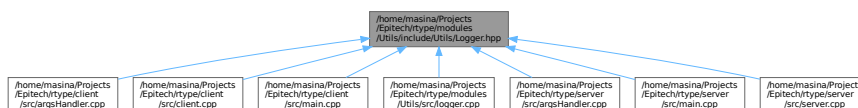
8.69 /home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/Logger.hpp File Reference

```
#include <array>
#include <chrono>
#include <iomanip>
#include <iostream>
```

Include dependency graph for Logger.hpp:



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



Classes

- class [utl::Logger](#)

Namespaces

- namespace [utl](#)

Enumerations

- enum class [utl::LogLevel](#) : uint8_t { [utl::INFO](#) , [utl::WARNING](#) }

8.70 Logger.hpp

[Go to the documentation of this file.](#)

```

00001 #pragma once
00002
00003 #include <array>
00004 #include <chrono>
00005 #include <iomanip>
00006 #include <iostream>
00007
00008 namespace utl
00009 {
00010
00011     enum class LogLevel : uint8_t
00012     {
00013         INFO,
00014         WARNING
00015     };
00016
00017     class Logger
00018     {
00019     public:
00020         Logger(const Logger &) = delete;
00021         Logger &operator=(const Logger &) = delete;
00022         Logger(Logger &&) = delete;
00023         Logger &operator=(Logger &&) = delete;
00024
00025         static void init();
00026
00027         template <typename Func> static void logExecutionTime(const std::string &message, Func &&func)
00028         {
00029             const auto start = std::chrono::high_resolution_clock::now();
00030             func();
00031             const auto end = std::chrono::high_resolution_clock::now();
00032             const auto duration = std::chrono::duration<float, std::milli>(end - start).count();
00033
00034             std::cout << getColorForDuration(duration)
00035                     << formatLogMessage(LogLevel::INFO, message + " took " + std::to_string(duration) + " ms")
00036                     << LOG_LEVEL_COLOR[COLOR_RESET];
00037         }
00038
00039         static void log(const std::string &message, const LogLevel &level)
00040         {
00041             std::cout << (level == LogLevel::INFO ? LOG_LEVEL_COLOR[COLOR_INFO] :
00042 LOG_LEVEL_COLOR[COLOR_WARNING])
00043                     << formatLogMessage(level, message) << LOG_LEVEL_COLOR[COLOR_RESET];
00044         }
00045
00046     private:
00047         enum ColorIndex : uint8_t
00048         {
00049             COLOR_ERROR,
00050             COLOR_INFO,
00051             COLOR_WARNING,
00052             COLOR_RESET
00053         };
00054
00055         static constexpr std::array<const char *, 4> LOG_LEVEL_COLOR = {
00056             "\033[31m", // ERROR/slow execution
00057             "\033[32m", // INFO/fast execution
00058             "\033[33m", // WARNING/medium execution
00059             "\033[0m\n" // RESET + newline
00060         };
00061
00062         static constexpr std::array<const char *, 2> LOG_LEVEL_STRING = {"INFO", "WARNING"};
00063
00064         Logger() = default;
00065         ~Logger() = default;
00066
00067         [[nodiscard]] static const char *getColorForDuration(const float duration)
00068         {
00069             return duration < 20.0F
00070                 ? LOG_LEVEL_COLOR[COLOR_INFO]
00071                 : (duration < 90.0F ? LOG_LEVEL_COLOR[COLOR_WARNING] :
00072 LOG_LEVEL_COLOR[COLOR_ERROR]);
00073         }
00074
00075         [[nodiscard]] static std::string formatLogMessage(LogLevel level, const std::string &message)
00076         {
00077             const auto inTime = std::chrono::system_clock::to_time_t(std::chrono::system_clock::now());
00078             std::ostringstream ss;
00079             ss << "[" << std::put_time(std::localtime(&inTime), "%Y-%m-%d %X") << " ";
00080             ss << "[" << LOG_LEVEL_STRING[static_cast<uint8_t>(level)] << " " << message;
00081             return ss.str();
00082         }
00083     };
00084 }

```

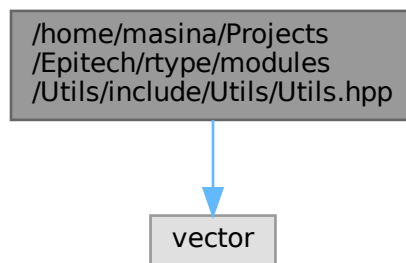
```
00081     }  
00082  
00083 }; // class Logger  
00084  
00085 } // namespace utl
```

8.71 /home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/Utils.hpp File Reference

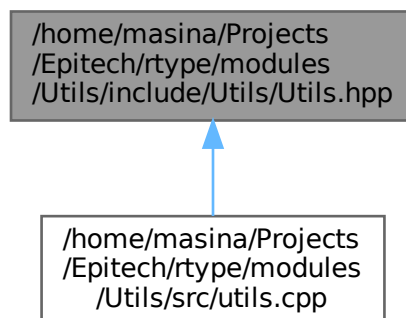
This file contains utility functions.

```
#include <vector>
```

Include dependency graph for Utils.hpp:



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



Namespaces

- namespace [utl](#)

Functions

- `std::vector< char > utl::readFile (const std::string &filename)`

8.71.1 Detailed Description

This file contains utility functions.

Definition in file [Utils.hpp](#).

8.72 Utils.hpp

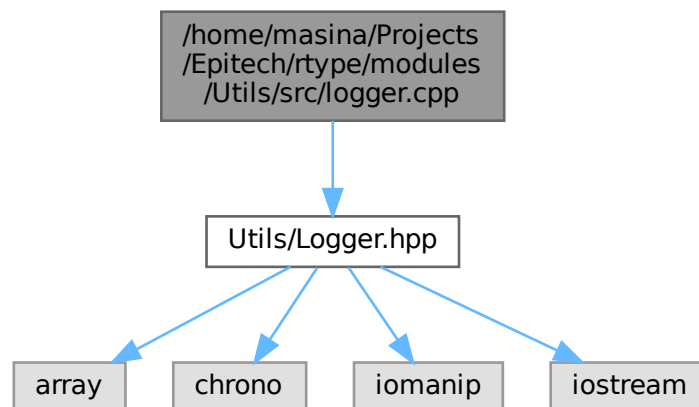
[Go to the documentation of this file.](#)

```
00001 ///  
00002 ///  
00003 ///  
00004 ///  
00005 ///  
00006 ///  
00007 #pragma once  
00008  
00009 #include <vector>  
00010  
00011 namespace utl  
00012 {  
00013  
00014     [[nodiscard]] std::vector<char> readFile(const std::string &filename);  
00015  
00016 } // namespace utl
```

8.73 /home/masina/Projects/Epitech/rtype/modules/
Utils/src/logger.cpp File Reference

```
#include "Utils/Logger.hpp"
```

Include dependency graph for logger.cpp:



8.74 logger.cpp

[Go to the documentation of this file.](#)

```

00001 #ifndef _WIN32
00002 #include <windows.h>
00003 #endif
00004
00005 #include "Utils/Logger.hpp"
00006
00007 void utl::Logger::init()
00008 {
00009     #ifndef _WIN32
00010         const HANDLE hOut = GetStdHandle(STD_OUTPUT_HANDLE);
00011         DWORD dwMode = 0;
00012         if (hOut != INVALID_HANDLE_VALUE && GetConsoleMode(hOut, &dwMode))
00013         {
00014             SetConsoleMode(hOut, dwMode | ENABLE_VIRTUAL_TERMINAL_PROCESSING);
00015         }
00016     #endif
00017 }

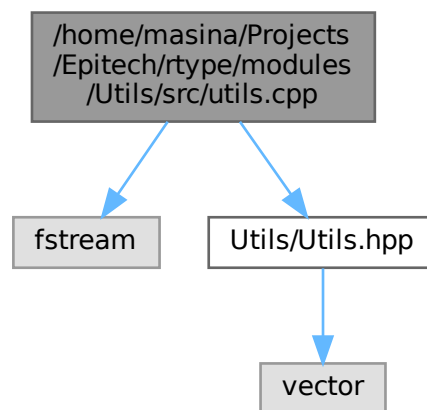
```

8.75 /home/masina/Projects/Epitech/rtype/modules/Utils/src/Utils.cpp File Reference

```
#include <fstream>
```

```
#include "Utils/Utils.hpp"
```

Include dependency graph for utils.cpp:



8.76 utils.cpp

[Go to the documentation of this file.](#)

```

00001 #include <fstream>
00002
00003 #include "Utils/Utils.hpp"
00004
00005 std::vector<char> utl::readFile(const std::string &filename)
00006 {
00007     std::ifstream file(filename, std::ios::binary | std::ios::ate);
00008     if (!file.is_open())

```



```
00009 {
00010     throw std::runtime_error("failed to open file " + filename);
00011 }
00012 const size_t fileSize = file.tellg();
00013 if (fileSize <= 0)
00014 {
00015     throw std::runtime_error("file " + filename + " is empty");
00016 }
00017 std::vector<char> buffer(fileSize);
00018 file.seekg(0, std::ios::beg);
00019 if (!file.read(buffer.data(), fileSize))
00020 {
00021     throw std::runtime_error("failed to read file " + filename);
00022 }
00023 return buffer;
00024 }
```

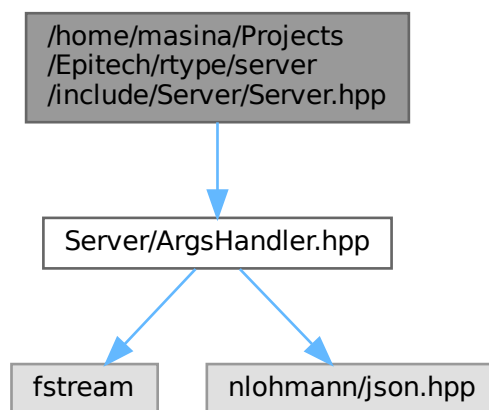
8.77 /home/masina/Projects/Epitech/rtype/README.md File Reference

8.78 /home/masina/Projects/Epitech/rtype/server/include/Server/Server.hpp File Reference

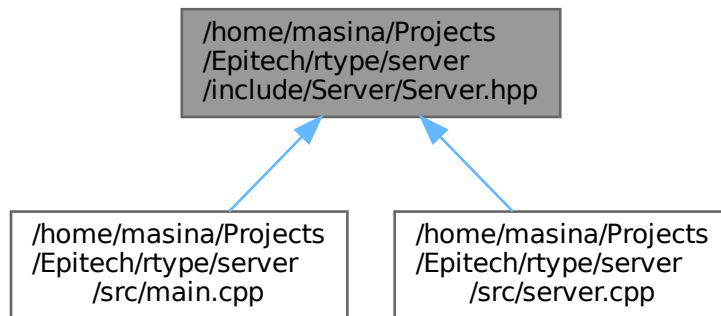
This file contains the Server class declaration.

```
#include "Server/ArgsHandler.hpp"
```

Include dependency graph for Server.hpp:



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



Classes

- class [srv::Server](#)
Class for the server.

Namespaces

- namespace [srv](#)

8.78.1 Detailed Description

This file contains the Server class declaration.

Definition in file [Server.hpp](#).

8.79 Server.hpp

[Go to the documentation of this file.](#)

```

00001 ///  

00002 ///  

00003 ///  

00004 ///  

00005 ///  

00006 ///  

00007 #pragma once  

00008  

00009 #include "Server/ArgsHandler.hpp"  

00010  

00011 namespace srv  

00012 {  

00013  

00014     ///  

00015     ///  

00016     ///  

00017     ///  

00018     ///  

00019     class Server  

00020     {
  
```

```

00021
00022     public:
00023         explicit Server(const ArgsConfig &config);
00024         ~Server() = default;
00025
00026         Server(const Server &) = delete;
00027         Server &operator=(const Server &) = delete;
00028         Server(Server &&) = delete;
00029         Server &operator=(Server &&) = delete;
00030
00031     private:
00032 }; // class Server
00033
00034 } // namespace srv

```

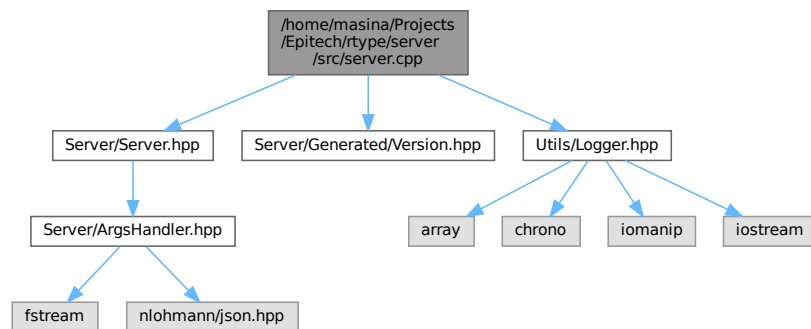
8.80 /home/masina/Projects/Epitech/rtype/server/src/server.cpp File Reference

```

#include "Server/Server.hpp"
#include "Server/Generated/Version.hpp"
#include "Utils/Logger.hpp"

```

Include dependency graph for server.cpp:



8.81 server.cpp

[Go to the documentation of this file.](#)

```

00001 #include "Server/Server.hpp"
00002 #include "Server/Generated/Version.hpp"
00003 #include "Utils/Logger.hpp"
00004
00005 srv::Server::Server(const ArgsConfig &config)
00006 {
00007     (void)config;
00008     utl::Logger::log("PROJECT INFO:", utl::LogLevel::INFO);
00009     std::cout << "  \tName: " PROJECT_NAME "\n"
00010               << "  \tVersion: " PROJECT_VERSION "\n"
00011               << "  \tBuild type: " BUILD_TYPE "\n"
00012               << "  \tGit tag: " GIT_TAG "\n"
00013               << "  \tGit commit hash: " GIT_COMMIT_HASH "\n";
00014 }

```


Index

/home/masina/Projects/Epitech/rtype/README.md	200, 201
/home/masina/Projects/Epitech/rtype/client/include/Client/Assets/Project/E	201, 202
/home/masina/Projects/Epitech/rtype/client/include/Client/Scene/Proj	203
/home/masina/Projects/Epitech/rtype/client/include/Client/Scene/Objec	204, 205
/home/masina/Projects/Epitech/rtype/client/include/Client/Scene/Proje	207, 208
/home/masina/Projects/Epitech/rtype/client/src/argsHandler.cpp	209, 210
/home/masina/Projects/Epitech/rtype/client/src/client.hpp	213, 214
/home/masina/Projects/Epitech/rtype/client/src/main.cpp	216, 217
/home/masina/Projects/Epitech/rtype/modules/Audio/SFMLAudio/SFMLAudi	218, 219
/home/masina/Projects/Epitech/rtype/modules/Audio/SFMLAudio/SFMLAudi	219, 220
/home/masina/Projects/Epitech/rtype/modules/ECS/Entity/ECS/Entities	220
/home/masina/Projects/Epitech/rtype/modules/ECS/Entity/ECS/EntitySp	151, 153
/home/masina/Projects/Epitech/rtype/modules/ECS/Entity/ECS/Registry	160, 162
/home/masina/Projects/Epitech/rtype/modules/ECS/Entity/ECS/Systems	221, 222
/home/masina/Projects/Epitech/rtype/server/src/argsHandler.cpp	165, 166
/home/masina/Projects/Epitech/rtype/server/src/main.cpp,	171, 172
/home/masina/Projects/Epitech/rtype/server/src/server.cpp,	223
/home/masina/Projects/Epitech/rtype/modules/Engine/GameClient/Engin	cli::AGameClient, 25
/home/masina/Projects/Epitech/rtype/modules/Engine/GameServer/Engin	srv::AGameServer, 28
/home/masina/Projects/Epitech/rtype/modules/Engine/ArgsHandler	cli::ArgsHandler, 34
/home/masina/Projects/Epitech/rtype/modules/Interfaces/InArg/H/Interfa	~AudioSystem
/home/masina/Projects/Epitech/rtype/modules/Interfaces/InAudio/System	eng::AudioSystem, 46
/home/masina/Projects/Epitech/rtype/modules/Interfaces/InGame/Inc	cli::Client, 50
/home/masina/Projects/Epitech/rtype/modules/Interfaces/InGame/Clock	utl::Clock, 53
	~Engine

- eng::Engine, 62
- ~FontSystem
 - ecs::FontSystem, 72
 - eng::FontSystem, 75
- ~IAudio
 - eng::IAudio, 78
- ~IGameClient
 - cli::IGameClient, 80
- ~IGameServer
 - srv::IGameServer, 82
- ~INetworkClient
 - eng::INetworkClient, 86
- ~INetworkServer
 - srv::INetworkServer, 87
- ~IPool
 - ecs::Registry::IPool, 88
- ~IRenderer
 - eng::IRenderer, 91
- ~ISystem
 - eng::ISystem, 95
- ~Logger
 - utl::Logger, 98
- ~Registry
 - ecs::Registry, 106
- ~SFMLAudio
 - eng::SFMLAudio, 123
- ~SFMLRenderer
 - eng::SFMLRenderer, 128
- ~Scene
 - eng::Scene, 114
- ~SceneManager
 - eng::SceneManager, 117
- ~Server
 - srv::Server, 120
- ~TextSyStem
 - ecs::TextSyStem, 139
 - eng::TextSyStem, 143
- A
 - eng, 19
- a
 - ecs::Color, 59
 - eng::Color, 60
- add
 - ecs::Registry::Pool< T >, 104
- addComponent
 - ecs::Registry, 107
- addScene
 - eng::SceneManager, 118
- addSystem
 - eng::Engine, 63
- APP_EXTENSION
 - argsHandler.cpp, 163, 165
- ArgsHandler
 - cli::ArgsHandler, 34
 - srv::ArgsHandler, 37
- argsHandler.cpp
 - APP_EXTENSION, 163, 165
 - HELP_MESSAGE, 163, 166
 - VERSION_MESSAGE, 163, 166
- AUDIO_BATTLE_THEME
 - cli::Paths::Audio, 15
- AUDIO_COIN
 - cli::Paths::Audio, 15
- AUDIO_TITLE
 - cli::Paths::Audio, 15
- AudioSystem
 - ecs::AudioSystem, 42, 43
 - eng::AudioSystem, 46
- B
 - eng, 19
- b
 - ecs::Color, 59
 - eng::Color, 60
- BUILD_TYPE
 - Version.hpp, 158, 161
- C
 - eng, 19
- clearWindow
 - eng::IRenderer, 91
 - eng::SFMLRenderer, 129
- cli, 13
 - json, 13
- cli::AGameClient, 23
 - ~AGameClient, 25
 - getName, 25
 - m_name, 25
 - setName, 25
- cli::ArgsConfig, 29
 - exit, 30
 - frameLimit, 30
 - fromFile, 29
 - fullscreen, 30
 - height, 30
 - width, 30
- cli::ArgsHandler, 33
 - ~ArgsHandler, 34
 - ArgsHandler, 34
 - operator=, 34
 - ParseArgs, 34
 - ParseEnv, 35
- cli::Client, 48
 - ~Client, 50
 - Client, 49, 50
 - m_engine, 51
 - m_game, 51
 - operator=, 50
- cli::Config, 14
- cli::Config::Audio, 14
 - DEFAULT_AUDIO_MUTED, 14
 - DEFAULT_AUDIO_VOLUME, 14
- cli::Config::Window, 14
 - DEFAULT_WINDOW_FRAME_LIMIT, 14
 - DEFAULT_WINDOW_FULLSCREEN, 14
 - DEFAULT_WINDOW_HEIGHT, 15
 - DEFAULT_WINDOW_WIDTH, 15

- cli::EnvConfig, 67
- cli::IGameClient, 79
 - ~IGameClient, 80
 - getName, 80
 - setName, 80
- cli::Paths, 15
- cli::Paths::Audio, 15
 - AUDIO_BATTLE_THEME, 15
 - AUDIO_COIN, 15
 - AUDIO_TITLE, 15
- cli::Paths::Fonts, 16
 - FONTS_RTYPE, 16
- Client
 - cli::Client, 49, 50
- Clock
 - utl::Clock, 53, 54
- Closed
 - eng, 18
- closeWindow
 - eng::IRenderer, 91
 - eng::SFMLRenderer, 129
- color
 - eng::Text, 137
- COLOR_ERROR
 - utl::Logger, 97
- COLOR_INFO
 - utl::Logger, 97
- COLOR_RESET
 - utl::Logger, 97
- COLOR_WARNING
 - utl::Logger, 97
- ColorIndex
 - utl::Logger, 97
- content
 - ecs::Text, 135
 - eng::Text, 137
- createAudio
 - eng::IAudio, 78
 - eng::SFMLAudio, 124
- createEntity
 - ecs::Registry, 107
- createFont
 - eng::IRenderer, 91
 - eng::SFMLRenderer, 129
- createText
 - eng::IRenderer, 91
 - eng::SFMLRenderer, 129
- createWindow
 - eng::IRenderer, 91
 - eng::SFMLRenderer, 129
- D
 - eng, 19
- data
 - ecs::Registry::Pool< T >, 105
- DEFAULT_AUDIO_MUTED
 - cli::Config::Audio, 14
- DEFAULT_AUDIO_VOLUME
 - cli::Config::Audio, 14
- DEFAULT_WINDOW_FRAME_LIMIT
 - cli::Config::Window, 14
- DEFAULT_WINDOW_FULLSCREEN
 - cli::Config::Window, 14
- DEFAULT_WINDOW_HEIGHT
 - cli::Config::Window, 15
- DEFAULT_WINDOW_WIDTH
 - cli::Config::Window, 15
- displayWindow
 - eng::IRenderer, 91
 - eng::SFMLRenderer, 130
- Down
 - eng, 19
- DrawCallback
 - ecs::TextSyStem, 139
- drawText
 - eng::IRenderer, 92
 - eng::SFMLRenderer, 130
- Duration
 - utl::Clock, 53
- E
 - eng, 19
- ecs, 16
 - Entity, 17
 - INVALID_ENTITY, 17
- ecs::Audio, 39
 - id, 39
 - loop, 39
 - path, 40
 - volume, 40
- ecs::AudioSystem, 42
 - ~AudioSystem, 42
 - AudioSystem, 42, 43
 - operator=, 43
 - update, 43
- ecs::Color, 58
 - a, 59
 - b, 59
 - g, 59
 - id, 59
 - r, 59
- ecs::Font, 69
 - id, 69
 - path, 69
- ecs::FontSystem, 71
 - ~FontSystem, 72
 - FontSystem, 72
 - operator=, 72
 - update, 73
- ecs::Registry, 105
 - ~Registry, 106
 - addComponent, 107
 - createEntity, 107
 - getAll, 108
 - getComponent, 108
 - getPool, 109
 - hasComponent, 110
 - m_components, 112

- m_entities, 112
- m_lastEntity, 112
- m_onComponentAddedCallbacks, 112
- m_onEntityCreatedCallbacks, 112
- onComponentAdded, 110
- onEntityCreated, 111
- operator=, 111
- Registry, 106, 107
- removeComponent, 111
- ecs::Registry::IPool, 87
 - ~IPool, 88
 - remove, 88
- ecs::Registry::Pool< T >, 102
 - add, 104
 - data, 105
 - get, 104
 - has, 104
 - remove, 104
- ecs::Sprite, 132
 - id, 133
 - path, 133
- ecs::Text, 134
 - content, 135
 - fontSize, 135
 - id, 135
- ecs::TextSyStem, 138
 - ~TextSyStem, 139
 - DrawCallback, 139
 - m_drawCallback, 140
 - operator=, 139
 - setDrawCallback, 140
 - TextSyStem, 139
 - update, 140
- ecs::Transform, 145
 - id, 146
 - rotation, 146
 - x, 146
 - y, 146
- ecs::Velocity, 147
 - id, 147
 - x, 147
 - y, 148
- eng, 17
 - A, 19
 - B, 19
 - C, 19
 - Closed, 18
 - D, 19
 - Down, 19
 - E, 19
 - Escape, 19
 - EventType, 18
 - F, 19
 - G, 19
 - H, 19
 - I, 19
 - J, 19
 - K, 19
 - Key, 18
 - KeyPressed, 18
 - KeyReleased, 18
 - L, 19
 - Left, 19
 - M, 19
 - N, 19
 - None, 18
 - Num0, 19
 - Num1, 19
 - Num2, 19
 - Num3, 19
 - Num4, 19
 - Num5, 19
 - Num6, 19
 - Num7, 19
 - Num8, 19
 - Num9, 19
 - O, 19
 - P, 19
 - Q, 19
 - R, 19
 - Right, 19
 - S, 19
 - scene_id_t, 18
 - Space, 19
 - T, 19
 - U, 19
 - Unknown, 19
 - Up, 19
 - V, 19
 - W, 19
 - X, 19
 - Y, 19
 - Z, 19
- eng::Audio, 40
 - loop, 41
 - name, 41
 - path, 41
 - volume, 41
- eng::AudioSystem, 44
 - ~AudioSystem, 46
 - AudioSystem, 46
 - isEnabled, 47
 - m_isEnable, 47
 - m_renderer, 47
 - operator=, 47
 - setEnabled, 47
 - update, 47
- eng::Color, 60
 - a, 60
 - b, 60
 - g, 60
 - r, 61
- eng::Engine, 61
 - ~Engine, 62
 - addSystem, 63
 - Engine, 62, 63

- getAudio, 63
- getClock, 63
- getNetworkClient, 64
- getRegistry, 64
- getRenderer, 64
- m_audio, 66
- m_clock, 66
- m_networkClient, 66
- m_registry, 66
- m_renderer, 66
- m_systems, 66
- operator=, 65
- updateSystems, 65
- eng::Event, 68
 - key, 68
 - type, 68
- eng::Font, 70
 - name, 71
 - path, 71
- eng::FontSystem, 73
 - ~FontSystem, 75
 - FontSystem, 75
 - isEnabled, 76
 - m_isEnable, 76
 - m_renderer, 76
 - operator=, 76
 - setEnabled, 76
 - update, 76
- eng::IAudio, 77
 - ~IAudio, 78
 - createAudio, 78
 - playAudio, 78
 - setLoop, 79
 - setVolume, 79
- eng::INetworkClient, 85
 - ~INetworkClient, 86
- eng::IRenderer, 88
 - ~IRenderer, 91
 - clearWindow, 91
 - closeWindow, 91
 - createFont, 91
 - createText, 91
 - createWindow, 91
 - displayWindow, 91
 - drawText, 92
 - pollEvent, 92
 - setFrameLimit, 92
 - setTextColor, 92
 - setTextContent, 92
 - setTextPosition, 93
 - windowIsOpen, 93
- eng::ISystem, 94
 - ~ISystem, 95
 - isEnabled, 95
 - update, 95
- eng::Scene, 113
 - ~Scene, 114
 - getDisplay, 114
 - getId, 114
 - getName, 114
 - m_display, 115
 - m_id, 115
 - m_name, 115
 - Scene, 114
 - setDisplay, 114
 - setName, 115
- eng::SceneManager, 116
 - ~SceneManager, 117
 - addScene, 118
 - getCurrentScene, 118
 - getScene, 118
 - m_currentSceneId, 118
 - m_scenes, 118
 - operator=, 118
 - SceneManager, 117
 - switchToScene, 118
- eng::SFMLAudio, 121
 - ~SFMLAudio, 123
 - createAudio, 124
 - operator=, 124
 - pImpl, 125
 - playAudio, 124
 - setLoop, 124
 - setVolume, 124
 - SFMLAudio, 123
- eng::SFMLAudio::Impl, 83
 - musics, 84
- eng::SFMLRenderer, 125
 - ~SFMLRenderer, 128
 - clearWindow, 129
 - closeWindow, 129
 - createFont, 129
 - createText, 129
 - createWindow, 129
 - displayWindow, 130
 - drawText, 130
 - m_impl, 132
 - operator=, 130
 - pollEvent, 130
 - setFrameLimit, 131
 - setTextColor, 131
 - setTextContent, 131
 - setTextPosition, 131
 - SFMLRenderer, 128, 129
 - windowIsOpen, 132
- eng::SFMLRenderer::Impl, 84
 - fonts, 85
 - texts, 85
 - window, 85
- eng::Text, 135
 - color, 137
 - content, 137
 - fontName, 137
 - name, 137
 - size, 137
 - x, 137

- y, [137](#)
- eng::TextSyStem, [141](#)
 - ~TextSyStem, [143](#)
 - isEnabled, [144](#)
 - m__isEnabled, [145](#)
 - m__renderer, [145](#)
 - operator=, [144](#)
 - setEnabled, [144](#)
 - TextSyStem, [143](#)
 - update, [144](#)
- Engine
 - eng::Engine, [62](#), [63](#)
- Entity
 - ecs, [17](#)
- Escape
 - eng, [19](#)
- EventType
 - eng, [18](#)
- exit
 - cli::ArgsConfig, [30](#)
 - srv::ArgsConfig, [32](#)
- F
 - eng, [19](#)
- fontName
 - eng::Text, [137](#)
- fonts
 - eng::SFMLRenderer::Impl, [85](#)
- FONTS_RTYPE
 - cli::Paths::Fonts, [16](#)
- fontSize
 - ecs::Text, [135](#)
- FontSystem
 - ecs::FontSystem, [72](#)
 - eng::FontSystem, [75](#)
- formatLogMessage
 - utl::Logger, [98](#)
- frameLimit
 - cli::ArgsConfig, [30](#)
- fromFile
 - cli::ArgsConfig, [29](#)
 - srv::ArgsConfig, [32](#)
- fullscreen
 - cli::ArgsConfig, [30](#)
- G
 - eng, [19](#)
- g
 - ecs::Color, [59](#)
 - eng::Color, [60](#)
- get
 - ecs::Registry::Pool< T >, [104](#)
- getAll
 - ecs::Registry, [108](#)
- getAudio
 - eng::Engine, [63](#)
- getClock
 - eng::Engine, [63](#)
- getColorForDuration
 - utl::Logger, [98](#)
- getComponent
 - ecs::Registry, [108](#)
- getCurrentScene
 - eng::SceneManager, [118](#)
- getDeltaSeconds
 - utl::Clock, [54](#)
- getDisplay
 - eng::Scene, [114](#)
- getElapsed
 - utl::Clock, [54](#)
- getId
 - eng::Scene, [114](#)
- getName
 - cli::AGameClient, [25](#)
 - cli::IGameClient, [80](#)
 - eng::Scene, [114](#)
 - srv::AGameServer, [28](#)
 - srv::IGameServer, [82](#)
- getNetworkClient
 - eng::Engine, [64](#)
- getPool
 - ecs::Registry, [109](#)
- getRegistry
 - eng::Engine, [64](#)
- getRenderer
 - eng::Engine, [64](#)
- getScene
 - eng::SceneManager, [118](#)
- GIT_COMMIT_HASH
 - Version.hpp, [158](#), [161](#)
- GIT_TAG
 - Version.hpp, [158](#), [161](#)
- H
 - eng, [19](#)
- has
 - ecs::Registry::Pool< T >, [104](#)
- hasComponent
 - ecs::Registry, [110](#)
- height
 - cli::ArgsConfig, [30](#)
- HELP_MESSAGE
 - argsHandler.cpp, [163](#), [166](#)
- host
 - srv::ArgsConfig, [32](#)
- I
 - eng, [19](#)
- id
 - ecs::Audio, [39](#)
 - ecs::Color, [59](#)
 - ecs::Font, [69](#)
 - ecs::Sprite, [133](#)
 - ecs::Text, [135](#)
 - ecs::Transform, [146](#)
 - ecs::Velocity, [147](#)
- INFO
 - utl, [21](#)

- init
 - utl::Logger, 98
- INVALID_ENTITY
 - ecs, 17
- isEnabled
 - eng::AudioSystem, 47
 - eng::FontSystem, 76
 - eng::ISystem, 95
 - eng::TextSyStem, 144
- J
 - eng, 19
- json
 - cli, 13
 - srv, 20
- K
 - eng, 19
- Key
 - eng, 18
- key
 - eng::Event, 68
- KeyPressed
 - eng, 18
- KeyReleased
 - eng, 18
- L
 - eng, 19
- Left
 - eng, 19
- log
 - utl::Logger, 99
- LOG_LEVEL_COLOR
 - utl::Logger, 101
- LOG_LEVEL_STRING
 - utl::Logger, 101
- logExecutionTime
 - utl::Logger, 100
- Logger
 - utl::Logger, 97
- LogLevel
 - utl, 20
- loop
 - ecs::Audio, 39
 - eng::Audio, 41
- M
 - eng, 19
- m_audio
 - eng::Engine, 66
- m_clock
 - eng::Engine, 66
- m_components
 - ecs::Registry, 112
- m_currentSceneId
 - eng::SceneManager, 118
- m_display
 - eng::Scene, 115
- m_drawCallback
 - ecs::TextSyStem, 140
- m_engine
 - cli::Client, 51
- m_entities
 - ecs::Registry, 112
- m_game
 - cli::Client, 51
- m_id
 - eng::Scene, 115
- m_impl
 - eng::SFMLRenderer, 132
- m_isEnable
 - eng::AudioSystem, 47
 - eng::FontSystem, 76
 - eng::TextSyStem, 145
- m_isPaused
 - utl::Clock, 57
- m_lastEntity
 - ecs::Registry, 112
- m_name
 - cli::AGameClient, 25
 - eng::Scene, 115
 - srv::AGameServer, 28
- m_networkClient
 - eng::Engine, 66
- m_onComponentAddedCallbacks
 - ecs::Registry, 112
- m_onEntityCreatedCallbacks
 - ecs::Registry, 112
- m_pausedDuration
 - utl::Clock, 57
- m_pausedTime
 - utl::Clock, 57
- m_registry
 - eng::Engine, 66
- m_renderer
 - eng::AudioSystem, 47
 - eng::Engine, 66
 - eng::FontSystem, 76
 - eng::TextSyStem, 145
- m_scenes
 - eng::SceneManager, 118
- m_start
 - utl::Clock, 58
- m_systems
 - eng::Engine, 66
- main
 - main.cpp, 170, 171
- main.cpp
 - main, 170, 171
- musics
 - eng::SFMLAudio::Impl, 84
- N
 - eng, 19
- name
 - eng::Audio, 41
 - eng::Font, 71

- eng::Text, 137
- None
 - eng, 18
- now
 - utl::Clock, 55
- Num0
 - eng, 19
- Num1
 - eng, 19
- Num2
 - eng, 19
- Num3
 - eng, 19
- Num4
 - eng, 19
- Num5
 - eng, 19
- Num6
 - eng, 19
- Num7
 - eng, 19
- Num8
 - eng, 19
- Num9
 - eng, 19
- O
 - eng, 19
- onComponentAdded
 - ecs::Registry, 110
- onEntityCreated
 - ecs::Registry, 111
- operator<<
 - utl::Clock, 57
- operator=
 - cli::ArgsHandler, 34
 - cli::Client, 50
 - ecs::AudioSystem, 43
 - ecs::FontSystem, 72
 - ecs::Registry, 111
 - ecs::TextSyStem, 139
 - eng::AudioSystem, 47
 - eng::Engine, 65
 - eng::FontSystem, 76
 - eng::SceneManager, 118
 - eng::SFMLAudio, 124
 - eng::SFMLRenderer, 130
 - eng::TextSyStem, 144
 - srv::ArgsHandler, 37
 - srv::Server, 120
 - utl::Clock, 55
 - utl::Logger, 100
- P
 - eng, 19
- ParseArgs
 - cli::ArgsHandler, 34
 - srv::ArgsHandler, 37
- ParseEnv
 - cli::ArgsHandler, 35
 - srv::ArgsHandler, 38
- path
 - ecs::Audio, 40
 - ecs::Font, 69
 - ecs::Sprite, 133
 - eng::Audio, 41
 - eng::Font, 71
- pause
 - utl::Clock, 56
- pImpl
 - eng::SFMLAudio, 125
- playAudio
 - eng::IAudio, 78
 - eng::SFMLAudio, 124
- pollEvent
 - eng::IRenderer, 92
 - eng::SFMLRenderer, 130
- port
 - srv::ArgsConfig, 32
- PROJECT_NAME
 - Version.hpp, 159, 161
- PROJECT_VERSION
 - Version.hpp, 159, 161
- PROJECT_VERSION_MAJOR
 - Version.hpp, 159, 161
- PROJECT_VERSION_MINOR
 - Version.hpp, 159, 161
- PROJECT_VERSION_PATCH
 - Version.hpp, 159, 161
- Q
 - eng, 19
- R
 - eng, 19
- r
 - ecs::Color, 59
 - eng::Color, 61
- r-type, 1
- readFile
 - utl, 21
- Registry
 - ecs::Registry, 106, 107
- remove
 - ecs::Registry::IPool, 88
 - ecs::Registry::Pool< T >, 104
- removeComponent
 - ecs::Registry, 111
- restart
 - utl::Clock, 56
- resume
 - utl::Clock, 56
- Right
 - eng, 19
- rotation
 - ecs::Transform, 146
- S

- eng, [19](#)
 - scancodeToKey
 - SFMLRenderer.cpp, [210](#)
 - Scene
 - eng::Scene, [114](#)
 - scene_id_t
 - eng, [18](#)
 - SceneManager
 - eng::SceneManager, [117](#)
 - Server
 - srv::Server, [119](#), [120](#)
 - setDisplay
 - eng::Scene, [114](#)
 - setDrawCallback
 - ecs::TextSyStem, [140](#)
 - setEnabled
 - eng::AudioSystem, [47](#)
 - eng::FontSystem, [76](#)
 - eng::TextSyStem, [144](#)
 - setFrameLimit
 - eng::IRenderer, [92](#)
 - eng::SFMLRenderer, [131](#)
 - setLoop
 - eng::IAudio, [79](#)
 - eng::SFMLAudio, [124](#)
 - setName
 - cli::AGameClient, [25](#)
 - cli::IGameClient, [80](#)
 - eng::Scene, [115](#)
 - srv::AGameServer, [28](#)
 - srv::IGameServer, [82](#)
 - setTextColor
 - eng::IRenderer, [92](#)
 - eng::SFMLRenderer, [131](#)
 - setTextContent
 - eng::IRenderer, [92](#)
 - eng::SFMLRenderer, [131](#)
 - setTextPosition
 - eng::IRenderer, [93](#)
 - eng::SFMLRenderer, [131](#)
 - setVolume
 - eng::IAudio, [79](#)
 - eng::SFMLAudio, [124](#)
 - SFMLAudio
 - eng::SFMLAudio, [123](#)
 - SFMLRenderer
 - eng::SFMLRenderer, [128](#), [129](#)
 - SFMLRenderer.cpp
 - scancodeToKey, [210](#)
 - size
 - eng::Text, [137](#)
 - Space
 - eng, [19](#)
 - srv, [20](#)
 - json, [20](#)
 - srv::AGameServer, [26](#)
 - ~AGameServer, [28](#)
 - getName, [28](#)
 - m_name, [28](#)
 - setName, [28](#)
 - srv::ArgsConfig, [31](#)
 - exit, [32](#)
 - fromFile, [32](#)
 - host, [32](#)
 - port, [32](#)
 - srv::ArgsHandler, [36](#)
 - ~ArgsHandler, [37](#)
 - ArgsHandler, [37](#)
 - operator=, [37](#)
 - ParseArgs, [37](#)
 - ParseEnv, [38](#)
 - srv::EnvConfig, [67](#)
 - srv::IGameServer, [81](#)
 - ~IGameServer, [82](#)
 - getName, [82](#)
 - setName, [82](#)
 - srv::INetworkServer, [86](#)
 - ~INetworkServer, [87](#)
 - srv::Server, [119](#)
 - ~Server, [120](#)
 - operator=, [120](#)
 - Server, [119](#), [120](#)
 - switchToScene
 - eng::SceneManager, [118](#)
- T
- eng, [19](#)
- texts
 - eng::SFMLRenderer::Impl, [85](#)
- TextSyStem
 - ecs::TextSyStem, [139](#)
 - eng::TextSyStem, [143](#)
- TimePoint
 - utl::Clock, [53](#)
- type
 - eng::Event, [68](#)
- U
- eng, [19](#)
- Unknown
 - eng, [19](#)
- Up
 - eng, [19](#)
- update
 - ecs::AudioSystem, [43](#)
 - ecs::FontSystem, [73](#)
 - ecs::TextSyStem, [140](#)
 - eng::AudioSystem, [47](#)
 - eng::FontSystem, [76](#)
 - eng::ISystem, [95](#)
 - eng::TextSyStem, [144](#)
- updateSystems
 - eng::Engine, [65](#)
- utl, [20](#)
 - INFO, [21](#)
 - LogLevel, [20](#)
 - readFile, [21](#)

- WARNING, [21](#)
- utl::Clock, [51](#)
 - ~Clock, [53](#)
 - Clock, [53](#), [54](#)
 - Duration, [53](#)
 - getDeltaSeconds, [54](#)
 - getElapsed, [54](#)
 - m_isPaused, [57](#)
 - m_pausedDuration, [57](#)
 - m_pausedTime, [57](#)
 - m_start, [58](#)
 - now, [55](#)
 - operator<<, [57](#)
 - operator=, [55](#)
 - pause, [56](#)
 - restart, [56](#)
 - resume, [56](#)
 - TimePoint, [53](#)
- utl::Logger, [95](#)
 - ~Logger, [98](#)
 - COLOR_ERROR, [97](#)
 - COLOR_INFO, [97](#)
 - COLOR_RESET, [97](#)
 - COLOR_WARNING, [97](#)
 - ColorIndex, [97](#)
 - formatLogMessage, [98](#)
 - getColorForDuration, [98](#)
 - init, [98](#)
 - log, [99](#)
 - LOG_LEVEL_COLOR, [101](#)
 - LOG_LEVEL_STRING, [101](#)
 - logExecutionTime, [100](#)
 - Logger, [97](#)
 - operator=, [100](#)
- V
 - eng, [19](#)
- Version.hpp
 - BUILD_TYPE, [158](#), [161](#)
 - GIT_COMMIT_HASH, [158](#), [161](#)
 - GIT_TAG, [158](#), [161](#)
 - PROJECT_NAME, [159](#), [161](#)
 - PROJECT_VERSION, [159](#), [161](#)
 - PROJECT_VERSION_MAJOR, [159](#), [161](#)
 - PROJECT_VERSION_MINOR, [159](#), [161](#)
 - PROJECT_VERSION_PATCH, [159](#), [161](#)
- VERSION_MESSAGE
 - argsHandler.cpp, [163](#), [166](#)
- volume
 - ecs::Audio, [40](#)
 - eng::Audio, [41](#)
- W
 - eng, [19](#)
- WARNING
 - utl, [21](#)
- width
 - cli::ArgsConfig, [30](#)
- window
 - eng::SFMLRenderer::Impl, [85](#)
- windowIsOpen
 - eng::IRenderer, [93](#)
 - eng::SFMLRenderer, [132](#)
- X
 - eng, [19](#)
- x
 - ecs::Transform, [146](#)
 - ecs::Velocity, [147](#)
 - eng::Text, [137](#)
- Y
 - eng, [19](#)
- y
 - ecs::Transform, [146](#)
 - ecs::Velocity, [148](#)
 - eng::Text, [137](#)
- Z
 - eng, [19](#)