

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 Build and Run	2
1.1.3.1 Prerequisites	2
1.1.3.2 Unix (Linux, macOS)	2
1.1.3.3 Windows	2
1.1.4 Documentation	2
1.1.5 External Libraries	2
1.1.6 Contributing	2
2 Namespace Index	3
2.1 Namespace List	3
3 Hierarchical Index	5
3.1 Class Hierarchy	5
4 Class Index	7
4.1 Class List	7
5 File Index	9
5.1 File List	9
6 Namespace Documentation	11
6.1 cli Namespace Reference	11
6.1.1 Typedef Documentation	11
6.1.1.1 json	11
6.2 cli::Config Namespace Reference	12
6.3 cli::Config::Audio Namespace Reference	12
6.3.1 Variable Documentation	12
6.3.1.1 DEFAULT_AUDIO_MUTED	12
6.3.1.2 DEFAULT_AUDIO_VOLUME	12
6.4 cli::Config::Window Namespace Reference	12
6.4.1 Variable Documentation	12
6.4.1.1 DEFAULT_WINDOW_FRAME_LIMIT	12
6.4.1.2 DEFAULT_WINDOW_FULLSCREEN	13
6.4.1.3 DEFAULT_WINDOW_HEIGHT	13
6.4.1.4 DEFAULT_WINDOW_WIDTH	13
6.5 cli::Path Namespace Reference	13
6.6 cli::Path::Audio Namespace Reference	13
6.6.1 Variable Documentation	13
6.6.1.1 AUDIO_BATTLE_THEME	13
6.6.1.2 AUDIO_COIN	13
6.6.1.3 AUDIO_TITLE	14

6.7 cli::Path::Font Namespace Reference	14
6.7.1 Variable Documentation	14
6.7.1.1 FONTS_RTYPE	14
6.8 cli::Path::Texture Namespace Reference	14
6.8.1 Variable Documentation	14
6.8.1.1 TEXTURE_PLAYER	14
6.9 ecs Namespace Reference	15
6.9.1 Typedef Documentation	15
6.9.1.1 Entity	15
6.9.2 Variable Documentation	15
6.9.2.1 INVALID_ENTITY	15
6.10 eng Namespace Reference	16
6.10.1 Typedef Documentation	17
6.10.1.1 id	17
6.10.2 Enumeration Type Documentation	17
6.10.2.1 EventType	17
6.10.2.2 Key	17
6.10.2.3 State	18
6.10.2.4 Status	18
6.11 gme Namespace Reference	19
6.12 srv Namespace Reference	19
6.12.1 Typedef Documentation	19
6.12.1.1 json	19
6.13 utl Namespace Reference	20
6.13.1 Enumeration Type Documentation	20
6.13.1.1 LogLevel	20
6.13.2 Function Documentation	20
6.13.2.1 readFile()	20
7 Class Documentation	21
7.1 gme::AGameClient Class Reference	21
7.1.1 Detailed Description	24
7.1.2 Constructor & Destructor Documentation	24
7.1.2.1 ~AGameClient()	24
7.1.3 Member Function Documentation	24
7.1.3.1 getName()	24
7.1.3.2 setName()	24
7.1.4 Member Data Documentation	25
7.1.4.1 m_name	25
7.2 gme::AGameServer Class Reference	25
7.2.1 Detailed Description	26
7.2.2 Constructor & Destructor Documentation	27
7.2.2.1 ~AGameServer()	27

7.2.3 Member Function Documentation	27
7.2.3.1 getName()	27
7.2.3.2 setName()	27
7.2.4 Member Data Documentation	27
7.2.4.1 m_name	27
7.3 cli::ArgsConfig Struct Reference	28
7.3.1 Detailed Description	28
7.3.2 Member Function Documentation	28
7.3.2.1 fromFile()	28
7.3.3 Member Data Documentation	29
7.3.3.1 exit	29
7.3.3.2 frameLimit	29
7.3.3.3 fullscreen	29
7.3.3.4 height	29
7.3.3.5 width	30
7.4 srv::ArgsConfig Struct Reference	30
7.4.1 Detailed Description	31
7.4.2 Member Function Documentation	31
7.4.2.1 fromFile()	31
7.4.3 Member Data Documentation	31
7.4.3.1 exit	31
7.4.3.2 host	31
7.4.3.3 port	32
7.5 cli::ArgsHandler Class Reference	32
7.5.1 Detailed Description	33
7.5.2 Constructor & Destructor Documentation	33
7.5.2.1 ArgsHandler() [1/3]	33
7.5.2.2 ~ArgsHandler()	33
7.5.2.3 ArgsHandler() [2/3]	33
7.5.2.4 ArgsHandler() [3/3]	33
7.5.3 Member Function Documentation	33
7.5.3.1 operator=() [1/2]	33
7.5.3.2 operator=() [2/2]	33
7.5.3.3 ParseArgs()	34
7.5.3.4 ParseEnv()	34
7.6 srv::ArgsHandler Class Reference	35
7.6.1 Detailed Description	35
7.6.2 Constructor & Destructor Documentation	36
7.6.2.1 ArgsHandler() [1/3]	36
7.6.2.2 ~ArgsHandler()	36
7.6.2.3 ArgsHandler() [2/3]	36
7.6.2.4 ArgsHandler() [3/3]	36
7.6.3 Member Function Documentation	36

7.6.3.1 operator=() [1/2]	36
7.6.3.2 operator=() [2/2]	36
7.6.3.3 ParseArgs()	37
7.6.3.4 ParseEnv()	37
7.7 eng::AScene Class Reference	38
7.7.1 Detailed Description	40
7.7.2 Constructor & Destructor Documentation	40
7.7.2.1 AScene() [1/3]	40
7.7.2.2 ~AScene()	40
7.7.2.3 AScene() [2/3]	40
7.7.2.4 AScene() [3/3]	40
7.7.3 Member Function Documentation	41
7.7.3.1 getId()	41
7.7.3.2 getName()	41
7.7.3.3 getRegistry()	41
7.7.3.4 operator=() [1/2]	41
7.7.3.5 operator=() [2/2]	41
7.7.3.6 setName()	41
7.7.4 Member Data Documentation	42
7.7.4.1 m_id	42
7.7.4.2 m_name	42
7.7.4.3 m_registry	42
7.7.4.4 s_nextId	42
7.8 eng::ASystem Class Reference	43
7.8.1 Detailed Description	44
7.8.2 Member Function Documentation	44
7.8.2.1 isEnabled()	44
7.8.2.2 setEnable()	44
7.8.3 Member Data Documentation	44
7.8.3.1 m_isEnable	44
7.9 ecs::Audio Struct Reference	45
7.9.1 Detailed Description	46
7.9.2 Member Data Documentation	47
7.9.2.1 loop	47
7.9.2.2 path	47
7.9.2.3 play	47
7.9.2.4 volume	47
7.10 cli::AudioSystem Class Reference	47
7.10.1 Detailed Description	50
7.10.2 Constructor & Destructor Documentation	50
7.10.2.1 AudioSystem() [1/3]	50
7.10.2.2 ~AudioSystem()	50
7.10.2.3 AudioSystem() [2/3]	50

7.10.2.4 AudioSystem() [3/3]	50
7.10.3 Member Function Documentation	51
7.10.3.1 operator==(1/2)	51
7.10.3.2 operator==(2/2)	51
7.10.3.3 update()	51
7.10.4 Member Data Documentation	51
7.10.4.1 m_audio	51
7.11 cli::Client Class Reference	52
7.11.1 Detailed Description	53
7.11.2 Constructor & Destructor Documentation	53
7.11.2.1 Client() [1/3]	53
7.11.2.2 ~Client()	54
7.11.2.3 Client() [2/3]	54
7.11.2.4 Client() [3/3]	54
7.11.3 Member Function Documentation	54
7.11.3.1 handleEvents()	54
7.11.3.2 lobbyScene()	54
7.11.3.3 operator==(1/2)	55
7.11.3.4 operator==(2/2)	55
7.11.3.5 run()	55
7.11.4 Member Data Documentation	55
7.11.4.1 m_engine	55
7.11.4.2 m_game	55
7.11.4.3 m_keysPressed	56
7.12 utl::Clock Class Reference	56
7.12.1 Detailed Description	57
7.12.2 Member Typedef Documentation	57
7.12.2.1 Duration	57
7.12.2.2 TimePoint	58
7.12.3 Constructor & Destructor Documentation	58
7.12.3.1 Clock() [1/3]	58
7.12.3.2 ~Clock()	58
7.12.3.3 Clock() [2/3]	58
7.12.3.4 Clock() [3/3]	58
7.12.4 Member Function Documentation	58
7.12.4.1 getDeltaSeconds()	58
7.12.4.2 getElapsed()	59
7.12.4.3 now()	59
7.12.4.4 operator==(1/2)	60
7.12.4.5 operator==(2/2)	60
7.12.4.6 pause()	60
7.12.4.7 restart()	60
7.12.4.8 resume()	61

7.12.5 Friends And Related Symbol Documentation	61
7.12.5.1 operator<<	61
7.12.6 Member Data Documentation	61
7.12.6.1 m_isPaused	61
7.12.6.2 m_pausedDuration	61
7.12.6.3 m_pausedTime	62
7.12.6.4 m_start	62
7.13 ecs::Color Struct Reference	62
7.13.1 Detailed Description	63
7.13.2 Member Data Documentation	64
7.13.2.1 a	64
7.13.2.2 b	64
7.13.2.3 g	64
7.13.2.4 r	64
7.14 eng::Color Struct Reference	64
7.14.1 Detailed Description	65
7.14.2 Member Data Documentation	65
7.14.2.1 a	65
7.14.2.2 b	65
7.14.2.3 g	65
7.14.2.4 r	65
7.15 eng::Engine Class Reference	66
7.15.1 Detailed Description	67
7.15.2 Constructor & Destructor Documentation	67
7.15.2.1 Engine() [1/3]	67
7.15.2.2 ~Engine()	68
7.15.2.3 Engine() [2/3]	68
7.15.2.4 Engine() [3/3]	68
7.15.3 Member Function Documentation	68
7.15.3.1 addSystem()	68
7.15.3.2 getAudio()	68
7.15.3.3 getClock()	69
7.15.3.4 getNetwork()	69
7.15.3.5 getRenderer()	69
7.15.3.6 getSceneManager()	69
7.15.3.7 getState()	70
7.15.3.8 operator=() [1/2]	70
7.15.3.9 operator=() [2/2]	70
7.15.3.10 render()	70
7.15.3.11 setState()	70
7.15.3.12 stop()	71
7.15.3.13 updateSystems()	71
7.15.4 Member Data Documentation	71

7.15.4.1	m_audio	71
7.15.4.2	m_clock	71
7.15.4.3	m_network	71
7.15.4.4	m_renderer	71
7.15.4.5	m_sceneManager	72
7.15.4.6	m_state	72
7.15.4.7	m_systems	72
7.16	ecs::Registry::EntityBuilder Class Reference	72
7.16.1	Detailed Description	74
7.16.2	Constructor & Destructor Documentation	74
7.16.2.1	EntityBuilder()	74
7.16.3	Member Function Documentation	74
7.16.3.1	build()	74
7.16.3.2	with()	74
7.16.4	Member Data Documentation	74
7.16.4.1	m_entity	74
7.16.4.2	m_registry	75
7.17	cli::EnvConfig Struct Reference	75
7.17.1	Detailed Description	75
7.18	srv::EnvConfig Struct Reference	75
7.18.1	Detailed Description	76
7.19	eng::Event Struct Reference	76
7.19.1	Detailed Description	76
7.19.2	Member Data Documentation	76
7.19.2.1	key	76
7.19.2.2	type	77
7.20	ecs::Font Struct Reference	77
7.20.1	Detailed Description	78
7.20.2	Member Data Documentation	79
7.20.2.1	path	79
7.21	eng::IAudio Class Reference	79
7.21.1	Detailed Description	80
7.21.2	Constructor & Destructor Documentation	80
7.21.2.1	~IAudio()	80
7.21.3	Member Function Documentation	80
7.21.3.1	createAudio()	80
7.21.3.2	isPlaying()	81
7.21.3.3	playAudio()	81
7.21.3.4	setLoop()	81
7.21.3.5	setVolume()	82
7.21.3.6	stopAudio()	82
7.22	ecs::IComponent Struct Reference	83
7.22.1	Detailed Description	83

7.22.2 Member Data Documentation	84
7.22.2.1 id	84
7.23 gme::IGameClient Class Reference	84
7.23.1 Detailed Description	86
7.23.2 Constructor & Destructor Documentation	86
7.23.2.1 ~IGameClient()	86
7.23.3 Member Function Documentation	86
7.23.3.1 getCurrentScene()	86
7.23.3.2 getName()	87
7.23.3.3 setName()	87
7.23.3.4 update()	87
7.24 gme::IGameServer Class Reference	87
7.24.1 Detailed Description	89
7.24.2 Constructor & Destructor Documentation	89
7.24.2.1 ~IGameServer()	89
7.24.3 Member Function Documentation	89
7.24.3.1 getName()	89
7.24.3.2 setName()	89
7.25 eng::SFMLAudio::Impl Struct Reference	90
7.25.1 Detailed Description	90
7.25.2 Member Data Documentation	91
7.25.2.1 musics	91
7.26 eng::SFMLRenderer::Impl Struct Reference	91
7.26.1 Detailed Description	91
7.26.2 Member Data Documentation	92
7.26.2.1 fonts	92
7.26.2.2 sprites	92
7.26.2.3 texts	92
7.26.2.4 textures	92
7.26.2.5 window	92
7.27 eng::INetworkClient Class Reference	92
7.27.1 Detailed Description	93
7.27.2 Constructor & Destructor Documentation	93
7.27.2.1 ~INetworkClient()	93
7.28 srv::INetworkServer Class Reference	93
7.28.1 Detailed Description	93
7.28.2 Constructor & Destructor Documentation	94
7.28.2.1 ~INetworkServer()	94
7.29 ecs::Registry::IPool Class Reference	94
7.29.1 Detailed Description	95
7.29.2 Constructor & Destructor Documentation	95
7.29.2.1 ~IPool()	95
7.29.3 Member Function Documentation	95

7.29.3.1 remove()	95
7.30 eng::IRenderer Class Reference	95
7.30.1 Detailed Description	98
7.30.2 Constructor & Destructor Documentation	98
7.30.2.1 ~IRenderer()	98
7.30.3 Member Function Documentation	98
7.30.3.1 clearWindow()	98
7.30.3.2 closeWindow()	98
7.30.3.3 createFont()	98
7.30.3.4 createSprite()	99
7.30.3.5 createText()	99
7.30.3.6 createTexture()	99
7.30.3.7 createWindow()	99
7.30.3.8 displayWindow()	99
7.30.3.9 drawPoint()	100
7.30.3.10 drawSprite()	100
7.30.3.11 drawText()	101
7.30.3.12 getWindowSize()	101
7.30.3.13 pollEvent()	101
7.30.3.14 setFrameLimit()	101
7.30.3.15 setSpriteFrame()	101
7.30.3.16 setSpritePosition()	102
7.30.3.17 setSpriteScale()	102
7.30.3.18 setSpriteTexture()	102
7.30.3.19 setTextColor()	103
7.30.3.20 setTextContent()	103
7.30.3.21 setTextPosition()	103
7.30.3.22 windowIsOpen()	104
7.31 eng::IScene Class Reference	104
7.31.1 Detailed Description	106
7.31.2 Constructor & Destructor Documentation	106
7.31.2.1 ~IScene()	106
7.31.3 Member Function Documentation	106
7.31.3.1 event()	106
7.31.3.2 getId()	107
7.31.3.3 getName()	107
7.31.3.4 getRegistry()	107
7.31.3.5 setName()	107
7.31.3.6 update()	107
7.32 gme::IScene Class Reference	108
7.32.1 Detailed Description	109
7.32.2 Constructor & Destructor Documentation	109
7.32.2.1 ~IScene()	109

7.32.3 Member Function Documentation	109
7.32.3.1 getEntities()	109
7.32.3.2 getEntitiesMutable()	109
7.32.3.3 getName()	109
7.33 eng::ISystem Class Reference	110
7.33.1 Detailed Description	111
7.33.2 Constructor & Destructor Documentation	111
7.33.2.1 ~ISystem()	111
7.33.3 Member Function Documentation	111
7.33.3.1 isEnabled()	111
7.33.3.2 setEnable()	111
7.33.3.3 update()	111
7.34 cli::Lobby Class Reference	112
7.34.1 Detailed Description	114
7.34.2 Constructor & Destructor Documentation	114
7.34.2.1 Lobby() [1/3]	114
7.34.2.2 ~Lobby()	115
7.34.2.3 Lobby() [2/3]	115
7.34.2.4 Lobby() [3/3]	115
7.34.3 Member Function Documentation	115
7.34.3.1 event()	115
7.34.3.2 operator=() [1/2]	115
7.34.3.3 operator=() [2/2]	115
7.34.3.4 update()	115
7.34.4 Member Data Documentation	116
7.34.4.1 m_fpsEntity	116
7.34.4.2 m_keysPressed	116
7.34.4.3 m_playerEntity	116
7.35 gme::LobbyScene Class Reference	116
7.35.1 Detailed Description	119
7.35.2 Constructor & Destructor Documentation	119
7.35.2.1 LobbyScene()	119
7.35.3 Member Function Documentation	119
7.35.3.1 addEntity()	119
7.35.3.2 getEntities()	119
7.35.3.3 getEntitiesMutable()	120
7.35.3.4 getName()	120
7.35.4 Member Data Documentation	120
7.35.4.1 m_entities	120
7.35.4.2 m_name	120
7.36 utl::Logger Class Reference	121
7.36.1 Detailed Description	122
7.36.2 Member Enumeration Documentation	122

7.36.2.1 ColorIndex	122
7.36.3 Constructor & Destructor Documentation	122
7.36.3.1 Logger() [1/3]	122
7.36.3.2 Logger() [2/3]	123
7.36.3.3 Logger() [3/3]	123
7.36.3.4 ~Logger()	123
7.36.4 Member Function Documentation	123
7.36.4.1 formatLogMessage()	123
7.36.4.2 getColorForDuration()	124
7.36.4.3 init()	124
7.36.4.4 log()	125
7.36.4.5 logExecutionTime()	125
7.36.4.6 operator=() [1/2]	126
7.36.4.7 operator=() [2/2]	126
7.36.5 Member Data Documentation	126
7.36.5.1 LOG_LEVEL_COLOR	126
7.36.5.2 LOG_LEVEL_STRING	126
7.37 ecs::Mob Struct Reference	127
7.37.1 Detailed Description	128
7.38 ecs::Pixel Struct Reference	129
7.38.1 Detailed Description	130
7.39 cli::PixelSystem Class Reference	131
7.39.1 Detailed Description	133
7.39.2 Constructor & Destructor Documentation	133
7.39.2.1 PixelSystem() [1/3]	133
7.39.2.2 ~PixelSystem()	133
7.39.2.3 PixelSystem() [2/3]	133
7.39.2.4 PixelSystem() [3/3]	133
7.39.3 Member Function Documentation	133
7.39.3.1 operator=() [1/2]	133
7.39.3.2 operator=() [2/2]	134
7.39.3.3 update()	134
7.39.4 Member Data Documentation	134
7.39.4.1 m_renderer	134
7.40 ecs::Player Struct Reference	135
7.40.1 Detailed Description	136
7.40.2 Member Data Documentation	137
7.40.2.1 is_cli	137
7.41 ecs::Registry::Pool< T > Class Template Reference	137
7.41.1 Detailed Description	139
7.41.2 Member Function Documentation	139
7.41.2.1 add()	139
7.41.2.2 get()	139

7.41.2.3	has()	139
7.41.2.4	remove()	139
7.41.3	Member Data Documentation	140
7.41.3.1	data	140
7.42	ecs::Rect Struct Reference	140
7.42.1	Detailed Description	141
7.42.2	Member Data Documentation	142
7.42.2.1	pos_x	142
7.42.2.2	pos_y	142
7.42.2.3	size_x	142
7.42.2.4	size_y	142
7.43	ecs::Registry Class Reference	142
7.43.1	Detailed Description	144
7.43.2	Constructor & Destructor Documentation	144
7.43.2.1	Registry() [1/3]	144
7.43.2.2	~Registry()	144
7.43.2.3	Registry() [2/3]	144
7.43.2.4	Registry() [3/3]	144
7.43.3	Member Function Documentation	145
7.43.3.1	addComponent()	145
7.43.3.2	createEntity()	145
7.43.3.3	getAll()	146
7.43.3.4	getComponent()	146
7.43.3.5	getPool()	147
7.43.3.6	hasComponent()	148
7.43.3.7	onComponentAdded()	148
7.43.3.8	operator=() [1/2]	148
7.43.3.9	operator=() [2/2]	148
7.43.3.10	removeComponent()	148
7.43.4	Member Data Documentation	149
7.43.4.1	m_components	149
7.43.4.2	m_entities	149
7.43.4.3	m_lastEntity	149
7.43.4.4	m_onComponentAddedCallbacks	149
7.44	gme::RTypeClient Class Reference	150
7.44.1	Detailed Description	152
7.44.2	Constructor & Destructor Documentation	152
7.44.2.1	RTypeClient() [1/3]	152
7.44.2.2	~RTypeClient()	152
7.44.2.3	RTypeClient() [2/3]	152
7.44.2.4	RTypeClient() [3/3]	153
7.44.3	Member Function Documentation	153
7.44.3.1	getCurrentScene()	153

7.44.3.2	<code>operator=()</code> [1/2]	153
7.44.3.3	<code>operator=()</code> [2/2]	153
7.44.3.4	<code>update()</code>	153
7.44.4	Member Data Documentation	153
7.44.4.1	<code>m_currentScene</code>	153
7.45	<code>gme::RTypeServer</code> Class Reference	154
7.45.1	Detailed Description	155
7.45.2	Constructor & Destructor Documentation	156
7.45.2.1	<code>RTypeServer()</code> [1/3]	156
7.45.2.2	<code>~RTypeServer()</code>	156
7.45.2.3	<code>RTypeServer()</code> [2/3]	156
7.45.2.4	<code>RTypeServer()</code> [3/3]	156
7.45.3	Member Function Documentation	156
7.45.3.1	<code>operator=()</code> [1/2]	156
7.45.3.2	<code>operator=()</code> [2/2]	156
7.46	<code>ecs::Scale</code> Struct Reference	157
7.46.1	Detailed Description	158
7.46.2	Member Data Documentation	159
7.46.2.1	<code>x</code>	159
7.46.2.2	<code>y</code>	159
7.47	<code>eng::SceneManager</code> Class Reference	159
7.47.1	Detailed Description	161
7.47.2	Constructor & Destructor Documentation	161
7.47.2.1	<code>SceneManager()</code> [1/3]	161
7.47.2.2	<code>~SceneManager()</code>	161
7.47.2.3	<code>SceneManager()</code> [2/3]	161
7.47.2.4	<code>SceneManager()</code> [3/3]	161
7.47.3	Member Function Documentation	161
7.47.3.1	<code>addScene()</code>	161
7.47.3.2	<code>getCurrentScene()</code>	162
7.47.3.3	<code>getScene()</code>	162
7.47.3.4	<code>operator=()</code> [1/2]	162
7.47.3.5	<code>operator=()</code> [2/2]	162
7.47.3.6	<code>switchToScene()</code>	162
7.47.4	Member Data Documentation	162
7.47.4.1	<code>m_currentSceneId</code>	162
7.47.4.2	<code>m_scenes</code>	163
7.48	<code>srv::Server</code> Class Reference	163
7.48.1	Detailed Description	163
7.48.2	Constructor & Destructor Documentation	164
7.48.2.1	<code>Server()</code> [1/3]	164
7.48.2.2	<code>~Server()</code>	164
7.48.2.3	<code>Server()</code> [2/3]	164

7.48.2.4	Server() [3/3]	164
7.48.3	Member Function Documentation	164
7.48.3.1	operator=() [1/2]	164
7.48.3.2	operator=() [2/2]	164
7.49	eng::SFMLAudio Class Reference	165
7.49.1	Detailed Description	167
7.49.2	Constructor & Destructor Documentation	167
7.49.2.1	SFMLAudio() [1/3]	167
7.49.2.2	~SFMLAudio()	167
7.49.2.3	SFMLAudio() [2/3]	167
7.49.2.4	SFMLAudio() [3/3]	167
7.49.3	Member Function Documentation	168
7.49.3.1	createAudio()	168
7.49.3.2	isPlaying()	168
7.49.3.3	operator=() [1/2]	168
7.49.3.4	operator=() [2/2]	168
7.49.3.5	playAudio()	168
7.49.3.6	setLoop()	169
7.49.3.7	setVolume()	169
7.49.3.8	stopAudio()	169
7.49.4	Member Data Documentation	169
7.49.4.1	pImpl	169
7.50	eng::SFMLRenderer Class Reference	170
7.50.1	Detailed Description	172
7.50.2	Constructor & Destructor Documentation	172
7.50.2.1	SFMLRenderer() [1/3]	172
7.50.2.2	~SFMLRenderer()	173
7.50.2.3	SFMLRenderer() [2/3]	173
7.50.2.4	SFMLRenderer() [3/3]	173
7.50.3	Member Function Documentation	173
7.50.3.1	clearWindow()	173
7.50.3.2	closeWindow()	173
7.50.3.3	createFont()	173
7.50.3.4	createSprite()	174
7.50.3.5	createText()	174
7.50.3.6	createTexture()	174
7.50.3.7	createWindow()	174
7.50.3.8	displayWindow()	175
7.50.3.9	drawPoint()	175
7.50.3.10	drawSprite()	175
7.50.3.11	drawText()	175
7.50.3.12	getWindowSize()	175
7.50.3.13	operator=() [1/2]	176

7.50.3.14	<code>operator=()</code> [2/2]	176
7.50.3.15	<code>pollEvent()</code>	176
7.50.3.16	<code>setFrameLimit()</code>	176
7.50.3.17	<code>setSpriteFrame()</code>	176
7.50.3.18	<code>setSpritePosition()</code>	177
7.50.3.19	<code>setSpriteScale()</code>	177
7.50.3.20	<code>setSpriteTexture()</code>	177
7.50.3.21	<code>setTextColor()</code>	177
7.50.3.22	<code>setTextContent()</code>	177
7.50.3.23	<code>setTextPosition()</code>	178
7.50.3.24	<code>windowIsOpen()</code>	178
7.50.4	Member Data Documentation	178
7.50.4.1	<code>m_impl</code>	178
7.51	<code>gme::Sprite</code> Struct Reference	178
7.51.1	Detailed Description	180
7.51.2	Member Data Documentation	180
7.51.2.1	<code>a</code>	180
7.51.2.2	<code>b</code>	180
7.51.2.3	<code>g</code>	180
7.51.2.4	<code>id</code>	180
7.51.2.5	<code>pos_x</code>	180
7.51.2.6	<code>pos_y</code>	181
7.51.2.7	<code>r</code>	181
7.51.2.8	<code>scale_x</code>	181
7.51.2.9	<code>scale_y</code>	181
7.51.2.10	<code>text_rect_fx</code>	181
7.51.2.11	<code>text_rect_fy</code>	181
7.51.2.12	<code>text_rect_x</code>	181
7.51.2.13	<code>text_rect_y</code>	181
7.51.2.14	<code>texture_path</code>	182
7.51.2.15	<code>type</code>	182
7.51.2.16	<code>v_x</code>	182
7.51.2.17	<code>v_y</code>	182
7.52	<code>cli::SpriteSystem</code> Class Reference	182
7.52.1	Detailed Description	185
7.52.2	Constructor & Destructor Documentation	185
7.52.2.1	<code>SpriteSystem()</code> [1/3]	185
7.52.2.2	<code>~SpriteSystem()</code>	185
7.52.2.3	<code>SpriteSystem()</code> [2/3]	185
7.52.2.4	<code>SpriteSystem()</code> [3/3]	185
7.52.3	Member Function Documentation	185
7.52.3.1	<code>operator=()</code> [1/2]	185
7.52.3.2	<code>operator=()</code> [2/2]	186

7.52.3.3	update()	186
7.52.4	Member Data Documentation	186
7.52.4.1	m_renderer	186
7.53	ecs::Text Struct Reference	187
7.53.1	Detailed Description	188
7.53.2	Member Data Documentation	189
7.53.2.1	content	189
7.53.2.2	font_size	189
7.54	eng::Text Struct Reference	189
7.54.1	Detailed Description	190
7.54.2	Member Data Documentation	190
7.54.2.1	color	190
7.54.2.2	content	190
7.54.2.3	font_name	190
7.54.2.4	name	190
7.54.2.5	size	191
7.54.2.6	x	191
7.54.2.7	y	191
7.55	cli::TextSyStem Class Reference	191
7.55.1	Detailed Description	194
7.55.2	Constructor & Destructor Documentation	194
7.55.2.1	TextSyStem() [1/3]	194
7.55.2.2	~TextSyStem()	194
7.55.2.3	TextSyStem() [2/3]	194
7.55.2.4	TextSyStem() [3/3]	194
7.55.3	Member Function Documentation	195
7.55.3.1	operator=() [1/2]	195
7.55.3.2	operator=() [2/2]	195
7.55.3.3	update()	195
7.55.4	Member Data Documentation	195
7.55.4.1	m_renderer	195
7.56	ecs::Texture Struct Reference	196
7.56.1	Detailed Description	197
7.56.2	Member Data Documentation	198
7.56.2.1	path	198
7.57	ecs::Transform Struct Reference	198
7.57.1	Detailed Description	199
7.57.2	Member Data Documentation	200
7.57.2.1	rotation	200
7.57.2.2	x	200
7.57.2.3	y	200
7.58	ecs::Velocity Struct Reference	200
7.58.1	Detailed Description	201

7.58.2 Member Data Documentation	202
7.58.2.1 x	202
7.58.2.2 y	202
7.59 eng::WindowSize Struct Reference	202
7.59.1 Detailed Description	202
7.59.2 Member Data Documentation	203
7.59.2.1 height	203
7.59.2.2 width	203
8 File Documentation	205
8.1 /home/masina/Projects/Epitech/rtype/client/include/Client/ArgsHandler.hpp File Reference	205
8.1.1 Detailed Description	206
8.2 ArgsHandler.hpp	206
8.3 /home/masina/Projects/Epitech/rtype/server/include/Server/ArgsHandler.hpp File Reference	207
8.3.1 Detailed Description	208
8.4 ArgsHandler.hpp	209
8.5 /home/masina/Projects/Epitech/rtype/client/include/Client/Client.hpp File Reference	209
8.5.1 Detailed Description	210
8.6 Client.hpp	211
8.7 /home/masina/Projects/Epitech/rtype/client/include/Client/Common.hpp File Reference	211
8.7.1 Detailed Description	212
8.8 Common.hpp	213
8.9 /home/masina/Projects/Epitech/rtype/client/include/Client/Generated/Version.hpp File Reference	213
8.9.1 Macro Definition Documentation	214
8.9.1.1 BUILD_TYPE	214
8.9.1.2 GIT_COMMIT_HASH	214
8.9.1.3 GIT_TAG	214
8.9.1.4 PROJECT_NAME	214
8.9.1.5 PROJECT_VERSION	214
8.9.1.6 PROJECT_VERSION_MAJOR	215
8.9.1.7 PROJECT_VERSION_MINOR	215
8.9.1.8 PROJECT_VERSION_PATCH	215
8.10 Version.hpp	215
8.11 /home/masina/Projects/Epitech/rtype/server/include/Server/Generated/Version.hpp File Reference	215
8.11.1 Macro Definition Documentation	216
8.11.1.1 BUILD_TYPE	216
8.11.1.2 GIT_COMMIT_HASH	216
8.11.1.3 GIT_TAG	216
8.11.1.4 PROJECT_NAME	216
8.11.1.5 PROJECT_VERSION	216

8.11.1.6 PROJECT_VERSION_MAJOR	216
8.11.1.7 PROJECT_VERSION_MINOR	217
8.11.1.8 PROJECT_VERSION_PATCH	217
8.12 Version.hpp	217
8.13 /home/masina/Projects/Epitech/rtype/client/include/Client/Scenes/Lobby.hpp File Reference	217
8.14 Lobby.hpp	218
8.15 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Systems.hpp File Reference	219
8.15.1 Detailed Description	220
8.16 Systems.hpp	220
8.17 /home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/Systems.hpp File Reference	222
8.17.1 Detailed Description	223
8.18 Systems.hpp	223
8.19 /home/masina/Projects/Epitech/rtype/client/src/argsHandler.cpp File Reference	224
8.19.1 Macro Definition Documentation	224
8.19.1.1 APP_EXTENSION	224
8.19.2 Variable Documentation	225
8.19.2.1 HELP_MESSAGE	225
8.19.2.2 VERSION_MESSAGE	225
8.20 argsHandler.cpp	225
8.21 /home/masina/Projects/Epitech/rtype/server/src/argsHandler.cpp File Reference	227
8.21.1 Macro Definition Documentation	227
8.21.1.1 APP_EXTENSION	227
8.21.2 Variable Documentation	227
8.21.2.1 HELP_MESSAGE	227
8.21.2.2 VERSION_MESSAGE	228
8.22 argsHandler.cpp	228
8.23 /home/masina/Projects/Epitech/rtype/client/src/client.cpp File Reference	229
8.23.1 Variable Documentation	230
8.23.1.1 DARK	230
8.24 client.cpp	230
8.25 /home/masina/Projects/Epitech/rtype/client/src/event.cpp File Reference	231
8.26 event.cpp	231
8.27 /home/masina/Projects/Epitech/rtype/client/src/main.cpp File Reference	231
8.27.1 Function Documentation	232
8.27.1.1 main()	232
8.28 main.cpp	233
8.29 /home/masina/Projects/Epitech/rtype/server/src/main.cpp File Reference	233
8.29.1 Function Documentation	234
8.29.1.1 main()	234
8.30 main.cpp	234
8.31 /home/masina/Projects/Epitech/rtype/client/src/scenes/lobby.cpp File Reference	235

8.31.1 Variable Documentation	235
8.31.1.1 WHITE	235
8.32 lobby.cpp	235
8.33 /home/masina/Projects/Epitech/rtype/modules/Audio/SFMLAudio/include/SFMLAudio/↔ SFMLAudio.hpp File Reference	238
8.33.1 Detailed Description	239
8.34 SFMLAudio.hpp	239
8.35 /home/masina/Projects/Epitech/rtype/modules/Audio/SFMLAudio/src/SFMLAudio.cpp File Reference	239
8.36 SFMLAudio.cpp	240
8.37 /home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.hpp File Reference	241
8.37.1 Detailed Description	242
8.38 Component.hpp	243
8.39 /home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Entity.hpp File Ref- erence	244
8.39.1 Detailed Description	245
8.40 Entity.hpp	245
8.41 /home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Registry.hpp File Reference	245
8.41.1 Detailed Description	247
8.42 Registry.hpp	247
8.43 /home/masina/Projects/Epitech/rtype/modules/ECS/src/registry.cpp File Reference . . .	249
8.44 registry.cpp	249
8.45 /home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/Engine.hpp File Reference	249
8.45.1 Detailed Description	250
8.46 Engine.hpp	250
8.47 /home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/IScene.hpp File Reference	251
8.47.1 Detailed Description	253
8.48 IScene.hpp	253
8.49 /home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/SceneManager.hpp File Reference	254
8.49.1 Detailed Description	255
8.50 SceneManager.hpp	255
8.51 /home/masina/Projects/Epitech/rtype/modules/Engine/src/Engine.cpp File Reference . .	256
8.52 Engine.cpp	257
8.53 /home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Client/include/R-Type↔ Client/LobbyScene.hpp File Reference	257
8.53.1 Detailed Description	258
8.54 LobbyScene.hpp	258
8.55 /home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Client/include/R-Type↔ Client/RTypeClient.hpp File Reference	259
8.55.1 Detailed Description	260
8.56 RTypeClient.hpp	260

8.57	/home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Client/src/rtype↵ Client.cpp File Reference	261
8.58	rtypeClient.cpp	261
8.59	/home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Server/include/R-Type↵ Server/RTypeServer.hpp File Reference	262
8.59.1	Detailed Description	262
8.60	RTypeServer.hpp	263
8.61	/home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Server/src/rtype↵ Server.cpp File Reference	263
8.62	rtypeServer.cpp	263
8.63	/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/AGame↵ Client.hpp File Reference	263
8.63.1	Detailed Description	265
8.64	AGameClient.hpp	265
8.65	/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/AGame↵ Server.hpp File Reference	265
8.65.1	Detailed Description	266
8.66	AGameServer.hpp	266
8.67	/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IAudio.hpp File Reference	267
8.67.1	Detailed Description	268
8.68	IAudio.hpp	268
8.69	/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IGame↵ Client.hpp File Reference	269
8.69.1	Detailed Description	270
8.70	IGameClient.hpp	270
8.71	/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IGame↵ Server.hpp File Reference	271
8.71.1	Detailed Description	272
8.72	IGameServer.hpp	272
8.73	/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/INetwork↵ Client.hpp File Reference	272
8.73.1	Detailed Description	273
8.74	INetworkClient.hpp	273
8.75	/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/INetwork↵ Server.hpp File Reference	274
8.75.1	Detailed Description	274
8.76	INetworkServer.hpp	274
8.77	/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IRenderer.hpp File Reference	275
8.77.1	Detailed Description	276
8.78	IRenderer.hpp	276
8.79	/home/masina/Projects/Epitech/rtype/modules/Renderer/SFMLRenderer/include/↵ SFMLRenderer/SFMLRenderer.hpp File Reference	278
8.79.1	Detailed Description	279
8.80	SFMLRenderer.hpp	279

8.81	/home/masina/Projects/Epitech/rtype/modules/Renderer/SFMLRenderer/src/SFMLRenderer.cpp File Reference	280
8.81.1	Function Documentation	281
8.81.1.1	scancodeToKey()	281
8.82	SFMLRenderer.cpp	281
8.83	/home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/Clock.hpp File Reference	285
8.83.1	Detailed Description	287
8.84	Clock.hpp	287
8.85	/home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/Logger.hpp File Reference	288
8.86	Logger.hpp	289
8.87	/home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/Utils.hpp File Reference	290
8.87.1	Detailed Description	291
8.88	Utils.hpp	291
8.89	/home/masina/Projects/Epitech/rtype/modules/Utils/src/logger.cpp File Reference	292
8.90	logger.cpp	292
8.91	/home/masina/Projects/Epitech/rtype/modules/Utils/src/utils.cpp File Reference	292
8.92	utils.cpp	293
8.93	/home/masina/Projects/Epitech/rtype/README.md File Reference	293
8.94	/home/masina/Projects/Epitech/rtype/server/include/Server/Server.hpp File Reference	293
8.94.1	Detailed Description	295
8.95	Server.hpp	295
8.96	/home/masina/Projects/Epitech/rtype/server/src/server.cpp File Reference	295
8.97	server.cpp	296
	Index	297

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

```
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
    end

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

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

R-Type
assets          # Game assets (images, sounds, etc.)
cmake            # Cmake configs
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
```

1.1.3 Build and Run

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
```

1.1.3.1 Prerequisites

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

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

1.1.3.2 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
```

1.1.3.3 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
```

1.1.4 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](#)

1.1.5 External Libraries

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

1.1.6 Contributing

Want to contribute? See [CONTRIBUTING.md](#).

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

cli	11
cli::Config	12
cli::Config::Audio	12
cli::Config::Window	12
cli::Path	13
cli::Path::Audio	13
cli::Path::Font	14
cli::Path::Texture	14
ecs	15
eng	16
gme	19
srv	19
utl	20

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

cli::ArgsConfig	28
srv::ArgsConfig	30
cli::ArgsHandler	32
srv::ArgsHandler	35
cli::Client	52
utl::Clock	56
eng::Color	64
eng::Engine	66
ecs::Registry::EntityBuilder	72
cli::EnvConfig	75
srv::EnvConfig	75
eng::Event	76
eng::IAudio	79
eng::SFMLAudio	165
ecs::IComponent	83
ecs::Audio	45
ecs::Color	62
ecs::Font	77
ecs::Mob	127
ecs::Pixel	129
ecs::Player	135
ecs::Rect	140
ecs::Scale	157
ecs::Text	187
ecs::Texture	196
ecs::Transform	198
ecs::Velocity	200
gme::IGameClient	84
gme::AGameClient	21
gme::RTypeClient	150
gme::IGameServer	87
gme::AGameServer	25
gme::RTypeServer	154
eng::SFMLAudio::Impl	90

eng::SFMLRenderer::Impl	91
eng::INetworkClient	92
srv::INetworkServer	93
ecs::Registry::IPool	94
ecs::Registry::Pool< T >	137
eng::IRenderer	95
eng::SFMLRenderer	170
eng::IScene	104
eng::AScene	38
cli::Lobby	112
gme::IScene	108
gme::LobbyScene	116
eng::ISystem	110
eng::ASystem	43
cli::AudioSystem	47
cli::PixelSystem	131
cli::SpriteSystem	182
cli::TextSyStem	191
utl::Logger	121
ecs::Registry	142
eng::SceneManager	159
srv::Server	163
gme::Sprite	178
eng::Text	189
eng::WindowSize	202

Chapter 4

Class Index

4.1 Class List

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

gme::AGameClient	
Abstraction for the games	21
gme::AGameServer	
Abstraction for the games	25
cli::ArgsConfig	28
srv::ArgsConfig	30
cli::ArgsHandler	
Class to handle command line arguments	32
srv::ArgsHandler	
Class to handle command line arguments	35
eng::AScene	
Class for scene	38
eng::ASystem	43
ecs::Audio	45
cli::AudioSystem	
Class for managing entities and their components	47
cli::Client	
Class for the client	52
utl::Clock	
Class for clock	56
ecs::Color	62
eng::Color	64
eng::Engine	
Class for the game engine	66
ecs::Registry::EntityBuilder	72
cli::EnvConfig	75
srv::EnvConfig	75
eng::Event	76
ecs::Font	77
eng::IAudio	
Interface for the audio	79
ecs::IComponent	83
gme::IGameClient	
Interface for the games	84
gme::IGameServer	
Interface for the games	87

eng::SFMLAudio::Impl	90
eng::SFMLRenderer::Impl	91
eng::INetworkClient	
Interface for the client network	92
srv::INetworkServer	
Interface for the server network	93
ecs::Registry::IPool	94
eng::IRenderer	
Interface for the renderer	95
eng::IScene	
Interface class for scene	104
gme::IScene	
Interface for scenes	108
eng::ISystem	110
cli::Lobby	
Lobby scene	112
gme::LobbyScene	
Class for the Lobby scene	116
utl::Logger	121
ecs::Mob	127
ecs::Pixel	129
cli::PixelSystem	131
ecs::Player	135
ecs::Registry::Pool< T >	137
ecs::Rect	140
ecs::Registry	
Class for managing entities and their components	142
gme::RTypeClient	
Class for the R-Type game	150
gme::RTypeServer	
Class for the R-Type game	154
ecs::Scale	157
eng::SceneManager	
Class for managing scenes	159
srv::Server	
Class for the server	163
eng::SFMLAudio	
Class for audio management	165
eng::SFMLRenderer	
Class for the R-Type game	170
gme::Sprite	178
cli::SpriteSystem	182
ecs::Text	187
eng::Text	189
cli::TextSyStem	
Class for managing entities and their components	191
ecs::Texture	196
ecs::Transform	198
ecs::Velocity	200
eng::WindowSize	202

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	205
This file contains the ArgsHandler class declaration	
/home/masina/Projects/Epitech/rtype/client/include/Client/Client.hpp	209
This file contains the Client class declaration	
/home/masina/Projects/Epitech/rtype/client/include/Client/Common.hpp	211
This file contains common definitions and constants	
/home/masina/Projects/Epitech/rtype/client/include/Client/Generated/Version.hpp	213
/home/masina/Projects/Epitech/rtype/client/include/Client/Scenes/Lobby.hpp	217
/home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Systems.hpp	219
This file contains the system definitions	
/home/masina/Projects/Epitech/rtype/client/src/argsHandler.cpp	224
/home/masina/Projects/Epitech/rtype/client/src/client.cpp	229
/home/masina/Projects/Epitech/rtype/client/src/event.cpp	231
/home/masina/Projects/Epitech/rtype/client/src/main.cpp	231
/home/masina/Projects/Epitech/rtype/client/src/scenes/lobby.cpp	235
/home/masina/Projects/Epitech/rtype/modules/Audio/SFMLAudio/include/SFMLAudio/SFMLAudio.hpp	238
SFMLAudio class declaration	
/home/masina/Projects/Epitech/rtype/modules/Audio/SFMLAudio/src/SFMLAudio.cpp	239
/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.hpp	241
This file contains the component definitions	
/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Entity.hpp	244
This file contains the entity definitions	
/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Registry.hpp	245
This file contains the Registry class declaration	
/home/masina/Projects/Epitech/rtype/modules/ECS/src/registry.cpp	249
/home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/Engine.hpp	249
This file contains the Engine class declaration	
/home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/IScene.hpp	251
This file contains the IScene class	
/home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/SceneManager.hpp	254
This file contains the SceneManager class declaration	
/home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/Systems.hpp	222
This file contains the system definitions	
/home/masina/Projects/Epitech/rtype/modules/Engine/src/Engine.cpp	256

/home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Client/include/R-Type↔ Client/ LobbyScene.hpp	
This file contains the lobby scene	257
/home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Client/include/R-Type↔ Client/ RTypeClient.hpp	
RType client class declaration	259
/home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Client/src/ rtypeClient.cpp	261
/home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Server/include/R-Type↔ Server/ RTypeServer.hpp	
RType client class declaration	262
/home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Server/src/ rtypeServer.cpp	263
/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/ AGameClient.hpp	
This file contains the game abstract class	263
/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/ AGameServer.hpp	
This file contains the game abstract class	265
/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/ IAudio.hpp	
This file contains the Audio interface	267
/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/ IGameClient.hpp	
This file contains the Game interface	269
/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/ IGameServer.hpp	
This file contains the Game interface	271
/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/ INetworkClient.hpp	
This file contains the client network interface	272
/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/ INetworkServer.hpp	
This file contains the server network interface	274
/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/ IRenderer.hpp	
This file contains the IRenderer class declaration	275
/home/masina/Projects/Epitech/rtype/modules/Renderer/SFMLRenderer/include/SFMLRenderer/ SFMLRenderer.	
SFMLRenderer class declaration with PImpl	278
/home/masina/Projects/Epitech/rtype/modules/Renderer/SFMLRenderer/src/ SFMLRenderer.cpp	280
/home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/ Clock.hpp	
This file contains the Clock class	285
/home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/ Logger.hpp	288
/home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/ Utils.hpp	
This file contains utility functions	290
/home/masina/Projects/Epitech/rtype/modules/Utils/src/ logger.cpp	292
/home/masina/Projects/Epitech/rtype/modules/Utils/src/ utils.cpp	292
/home/masina/Projects/Epitech/rtype/server/include/Server/ ArgsHandler.hpp	
This file contains the ArgsHandler class declaration	207
/home/masina/Projects/Epitech/rtype/server/include/Server/ Server.hpp	
This file contains the Server class declaration	293
/home/masina/Projects/Epitech/rtype/server/include/Server/Generated/ Version.hpp	215
/home/masina/Projects/Epitech/rtype/server/src/ argsHandler.cpp	227
/home/masina/Projects/Epitech/rtype/server/src/ main.cpp	233
/home/masina/Projects/Epitech/rtype/server/src/ server.cpp	295

Chapter 6

Namespace Documentation

6.1 cli Namespace Reference

Namespaces

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

Classes

- struct [ArgsConfig](#)
- class [ArgsHandler](#)
Class to handle command line arguments.
- class [AudioSystem](#)
Class for managing entities and their components.
- class [Client](#)
Class for the client.
- struct [EnvConfig](#)
- class [Lobby](#)
[Lobby](#) scene.
- class [PixelSystem](#)
- class [SpriteSystem](#)
- class [TextSyStem](#)
Class for managing entities and their components.

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 23 of file [Common.hpp](#).

6.3.1.2 DEFAULT_AUDIO_VOLUME

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

Definition at line 22 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 17 of file [Common.hpp](#).

6.4.1.2 DEFAULT_WINDOW_FULLSCREEN

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

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

6.4.1.3 DEFAULT_WINDOW_HEIGHT

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

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

6.4.1.4 DEFAULT_WINDOW_WIDTH

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

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

6.5 cli::Path Namespace Reference

Namespaces

- namespace [Audio](#)
- namespace [Font](#)
- namespace [Texture](#)

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

Definition at line 32 of file [Common.hpp](#).

6.6.1.2 AUDIO_COIN

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

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

6.6.1.3 AUDIO_TITLE

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

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

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

6.7 cli::Path::Font 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::Path::Font::FONTS_RTYPE = "assets/fonts/r-type.otf" [inline], [constexpr]
```

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

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

6.8 cli::Path::Texture Namespace Reference

Variables

- constexpr auto [TEXTURE_PLAYER](#) = "assets/sprites/r-typesheet42.gif"

6.8.1 Variable Documentation

6.8.1.1 TEXTURE_PLAYER

```
auto cli::Path::Texture::TEXTURE_PLAYER = "assets/sprites/r-typesheet42.gif" [inline], [constexpr]
```

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

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

6.9 ecs Namespace Reference

Classes

- struct [Audio](#)
- struct [Color](#)
- struct [Font](#)
- struct [IComponent](#)
- struct [Mob](#)
- struct [Pixel](#)
- struct [Player](#)
- struct [Rect](#)
- class [Registry](#)
 - Class for managing entities and their components.
- struct [Scale](#)
- struct [Text](#)
- struct [Texture](#)
- struct [Transform](#)
- struct [Velocity](#)

Typedefs

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

Variables

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

6.9.1 Typedef Documentation

6.9.1.1 Entity

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

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

6.9.2 Variable Documentation

6.9.2.1 INVALID_ENTITY

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

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

6.10 eng Namespace Reference

Classes

- class [AScene](#)
Class for scene.
- class [ASystem](#)
- struct [Color](#)
- class [Engine](#)
Class for the game engine.
- struct [Event](#)
- class [IAudio](#)
Interface for the audio.
- class [INetworkClient](#)
Interface for the client network.
- class [IRenderer](#)
Interface for the renderer.
- class [IScene](#)
interface class for scene
- class [ISystem](#)
- class [SceneManager](#)
Class for managing scenes.
- class [SFMLAudio](#)
Class for audio management.
- class [SFMLRenderer](#)
Class for the R-Type game.
- struct [Text](#)
- struct [WindowSize](#)

Typedefs

- using [id](#) = unsigned int

Enumerations

- enum [State](#) : unsigned char { [STOP](#) = 0 , [RUN](#) = 1 , [DEFAULT](#) = 2 }
- enum class [Status](#) { [Stopped](#) , [Paused](#) , [Playing](#) }
- 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](#) }

6.10.1 Typedef Documentation

6.10.1.1 id

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

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

6.10.2 Enumeration Type Documentation

6.10.2.1 EventType

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

Enumerator

Closed	
KeyPressed	
KeyReleased	
None	

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

6.10.2.2 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	

Enumerator

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 31 of file [IRenderer.hpp](#).

6.10.2.3 State

enum [eng::State](#) : unsigned char

Enumerator

STOP	
RUN	
DEFAULT	

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

6.10.2.4 Status

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

Enumerator

Stopped	
Paused	
Playing	

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

6.11 gme Namespace Reference

Classes

- class [AGameClient](#)
Abstraction for the games.
- class [AGameServer](#)
Abstraction for the games.
- class [IGameClient](#)
Interface for the games.
- class [IGameServer](#)
Interface for the games.
- class [IScene](#)
Interface for scenes.
- class [LobbyScene](#)
Class for the Lobby scene.
- class [RTypeClient](#)
Class for the R-Type game.
- class [RTypeServer](#)
Class for the R-Type game.
- struct [Sprite](#)

6.12 srv Namespace Reference

Classes

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

Typedefs

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

6.12.1 Typedef Documentation

6.12.1.1 json

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

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

6.13 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)

6.13.1 Enumeration Type Documentation

6.13.1.1 LogLevel

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

Enumerator

INFO	
WARNING	

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

6.13.2 Function Documentation

6.13.2.1 readFile()

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

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

Chapter 7

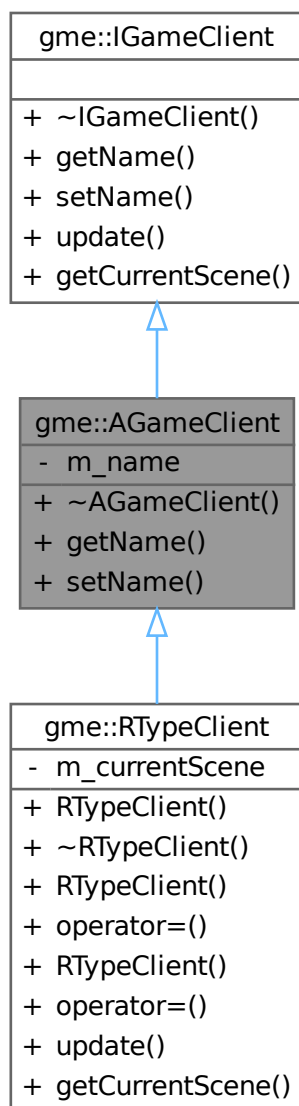
Class Documentation

7.1 gme::AGameClient Class Reference

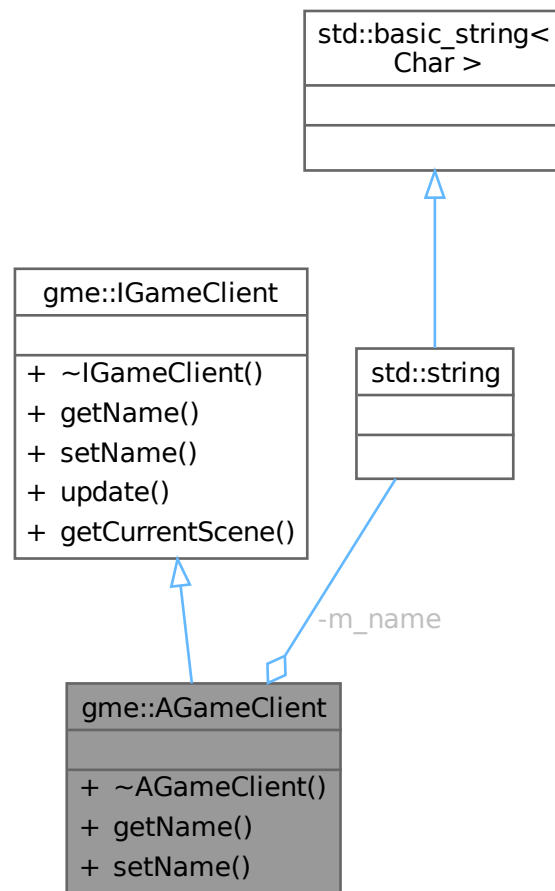
Abstraction for the games.

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

Inheritance diagram for gme::AGameClient:



Collaboration diagram for gme::AGameClient:



Public Member Functions

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

Public Member Functions inherited from `gme::IGameClient`

- virtual `~IGameClient` ()=default
- virtual void `update` (float `deltaTime`, unsigned int `width`, unsigned int `height`)=0
- virtual const `IScene` & `getCurrentScene` () const =0

Private Attributes

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

7.1.1 Detailed Description

Abstraction for the games.

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

7.1.2 Constructor & Destructor Documentation

7.1.2.1 ~AGameClient()

`gme::AGameClient::~~AGameClient ()` [override], [default]

7.1.3 Member Function Documentation

7.1.3.1 getName()

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

Implements [gme::IGameClient](#).

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

References [m_name](#).

7.1.3.2 setName()

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

Implements [gme::IGameClient](#).

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

References [m_name](#).

Referenced by [gme::RTypeClient::RTypeClient\(\)](#).

Here is the caller graph for this function:



7.1.4 Member Data Documentation

7.1.4.1 m_name

`std::string gme::AGameClient::m_name = "default_name" [private]`

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

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

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

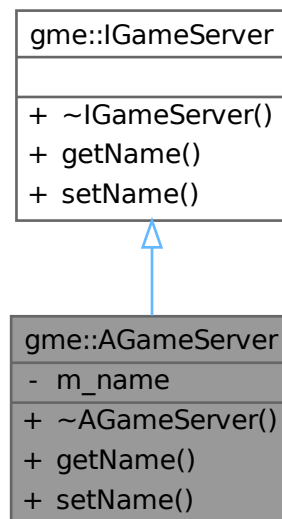
- [/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/AGameClient.hpp](#)

7.2 gme::AGameServer Class Reference

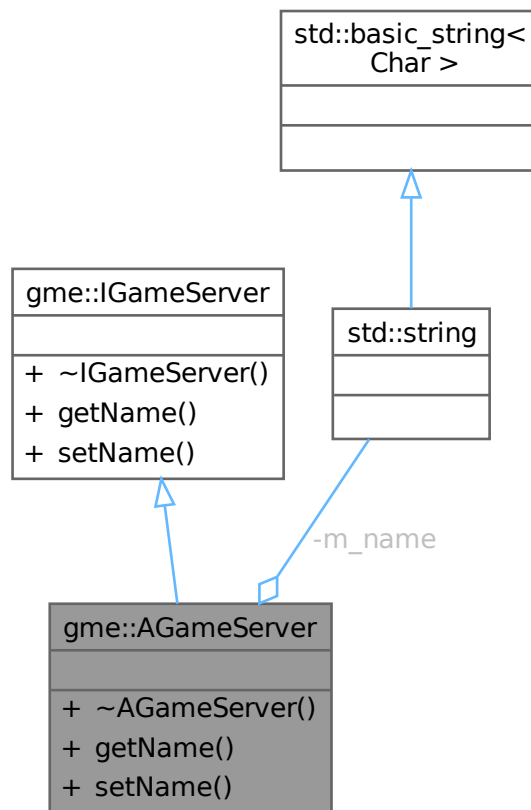
Abstraction for the games.

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

Inheritance diagram for gme::AGameServer:



Collaboration diagram for `gme::AGameServer`:



Public Member Functions

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

Public Member Functions inherited from `gme::IGameServer`

- virtual `~IGameServer()`=default

Private Attributes

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

7.2.1 Detailed Description

Abstraction for the games.

Definition at line 19 of file `AGameServer.hpp`.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 ~AGameServer()

`gme::AGameServer::~~AGameServer ()` [override], [default]

7.2.3 Member Function Documentation

7.2.3.1 getName()

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

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

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

References [m_name](#).

7.2.3.2 setName()

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

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

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

References [m_name](#).

7.2.4 Member Data Documentation

7.2.4.1 m_name

`std::string gme::AGameServer::m_name = "default_name"` [private]

Definition at line 28 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()

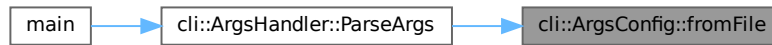
[cli::ArgsConfig](#) cli::ArgsConfig::fromFile (
const std::string & path) [static]

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\(\)](#).

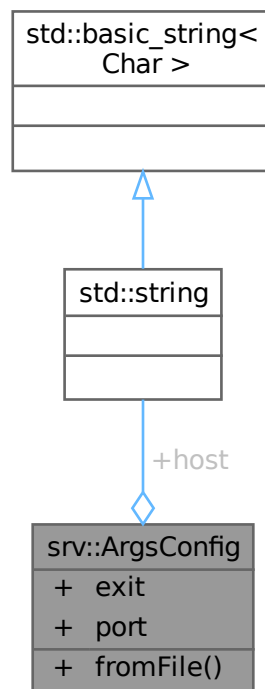
The documentation for this struct was generated from the following files:

- [/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
<div>+ ArgsHandler() + ~ArgsHandler() + ArgsHandler() + operator=() + ArgsHandler() + operator=() + ParseArgs() + ParseEnv()</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[])

7.5.1 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()

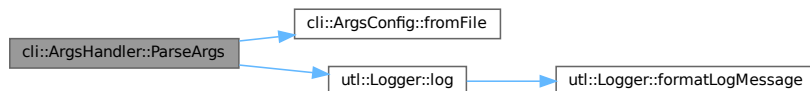
```
cli::ArgsConfig cli::ArgsHandler::ParseArgs (
    int argc,
    const char *const argv[]) [static]
```

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:

srv::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[])

7.6.1 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:

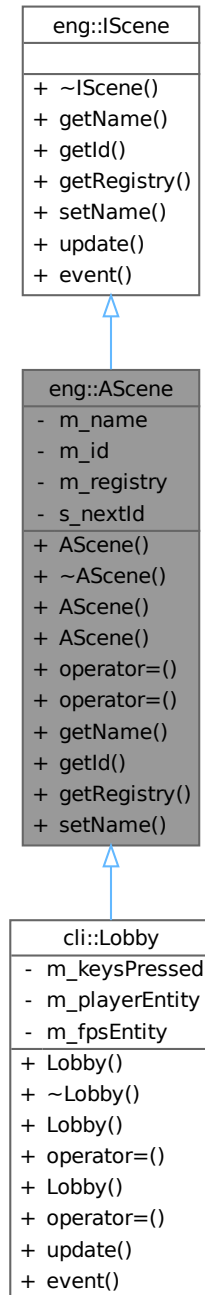
- [/home/masina/Projects/Epitech/rtype/server/include/Server/ArgsHandler.hpp](#)
- [/home/masina/Projects/Epitech/rtype/server/src/argsHandler.cpp](#)

7.7 eng::AScene Class Reference

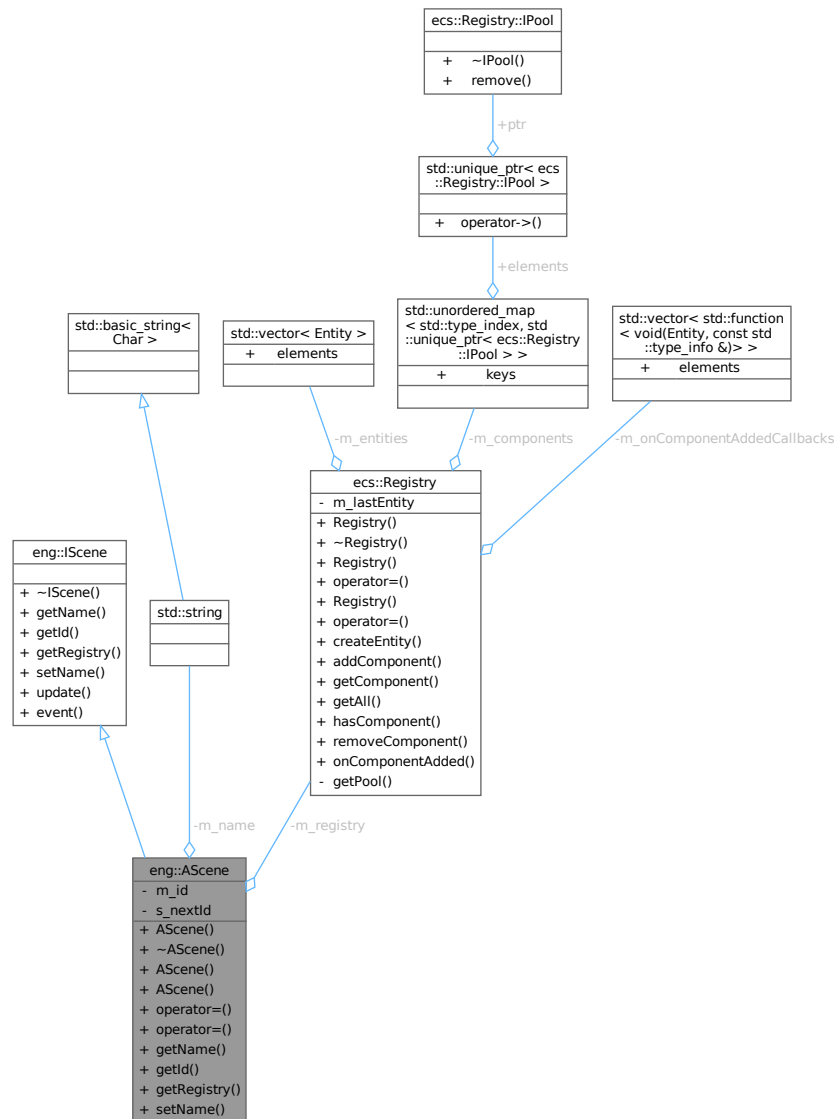
Class for scene.

```
#include <IScene.hpp>
```

Inheritance diagram for eng::AScene:



Collaboration diagram for eng::AScene:



Public Member Functions

- `AScene()`
- `~AScene()` override=default
- `AScene(const AScene &other)=delete`
- `AScene(AScene &&other)=delete`
- `AScene &operator=(const AScene &other)=delete`
- `AScene &operator=(AScene &&other)=delete`
- `std::string &getName()` override
- `id getId()` const override
- `ecs::Registry &getRegistry()` override
- `void setName(const std::string &newName)` override

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

- virtual [~IScene](#) ()=default
- virtual void [update](#) (float dt, const [WindowSize](#) &size)=0
- virtual void [event](#) (const [Event](#) &event)=0

Private Attributes

- std::string [m_name](#) = "default_name"
- [id](#) [m_id](#) = 1
- [ecs::Registry](#) [m_registry](#)

Static Private Attributes

- static [id](#) [s_nextId](#) = 1

7.7.1 Detailed Description

Class for scene.

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

7.7.2 Constructor & Destructor Documentation

7.7.2.1 AScene() [1/3]

[eng::AScene::AScene](#) () [inline]

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

7.7.2.2 ~AScene()

[eng::AScene::~~AScene](#) () [override], [default]

7.7.2.3 AScene() [2/3]

[eng::AScene::AScene](#) (
const [AScene](#) & other) [delete]

7.7.2.4 AScene() [3/3]

[eng::AScene::AScene](#) (
[AScene](#) && other) [delete]

7.7.3 Member Function Documentation

7.7.3.1 getId()

[id](#) `eng::AScene::getId () const` [inline], [nodiscard], [override], [virtual]

Implements [eng::IScene](#).

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

References [m_id](#).

7.7.3.2 getName()

`std::string & eng::AScene::getName ()` [inline], [nodiscard], [override], [virtual]

Implements [eng::IScene](#).

Definition at line 56 of file [IScene.hpp](#).

References [m_name](#).

7.7.3.3 getRegistry()

[ecs::Registry](#) & `eng::AScene::getRegistry ()` [inline], [nodiscard], [override], [virtual]

Implements [eng::IScene](#).

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

References [m_registry](#).

7.7.3.4 operator=() [1/2]

[AScene](#) & `eng::AScene::operator= (`
 [AScene](#) && other) [delete]

7.7.3.5 operator=() [2/2]

[AScene](#) & `eng::AScene::operator= (`
 const [AScene](#) & other) [delete]

7.7.3.6 setName()

`void eng::AScene::setName (`
 const std::string & newName) [inline], [override], [virtual]

Implements [eng::IScene](#).

Definition at line 60 of file [IScene.hpp](#).

References [m_name](#).

7.7.4 Member Data Documentation

7.7.4.1 m_id

`id eng::AScene::m_id = 1 [private]`

Definition at line 64 of file [IScene.hpp](#).

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

7.7.4.2 m_name

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

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

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

7.7.4.3 m_registry

`ecs::Registry eng::AScene::m_registry [private]`

Definition at line 65 of file [IScene.hpp](#).

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

7.7.4.4 s_nextId

`id eng::AScene::s_nextId = 1 [inline], [static], [private]`

Definition at line 66 of file [IScene.hpp](#).

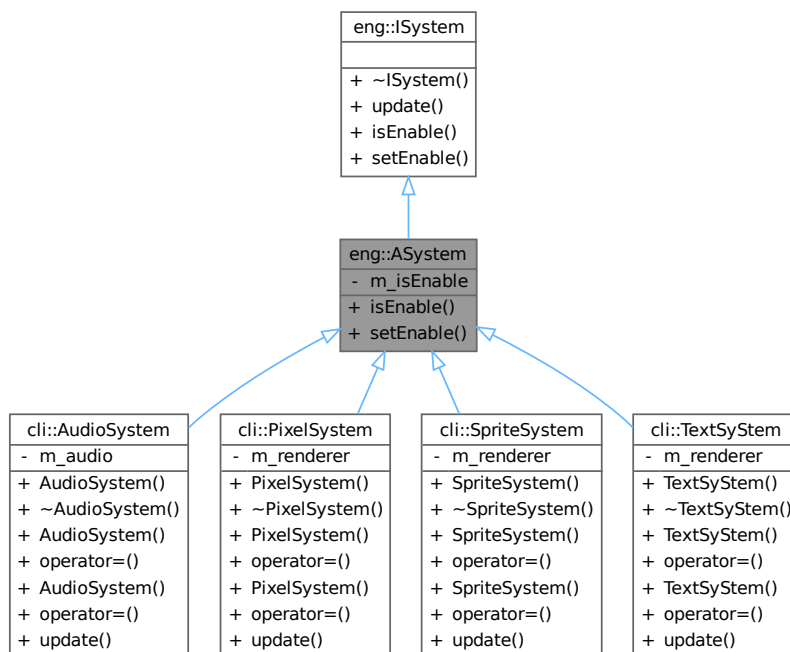
The documentation for this class was generated from the following file:

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

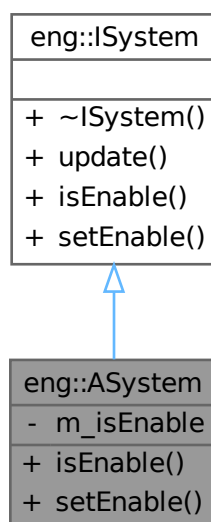
7.8 eng::ASystem Class Reference

#include <Systems.hpp>

Inheritance diagram for eng::ASystem:



Collaboration diagram for eng::ASystem:



Public Member Functions

- bool [isEnabled](#) () override
- void [setEnabled](#) (const bool enable) override

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

- virtual [~ISystem](#) ()=default
- virtual void [update](#) ([ecs::Registry](#) ®istry, float dt)=0

Private Attributes

- bool [m_isEnable](#) = true

7.8.1 Detailed Description

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

7.8.2 Member Function Documentation

7.8.2.1 isEnabled()

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

Implements [eng::ISystem](#).

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

References [m_isEnable](#).

7.8.2.2 setEnabled()

void [eng::ASystem::setEnabled](#) (
 const bool enable) [inline], [override], [virtual]

Implements [eng::ISystem](#).

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

References [m_isEnable](#).

7.8.3 Member Data Documentation

7.8.3.1 m_isEnable

bool [eng::ASystem::m_isEnable](#) = true [private]

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

Referenced by [isEnabled\(\)](#), and [setEnabled\(\)](#).

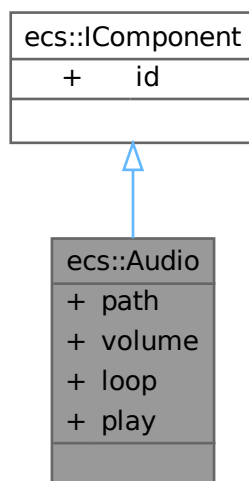
The documentation for this class was generated from the following file:

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

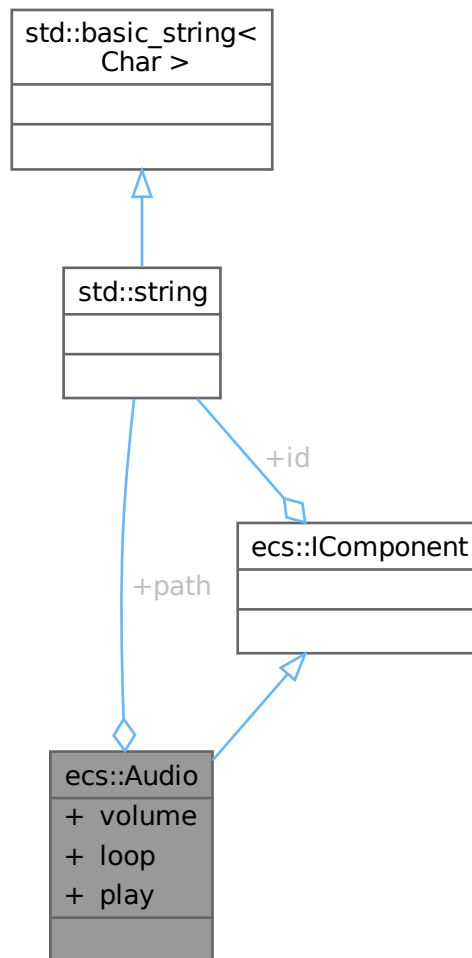
7.9 ecs::Audio Struct Reference

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

Inheritance diagram for ecs::Audio:



Collaboration diagram for `ecs::Audio`:



Public Attributes

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

Public Attributes inherited from [ecs::IComponent](#)

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

7.9.1 Detailed Description

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

7.9.2 Member Data Documentation

7.9.2.1 loop

bool ecs::Audio::loop

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

7.9.2.2 path

std::string ecs::Audio::path

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

7.9.2.3 play

bool ecs::Audio::play

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

7.9.2.4 volume

float ecs::Audio::volume

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

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

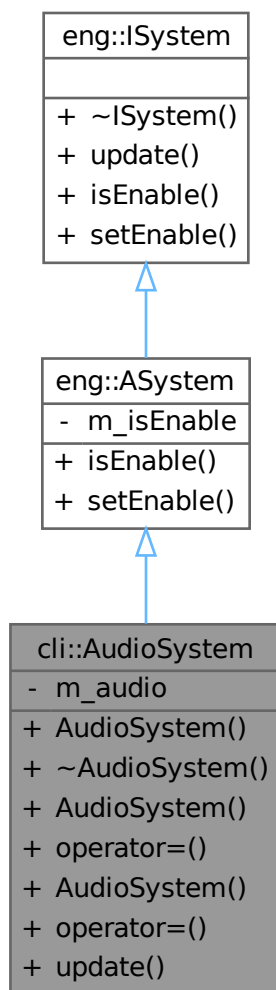
- [/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.hpp](#)

7.10 cli::AudioSystem Class Reference

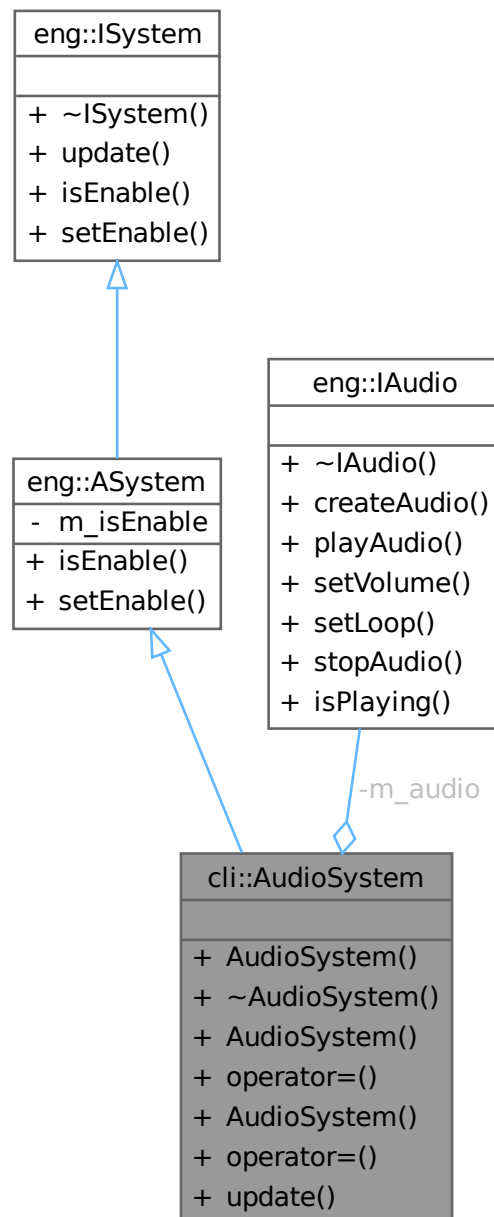
Class for managing entities and their components.

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

Inheritance diagram for cli::AudioSystem:



Collaboration diagram for cli::AudioSystem:



Public Member Functions

- [AudioSystem](#) ([eng::IAudio](#) &audio)
- [~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

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

- bool [isEnabled](#) () override
- void [setEnabled](#) (const bool enable) override

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

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

Private Attributes

- [eng::IAudio](#) & [m_audio](#)

7.10.1 Detailed Description

Class for managing entities and their components.

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

7.10.2 Constructor & Destructor Documentation

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

```
cli::AudioSystem::AudioSystem (  
    eng::IAudio & audio)    [inline], [explicit]
```

Definition at line 70 of file [Systems.hpp](#).

7.10.2.2 [~AudioSystem\(\)](#)

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

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

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

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

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

7.10.3 Member Function Documentation

7.10.3.1 operator=() [1/2]

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

7.10.3.2 operator=() [2/2]

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

7.10.3.3 update()

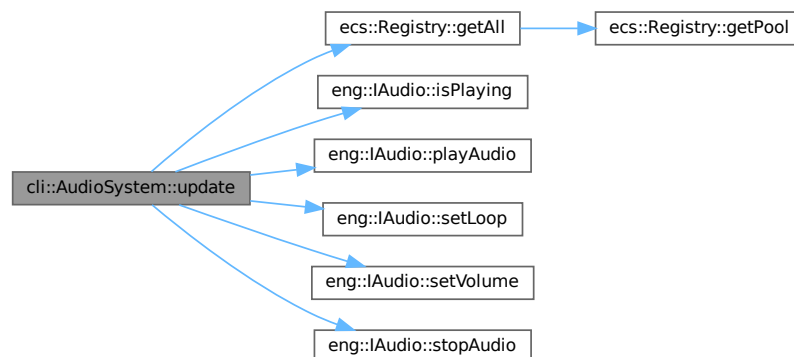
void cli::AudioSystem::update (
[ecs::Registry](#) & registry,
float dt) [inline], [override], [virtual]

Implements [eng::ISystem](#).

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

References [ecs::Registry::getAll\(\)](#), [eng::IAudio::isPlaying\(\)](#), [m_audio](#), [eng::IAudio::playAudio\(\)](#), [eng::Playing](#), [eng::IAudio::setLoop\(\)](#), [eng::IAudio::setVolume\(\)](#), [eng::IAudio::stopAudio\(\)](#), and [eng::Stopped](#).

Here is the call graph for this function:



7.10.4 Member Data Documentation

7.10.4.1 m_audio

[eng::IAudio](#)& cli::AudioSystem::m_audio [private]

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

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

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

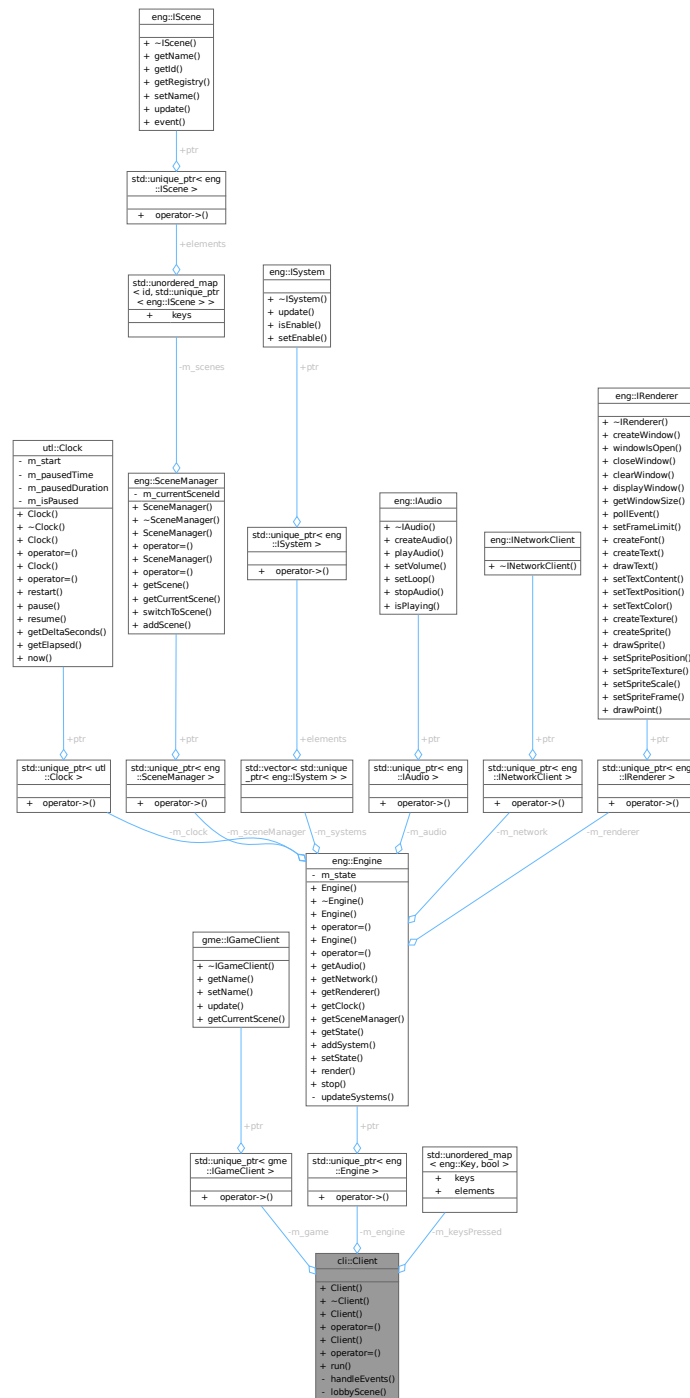
- `/home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Systems.hpp`

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
- void [run](#) ()

Private Member Functions

- void [handleEvents](#) ([eng::Event](#) &event)
- [eng::IScene](#) & [lobbyScene](#) ()

Private Attributes

- std::unique_ptr< [gme::IGameClient](#) > [m_game](#)
- std::unique_ptr< [eng::Engine](#) > [m_engine](#)
- std::unordered_map< [eng::Key](#), bool > [m_keysPressed](#)

7.11.1 Detailed Description

Class for the client.

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

7.11.2 Constructor & Destructor Documentation

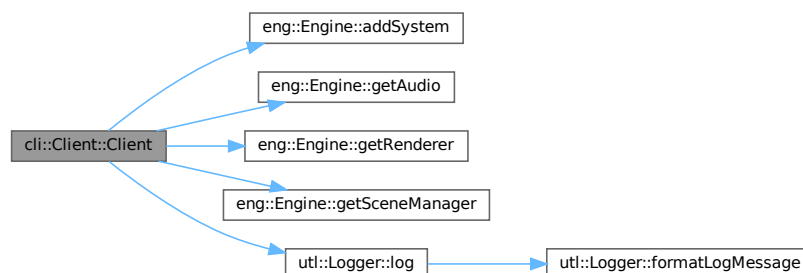
7.11.2.1 Client() [1/3]

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

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

References [eng::Engine::addSystem\(\)](#), [BUILD_TYPE](#), [cli::ArgsConfig::frameLimit](#), [cli::ArgsConfig::fullscreen](#), [eng::Engine::getAudio\(\)](#), [eng::Engine::getRenderer\(\)](#), [eng::Engine::getSceneManager\(\)](#), [GIT_COMMIT_HASH](#), [GIT_TAG](#), [cli::ArgsConfig::height](#), [utl::INFO](#), [utl::Logger::log\(\)](#), [m_engine](#), [m_game](#), [PROJECT_NAME](#), [PROJECT_VERSION](#), 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

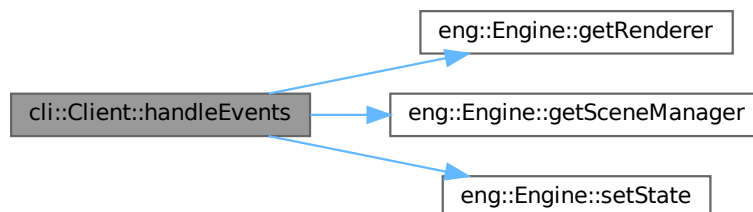
7.11.3.1 handleEvents()

void cli::Client::handleEvents (
[eng::Event](#) & event) [private]

Definition at line 3 of file [event.cpp](#).

References [eng::Closed](#), [eng::Escape](#), [eng::Engine::getRenderer\(\)](#), [eng::Engine::getSceneManager\(\)](#), [eng::Event::key](#), [eng::KeyPressed](#), [eng::KeyReleased](#), [m_engine](#), [m_keysPressed](#), [eng::Engine::setState\(\)](#), [eng::STOP](#), and [eng::Event::type](#).

Here is the call graph for this function:



7.11.3.2 lobbyScene()

[eng::IScene](#) & cli::Client::lobbyScene () [private]

7.11.3.3 operator=() [1/2]

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

7.11.3.4 operator=() [2/2]

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

7.11.3.5 run()

void cli::Client::run ()

Definition at line 40 of file [client.cpp](#).

References [DARK](#), and [eng::RUN](#).

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

Here is the caller graph for this function:



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 43 of file [Client.hpp](#).

Referenced by [Client\(\)](#), and [handleEvents\(\)](#).

7.11.4.2 m_game

std::unique_ptr<[gme::IGameClient](#)> cli::Client::m_game [private]

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

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

7.11.4.3 m_keysPressed

std::unordered_map<eng::Key, bool> cli::Client::m_keysPressed [private]

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

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

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](#)
- /home/masina/Projects/Epitech/rtype/client/src/[event.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

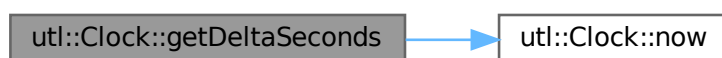
7.12.4.1 getDeltaSeconds()

```
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\(\)](#).

Here is the call 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:



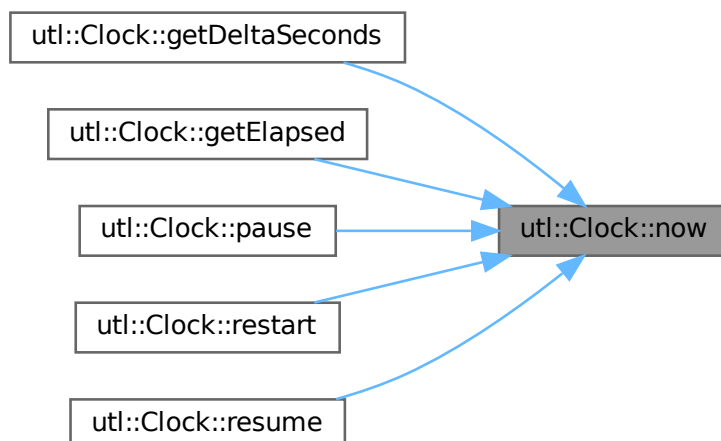
7.12.4.3 now()

```
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\(\)](#).

Here is the call 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\(\)](#).

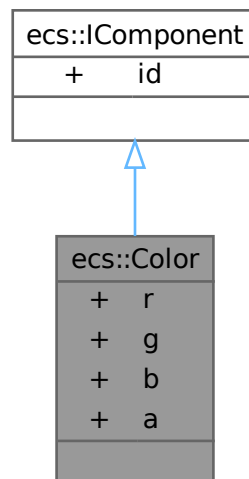
The documentation for this class was generated from the following file:

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

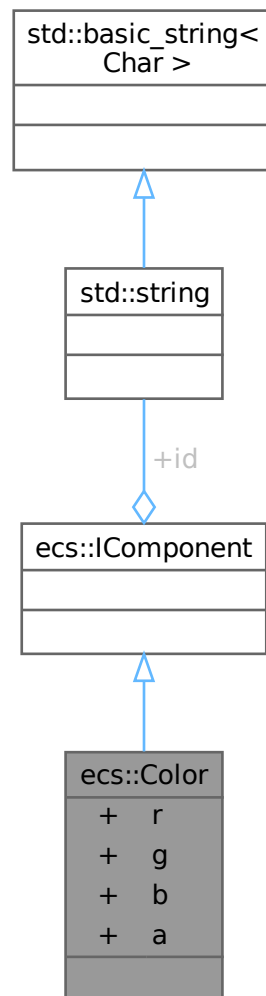
7.13 ecs::Color Struct Reference

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

Inheritance diagram for `ecs::Color`:



Collaboration diagram for `ecs::Color`:



Public Attributes

- unsigned char `r` {}
- unsigned char `g` {}
- unsigned char `b` {}
- unsigned char `a` {}

Public Attributes inherited from `ecs::IComponent`

- std::string `id`

7.13.1 Detailed Description

Definition at line 24 of file `Component.hpp`.

7.13.2 Member Data Documentation

7.13.2.1 a

```
unsigned char ecs::Color::a {}
```

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

7.13.2.2 b

```
unsigned char ecs::Color::b {}
```

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

7.13.2.3 g

```
unsigned char ecs::Color::g {}
```

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

7.13.2.4 r

```
unsigned char ecs::Color::r {}
```

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

Referenced by [cli::PixelSystem::update\(\)](#).

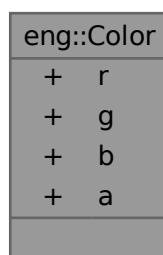
The documentation for this struct was generated from the following file:

- [/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

- unsigned char [r](#)
- unsigned char [g](#)
- unsigned char [b](#)
- unsigned char [a](#)

7.14.1 Detailed Description

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

7.14.2 Member Data Documentation

7.14.2.1 [a](#)

unsigned char eng::Color::a

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

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

7.14.2.2 [b](#)

unsigned char eng::Color::b

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

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

7.14.2.3 [g](#)

unsigned char eng::Color::g

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

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

7.14.2.4 [r](#)

unsigned char eng::Color::r

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

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

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

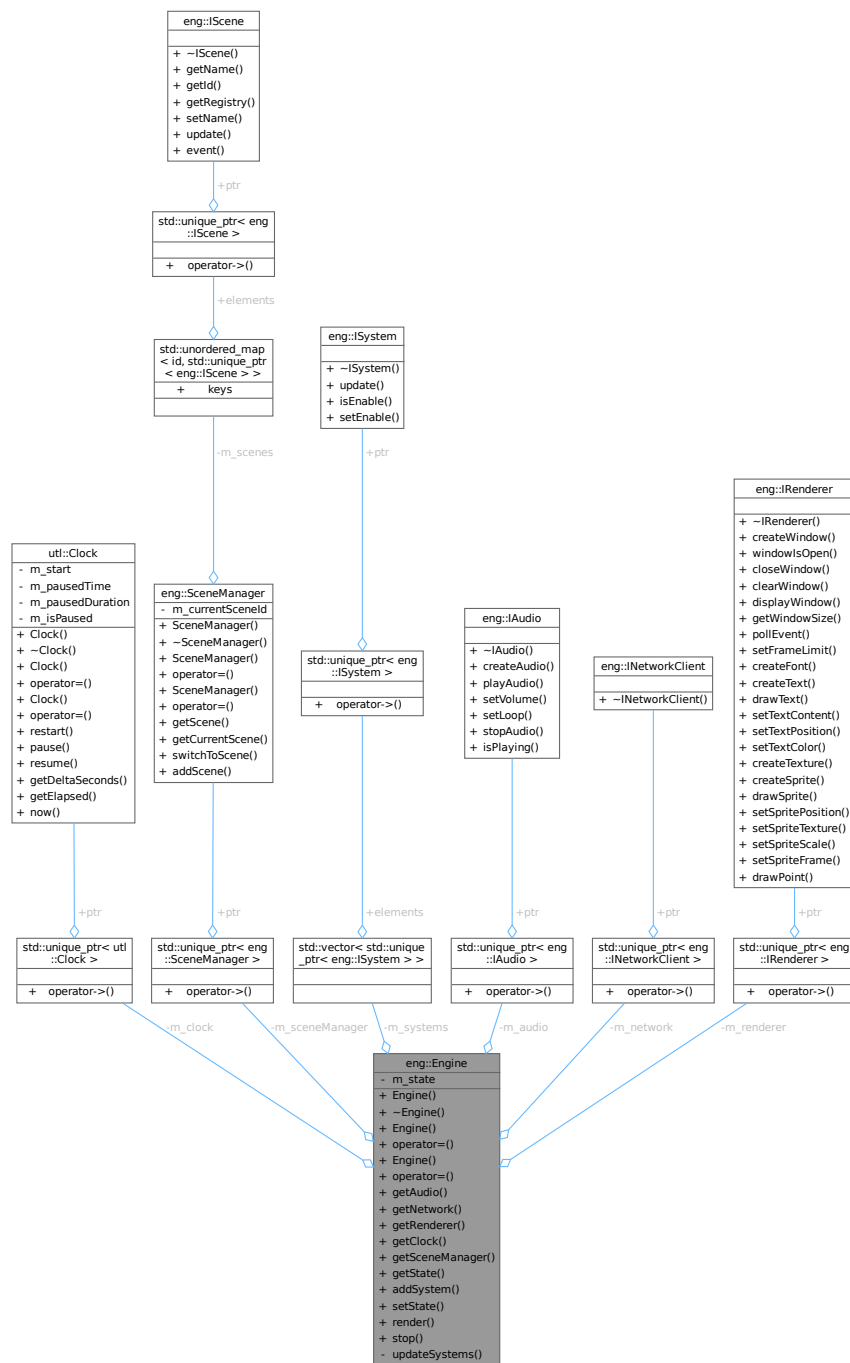
- [/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IRenderer.hpp](#)

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](#) > & [getNetwork](#) ()
- std::unique_ptr< [IRenderer](#) > & [getRenderer](#) ()
- std::unique_ptr< [utl::Clock](#) > & [getClock](#) ()
- std::unique_ptr< [SceneManager](#) > & [getSceneManager](#) ()
- [State](#) [getState](#) () const
- void [addSystem](#) (std::unique_ptr< [ISystem](#) > system)
- void [setState](#) (const [State](#) newState)
- void [render](#) ([ecs::Registry](#) ®istry, [Color](#) clearColor, float dt) const
- void [stop](#) () const

Private Member Functions

- void [updateSystems](#) ([ecs::Registry](#) ®istry, float dt) const

Private Attributes

- [State](#) [m_state](#) = [RUN](#)
- std::unique_ptr< [utl::Clock](#) > [m_clock](#)
- std::unique_ptr< [SceneManager](#) > [m_sceneManager](#)
- std::vector< std::unique_ptr< [ISystem](#) > > [m_systems](#)
- std::unique_ptr< [IAudio](#) > [m_audio](#)
- std::unique_ptr< [INetworkClient](#) > [m_network](#)
- std::unique_ptr< [IRenderer](#) > [m_renderer](#)

7.15.1 Detailed Description

Class for the game engine.

Definition at line 35 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 3 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 56 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 49 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 52 of file [Engine.hpp](#).

References [m_clock](#).

7.15.3.4 getNetwork()

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

Definition at line 50 of file [Engine.hpp](#).

References [m_network](#).

7.15.3.5 getRenderer()

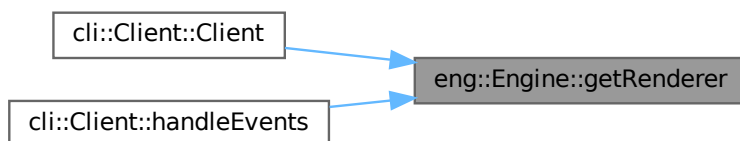
`std::unique_ptr< IRenderer > & eng::Engine::getRenderer ()` [inline]

Definition at line 51 of file [Engine.hpp](#).

References [m_renderer](#).

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

Here is the caller graph for this function:



7.15.3.6 getSceneManager()

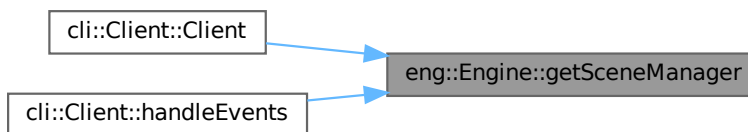
`std::unique_ptr< SceneManager > & eng::Engine::getSceneManager ()` [inline]

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

References [m_sceneManager](#).

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

Here is the caller graph for this function:



7.15.3.7 getState()

```
State eng::Engine::getState () const [inline]
```

Definition at line 54 of file [Engine.hpp](#).

References [m_state](#).

7.15.3.8 operator=() [1/2]

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

7.15.3.9 operator=() [2/2]

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

7.15.3.10 render()

```
void eng::Engine::render (  
    ecs::Registry & registry,  
    Color clearColor,  
    float dt) const
```

Definition at line 19 of file [Engine.cpp](#).

7.15.3.11 setState()

```
void eng::Engine::setState (  
    const State newState) [inline]
```

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

References [m_state](#).

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

Here is the caller graph for this function:



7.15.3.12 stop()

void eng::Engine::stop () const [inline]

Definition at line 60 of file [Engine.hpp](#).

References [m_renderer](#).

7.15.3.13 updateSystems()

void eng::Engine::updateSystems (
 [ecs::Registry](#) & registry,
 float dt) const [private]

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

7.15.4 Member Data Documentation

7.15.4.1 m_audio

std::unique_ptr<[IAudio](#)> eng::Engine::m_audio [private]

Definition at line 69 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 66 of file [Engine.hpp](#).

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

7.15.4.3 m_network

std::unique_ptr<[INetworkClient](#)> eng::Engine::m_network [private]

Definition at line 70 of file [Engine.hpp](#).

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

7.15.4.4 m_renderer

std::unique_ptr<[IRenderer](#)> eng::Engine::m_renderer [private]

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

Referenced by [getRenderer\(\)](#), and [stop\(\)](#).

7.15.4.5 m_sceneManager

`std::unique_ptr<SceneManager> eng::Engine::m_sceneManager` [private]

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

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

7.15.4.6 m_state

`State eng::Engine::m_state = RUN` [private]

Definition at line 65 of file [Engine.hpp](#).

Referenced by [getState\(\)](#), and [setState\(\)](#).

7.15.4.7 m_systems

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

Definition at line 68 of file [Engine.hpp](#).

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

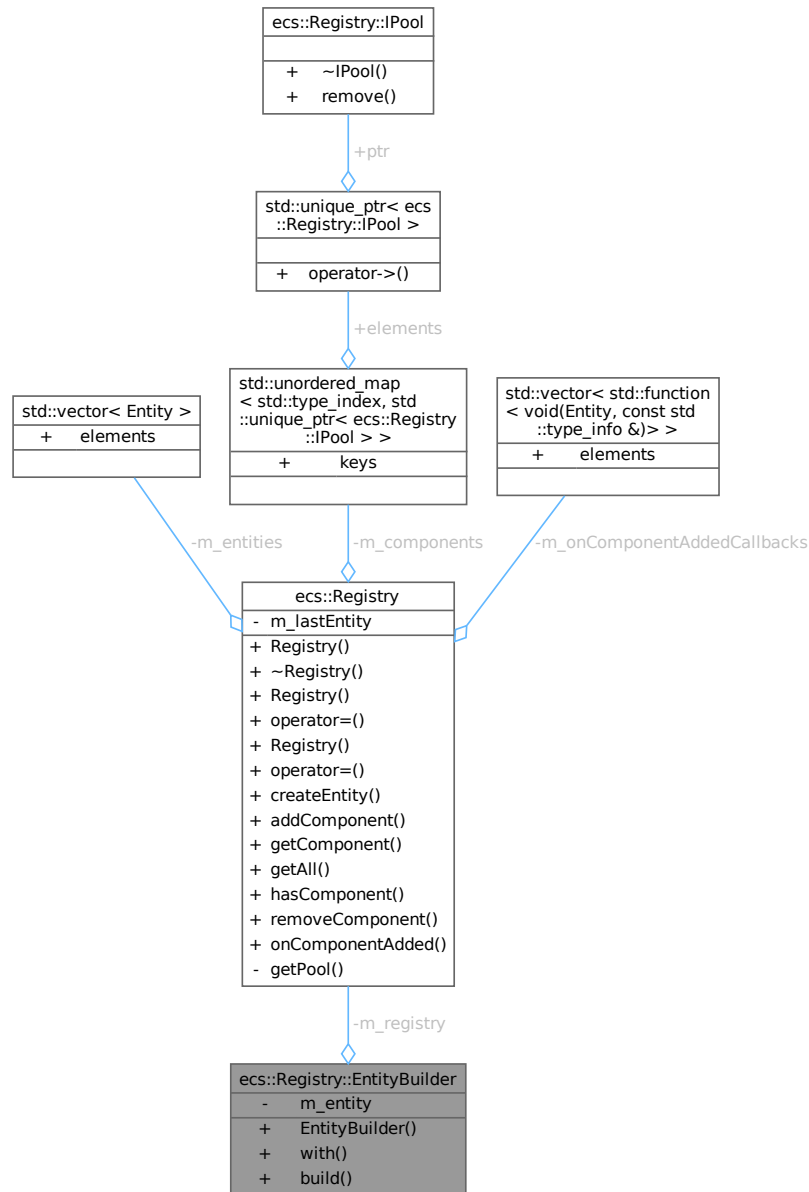
The documentation for this class was generated from the following files:

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

7.16 ecs::Registry::EntityBuilder Class Reference

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


Collaboration diagram for ecs::Registry::EntityBuilder:



Public Member Functions

- [EntityBuilder](#) ([Registry](#) ®, [Entity](#) e)
- `template<typename T , typename... Args>`
[EntityBuilder](#) & [with](#) (Args &&...args)
- [Entity build](#) () const

Private Attributes

- [Registry](#) & [m_registry](#)
- [Entity](#) [m_entity](#)

7.16.1 Detailed Description

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

7.16.2 Constructor & Destructor Documentation

7.16.2.1 EntityBuilder()

```
ecs::Registry::EntityBuilder::EntityBuilder (
    Registry & reg,
    Entity e) [inline]
```

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

7.16.3 Member Function Documentation

7.16.3.1 build()

```
Entity ecs::Registry::EntityBuilder::build () const [inline]
```

Definition at line 46 of file [Registry.hpp](#).

References [m_entity](#).

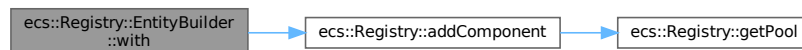
7.16.3.2 with()

```
template<typename T , typename... Args>
EntityBuilder & ecs::Registry::EntityBuilder::with (
    Args &&... args) [inline]
```

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

References [ecs::Registry::addComponent\(\)](#), [m_entity](#), and [m_registry](#).

Here is the call graph for this function:



7.16.4 Member Data Documentation

7.16.4.1 m_entity

```
Entity ecs::Registry::EntityBuilder::m_entity [private]
```

Definition at line 50 of file [Registry.hpp](#).

Referenced by [build\(\)](#), and [with\(\)](#).

7.16.4.2 m_registry

[Registry&](#) ecs::Registry::EntityBuilder::m_registry [private]

Definition at line 49 of file [Registry.hpp](#).

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

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

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

7.17 cli::EnvConfig Struct Reference

#include <ArgsHandler.hpp>

Collaboration diagram for cli::EnvConfig:



7.17.1 Detailed Description

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

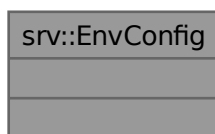
The documentation for this struct was generated from the following file:

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

7.18 srv::EnvConfig Struct Reference

#include <ArgsHandler.hpp>

Collaboration diagram for srv::EnvConfig:



7.18.1 Detailed Description

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

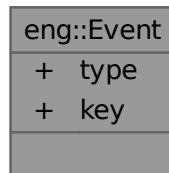
The documentation for this struct was generated from the following file:

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

7.19 eng::Event Struct Reference

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

Collaboration diagram for eng::Event:



Public Attributes

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

7.19.1 Detailed Description

Definition at line 85 of file [IRenderer.hpp](#).

7.19.2 Member Data Documentation

7.19.2.1 key

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

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

Referenced by [cli::Lobby::event\(\)](#), and [cli::Client::handleEvents\(\)](#).

7.19.2.2 type

`EventType` eng::Event::type = `EventType::None`

Definition at line 87 of file `IRenderer.hpp`.

Referenced by `cli::Lobby::event()`, and `cli::Client::handleEvents()`.

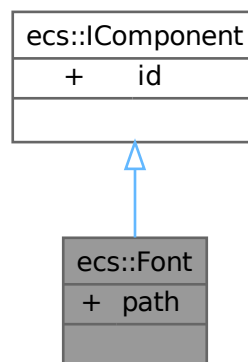
The documentation for this struct was generated from the following file:

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

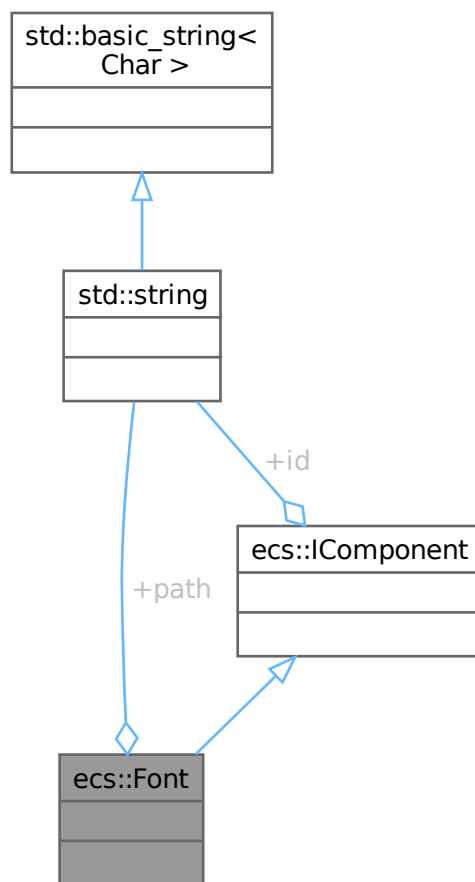
7.20 ecs::Font Struct Reference

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

Inheritance diagram for `ecs::Font`:



Collaboration diagram for `ecs::Font`:



Public Attributes

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

Public Attributes inherited from [ecs::IComponent](#)

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

7.20.1 Detailed Description

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

7.20.2 Member Data Documentation

7.20.2.1 path

std::string ecs::Font::path

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

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

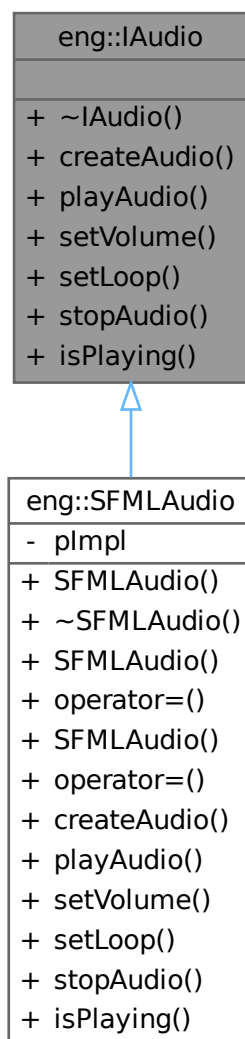
- [/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.hpp](#)

7.21 eng::IAudio Class Reference

Interface for the audio.

#include <IAudio.hpp>

Inheritance diagram for eng::IAudio:



Collaboration diagram for `eng::IAudio`:

eng::IAudio
<ul style="list-style-type: none"> + ~IAudio() + createAudio() + playAudio() + setVolume() + setLoop() + stopAudio() + isPlaying()

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
- virtual void [stopAudio](#) (const std::string &name)=0
- virtual [Status](#) [isPlaying](#) (const std::string &name)=0

7.21.1 Detailed Description

Interface for the audio.

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

7.21.2 Constructor & Destructor Documentation

7.21.2.1 ~IAudio()

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

7.21.3 Member Function Documentation

7.21.3.1 createAudio()

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

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

7.21.3.2 isPlaying()

```
virtual Status eng::IAudio::isPlaying (  
    const std::string & name) [pure virtual]
```

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

Referenced by [cli::AudioSystem::update\(\)](#).

Here is the caller graph for this function:



7.21.3.3 playAudio()

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

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

Referenced by [cli::AudioSystem::update\(\)](#).

Here is the caller graph for this function:



7.21.3.4 setLoop()

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

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

Referenced by [cli::AudioSystem::update\(\)](#).

Here is the caller graph for this function:



7.21.3.5 setVolume()

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

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

Referenced by [cli::AudioSystem::update\(\)](#).

Here is the caller graph for this function:



7.21.3.6 stopAudio()

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

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

Referenced by [cli::AudioSystem::update\(\)](#).

Here is the caller graph for this function:



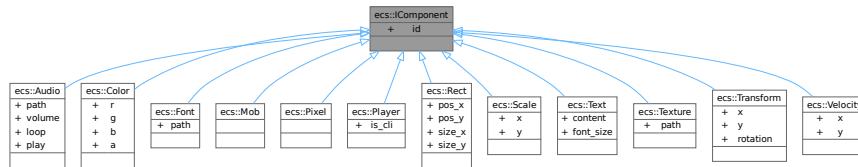
The documentation for this class was generated from the following file:

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

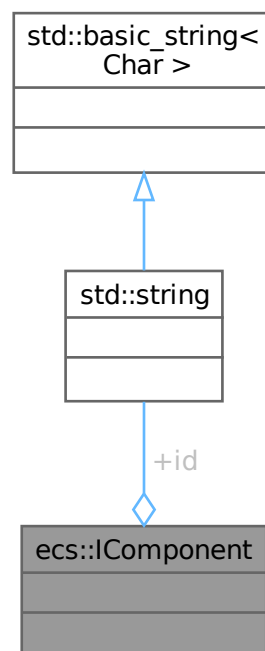
7.22 ecs::IComponent Struct Reference

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

Inheritance diagram for ecs::IComponent:



Collaboration diagram for ecs::IComponent:



Public Attributes

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

7.22.1 Detailed Description

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

7.22.2 Member Data Documentation

7.22.2.1 id

std::string ecs::IComponent::id

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

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

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

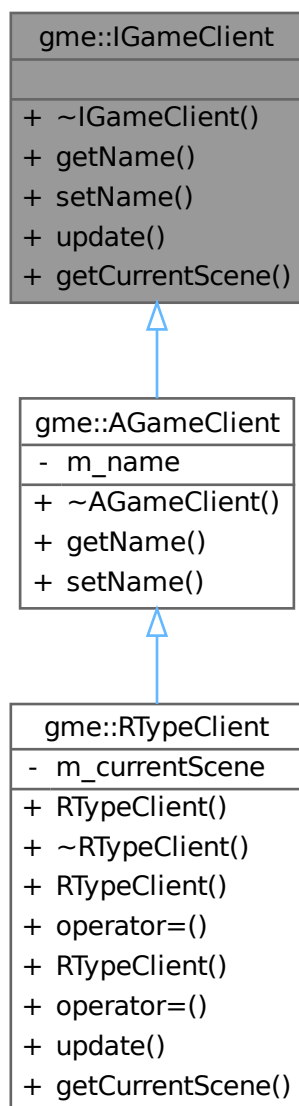
- [/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.hpp](#)

7.23 gme::IGameClient Class Reference

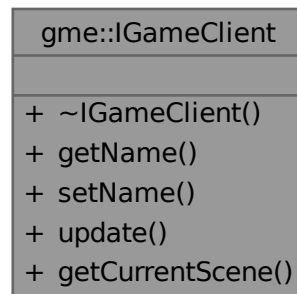
Interface for the games.

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

Inheritance diagram for gme::IGameClient:



Collaboration diagram for `gme::IGameClient`:



Public Member Functions

- virtual [~IGameClient](#) ()=default
- virtual `std::string` & [getName](#) ()=0
- virtual void [setName](#) (const `std::string` &newName)=0
- virtual void [update](#) (float deltaTime, unsigned int width, unsigned int height)=0
- virtual const [IScene](#) & [getCurrentScene](#) () const =0

7.23.1 Detailed Description

Interface for the games.

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

7.23.2 Constructor & Destructor Documentation

7.23.2.1 ~IGameClient()

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

7.23.3 Member Function Documentation

7.23.3.1 getCurrentScene()

virtual const [IScene](#) & `gme::IGameClient::getCurrentScene` () const [nodiscard], [pure virtual]

Implemented in [gme::RTypeClient](#).

7.23.3.2 getName()

virtual std::string & gme::IGameClient::getName () [nodiscard], [pure virtual]

Implemented in [gme::AGameClient](#).

7.23.3.3 setName()

virtual void gme::IGameClient::setName (
 const std::string & newName) [pure virtual]

Implemented in [gme::AGameClient](#).

7.23.3.4 update()

virtual void gme::IGameClient::update (
 float deltaTime,
 unsigned int width,
 unsigned int height) [pure virtual]

Implemented in [gme::RTypeClient](#).

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

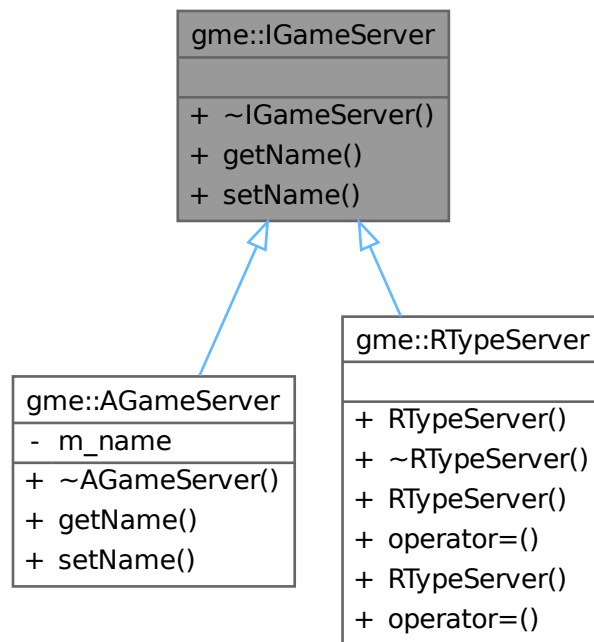
- [/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IGameClient.hpp](#)

7.24 gme::IGameServer Class Reference

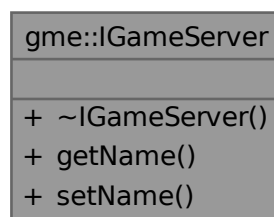
Interface for the games.

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

Inheritance diagram for gme::IGameServer:



Collaboration diagram for gme::IGameServer:



Public Member Functions

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

7.24.1 Detailed Description

Interface for the games.

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

7.24.2 Constructor & Destructor Documentation

7.24.2.1 ~IGameServer()

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

7.24.3 Member Function Documentation

7.24.3.1 getName()

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

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

7.24.3.2 setName()

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

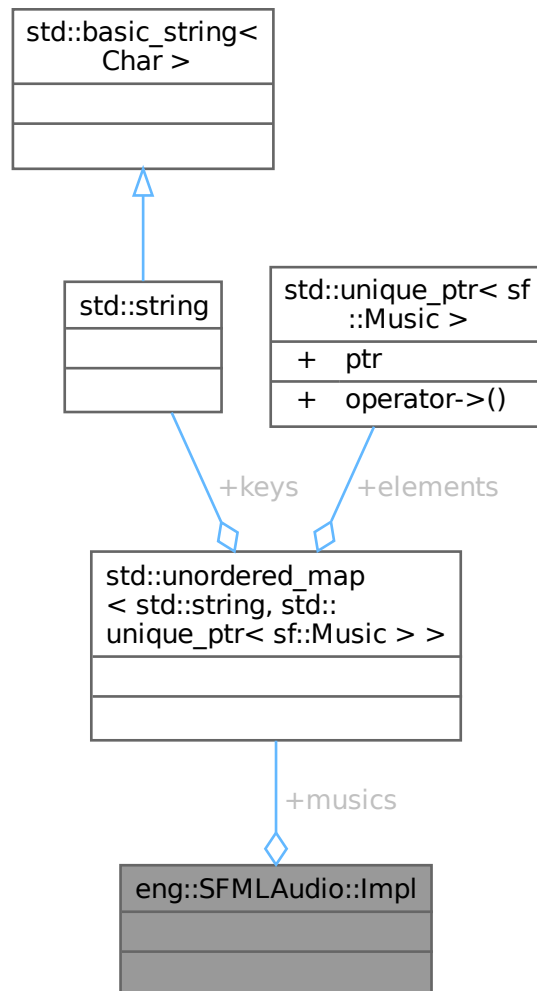
Reimplemented in [gme::AGameServer](#).

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

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

7.25 eng::SFMLAudio::Impl Struct Reference

Collaboration diagram for eng::SFMLAudio::Impl:



Public Attributes

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

7.25.1 Detailed Description

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

7.25.2 Member Data Documentation

7.25.2.1 musics

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

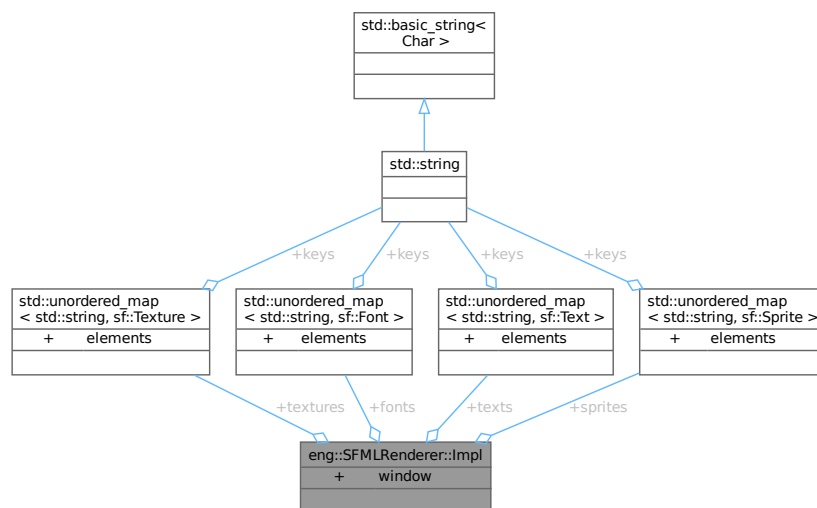
Definition at line 11 of file [SFMLAudio.cpp](#).

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

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

7.26 eng::SFMLRenderer::Impl Struct Reference

Collaboration diagram for eng::SFMLRenderer::Impl:



Public Attributes

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

7.26.1 Detailed Description

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

7.26.2 Member Data Documentation

7.26.2.1 fonts

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

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

7.26.2.2 sprites

`std::unordered_map<std::string, sf::Sprite> eng::SFMLRenderer::Impl::sprites`

Definition at line 16 of file [SFMLRenderer.cpp](#).

7.26.2.3 texts

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

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

7.26.2.4 textures

`std::unordered_map<std::string, sf::Texture> eng::SFMLRenderer::Impl::textures`

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

7.26.2.5 window

`sf::RenderWindow eng::SFMLRenderer::Impl::window`

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

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

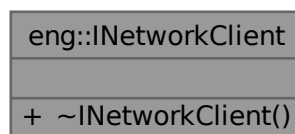
- [/home/masina/Projects/Epitech/rtype/modules/Renderer/SFMLRenderer/src/SFMLRenderer.cpp](#)

7.27 eng::INetworkClient Class Reference

Interface for the client network.

`#include <INetworkClient.hpp>`

Collaboration diagram for `eng::INetworkClient`:



Public Member Functions

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

7.27.1 Detailed Description

Interface for the client network.

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

7.27.2 Constructor & Destructor Documentation

7.27.2.1 ~INetworkClient()

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

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

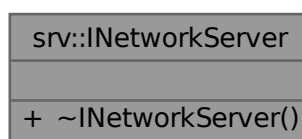
- /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/[INetworkClient.hpp](#)

7.28 srv::INetworkServer Class Reference

Interface for the server network.

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

Collaboration diagram for srv::INetworkServer:



Public Member Functions

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

7.28.1 Detailed Description

Interface for the server network.

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

7.28.2 Constructor & Destructor Documentation

7.28.2.1 ~INetworkServer()

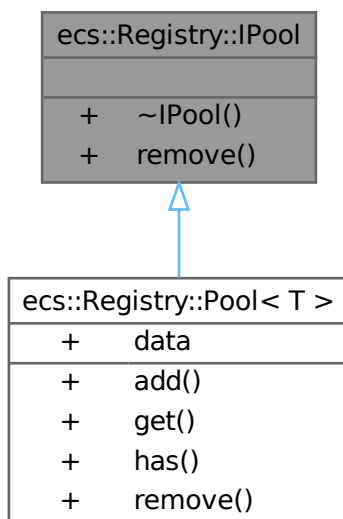
virtual srv::INetworkServer::~INetworkServer () [virtual], [default]

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

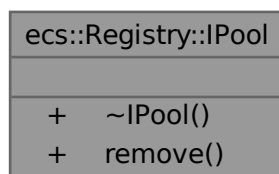
- /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/[INetworkServer.hpp](#)

7.29 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

7.29.1 Detailed Description

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

7.29.2 Constructor & Destructor Documentation

7.29.2.1 ~IPool()

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

7.29.3 Member Function Documentation

7.29.3.1 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:

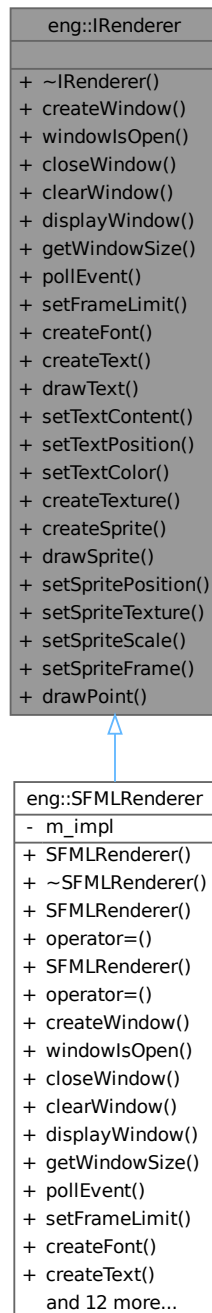
- `/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Registry.hpp`

7.30 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() + clearWindow() + displayWindow() + getWindowSize() + pollEvent() + setFrameLimit() + createFont() + createText() + drawText() + setTextContent() + setTextPosition() + setTextColor() + createTexture() + createSprite() + drawSprite() + setSpritePosition() + setSpriteTexture() + setSpriteScale() + setSpriteFrame() + drawPoint()

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 [clearWindow](#) ([Color](#) color)=0
- virtual void [displayWindow](#) ()=0
- virtual [WindowSize](#) [getWindowSize](#) ()=0
- virtual bool [pollEvent](#) ([Event](#) &event)=0
- virtual void [setFrameLimit](#) (unsigned int frameLimit)=0
- virtual void [createFont](#) (const std::string &name, const std::string &path)=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, float x, float y)=0
- virtual void [setTextColor](#) (const std::string &name, [Color](#) color)=0
- virtual void [createTexture](#) (const std::string &name, const std::string &path)=0
- virtual void [createSprite](#) (const std::string &name, const std::string &textureName, float x, float y, float scale_x=1, float scale_y=1, int fx=0, int fy=0, int fnx=-1, int fny=-1)=0
- virtual void [drawSprite](#) (const std::string &name)=0
- virtual void [setSpritePosition](#) (const std::string &name, float x, float y)=0
- virtual void [setSpriteTexture](#) (const std::string &name, const std::string &path)=0
- virtual void [setSpriteScale](#) (const std::string &name, int x, int y)=0
- virtual void [setSpriteFrame](#) (const std::string &name, int fx, int fy, int fnx, int fny)=0
- virtual void [drawPoint](#) (float x, float y, [Color](#) color)=0

7.30.1 Detailed Description

Interface for the renderer.

Definition at line 102 of file [IRenderer.hpp](#).

7.30.2 Constructor & Destructor Documentation

7.30.2.1 ~IRenderer()

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

7.30.3 Member Function Documentation

7.30.3.1 clearWindow()

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

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

7.30.3.2 closeWindow()

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

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

7.30.3.3 createFont()

virtual void eng::IRenderer::createFont (
const std::string & name,
const std::string & path) [pure virtual]

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

7.30.3.4 createSprite()

```
virtual void eng::IRenderer::createSprite (  
    const std::string & name,  
    const std::string & textureName,  
    float x,  
    float y,  
    float scale_x = 1,  
    float scale_y = 1,  
    int fx = 0,  
    int fy = 0,  
    int fnx = -1,  
    int fny = -1) [pure virtual]
```

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

7.30.3.5 createText()

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

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

7.30.3.6 createTexture()

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

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

7.30.3.7 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](#).

7.30.3.8 displayWindow()

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

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

7.30.3.9 drawPoint()

```
virtual void eng::IRenderer::drawPoint (  
    float x,  
    float y,  
    Color color) [pure virtual]
```

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

Referenced by [cli::PixelSystem::update\(\)](#).

Here is the caller graph for this function:



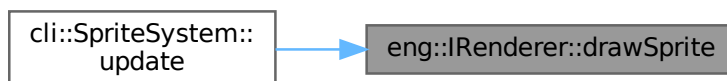
7.30.3.10 drawSprite()

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

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

Referenced by [cli::SpriteSystem::update\(\)](#).

Here is the caller graph for this function:



7.30.3.11 drawText()

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

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

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

Here is the caller graph for this function:



7.30.3.12 getWindowSize()

```
virtual WindowSize eng::IRenderer::getWindowSize () [nodiscard], [pure virtual]
```

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

7.30.3.13 pollEvent()

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

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

7.30.3.14 setFrameLimit()

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

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

7.30.3.15 setSpriteFrame()

```
virtual void eng::IRenderer::setSpriteFrame (  
    const std::string & name,  
    int fx,  
    int fy,  
    int fnx,  
    int fny) [pure virtual]
```

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

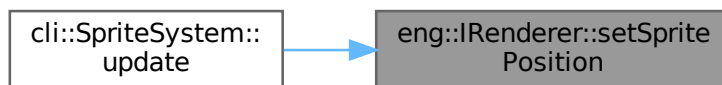
7.30.3.16 setSpritePosition()

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

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

Referenced by [cli::SpriteSystem::update\(\)](#).

Here is the caller graph for this function:



7.30.3.17 setSpriteScale()

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

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

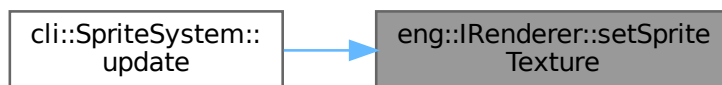
7.30.3.18 setSpriteTexture()

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

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

Referenced by [cli::SpriteSystem::update\(\)](#).

Here is the caller graph for this function:



7.30.3.19 setTextColor()

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

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

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

Here is the caller graph for this function:



7.30.3.20 setTextContent()

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

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

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

Here is the caller graph for this function:



7.30.3.21 setPosition()

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

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

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

Here is the caller graph for this function:



7.30.3.22 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:

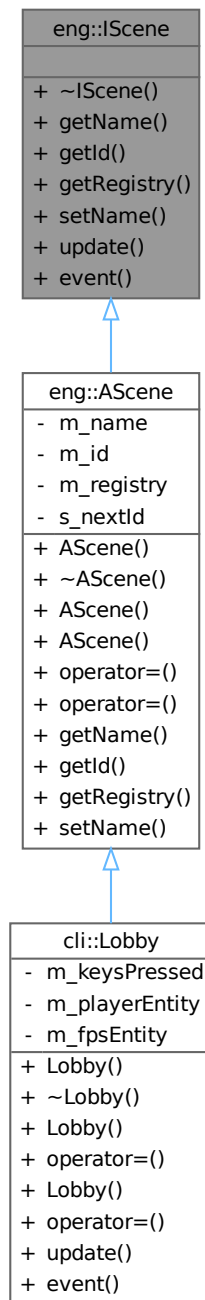
- `/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IRenderer.hpp`

7.31 eng::IScene Class Reference

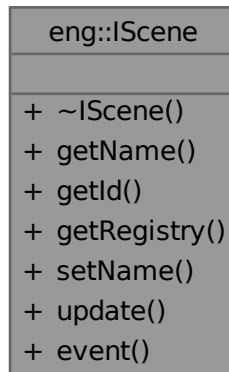
interface class for scene

```
#include <IScene.hpp>
```


Inheritance diagram for eng::IScene:



Collaboration diagram for `eng::IScene`:



Public Member Functions

- virtual `~IScene()`=default
- virtual `std::string & getName()`=0
- virtual `id getId()` const =0
- virtual `ecs::Registry & getRegistry()`=0
- virtual void `setName` (const `std::string &newName`)=0
- virtual void `update` (float dt, const `WindowSize &size`)=0
- virtual void `event` (const `Event &event`)=0

7.31.1 Detailed Description

interface class for scene

Definition at line 24 of file `IScene.hpp`.

7.31.2 Constructor & Destructor Documentation

7.31.2.1 ~IScene()

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

7.31.3 Member Function Documentation

7.31.3.1 event()

virtual void `eng::IScene::event` (
 const `Event &event`) [pure virtual]

Implemented in `cli::Lobby`.

7.31.3.2 getId()

virtual [id](#) eng::IScene::getId () const [nodiscard], [pure virtual]

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

7.31.3.3 getName()

virtual std::string & eng::IScene::getName () [nodiscard], [pure virtual]

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

7.31.3.4 getRegistry()

virtual [ecs::Registry](#) & eng::IScene::getRegistry () [nodiscard], [pure virtual]

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

7.31.3.5 setName()

virtual void eng::IScene::setName (
 const std::string & newName) [pure virtual]

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

7.31.3.6 update()

virtual void eng::IScene::update (
 float dt,
 const [WindowSize](#) & size) [pure virtual]

Implemented in [cli::Lobby](#).

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

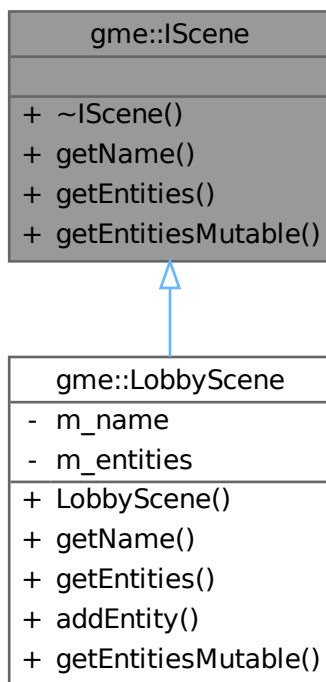
- [/home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/IScene.hpp](#)

7.32 gme::IScene Class Reference

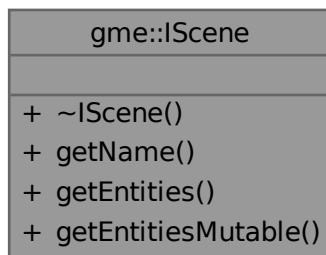
Interface for scenes.

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

Inheritance diagram for gme::IScene:



Collaboration diagram for gme::IScene:



Public Member Functions

- virtual [~IScene](#) ()=default
- virtual const std::string & [getName](#) () const =0
- virtual const std::vector< [Sprite](#) > & [getEntities](#) () const =0
- virtual std::vector< [Sprite](#) > & [getEntitiesMutable](#) ()=0

7.32.1 Detailed Description

Interface for scenes.

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

7.32.2 Constructor & Destructor Documentation

7.32.2.1 ~IScene()

virtual gme::IScene::~~IScene () [virtual], [default]

7.32.3 Member Function Documentation

7.32.3.1 getEntities()

virtual const std::vector< [Sprite](#) > & gme::IScene::getEntities () const [nodiscard], [pure virtual]

Implemented in [gme::LobbyScene](#).

7.32.3.2 getEntitiesMutable()

virtual std::vector< [Sprite](#) > & gme::IScene::getEntitiesMutable () [nodiscard], [pure virtual]

Implemented in [gme::LobbyScene](#).

7.32.3.3 getName()

virtual const std::string & gme::IScene::getName () const [nodiscard], [pure virtual]

Implemented in [gme::LobbyScene](#).

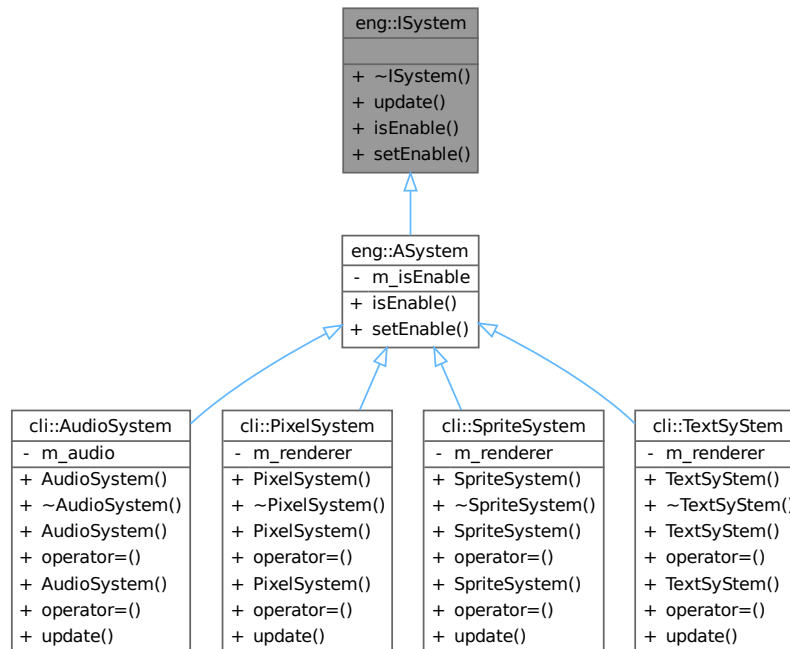
The documentation for this class was generated from the following file:

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

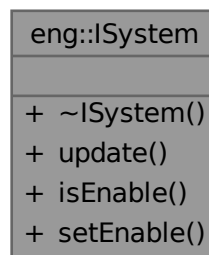
7.33 eng::ISystem Class Reference

```
#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
- virtual void `setEnable` (bool enable)=0

7.33.1 Detailed Description

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

7.33.2 Constructor & Destructor Documentation

7.33.2.1 ~ISystem()

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

7.33.3 Member Function Documentation

7.33.3.1 isEnabled()

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

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

7.33.3.2 setEnable()

virtual void eng::ISystem::setEnable (
 bool enable) [pure virtual]

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

7.33.3.3 update()

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

Implemented in [cli::AudioSystem](#), [cli::PixelSystem](#), [cli::SpriteSystem](#), and [cli::TextSyStem](#).

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

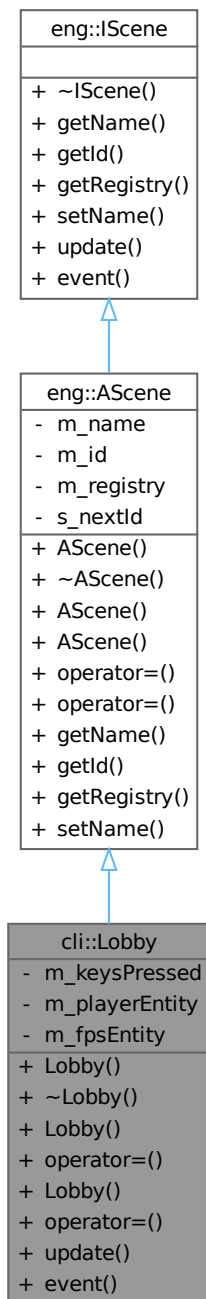
- [/home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/Systems.hpp](#)

7.34 cli::Lobby Class Reference

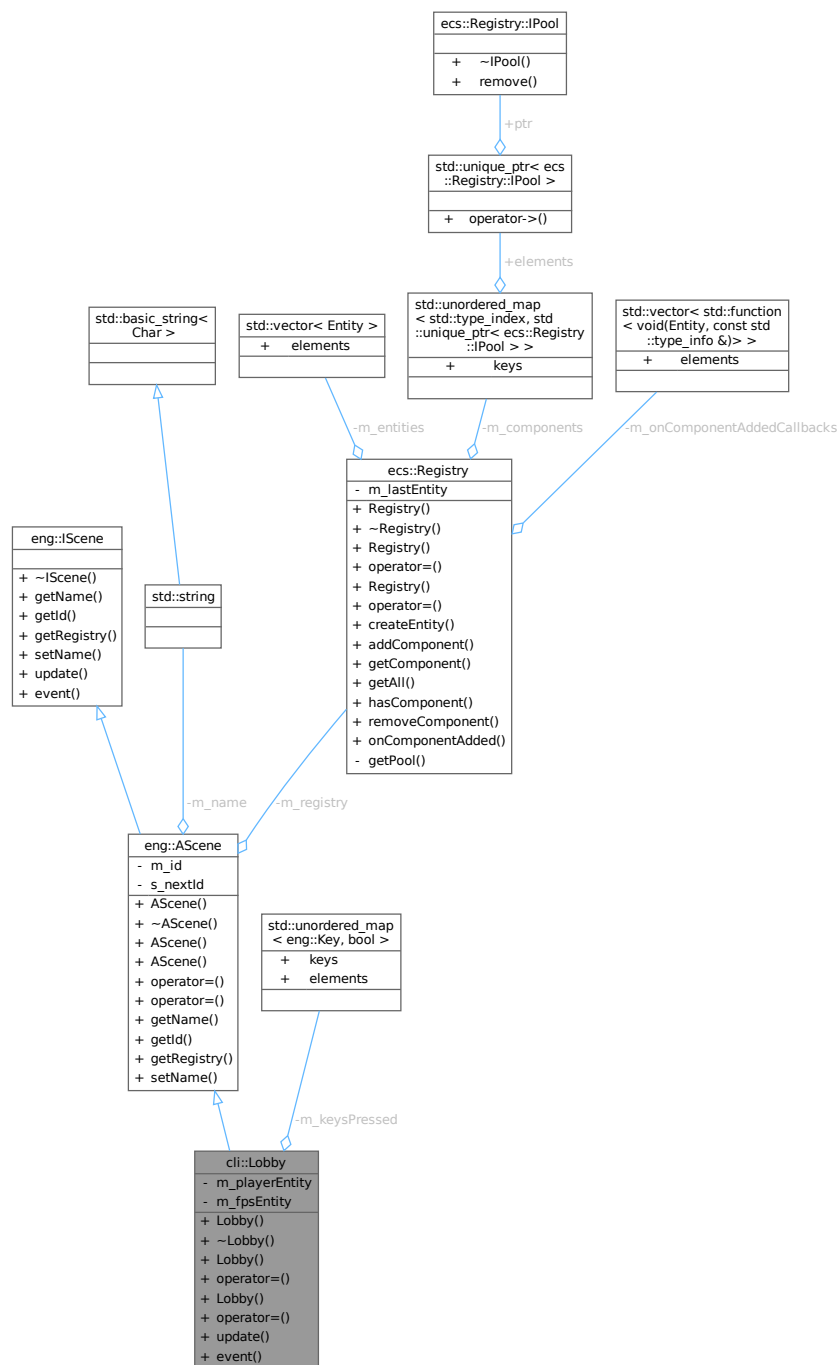
[Lobby](#) scene.

```
#include <Lobby.hpp>
```

Inheritance diagram for cli::Lobby:



Collaboration diagram for cli::Lobby:



Public Member Functions

- `Lobby` (const std::unique_ptr< eng::IRenderer > &renderer, const std::unique_ptr< eng::IAudio > &audio)
- `~Lobby` () override=default
- `Lobby` (const Lobby &other)=delete
- `Lobby` & operator= (const Lobby &other)=delete

- [Lobby](#) ([Lobby](#) &&other)=delete
- [Lobby](#) & [operator=](#) ([Lobby](#) &&other)=delete
- void [update](#) (float dt, const [eng::WindowSize](#) &size) override
- void [event](#) (const [eng::Event](#) &event) override

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

- [AScene](#) ()
- [~AScene](#) () override=default
- [AScene](#) (const [AScene](#) &other)=delete
- [AScene](#) ([AScene](#) &&other)=delete
- [AScene](#) & [operator=](#) (const [AScene](#) &other)=delete
- [AScene](#) & [operator=](#) ([AScene](#) &&other)=delete
- std::string & [getName](#) () override
- [id](#) [getId](#) () const override
- [ecs::Registry](#) & [getRegistry](#) () override
- void [setName](#) (const std::string &newName) override

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

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

Private Attributes

- std::unordered_map< [eng::Key](#), bool > [m_keysPressed](#)
- [ecs::Entity](#) [m_playerEntity](#)
- [ecs::Entity](#) [m_fpsEntity](#)

7.34.1 Detailed Description

[Lobby](#) scene.

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

7.34.2 Constructor & Destructor Documentation

7.34.2.1 [Lobby](#)() [1/3]

```
cli::Lobby::Lobby (
    const std::unique_ptr< eng::IRenderer > & renderer,
    const std::unique_ptr< eng::IAudio > & audio)
```

Definition at line 8 of file [lobby.cpp](#).

References [eng::Color::a](#), [cli::Path::Audio::AUDIO_TITLE](#), [eng::Color::b](#), [cli::Path::Font::FONTS_RTYPE](#), [eng::Color::g](#), [ecs::IComponent::id](#), [m_fpsEntity](#), [m_playerEntity](#), [eng::Color::r](#), [cli::Path::Texture::TEXTURE_PLAYER_WHITE](#), and [ecs::Scale::x](#).

7.34.2.2 ~Lobby()

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

7.34.2.3 Lobby() [2/3]

cli::Lobby::Lobby (
const [Lobby](#) & other) [delete]

7.34.2.4 Lobby() [3/3]

cli::Lobby::Lobby (
[Lobby](#) && other) [delete]

7.34.3 Member Function Documentation

7.34.3.1 event()

void cli::Lobby::event (
const [eng::Event](#) & event) [override], [virtual]

Implements [eng::IScene](#).

Definition at line 156 of file [lobby.cpp](#).

References [eng::Down](#), [eng::Event::key](#), [eng::KeyPressed](#), [eng::KeyReleased](#), [eng::Left](#), [eng::Right](#), [eng::Event::type](#), and [eng::Up](#).

7.34.3.2 operator=() [1/2]

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

7.34.3.3 operator=() [2/2]

[Lobby](#) & cli::Lobby::operator= (
[Lobby](#) && other) [delete]

7.34.3.4 update()

void cli::Lobby::update (
float dt,
const [eng::WindowSize](#) & size) [override], [virtual]

Implements [eng::IScene](#).

Definition at line 106 of file [lobby.cpp](#).

References [eng::Down](#), [eng::WindowSize::height](#), [eng::Left](#), [eng::Right](#), [eng::Up](#), [eng::WindowSize::width](#), [ecs::Transform::x](#), [ecs::Velocity::x](#), and [ecs::Transform::y](#).

7.34.4 Member Data Documentation

7.34.4.1 m_fpsEntity

[ecs::Entity](#) cli::Lobby::m_fpsEntity [private]

Definition at line 39 of file [Lobby.hpp](#).

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

7.34.4.2 m_keysPressed

std::unordered_map<[eng::Key](#), bool> cli::Lobby::m_keysPressed [private]

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

7.34.4.3 m_playerEntity

[ecs::Entity](#) cli::Lobby::m_playerEntity [private]

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

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

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

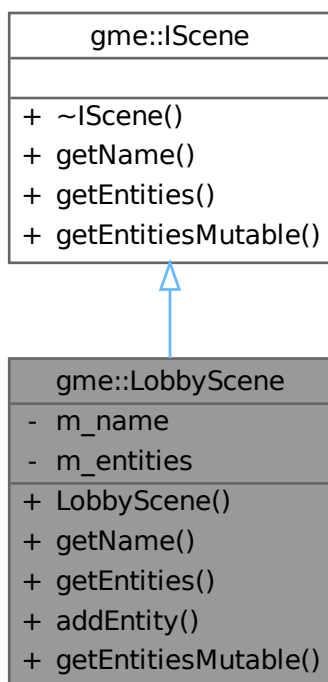
- [/home/masina/Projects/Epitech/rtype/client/include/Client/Scenes/Lobby.hpp](#)
- [/home/masina/Projects/Epitech/rtype/client/src/scenes/lobby.cpp](#)

7.35 gme::LobbyScene Class Reference

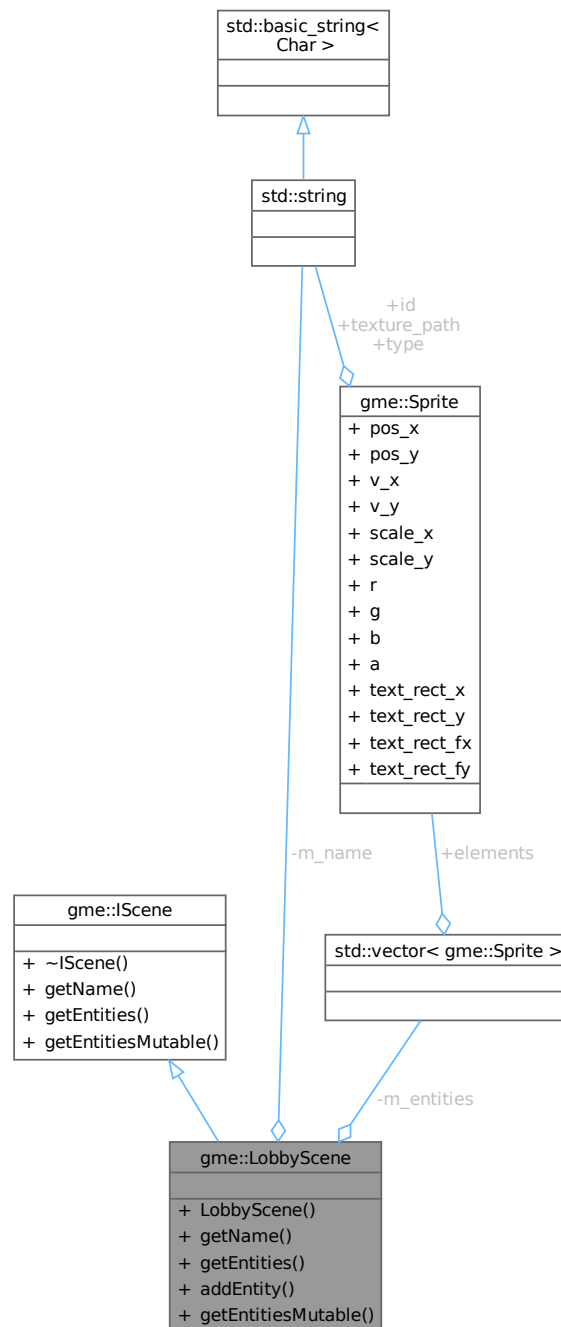
Class for the Lobby scene.

```
#include <LobbyScene.hpp>
```

Inheritance diagram for gme::LobbyScene:



Collaboration diagram for gme::LobbyScene:



Public Member Functions

- [LobbyScene](#) ()
- const std::string & [getName](#) () const override
- const std::vector< [Sprite](#) > & [getEntities](#) () const override
- void [addEntity](#) (const [Sprite](#) &e)
- std::vector< [Sprite](#) > & [getEntitiesMutable](#) () override

Public Member Functions inherited from [gme::IScene](#)

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

Private Attributes

- std::string [m_name](#)
- std::vector< [Sprite](#) > [m_entities](#)

7.35.1 Detailed Description

Class for the Lobby scene.

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

7.35.2 Constructor & Destructor Documentation

7.35.2.1 LobbyScene()

[gme::LobbyScene::LobbyScene](#) () [inline]

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

7.35.3 Member Function Documentation

7.35.3.1 addEntity()

void [gme::LobbyScene::addEntity](#) (
 const [Sprite](#) & e) [inline]

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

References [m_entities](#).

7.35.3.2 getEntities()

const std::vector< [Sprite](#) > & [gme::LobbyScene::getEntities](#) () const [inline], [nodiscard], [override], [virtual]

Implements [gme::IScene](#).

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

References [m_entities](#).

7.35.3.3 getEntitiesMutable()

`std::vector< Sprite > & gme::LobbyScene::getEntitiesMutable ()` `[inline]`, `[override]`, `[virtual]`

Implements [gme::IScene](#).

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

References [m_entities](#).

7.35.3.4 getName()

`const std::string & gme::LobbyScene::getName () const` `[inline]`, `[nodiscard]`, `[override]`, `[virtual]`

Implements [gme::IScene](#).

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

References [m_name](#).

7.35.4 Member Data Documentation

7.35.4.1 m_entities

`std::vector<Sprite> gme::LobbyScene::m_entities` `[private]`

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

Referenced by [addEntity\(\)](#), [getEntities\(\)](#), and [getEntitiesMutable\(\)](#).

7.35.4.2 m_name

`std::string gme::LobbyScene::m_name` `[private]`

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

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

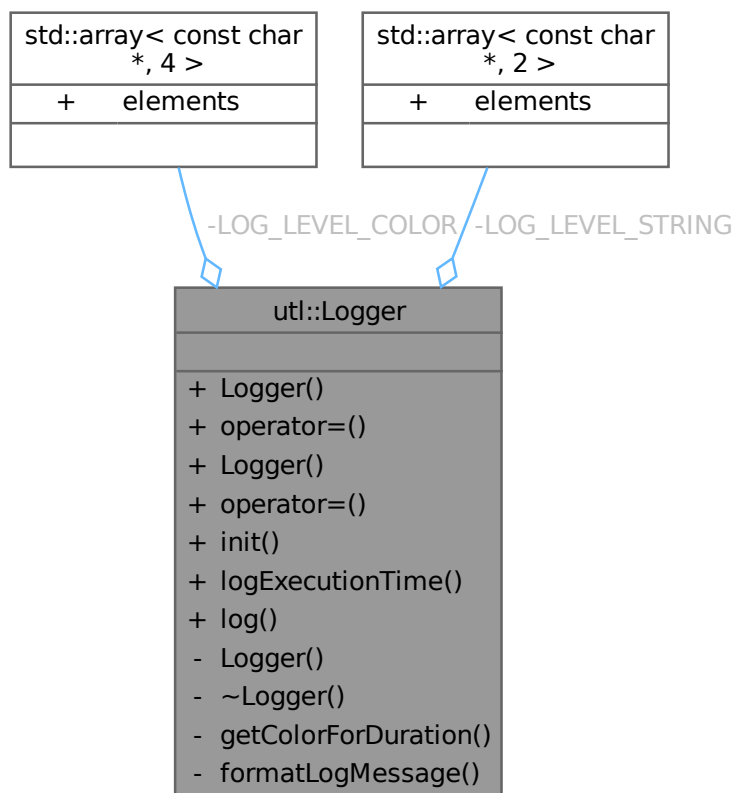
The documentation for this class was generated from the following file:

- [/home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Client/include/R-TypeClient/LobbyScene.hpp](#)

7.36 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"}

7.36.1 Detailed Description

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

7.36.2 Member Enumeration Documentation

7.36.2.1 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](#).

7.36.3 Constructor & Destructor Documentation

7.36.3.1 Logger() [1/3]

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

7.36.3.2 Logger() [2/3]

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

7.36.3.3 Logger() [3/3]

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

7.36.3.4 ~Logger()

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

7.36.4 Member Function Documentation

7.36.4.1 formatLogMessage()

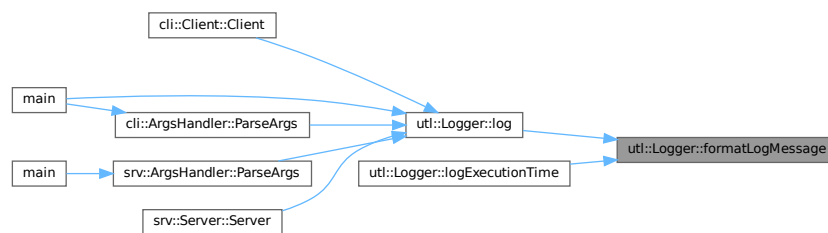
```
static std::string utl::Logger::formatLogMessage (
    LogLevel level,
    const std::string & message) [inline], [static], [nodiscard], [private]
```

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:



7.36.4.2 getColorForDuration()

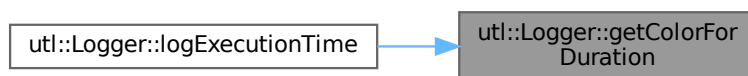
```
static const char * utl::Logger::getColorForDuration (  
    const float duration)  [inline], [static], [nodiscard], [private]
```

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:



7.36.4.3 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:



7.36.4.4 log()

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

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

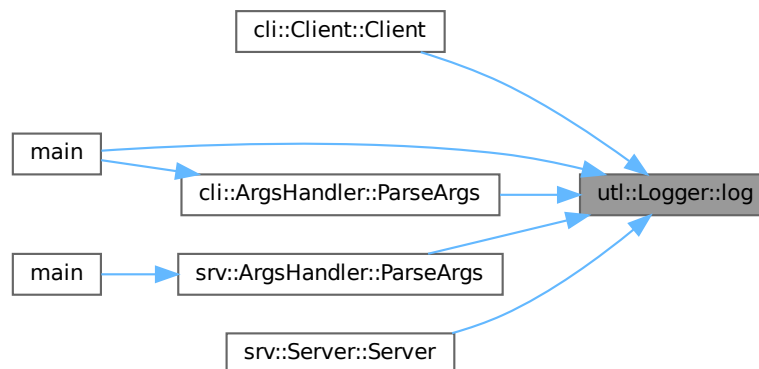
References [COLOR_INFO](#), [COLOR_RESET](#), [COLOR_WARNING](#), [formatLogMessage\(\)](#), [utl::INFO](#), and [LOG_LEVEL_COLOR](#).

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:



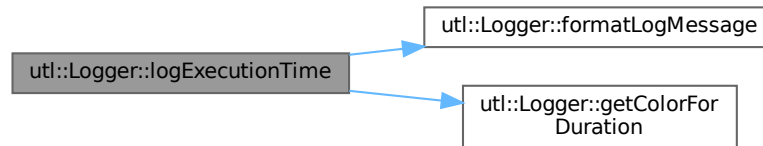
7.36.4.5 logExecutionTime()

```
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:



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

```

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

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

```

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

7.36.5 Member Data Documentation

7.36.5.1 `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\(\)](#).

7.36.5.2 `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\(\)](#).

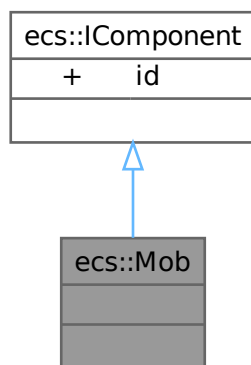
The documentation for this class was generated from the following files:

- `/home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/Logger.hpp`
- `/home/masina/Projects/Epitech/rtype/modules/Utils/src/logger.cpp`

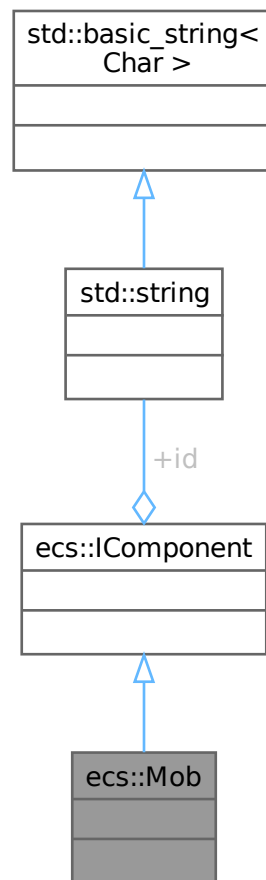
7.37 ecs::Mob Struct Reference

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

Inheritance diagram for ecs::Mob:



Collaboration diagram for `ecs::Mob`:



Additional Inherited Members

Public Attributes inherited from [ecs::IComponent](#)

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

7.37.1 Detailed Description

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

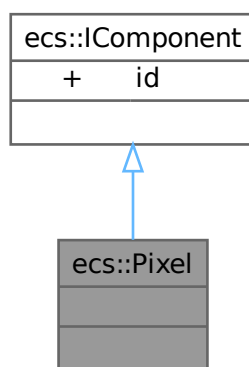
The documentation for this struct was generated from the following file:

- `/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.hpp`

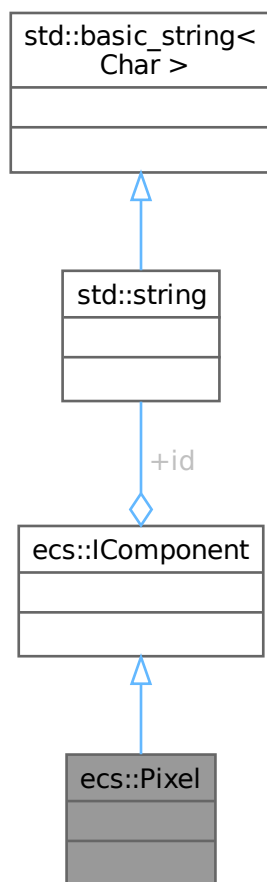
7.38 ecs::Pixel Struct Reference

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

Inheritance diagram for ecs::Pixel:



Collaboration diagram for `ecs::Pixel`:



Additional Inherited Members

Public Attributes inherited from [ecs::IComponent](#)

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

7.38.1 Detailed Description

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

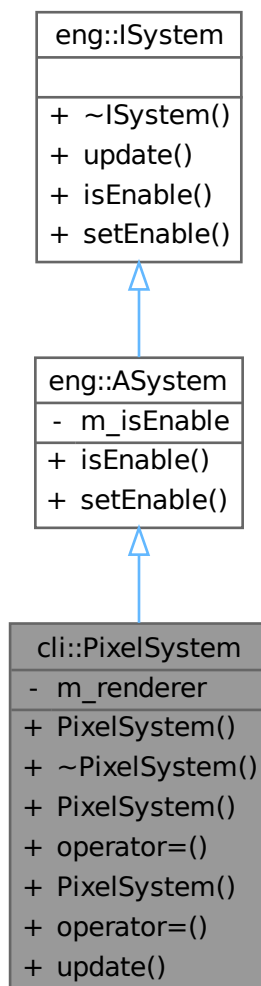
The documentation for this struct was generated from the following file:

- `/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.hpp`

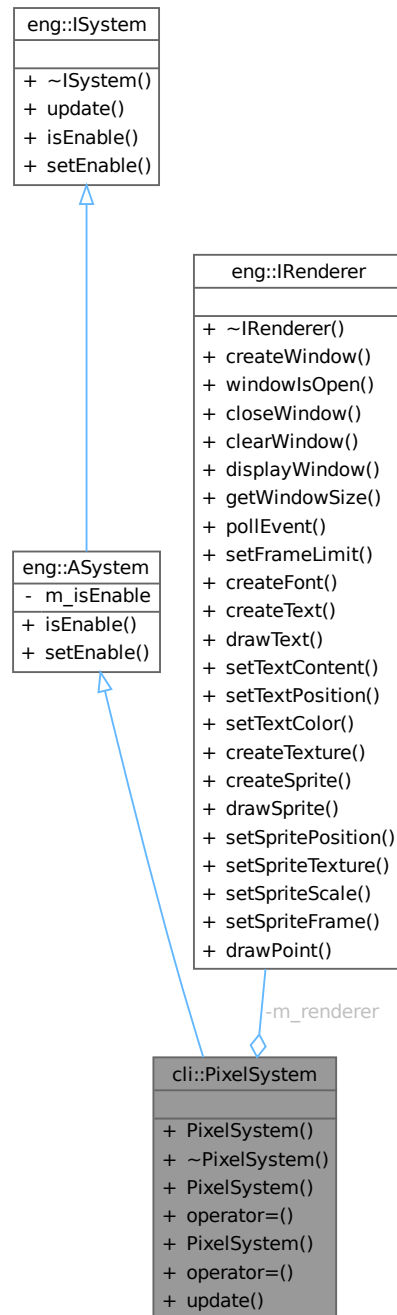
7.39 cli::PixelSystem Class Reference

#include <Systems.hpp>

Inheritance diagram for cli::PixelSystem:



Collaboration diagram for cli::PixelSystem:



Public Member Functions

- `PixelSystem` (`eng::IRenderer &renderer`)
- `~PixelSystem` () override=default
- `PixelSystem` (`const SpriteSystem &`)=delete
- `PixelSystem & operator=` (`const SpriteSystem &`)=delete
- `PixelSystem` (`SpriteSystem &&`)=delete
- `PixelSystem & operator=` (`SpriteSystem &&`)=delete
- `void update` (`ecs::Registry ®istry`, `float dt`) override

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

- bool [isEnabled](#) () override
- void [setEnabled](#) (const bool enable) override

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

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

Private Attributes

- [eng::IRenderer](#) & [m_renderer](#)

7.39.1 Detailed Description

Definition at line 143 of file [Systems.hpp](#).

7.39.2 Constructor & Destructor Documentation

7.39.2.1 PixelSystem() [1/3]

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

Definition at line 146 of file [Systems.hpp](#).

7.39.2.2 ~PixelSystem()

```
cli::PixelSystem::~~PixelSystem () [override], [default]
```

7.39.2.3 PixelSystem() [2/3]

```
cli::PixelSystem::PixelSystem (
    const SpriteSystem & ) [explicit], [delete]
```

7.39.2.4 PixelSystem() [3/3]

```
cli::PixelSystem::PixelSystem (
    SpriteSystem && ) [explicit], [delete]
```

7.39.3 Member Function Documentation

7.39.3.1 operator=() [1/2]

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

7.39.3.2 operator=() [2/2]

[PixelSystem](#) & cli::PixelSystem::operator= (
[SpriteSystem](#) &&) [delete]

7.39.3.3 update()

```
void cli::PixelSystem::update (  

    ecs::Registry & registry,  

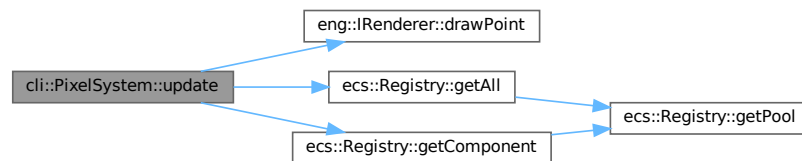
    float dt) [inline], [override], [virtual]
```

Implements [eng::ISystem](#).

Definition at line 154 of file [Systems.hpp](#).

References [eng::IRenderer::drawPoint\(\)](#), [ecs::Registry::getAll\(\)](#), [ecs::Registry::getComponent\(\)](#), [m_renderer](#), and [ecs::Color::r](#).

Here is the call graph for this function:



7.39.4 Member Data Documentation

7.39.4.1 m_renderer

[eng::IRenderer](#)& cli::PixelSystem::m_renderer [private]

Definition at line 166 of file [Systems.hpp](#).

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

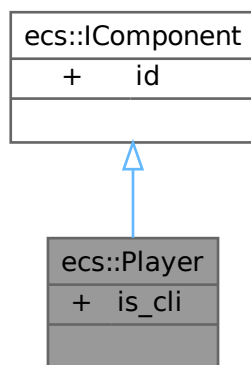
The documentation for this class was generated from the following file:

- `/home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Systems.hpp`

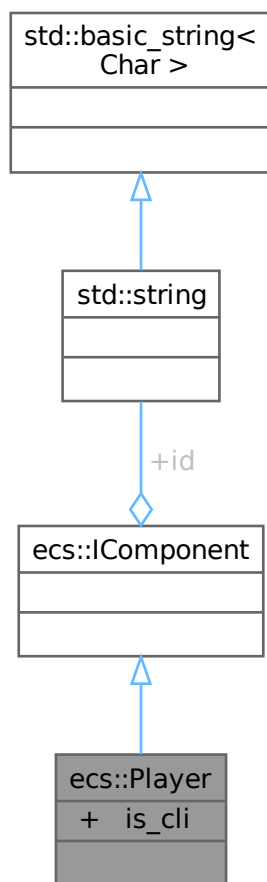
7.40 ecs::Player Struct Reference

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

Inheritance diagram for ecs::Player:



Collaboration diagram for `ecs::Player`:



Public Attributes

- `bool is_cli {}`

Public Attributes inherited from `ecs::IComponent`

- `std::string id`

7.40.1 Detailed Description

Definition at line 39 of file [Component.hpp](#).

7.40.2 Member Data Documentation

7.40.2.1 is_cli

```
bool ecs::Player::is_cli {}
```

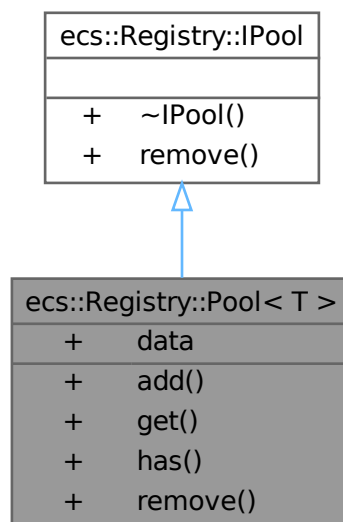
Definition at line 41 of file [Component.hpp](#).

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

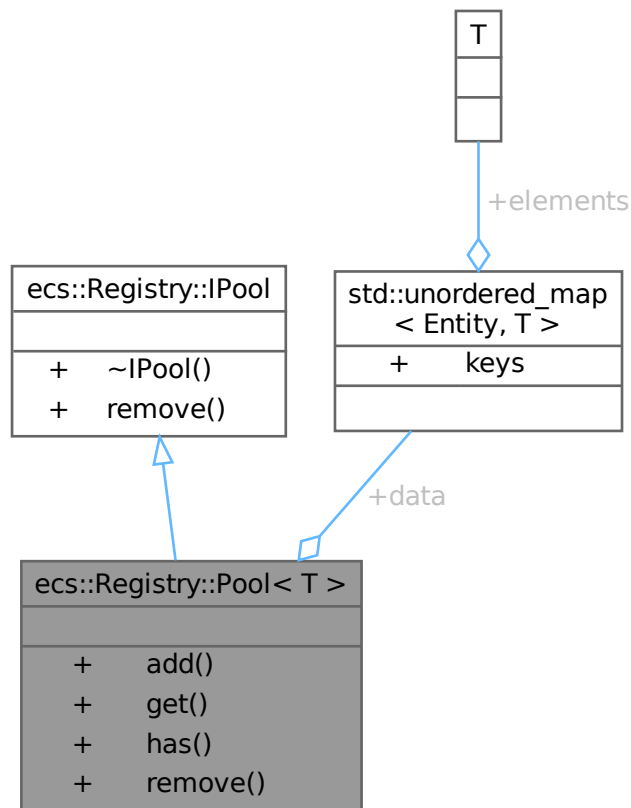
- [/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.hpp](#)

7.41 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`

7.41.1 Detailed Description

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

Definition at line 104 of file [Registry.hpp](#).

7.41.2 Member Function Documentation

7.41.2.1 add()

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

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

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

7.41.2.2 get()

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

Definition at line 114 of file [Registry.hpp](#).

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

7.41.2.3 has()

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

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

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

7.41.2.4 remove()

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

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

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

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

7.41.3 Member Data Documentation

7.41.3.1 data

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

Definition at line 107 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\(\)](#).

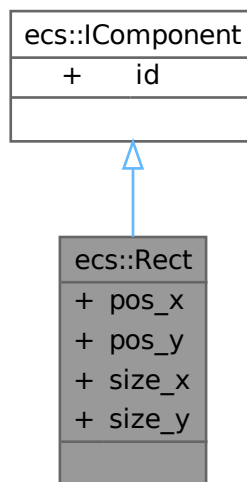
The documentation for this class was generated from the following file:

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

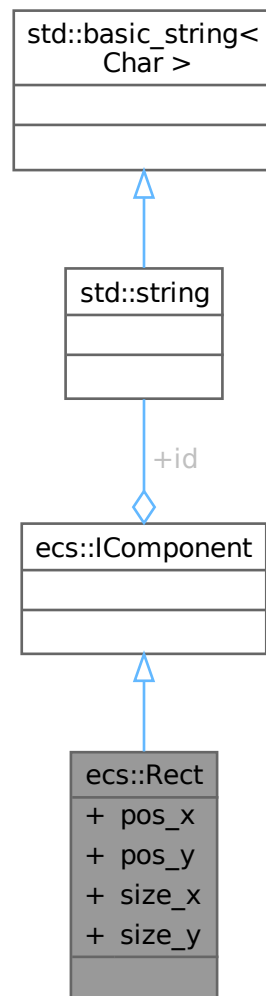
7.42 ecs::Rect Struct Reference

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

Inheritance diagram for `ecs::Rect`:



Collaboration diagram for `ecs::Rect`:



Public Attributes

- float `pos_x` {}
- float `pos_y` {}
- int `size_x` {}
- int `size_y` {}

Public Attributes inherited from `ecs::IComponent`

- `std::string id`

7.42.1 Detailed Description

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

7.42.2 Member Data Documentation

7.42.2.1 pos_x

```
float ecs::Rect::pos_x {}
```

Definition at line 49 of file [Component.hpp](#).

7.42.2.2 pos_y

```
float ecs::Rect::pos_y {}
```

Definition at line 49 of file [Component.hpp](#).

7.42.2.3 size_x

```
int ecs::Rect::size_x {}
```

Definition at line 50 of file [Component.hpp](#).

7.42.2.4 size_y

```
int ecs::Rect::size_y {}
```

Definition at line 50 of file [Component.hpp](#).

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

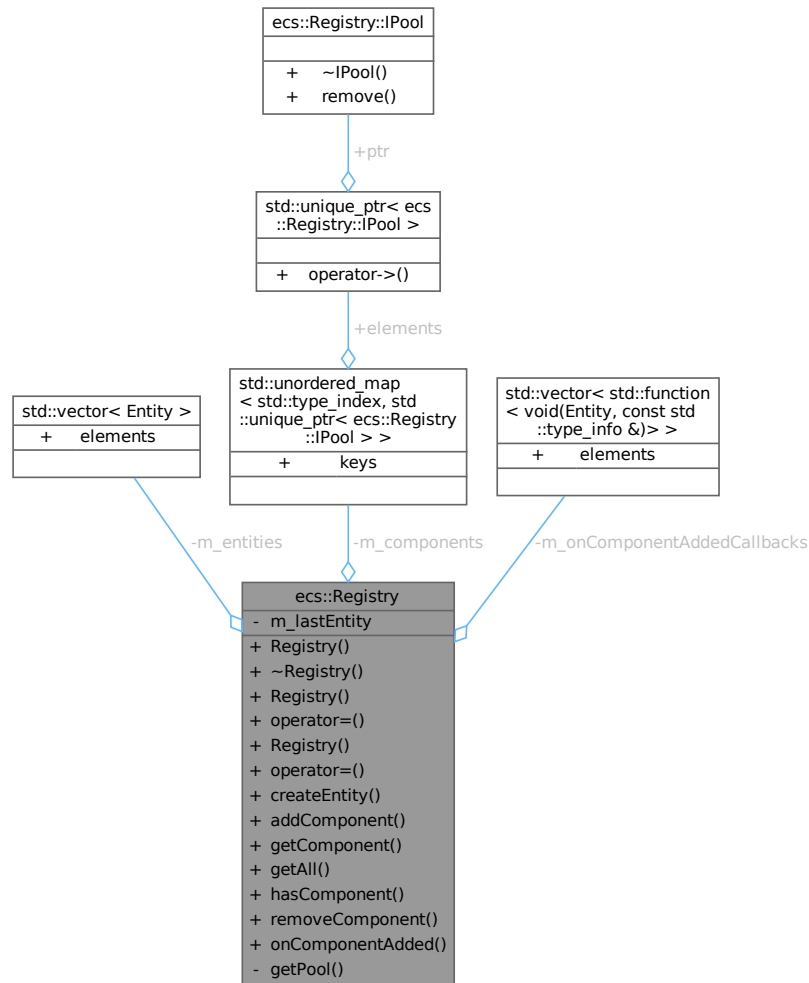
- [/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.hpp](#)

7.43 ecs::Registry Class Reference

Class for managing entities and their components.

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

Collaboration diagram for ecs::Registry:



Classes

- class [EntityBuilder](#)
- 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
- [EntityBuilder](#) [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 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, const std::type_info &)> > m_onComponentAddedCallbacks`

7.43.1 Detailed Description

Class for managing entities and their components.

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

7.43.2 Constructor & Destructor Documentation

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

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

7.43.2.2 [~Registry](#)()

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

7.43.2.3 [Registry](#)() [2/3]

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

7.43.2.4 [Registry](#)() [3/3]

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

7.43.3 Member Function Documentation

7.43.3.1 addComponent()

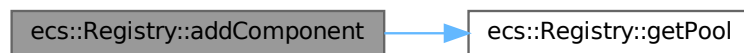
```
template<typename T , typename... Args>  
T & ecs::Registry::addComponent (   
    Entity e,  
    Args &&... args) [inline]
```

Definition at line 60 of file [Registry.hpp](#).

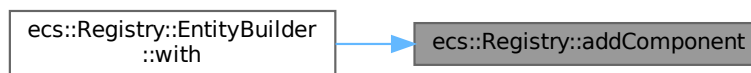
References [getPool\(\)](#), and [m_onComponentAddedCallbacks](#).

Referenced by [ecs::Registry::EntityBuilder::with\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.43.3.2 createEntity()

```
EntityBuilder ecs::Registry::createEntity () [inline]
```

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

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

7.43.3.3 getAll()

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

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

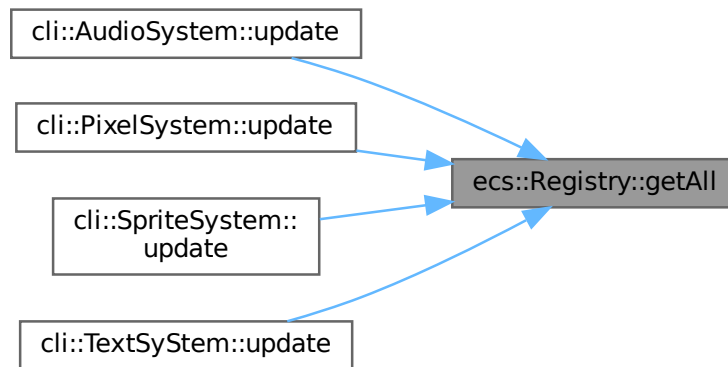
References [getPool\(\)](#).

Referenced by [cli::AudioSystem::update\(\)](#), [cli::PixelSystem::update\(\)](#), [cli::SpriteSystem::update\(\)](#), and [cli::TextSyStem::update\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.43.3.4 getComponent()

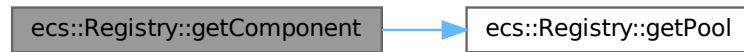
```
template<typename T >
T * ecs::Registry::getComponent (
    Entity e) [inline]
```

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

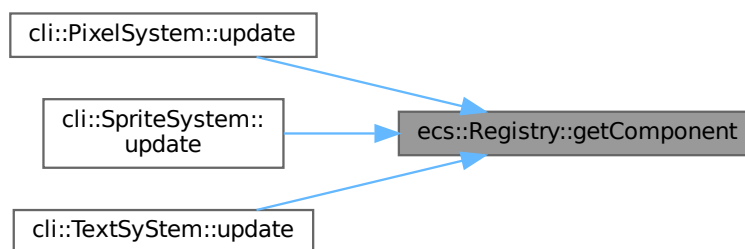
References [getPool\(\)](#).

Referenced by [cli::PixelSystem::update\(\)](#), [cli::SpriteSystem::update\(\)](#), and [cli::TextSyStem::update\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.43.3.5 getPool()

```
template<typename T >
```

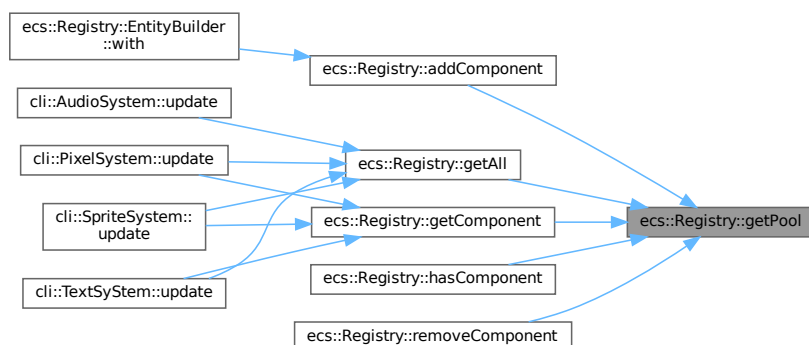
```
Pool< T > & ecs::Registry::getPool () [inline], [private]
```

Definition at line 129 of file [Registry.hpp](#).

References [m_components](#).

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

Here is the caller graph for this function:



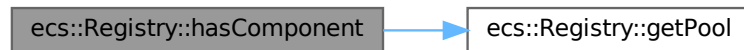
7.43.3.6 hasComponent()

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

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

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

Here is the call graph for this function:



7.43.3.7 onComponentAdded()

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

Definition at line 91 of file [Registry.hpp](#).

References [m_onComponentAddedCallbacks](#).

7.43.3.8 operator=() [1/2]

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

7.43.3.9 operator=() [2/2]

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

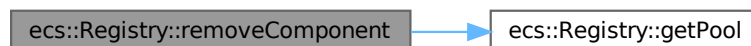
7.43.3.10 removeComponent()

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

Definition at line 85 of file [Registry.hpp](#).

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

Here is the call graph for this function:



7.43.4 Member Data Documentation

7.43.4.1 m_components

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

Definition at line 140 of file [Registry.hpp](#).

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

7.43.4.2 m_entities

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

Definition at line 139 of file [Registry.hpp](#).

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

7.43.4.3 m_lastEntity

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

Definition at line 138 of file [Registry.hpp](#).

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

7.43.4.4 m_onComponentAddedCallbacks

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

Definition at line 141 of file [Registry.hpp](#).

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

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

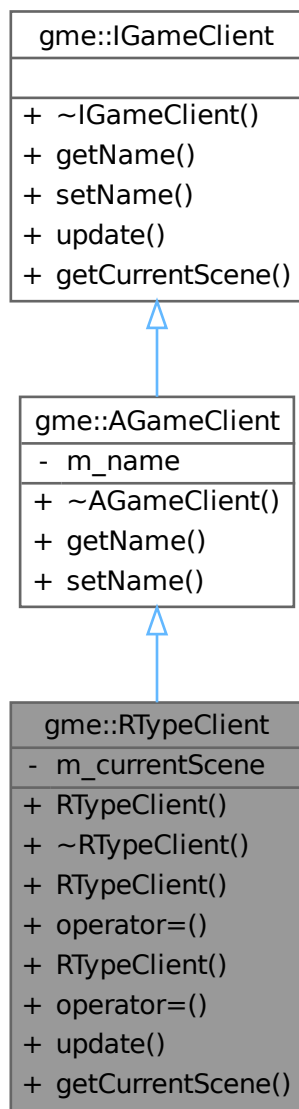
- [/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Registry.hpp](#)

7.44 gme::RTypeClient Class Reference

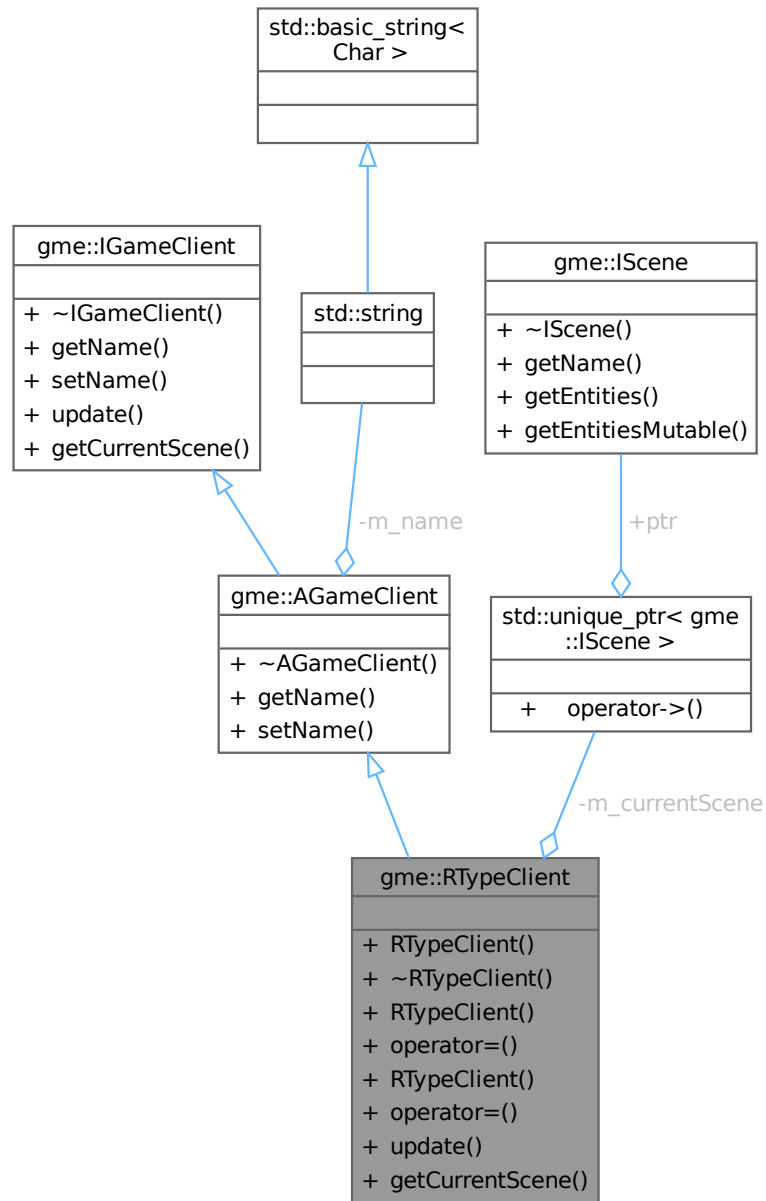
Class for the R-Type game.

```
#include <RTypeClient.hpp>
```

Inheritance diagram for gme::RTypeClient:



Collaboration diagram for gme::RTypeClient:



Public Member Functions

- [RTypeClient](#) ()
- [~RTypeClient](#) () override=default
- [RTypeClient](#) (const [RTypeClient](#) &)=delete
- [RTypeClient](#) & [operator=](#) (const [RTypeClient](#) &)=delete
- [RTypeClient](#) ([RTypeClient](#) &&)=delete
- [RTypeClient](#) & [operator=](#) ([RTypeClient](#) &&)=delete
- void [update](#) (float deltaTime, unsigned int width, unsigned int height) override
- const [IScene](#) & [getCurrentScene](#) () const override

Public Member Functions inherited from [gme::AGameClient](#)

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

Public Member Functions inherited from [gme::IGameClient](#)

- `virtual` [~IGameClient](#) ()=default

Private Attributes

- `std::unique_ptr<` [IScene](#) `>` [m_currentScene](#)

7.44.1 Detailed Description

Class for the R-Type game.

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

7.44.2 Constructor & Destructor Documentation

7.44.2.1 RTypeClient() [1/3]

`gme::RTypeClient::RTypeClient ()`

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

References [gme::AGameClient::setName\(\)](#).

Here is the call graph for this function:



7.44.2.2 ~RTypeClient()

`gme::RTypeClient::~~RTypeClient ()` [override], [default]

7.44.2.3 RTypeClient() [2/3]

`gme::RTypeClient::RTypeClient (`
 const [RTypeClient](#) &) [delete]

7.44.2.4 RTypeClient() [3/3]

`gme::RTypeClient::RTypeClient (`
 [RTypeClient](#) &&) [delete]

7.44.3 Member Function Documentation

7.44.3.1 getCurrentScene()

`const IScene & gme::RTypeClient::getCurrentScene () const` [inline], [nodiscard], [override], [virtual]

Implements [gme::IGameClient](#).

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

References [m_currentScene](#).

7.44.3.2 operator=() [1/2]

`RTypeClient & gme::RTypeClient::operator= (`
 const [RTypeClient](#) &) [delete]

7.44.3.3 operator=() [2/2]

`RTypeClient & gme::RTypeClient::operator= (`
 [RTypeClient](#) &&) [delete]

7.44.3.4 update()

`void gme::RTypeClient::update (`
 float deltaTime,
 unsigned int width,
 unsigned int height) [override], [virtual]

Implements [gme::IGameClient](#).

Definition at line 6 of file [rtypeClient.cpp](#).

7.44.4 Member Data Documentation

7.44.4.1 m_currentScene

`std::unique_ptr<IScene> gme::RTypeClient::m_currentScene` [private]

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

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

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

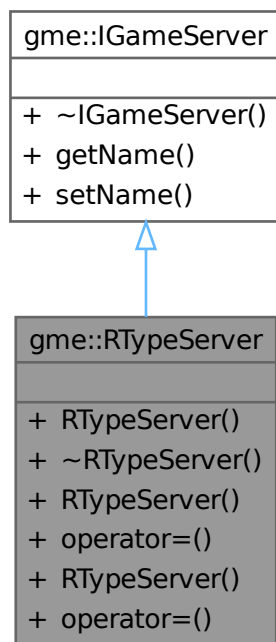
- [/home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Client/include/R-Type/Client/RTypeClient.hpp](#)
- [/home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Client/src/rtypeClient.cpp](#)

7.45 gme::RTypeServer Class Reference

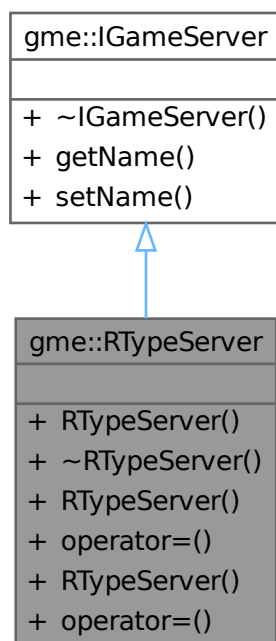
Class for the R-Type game.

```
#include <RTypeServer.hpp>
```

Inheritance diagram for gme::RTypeServer:



Collaboration diagram for gme::RTypeServer:



Public Member Functions

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

Public Member Functions inherited from [gme::IGameServer](#)

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

7.45.1 Detailed Description

Class for the R-Type game.

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

7.45.2 Constructor & Destructor Documentation

7.45.2.1 RTypeServer() [1/3]

`gme::RTypeServer::RTypeServer ()` [default]

7.45.2.2 ~RTypeServer()

`gme::RTypeServer::~~RTypeServer ()` [override], [default]

7.45.2.3 RTypeServer() [2/3]

`gme::RTypeServer::RTypeServer (
 const RTypeServer &)` [delete]

7.45.2.4 RTypeServer() [3/3]

`gme::RTypeServer::RTypeServer (
 RTypeServer &&)` [delete]

7.45.3 Member Function Documentation

7.45.3.1 operator=() [1/2]

`RTypeServer & gme::RTypeServer::operator= (
 const RTypeServer &)` [delete]

7.45.3.2 operator=() [2/2]

`RTypeServer & gme::RTypeServer::operator= (
 RTypeServer &&)` [delete]

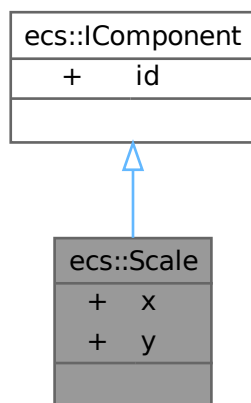
The documentation for this class was generated from the following file:

- `/home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Server/include/R-TypeServer/RTypeServer.hpp`

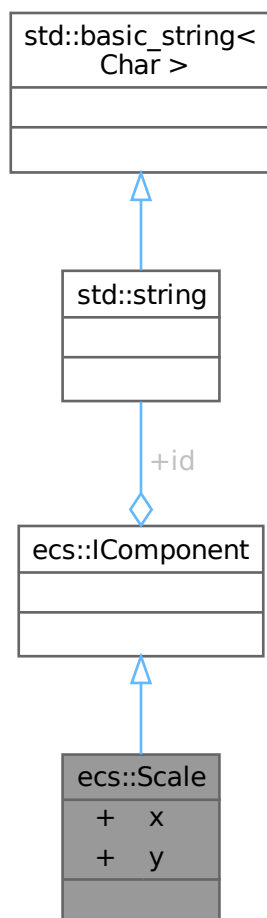
7.46 ecs::Scale Struct Reference

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

Inheritance diagram for ecs::Scale:



Collaboration diagram for `ecs::Scale`:



Public Attributes

- float `x` {}
- float `y` {}

Public Attributes inherited from `ecs::IComponent`

- `std::string id`

7.46.1 Detailed Description

Definition at line 52 of file `Component.hpp`.

7.46.2 Member Data Documentation

7.46.2.1 x

```
float ecs::Scale::x {}
```

Definition at line 54 of file [Component.hpp](#).

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

7.46.2.2 y

```
float ecs::Scale::y {}
```

Definition at line 54 of file [Component.hpp](#).

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

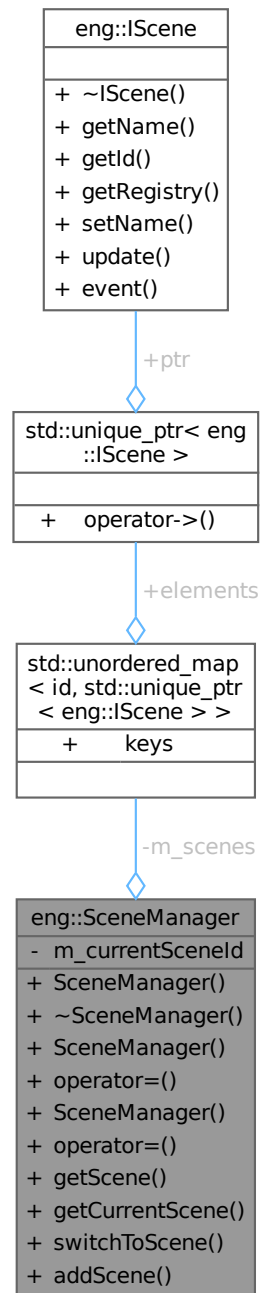
- [/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.hpp](#)

7.47 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< IScene > & getScene (const id sceneId)`
- `std::unique_ptr< IScene > & getCurrentScene ()`
- `void switchToScene (const id sceneId)`
- `void addScene (std::unique_ptr< IScene > scene)`

Private Attributes

- `std::unordered_map< id, std::unique_ptr< IScene > > m_scenes`
- `id m_currentSceneId = 1`

7.47.1 Detailed Description

Class for managing scenes.

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

7.47.2 Constructor & Destructor Documentation

7.47.2.1 SceneManager() [1/3]

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

7.47.2.2 ~SceneManager()

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

7.47.2.3 SceneManager() [2/3]

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

7.47.2.4 SceneManager() [3/3]

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

7.47.3 Member Function Documentation

7.47.3.1 addScene()

`void eng::SceneManager::addScene (
std::unique_ptr< IScene > scene)` [inline]

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

References [m_scenes](#).

7.47.3.2 getCurrentScene()

```
std::unique_ptr< IScene > & eng::SceneManager::getCurrentScene () [inline]
```

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

References [m_currentSceneId](#), and [m_scenes](#).

7.47.3.3 getScene()

```
std::unique_ptr< IScene > & eng::SceneManager::getScene (  
    const id sceneId) [inline]
```

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

References [m_scenes](#).

7.47.3.4 operator=() [1/2]

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

7.47.3.5 operator=() [2/2]

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

7.47.3.6 switchToScene()

```
void eng::SceneManager::switchToScene (  
    const id sceneId) [inline]
```

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

References [m_currentSceneId](#).

7.47.4 Member Data Documentation

7.47.4.1 m_currentSceneId

```
id eng::SceneManager::m_currentSceneId = 1 [private]
```

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

Referenced by [getCurrentScene\(\)](#), and [switchToScene\(\)](#).

7.47.4.2 m_scenes

std::unordered_map<id, std::unique_ptr<IScene> > eng::SceneManager::m_scenes [private]

Definition at line 56 of file [SceneManager.hpp](#).

Referenced by [addScene\(\)](#), [getCurrentScene\(\)](#), and [getScene\(\)](#).

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

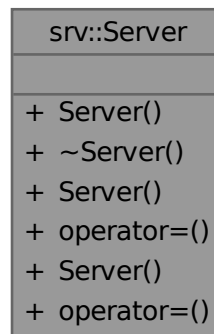
- [/home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/SceneManager.hpp](#)

7.48 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

7.48.1 Detailed Description

Class for the server.

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

7.48.2 Constructor & Destructor Documentation

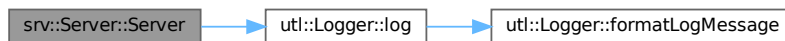
7.48.2.1 Server() [1/3]

```
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:



7.48.2.2 ~Server()

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

7.48.2.3 Server() [2/3]

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

7.48.2.4 Server() [3/3]

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

7.48.3 Member Function Documentation

7.48.3.1 operator=() [1/2]

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

7.48.3.2 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](#)

7.49 eng::SFMLAudio Class Reference

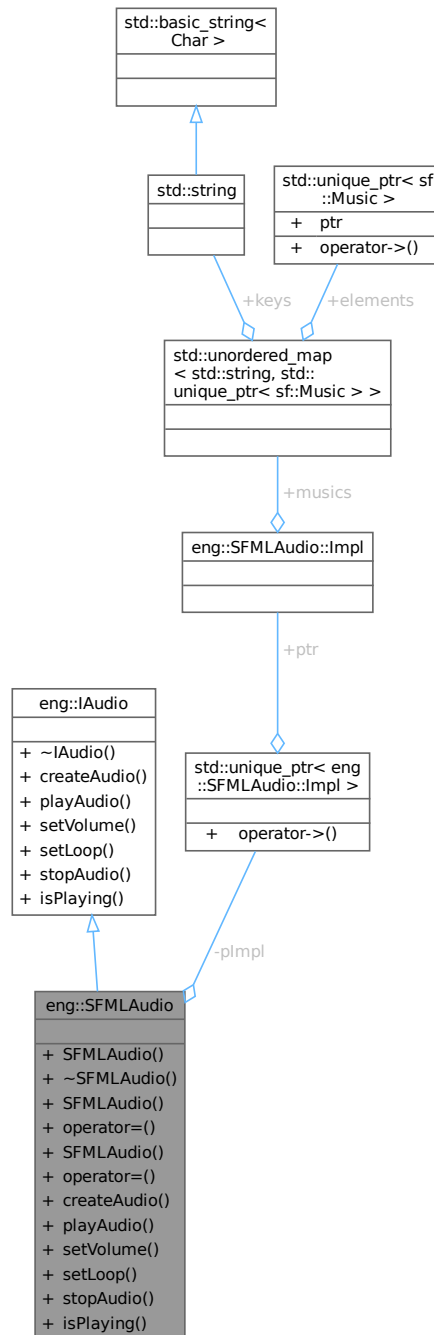
Class for audio management.

```
#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
- void [stopAudio](#) (const std::string &name) override
- [Status](#) [isPlaying](#) (const std::string &name) override

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

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

Private Attributes

- std::unique_ptr< [Impl](#) > [pImpl](#)

7.49.1 Detailed Description

Class for audio management.

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

7.49.2 Constructor & Destructor Documentation

7.49.2.1 SFMLAudio() [1/3]

[eng::SFMLAudio::SFMLAudio](#) ()

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

7.49.2.2 ~SFMLAudio()

[eng::SFMLAudio::~SFMLAudio](#) () [override], [default]

7.49.2.3 SFMLAudio() [2/3]

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

7.49.2.4 SFMLAudio() [3/3]

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

7.49.3 Member Function Documentation

7.49.3.1 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](#).

7.49.3.2 isPlaying()

```
Status eng::SFMLAudio::isPlaying (
    const std::string & name) [override], [virtual]
```

Implements [eng::IAudio](#).

Definition at line 56 of file [SFMLAudio.cpp](#).

References [eng::Paused](#), [pImpl](#), [eng::Playing](#), and [eng::Stopped](#).

7.49.3.3 operator=() [1/2]

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

7.49.3.4 operator=() [2/2]

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

7.49.3.5 playAudio()

```
void eng::SFMLAudio::playAudio (
    const std::string & name) [override], [virtual]
```

Implements [eng::IAudio](#).

Definition at line 30 of file [SFMLAudio.cpp](#).

References [pImpl](#).

7.49.3.6 setLoop()

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

Implements [eng::IAudio](#).

Definition at line 42 of file [SFMLAudio.cpp](#).

References [pImpl](#).

7.49.3.7 setVolume()

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

Implements [eng::IAudio](#).

Definition at line 36 of file [SFMLAudio.cpp](#).

References [pImpl](#).

7.49.3.8 stopAudio()

```
void eng::SFMLAudio::stopAudio (
    const std::string & name)    [override], [virtual]
```

Implements [eng::IAudio](#).

Definition at line 48 of file [SFMLAudio.cpp](#).

References [pImpl](#).

7.49.4 Member Data Documentation

7.49.4.1 pImpl

```
std::unique_ptr<Impl> eng::SFMLAudio::pImpl    [private]
```

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

Referenced by [createAudio\(\)](#), [isPlaying\(\)](#), [playAudio\(\)](#), [setLoop\(\)](#), [setVolume\(\)](#), and [stopAudio\(\)](#).

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

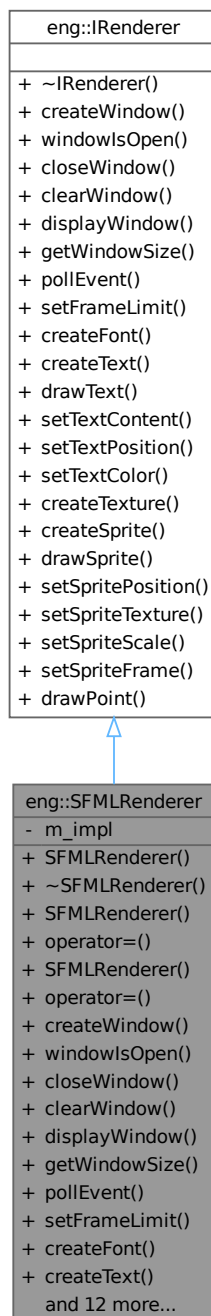
- [/home/masina/Projects/Epitech/rtype/modules/Audio/SFMLAudio/include/SFMLAudio/SFMLAudio.hpp](#)
- [/home/masina/Projects/Epitech/rtype/modules/Audio/SFMLAudio/src/SFMLAudio.cpp](#)

7.50 eng::SFMLRenderer Class Reference

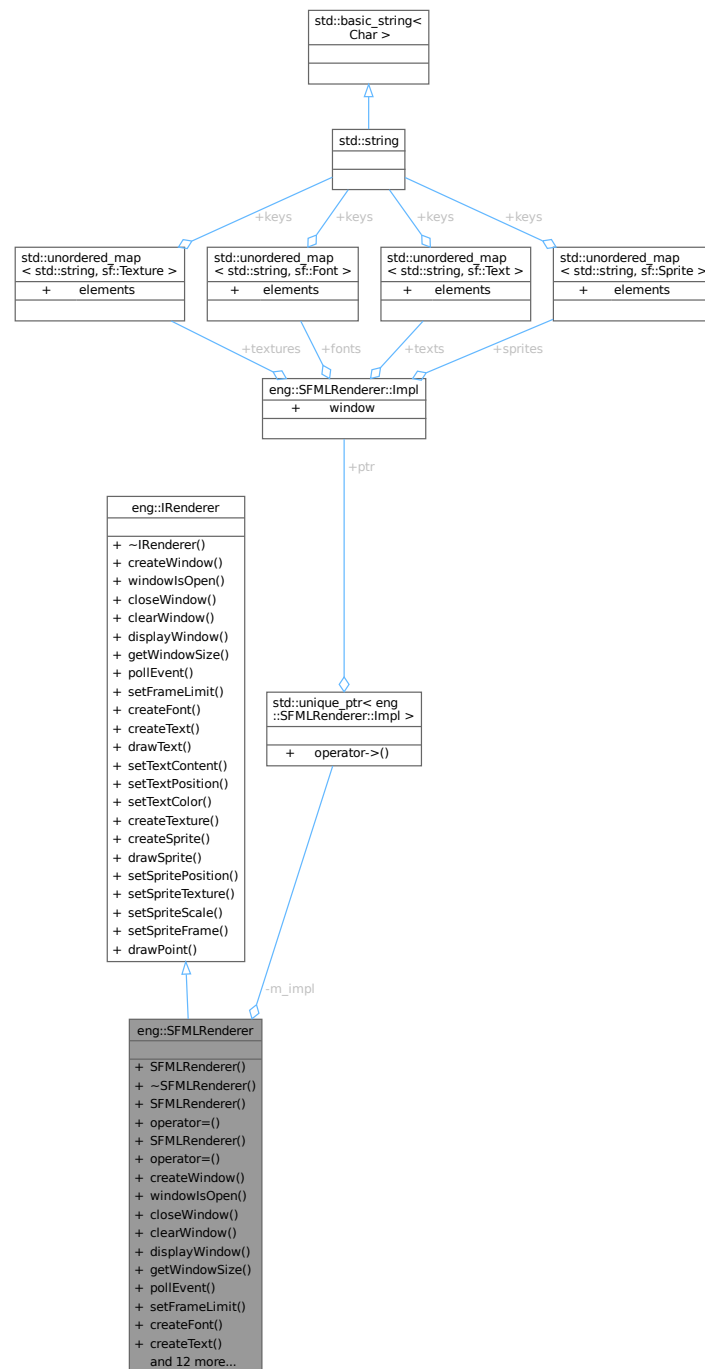
Class for the R-Type game.

```
#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
- bool [windowIsOpen](#) () const override
- void [closeWindow](#) () override
- void [clearWindow](#) ([Color](#) color) override
- void [displayWindow](#) () override
- [WindowSize](#) [getWindowSize](#) () override
- bool [pollEvent](#) ([Event](#) &event) override
- void [setFrameLimit](#) (unsigned int frameLimit) override
- void [createFont](#) (const std::string &name, const std::string &path) override
- void [createText](#) ([Text](#) text) override
- void [setTextContent](#) (const std::string &name, const std::string &content) override
- void [setTextPosition](#) (const std::string &name, float x, float y) override
- void [setTextColor](#) (const std::string &name, [Color](#) color) override
- void [drawText](#) (const std::string &name) override
- void [createTexture](#) (const std::string &name, const std::string &path) override
- void [createSprite](#) (const std::string &name, const std::string &textureName, float x, float y, float scale_x, float scale_y, int fx, int fy, int fnx, int fny) override
- void [setSpritePosition](#) (const std::string &name, float x, float y) override
- void [setSpriteTexture](#) (const std::string &name, const std::string &path) override
- void [setSpriteFrame](#) (const std::string &name, int fx, int fy, int fnx, int fny) override
- void [setSpriteScale](#) (const std::string &name, int x, int y) override
- void [drawSprite](#) (const std::string &name) override
- void [drawPoint](#) (float x, float y, [Color](#) color) override

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

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

Private Attributes

- std::unique_ptr< [Impl](#) > [m_impl](#)

7.50.1 Detailed Description

Class for the R-Type game.

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

7.50.2 Constructor & Destructor Documentation

7.50.2.1 SFMLRenderer() [1/3]

[eng::SFMLRenderer::SFMLRenderer](#) ()

Definition at line 19 of file [SFMLRenderer.cpp](#).

7.50.2.2 ~SFMLRenderer()

eng::SFMLRenderer::~SFMLRenderer () [override], [default]

7.50.2.3 SFMLRenderer() [2/3]

eng::SFMLRenderer::~SFMLRenderer (
 const [SFMLRenderer](#) &) [delete]

7.50.2.4 SFMLRenderer() [3/3]

eng::SFMLRenderer::~SFMLRenderer (
 [SFMLRenderer](#) &&) [delete]

7.50.3 Member Function Documentation

7.50.3.1 clearWindow()

void eng::SFMLRenderer::clearWindow (
 [Color](#) color) [override], [virtual]

Implements [eng::IRenderer](#).

Definition at line 106 of file [SFMLRenderer.cpp](#).

References [eng::Color::a](#), [eng::Color::b](#), [eng::Color::g](#), and [eng::Color::r](#).

7.50.3.2 closeWindow()

void eng::SFMLRenderer::closeWindow () [override], [virtual]

Implements [eng::IRenderer](#).

Definition at line 33 of file [SFMLRenderer.cpp](#).

7.50.3.3 createFont()

void eng::SFMLRenderer::createFont (
 const std::string & name,
 const std::string & path) [override], [virtual]

Implements [eng::IRenderer](#).

Definition at line 37 of file [SFMLRenderer.cpp](#).

7.50.3.4 createSprite()

```
void eng::SFMLRenderer::createSprite (  
    const std::string & name,  
    const std::string & textureName,  
    float x,  
    float y,  
    float scale_x,  
    float scale_y,  
    int fx,  
    int fy,  
    int fnx,  
    int fny) [override], [virtual]
```

Implements [eng::IRenderer](#).

Definition at line 241 of file [SFMLRenderer.cpp](#).

7.50.3.5 createText()

```
void eng::SFMLRenderer::createText (  
    Text text) [override], [virtual]
```

Implements [eng::IRenderer](#).

Definition at line 47 of file [SFMLRenderer.cpp](#).

References [eng::Color::a](#), [eng::Color::b](#), [eng::Text::color](#), [eng::Text::content](#), [eng::Text::font_name](#), [eng::Color::g](#), [eng::Text::name](#), [eng::Color::r](#), [eng::Text::size](#), [eng::Text::x](#), and [eng::Text::y](#).

7.50.3.6 createTexture()

```
void eng::SFMLRenderer::createTexture (  
    const std::string & name,  
    const std::string & path) [override], [virtual]
```

Implements [eng::IRenderer](#).

Definition at line 260 of file [SFMLRenderer.cpp](#).

7.50.3.7 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 23 of file [SFMLRenderer.cpp](#).

7.50.3.8 displayWindow()

void eng::SFMLRenderer::displayWindow () [override], [virtual]

Implements [eng::IRenderer](#).

Definition at line 111 of file [SFMLRenderer.cpp](#).

7.50.3.9 drawPoint()

```
void eng::SFMLRenderer::drawPoint (
    float x,
    float y,
    Color color) [override], [virtual]
```

Implements [eng::IRenderer](#).

Definition at line 343 of file [SFMLRenderer.cpp](#).

References [eng::Color::a](#), [eng::Color::b](#), [eng::Color::g](#), and [eng::Color::r](#).

7.50.3.10 drawSprite()

```
void eng::SFMLRenderer::drawSprite (
    const std::string & name) [override], [virtual]
```

Implements [eng::IRenderer](#).

Definition at line 275 of file [SFMLRenderer.cpp](#).

7.50.3.11 drawText()

```
void eng::SFMLRenderer::drawText (
    const std::string & name) [override], [virtual]
```

Implements [eng::IRenderer](#).

Definition at line 94 of file [SFMLRenderer.cpp](#).

7.50.3.12 getWindowSize()

[eng::WindowSize](#) eng::SFMLRenderer::getWindowSize () [override], [virtual]

Implements [eng::IRenderer](#).

Definition at line 349 of file [SFMLRenderer.cpp](#).

References [eng::WindowSize::width](#).

7.50.3.13 operator=() [1/2]

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

7.50.3.14 operator=() [2/2]

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

7.50.3.15 pollEvent()

bool eng::SFMLRenderer::pollEvent (
[Event](#) & event) [override], [virtual]

Implements [eng::IRenderer](#).

Definition at line 207 of file [SFMLRenderer.cpp](#).

References [eng::Closed](#), [eng::KeyPressed](#), [eng::KeyReleased](#), [eng::None](#), and [scancodeToKey\(\)](#).

Here is the call graph for this function:



7.50.3.16 setFrameLimit()

void eng::SFMLRenderer::setFrameLimit (
 unsigned int frameLimit) [override], [virtual]

Implements [eng::IRenderer](#).

Definition at line 35 of file [SFMLRenderer.cpp](#).

7.50.3.17 setSpriteFrame()

void eng::SFMLRenderer::setSpriteFrame (
 const std::string & name,
 int fx,
 int fy,
 int fnx,
 int fny) [override], [virtual]

Implements [eng::IRenderer](#).

Definition at line 319 of file [SFMLRenderer.cpp](#).

7.50.3.18 setSpritePosition()

```
void eng::SFMLRenderer::setSpritePosition (  
    const std::string & name,  
    float x,  
    float y) [override], [virtual]
```

Implements [eng::IRenderer](#).

Definition at line 287 of file [SFMLRenderer.cpp](#).

7.50.3.19 setSpriteScale()

```
void eng::SFMLRenderer::setSpriteScale (  
    const std::string & name,  
    int x,  
    int y) [override], [virtual]
```

Implements [eng::IRenderer](#).

Definition at line 331 of file [SFMLRenderer.cpp](#).

7.50.3.20 setSpriteTexture()

```
void eng::SFMLRenderer::setSpriteTexture (  
    const std::string & name,  
    const std::string & path) [override], [virtual]
```

Implements [eng::IRenderer](#).

Definition at line 299 of file [SFMLRenderer.cpp](#).

7.50.3.21 setTextColor()

```
void eng::SFMLRenderer::setTextColor (  
    const std::string & name,  
    Color color) [override], [virtual]
```

Implements [eng::IRenderer](#).

Definition at line 82 of file [SFMLRenderer.cpp](#).

References [eng::Color::a](#), [eng::Color::b](#), [eng::Color::g](#), and [eng::Color::r](#).

7.50.3.22 setTextContent()

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

Implements [eng::IRenderer](#).

Definition at line 58 of file [SFMLRenderer.cpp](#).

7.50.3.23 setTexturePosition()

```
void eng::SFMLRenderer::setTexturePosition (
    const std::string & name,
    float x,
    float y)  [override], [virtual]
```

Implements [eng::IRenderer](#).

Definition at line 70 of file [SFMLRenderer.cpp](#).

7.50.3.24 windowIsOpen()

```
bool eng::SFMLRenderer::windowIsOpen () const  [override], [virtual]
```

Implements [eng::IRenderer](#).

Definition at line 31 of file [SFMLRenderer.cpp](#).

7.50.4 Member Data Documentation

7.50.4.1 m_impl

```
std::unique_ptr<Impl> eng::SFMLRenderer::m_impl  [private]
```

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

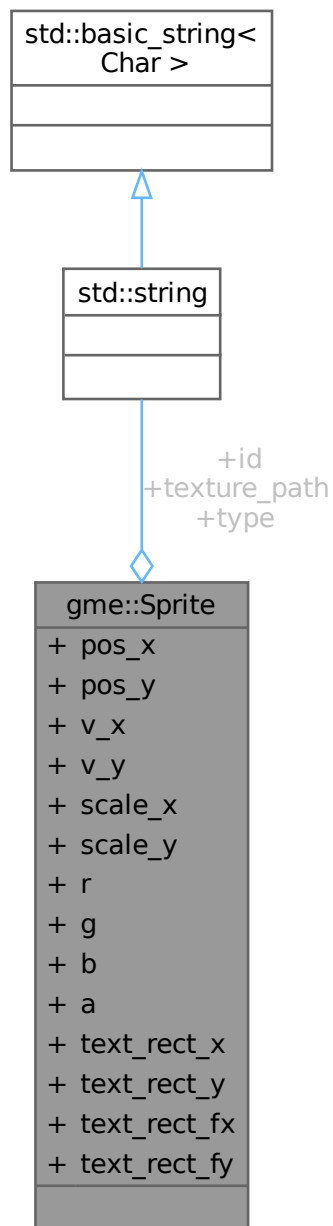
The documentation for this class was generated from the following files:

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

7.51 gme::Sprite Struct Reference

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

Collaboration diagram for gme::Sprite:



Public Attributes

- std::string [type](#)
- float [pos_x](#) = 0.F
- float [pos_y](#) = 0.F
- float [v_x](#) = 0.F
- float [v_y](#) = 0.F
- float [scale_x](#) = 1.F

- float `scale_y` = 1.F
- unsigned char `r` = 255u
- unsigned char `g` = 255u
- unsigned char `b` = 255u
- unsigned char `a` = 255u
- std::string `texture_path` = ""
- float `text_rect_x` = 0.F
- float `text_rect_y` = 0.F
- int `text_rect_fx` = 0
- int `text_rect_fy` = 0
- std::string `id`

7.51.1 Detailed Description

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

7.51.2 Member Data Documentation

7.51.2.1 a

unsigned char gme::Sprite::a = 255u

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

7.51.2.2 b

unsigned char gme::Sprite::b = 255u

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

7.51.2.3 g

unsigned char gme::Sprite::g = 255u

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

7.51.2.4 id

std::string gme::Sprite::id

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

7.51.2.5 pos_x

float gme::Sprite::pos_x = 0.F

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

7.51.2.6 pos_y

```
float gme::Sprite::pos_y = 0.F
```

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

7.51.2.7 r

```
unsigned char gme::Sprite::r = 255u
```

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

7.51.2.8 scale_x

```
float gme::Sprite::scale_x = 1.F
```

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

7.51.2.9 scale_y

```
float gme::Sprite::scale_y = 1.F
```

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

7.51.2.10 text_rect_fx

```
int gme::Sprite::text_rect_fx = 0
```

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

7.51.2.11 text_rect_fy

```
int gme::Sprite::text_rect_fy = 0
```

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

7.51.2.12 text_rect_x

```
float gme::Sprite::text_rect_x = 0.F
```

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

7.51.2.13 text_rect_y

```
float gme::Sprite::text_rect_y = 0.F
```

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

7.51.2.14 texture_path

```
std::string gme::Sprite::texture_path = ""
```

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

7.51.2.15 type

```
std::string gme::Sprite::type
```

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

7.51.2.16 v_x

```
float gme::Sprite::v_x = 0.F
```

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

7.51.2.17 v_y

```
float gme::Sprite::v_y = 0.F
```

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

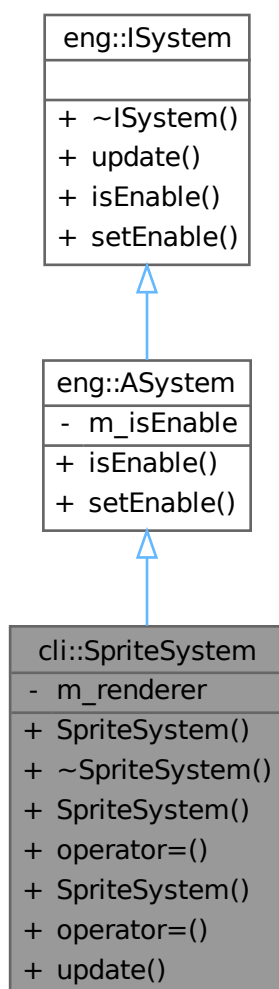
The documentation for this struct was generated from the following file:

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

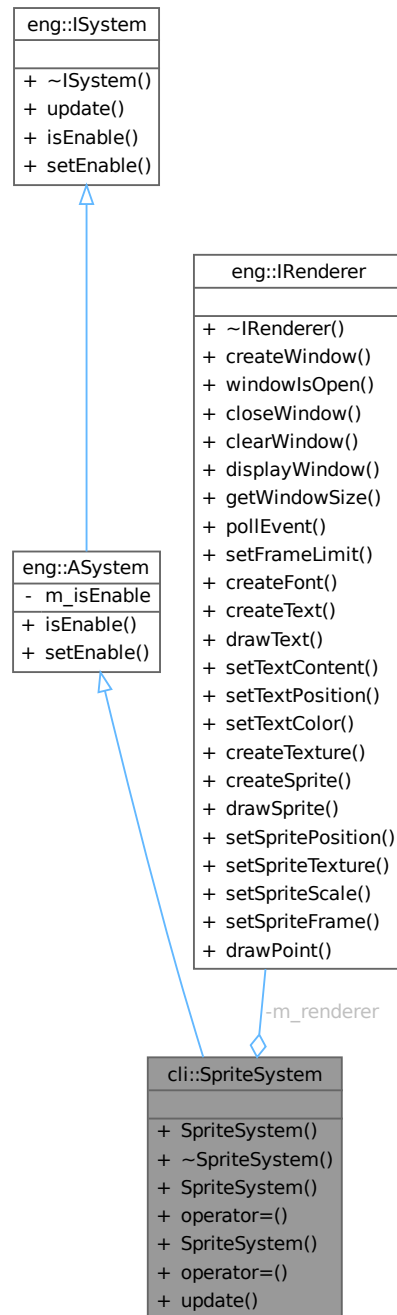
7.52 cli::SpriteSystem Class Reference

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

Inheritance diagram for cli::SpriteSystem:



Collaboration diagram for cli::SpriteSystem:



Public Member Functions

- `SpriteSystem` (`eng::IRenderer &renderer`)
- `~SpriteSystem` () override=default
- `SpriteSystem` (`const SpriteSystem &`)=delete
- `SpriteSystem & operator=` (`const SpriteSystem &`)=delete
- `SpriteSystem` (`SpriteSystem &&`)=delete
- `SpriteSystem & operator=` (`SpriteSystem &&`)=delete
- `void update` (`ecs::Registry ®istry`, `float dt`) override

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

- bool [isEnabled](#) () override
- void [setEnabled](#) (const bool enable) override

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

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

Private Attributes

- [eng::IRenderer](#) & [m_renderer](#)

7.52.1 Detailed Description

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

7.52.2 Constructor & Destructor Documentation

7.52.2.1 SpriteSystem() [1/3]

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

Definition at line 103 of file [Systems.hpp](#).

7.52.2.2 ~SpriteSystem()

```
cli::SpriteSystem::~SpriteSystem () [override], [default]
```

7.52.2.3 SpriteSystem() [2/3]

```
cli::SpriteSystem::SpriteSystem (  
    const SpriteSystem & ) [delete]
```

7.52.2.4 SpriteSystem() [3/3]

```
cli::SpriteSystem::SpriteSystem (  
    SpriteSystem && ) [delete]
```

7.52.3 Member Function Documentation

7.52.3.1 operator=() [1/2]

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

7.52.3.2 operator=() [2/2]

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

7.52.3.3 update()

```
void cli::SpriteSystem::update (  

    ecs::Registry & registry,  

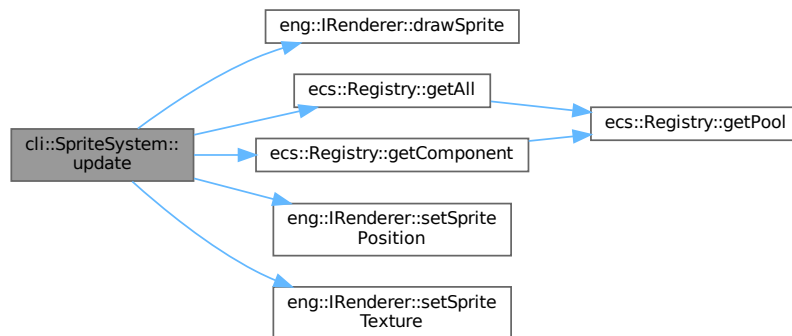
    float dt) [inline], [override], [virtual]
```

Implements [eng::ISystem](#).

Definition at line 111 of file [Systems.hpp](#).

References [eng::IRenderer::drawSprite\(\)](#), [ecs::Registry::getAll\(\)](#), [ecs::Registry::getComponent\(\)](#), [m_renderer](#), [eng::IRenderer::setSpritePosition\(\)](#), [eng::IRenderer::setSpriteTexture\(\)](#), and [ecs::Velocity::x](#).

Here is the call graph for this function:



7.52.4 Member Data Documentation

7.52.4.1 m_renderer

[eng::IRenderer](#)& cli::SpriteSystem::m_renderer [private]

Definition at line 140 of file [Systems.hpp](#).

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

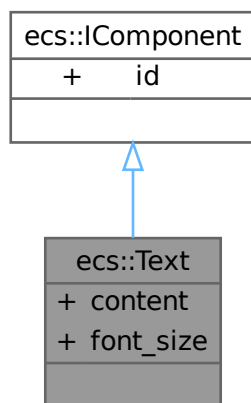
The documentation for this class was generated from the following file:

- `/home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Systems.hpp`

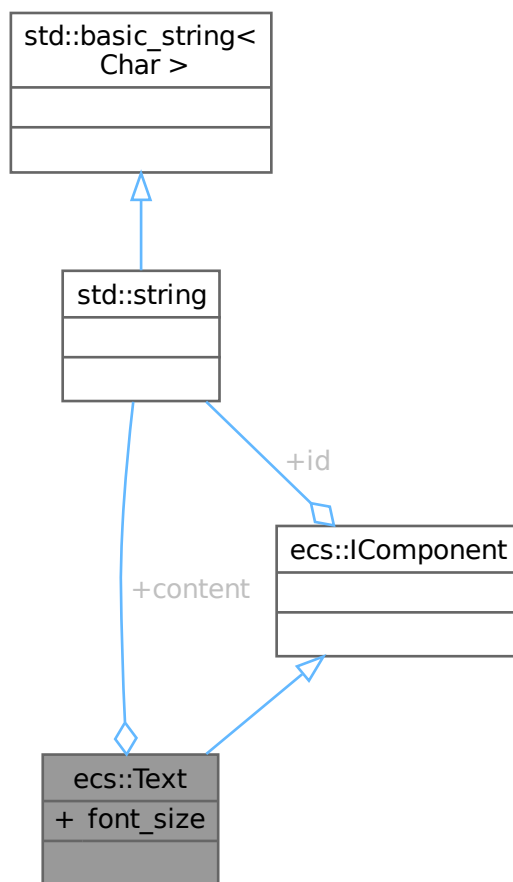
7.53 ecs::Text Struct Reference

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

Inheritance diagram for ecs::Text:



Collaboration diagram for `ecs::Text`:



Public Attributes

- `std::string` `content`
- `unsigned int` `font_size`

Public Attributes inherited from `ecs::IComponent`

- `std::string` `id`

7.53.1 Detailed Description

Definition at line 56 of file `Component.hpp`.

7.53.2 Member Data Documentation

7.53.2.1 content

std::string ecs::Text::content

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

7.53.2.2 font_size

unsigned int ecs::Text::font_size

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

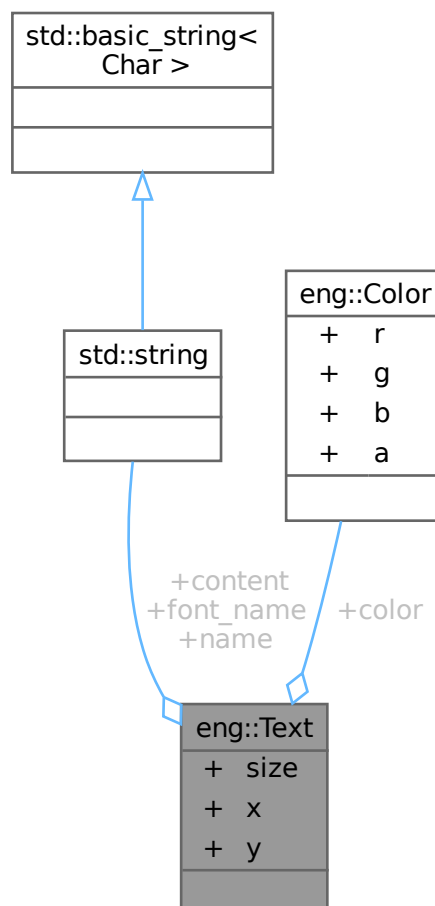
The documentation for this struct was generated from the following file:

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

7.54 eng::Text Struct Reference

#include <IRenderer.hpp>

Collaboration diagram for eng::Text:



Public Attributes

- `std::string` [font_name](#)
- [Color](#) `color`
- `std::string` `content`
- `unsigned int` `size`
- `float` `x`
- `float` `y`
- `std::string` `name`

7.54.1 Detailed Description

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

7.54.2 Member Data Documentation

7.54.2.1 color

[Color](#) `eng::Text::color`

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

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

7.54.2.2 content

`std::string` `eng::Text::content`

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

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

7.54.2.3 font_name

`std::string` `eng::Text::font_name`

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

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

7.54.2.4 name

`std::string` `eng::Text::name`

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

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

7.54.2.5 size

unsigned int eng::Text::size

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

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

7.54.2.6 x

float eng::Text::x

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

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

7.54.2.7 y

float eng::Text::y

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

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

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

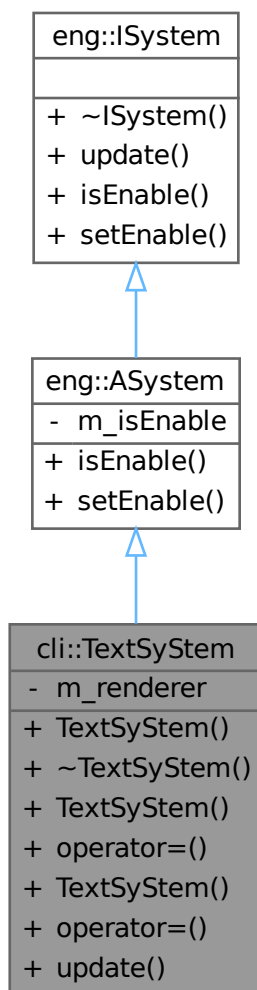
- [/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IRenderer.hpp](#)

7.55 cli::TextSyStem Class Reference

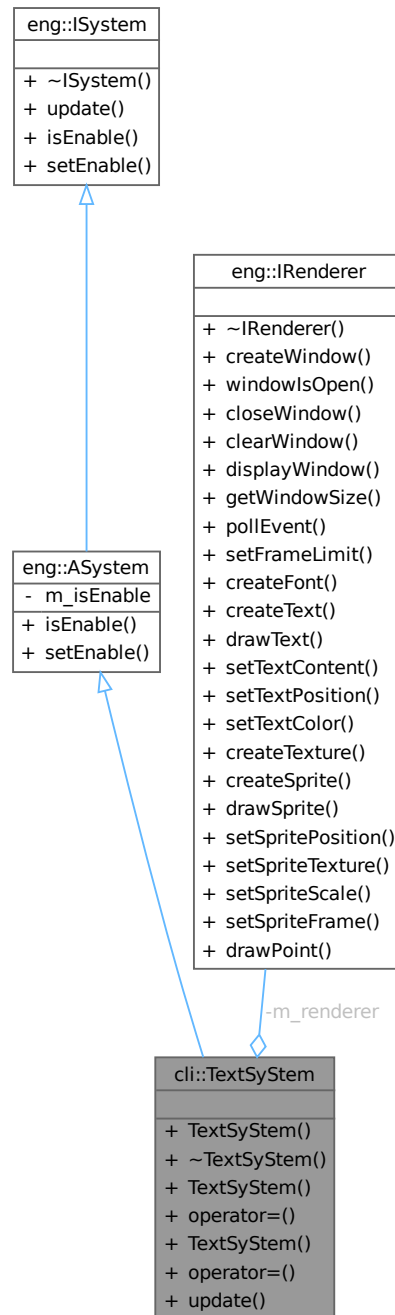
Class for managing entities and their components.

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

Inheritance diagram for cli::TextSyStem:



Collaboration diagram for cli::TextSyStem:



Public Member Functions

- `TextSyStem` (`eng::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

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

- bool [isEnabled](#) () override
- void [setEnabled](#) (const bool enable) override

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

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

Private Attributes

- [eng::IRenderer](#) & [m_renderer](#)

7.55.1 Detailed Description

Class for managing entities and their components.

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

7.55.2 Constructor & Destructor Documentation

7.55.2.1 [TextSyStem](#)() [1/3]

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

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

7.55.2.2 [~TextSyStem](#)()

```
cli::TextSyStem::~~TextSyStem () [override], [default]
```

7.55.2.3 [TextSyStem](#)() [2/3]

```
cli::TextSyStem::TextSyStem (  
    const TextSyStem & ) [delete]
```

7.55.2.4 [TextSyStem](#)() [3/3]

```
cli::TextSyStem::TextSyStem (  
    TextSyStem && ) [delete]
```

7.55.3 Member Function Documentation

7.55.3.1 operator=() [1/2]

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

7.55.3.2 operator=() [2/2]

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

7.55.3.3 update()

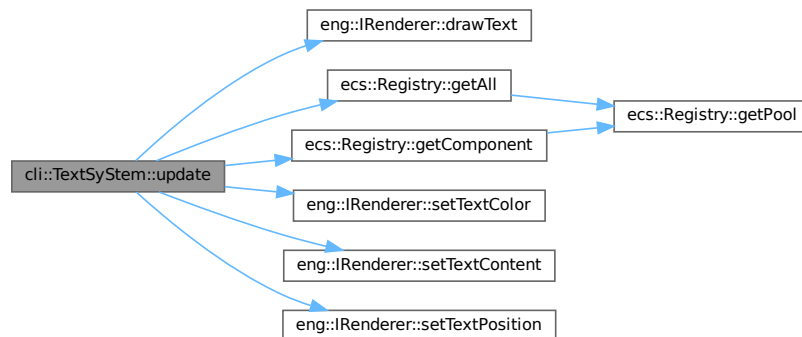
void cli::TextSyStem::update (
 [ecs::Registry](#) & registry,
 float dt) [inline], [override], [virtual]

Implements [eng::ISystem](#).

Definition at line 35 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:



7.55.4 Member Data Documentation

7.55.4.1 m_renderer

[eng::IRenderer](#)& cli::TextSyStem::m_renderer [private]

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

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

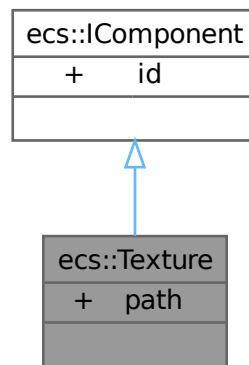
The documentation for this class was generated from the following file:

- `/home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Systems.hpp`

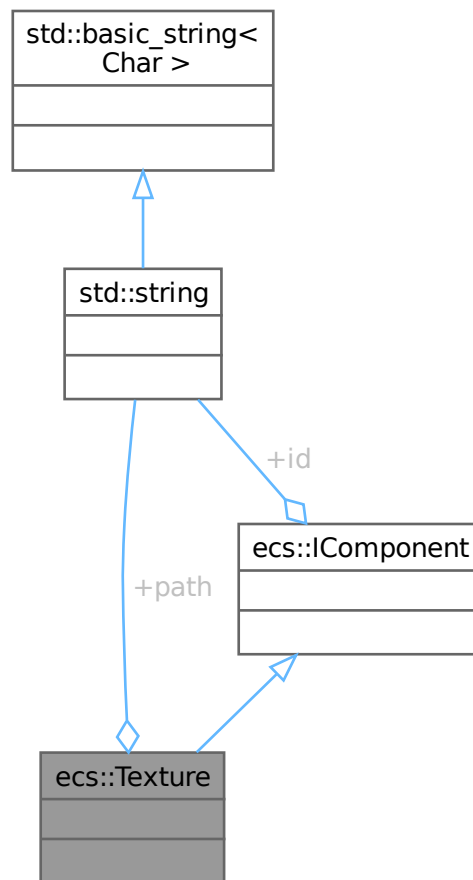
7.56 ecs::Texture Struct Reference

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

Inheritance diagram for ecs::Texture:



Collaboration diagram for `ecs::Texture`:



Public Attributes

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

Public Attributes inherited from [ecs::IComponent](#)

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

7.56.1 Detailed Description

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

7.56.2 Member Data Documentation

7.56.2.1 path

`std::string ecs::Texture::path`

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

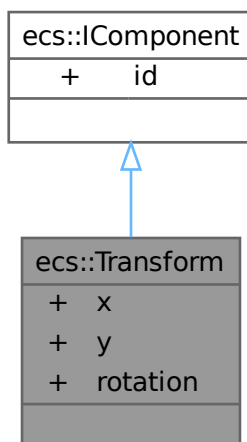
The documentation for this struct was generated from the following file:

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

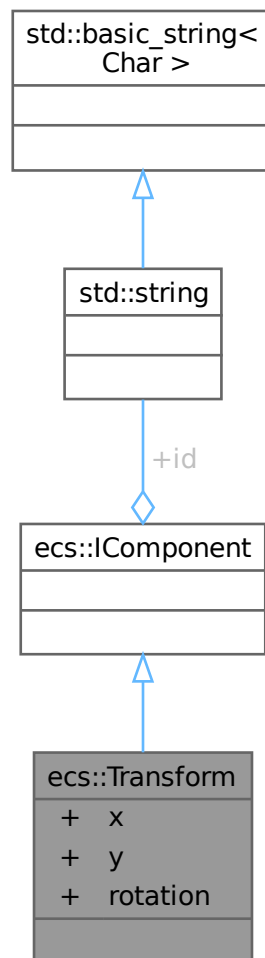
7.57 ecs::Transform Struct Reference

`#include <Component.hpp>`

Inheritance diagram for `ecs::Transform`:



Collaboration diagram for `ecs::Transform`:



Public Attributes

- float `x` {}
- float `y` {}
- float `rotation` {}

Public Attributes inherited from `ecs::IComponent`

- std::string `id`

7.57.1 Detailed Description

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

7.57.2 Member Data Documentation

7.57.2.1 rotation

```
float ecs::Transform::rotation {}
```

Definition at line 70 of file [Component.hpp](#).

7.57.2.2 x

```
float ecs::Transform::x {}
```

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

Referenced by [cli::Lobby::update\(\)](#).

7.57.2.3 y

```
float ecs::Transform::y {}
```

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

Referenced by [cli::Lobby::update\(\)](#).

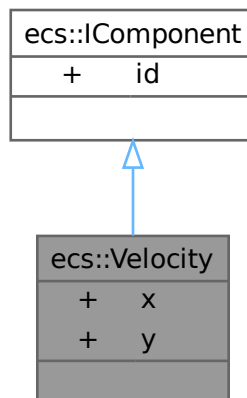
The documentation for this struct was generated from the following file:

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

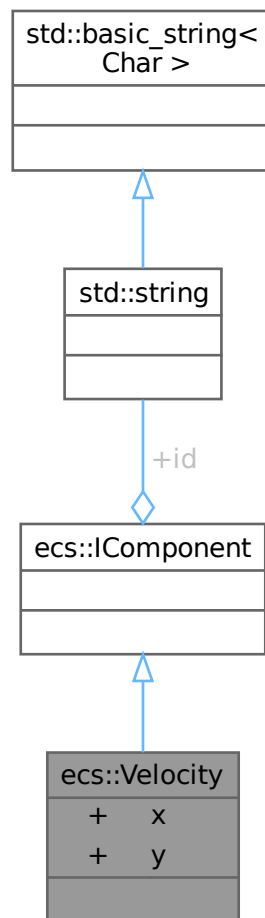
7.58 ecs::Velocity Struct Reference

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

Inheritance diagram for `ecs::Velocity`:



Collaboration diagram for ecs::Velocity:



Public Attributes

- float `x` {}
- float `y` {}

Public Attributes inherited from `ecs::IComponent`

- `std::string id`

7.58.1 Detailed Description

Definition at line 72 of file `Component.hpp`.

7.58.2 Member Data Documentation

7.58.2.1 x

```
float ecs::Velocity::x {}
```

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

Referenced by [cli::Lobby::update\(\)](#), and [cli::SpriteSystem::update\(\)](#).

7.58.2.2 y

```
float ecs::Velocity::y {}
```

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

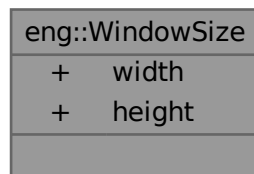
The documentation for this struct was generated from the following file:

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

7.59 eng::WindowSize Struct Reference

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

Collaboration diagram for eng::WindowSize:



Public Attributes

- unsigned int [width](#)
- unsigned int [height](#)

7.59.1 Detailed Description

Definition at line 91 of file [IRenderer.hpp](#).

7.59.2 Member Data Documentation

7.59.2.1 height

unsigned int eng::WindowSize::height

Definition at line 94 of file [IRenderer.hpp](#).

Referenced by [cli::Lobby::update\(\)](#).

7.59.2.2 width

unsigned int eng::WindowSize::width

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

Referenced by [eng::SFMLRenderer::getWindowSize\(\)](#), and [cli::Lobby::update\(\)](#).

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

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

Chapter 8

File Documentation

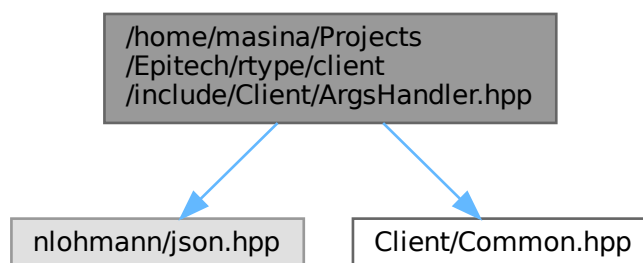
8.1 /home/masina/Projects/Epitech/rtype/client/include/Client/ArgsHandler.hpp File Reference

This file contains the ArgsHandler class declaration.

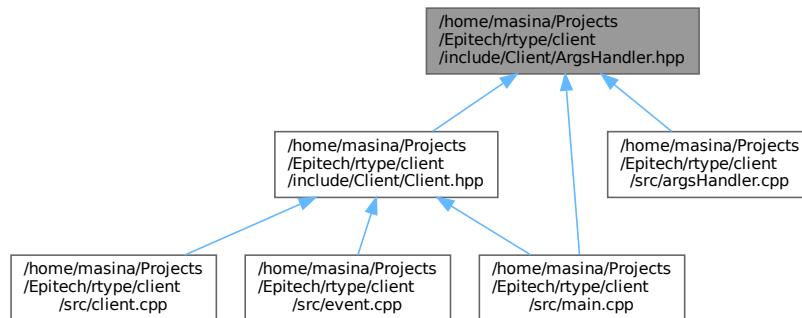
```
#include <nlohmann/json.hpp>
```

```
#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 /// @file ArgsHandler.hpp
00003 /// @brief This file contains the ArgsHandler class declaration
00004 /// @namespace cli
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     /// @class ArgsHandler
00033     /// @brief Class to handle command line arguments
00034     /// @namespace cli
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
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 cli
00054 }

```

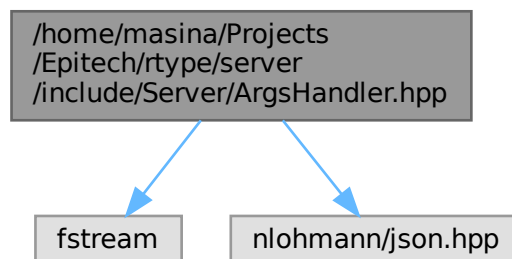
8.3 /home/masina/Projects/Epitech/rtype/server/include/Server/ArgsHandler.hpp File Reference ↵

This file contains the ArgsHandler class declaration.

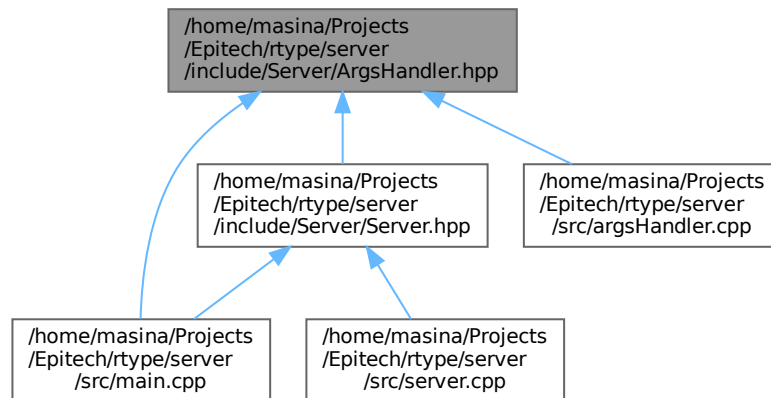
```
#include <fstream>
```

```
#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     public:  

00038         ArgsHandler() = default;  

00039         ~ArgsHandler() = default;  

00040         ///  

00041         ArgsHandler(const ArgsHandler &) = delete;  

00042         ArgsHandler &operator=(const ArgsHandler &) = delete;  

00043         ArgsHandler(ArgsHandler &&) = delete;  

00044         ArgsHandler &operator=(ArgsHandler &&) = delete;  

00045         ///  

00046         static ArgsConfig ParseArgs(int argc, const char *const argv[]);  

00047         static EnvConfig ParseEnv(const char *const env[]);  

00048         ///  

00049     private:  

00050     }; // class ArgsHandler  

00051 }  

00052 }  

00053 } // namespace srv

```

8.5 /home/masina/Projects/Epitech/rtype/client/include/Client/↵ Client.hpp File Reference

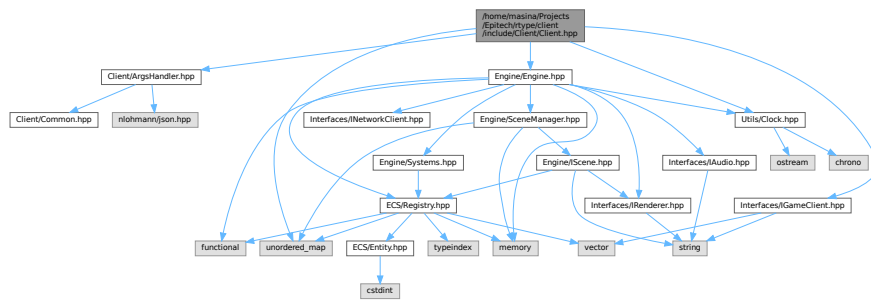
This file contains the Client class declaration.

```

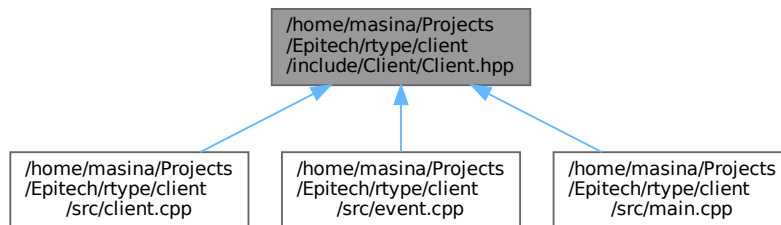
#include <unordered_map>
#include "Client/ArgsHandler.hpp"
#include "Engine/Engine.hpp"
#include "Interfaces/IGameClient.hpp"
#include "Utils/Clock.hpp"

```

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 <unordered_map>  

00010  

00011 #include "Client/ArgsHandler.hpp"  

00012 #include "Engine/Engine.hpp"  

00013 #include "Interfaces/IGameClient.hpp"  

00014 #include "Utils/Clock.hpp"  

00015  

00016 namespace cli  

00017 {  

00018  

00019     ///  

00020     ///  

00021     ///  

00022     ///  

00023     ///  

00024     class Client  

00025     {  

00026  

00027     public:  

00028         explicit Client(const ArgsConfig &cfg);  

00029         ~Client() = default;  

00030  

00031         Client(const Client &) = delete;  

00032         Client &operator=(const Client &) = delete;  

00033         Client(Client &&) = delete;  

00034         Client &operator=(Client &&) = delete;  

00035  

00036         void run();  

00037  

00038     private:  

00039         void handleEvents(eng::Event &event);  

00040         eng::IScene &lobbyScene();  

00041  

00042         std::unique_ptr<gme::IGameClient> m_game;  

00043         std::unique_ptr<eng::Engine> m_engine;  

00044         std::unordered_map<eng::Key, bool> m_keysPressed;  

00045     }; // class Client  

00046  

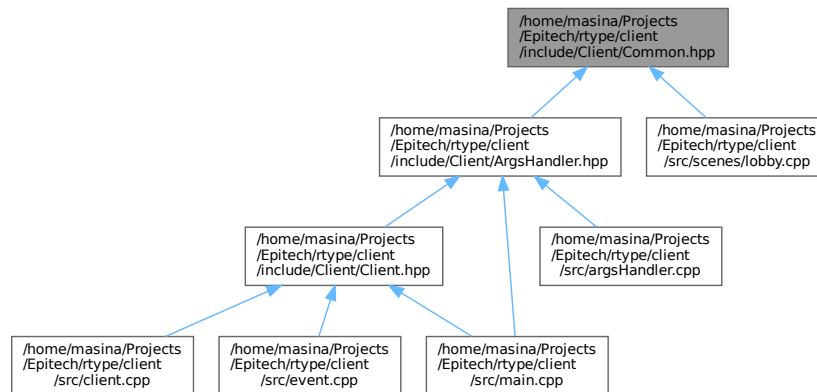
00047 } // namespace cli

```

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::Config`
- namespace `cli::Config::Window`
- namespace `cli::Config::Audio`
- namespace `cli::Path`
- namespace `cli::Path::Audio`
- namespace `cli::Path::Font`
- namespace `cli::Path::Texture`

Variables

- 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
- constexpr auto `cli::Path::Audio::AUDIO_TITLE` = "assets/audio/title.mp3"
- constexpr auto `cli::Path::Audio::AUDIO_COIN` = "assets/audio/coin.mp3"
- constexpr auto `cli::Path::Audio::AUDIO_BATTLE_THEME` = "assets/audio/battle_theme.mp3"
- constexpr auto `cli::Path::Font::FONTS_RTYPE` = "assets/fonts/r-type.otf"
- constexpr auto `cli::Path::Texture::TEXTURE_PLAYER` = "assets/sprites/r-typesheet42.gif"

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.](#)

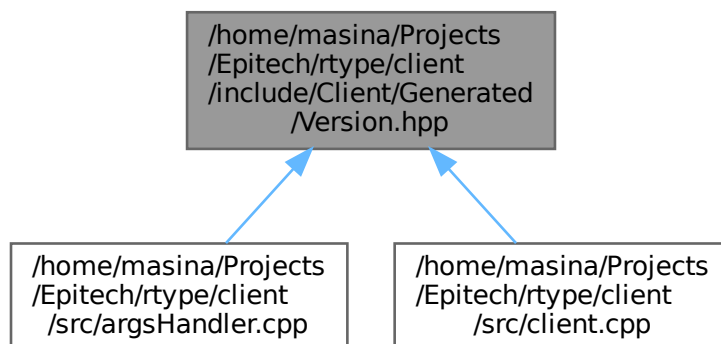
```

00001 ///
00002 /// @file Common.hpp
00003 /// @brief This file contains common definitions and constants
00004 /// @namespace cli
00005 ///
00006
00007 #pragma once
00008
00009 namespace cli
00010 {
00011     namespace Config
00012     {
00013         namespace Window
00014         {
00015             inline constexpr auto DEFAULT_WINDOW_WIDTH = 960;
00016             inline constexpr auto DEFAULT_WINDOW_HEIGHT = 540;
00017             inline constexpr auto DEFAULT_WINDOW_FRAME_LIMIT = 240;
00018             inline constexpr auto DEFAULT_WINDOW_FULLSCREEN = false;
00019         } // namespace Window
00020         namespace Audio
00021         {
00022             inline constexpr auto DEFAULT_AUDIO_VOLUME = 50; // unused
00023             inline constexpr auto DEFAULT_AUDIO_MUTED = false; // unused
00024         } // namespace Audio
00025     } // namespace Config
00026     namespace Path
00027     {
00028         namespace Audio
00029         {
00030             inline constexpr auto AUDIO_TITLE = "assets/audio/title.mp3";
00031             inline constexpr auto AUDIO_COIN = "assets/audio/coin.mp3";
00032             inline constexpr auto AUDIO_BATTLE_THEME = "assets/audio/battle_theme.mp3";
00033         } // namespace Audio
00034         namespace Font
00035         {
00036             inline constexpr auto FONTS_RTYPE = "assets/fonts/r-type.otf";
00037         } // namespace Font
00038         namespace Texture
00039         {
00040             inline constexpr auto TEXTURE_PLAYER = "assets/sprites/r-typesheet42.gif";
00041         } // namespace Texture
00042     } // namespace Path
00043 } // 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 "063d861"`
- `#define GIT_TAG "063d861"`
- `#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 "063d861"
```

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 "063d861"
```

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](#).

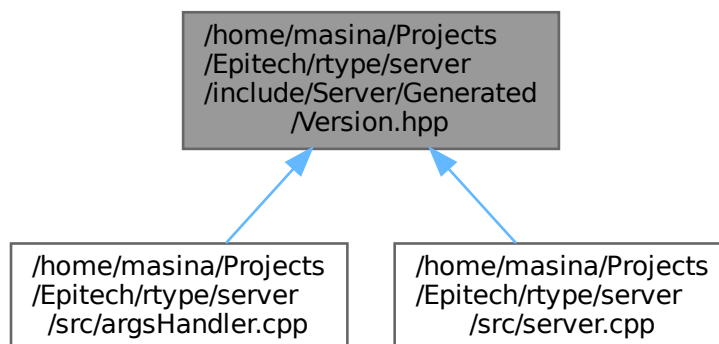
8.10 Version.hpp

[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 "063d861"
00014 #define GIT_TAG "063d861"
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 "063d861"`
- `#define GIT_TAG "063d861"`
- `#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 "063d861"
```

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

8.11.1.3 GIT_TAG

```
#define GIT_TAG "063d861"
```

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](#).

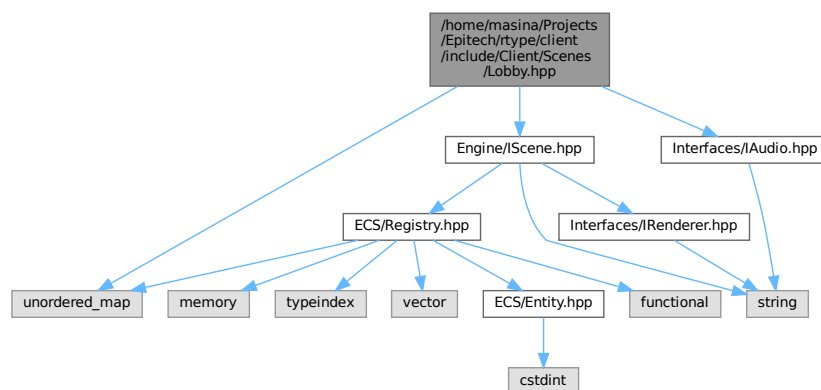
8.12 Version.hpp

[Go to the documentation of this file.](#)

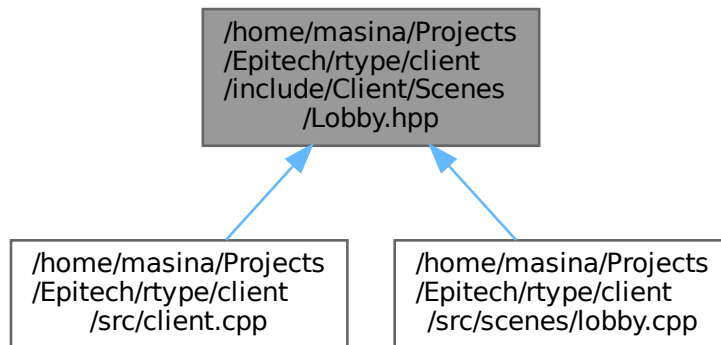
```
00001 #pragma once
00002
00003 //
00004 // =====
00005 // DO NOT EDIT THIS FILE MANUALLY. IT IS GENERATED BY CMAKE DURING THE BUILD PROCESS.
00006 // =====
00007 #define PROJECT_NAME "r-type_server"
00008 #define PROJECT_VERSION "0.0.0"
00009 #define PROJECT_VERSION_MAJOR "0"
00010 #define PROJECT_VERSION_MINOR "0"
00011 #define PROJECT_VERSION_PATCH "0"
00012
00013 #define GIT_COMMIT_HASH "063d861"
00014 #define GIT_TAG "063d861"
00015 #define BUILD_TYPE "Release"
```

8.13 /home/masina/Projects/Epitech/rtype/client/include/Client/Scenes/Lobby.hpp File Reference

```
#include <unordered_map>
#include "Engine/IScene.hpp"
#include "Interfaces/IAudio.hpp"
Include dependency graph for Lobby.hpp:
```



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



Classes

- class `cli::Lobby`
 Lobby scene.

Namespaces

- namespace `eng`
- namespace `cli`

8.14 Lobby.hpp

[Go to the documentation of this file.](#)

```

00001 ///  

00002 ///  

00003 ///  

00004 ///  

00005 ///  

00006 ///  

00007 #pragma once  

00008  

00009 #include <unordered_map>  

00010  

00011 #include "Engine/IScene.hpp"  

00012 #include "Interfaces/IAudio.hpp"  

00013  

00014 namespace cli  

00015 {  

00016     ///  

00017     ///  

00018     ///  

00019     ///  

00020     ///  

00021     class Lobby final : public eng::AScene  

00022     {  

00023     public:  

00024         Lobby(const std::unique_ptr<eng::IRenderer> &renderer, const std::unique_ptr<eng::IAudio> &audio);  

00025         ~Lobby() override = default;  

00026  

00027         Lobby(const Lobby &other) = delete;  

00028         Lobby &operator=(const Lobby &other) = delete;  

00029         Lobby(Lobby &&other) = delete;  


```

```

00030         Lobby &operator=(Lobby &&other) = delete;
00031
00032         void update(float dt, const eng::WindowSize &size) override;
00033         void event(const eng::Event &event) override;
00034
00035     private:
00036         std::unordered_map<eng::Key, bool> m_keysPressed;
00037
00038         ecs::Entity m_playerEntity;
00039         ecs::Entity m_fpsEntity;
00040     }; // class Lobby
00041 } // namespace cli

```

8.15 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Systems.hpp File Reference

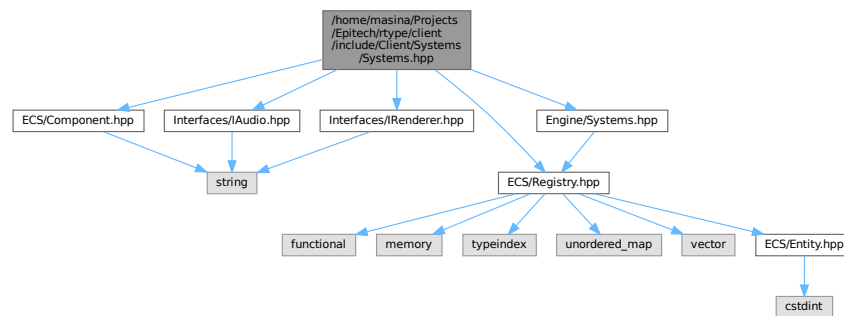
This file contains the system definitions.

```

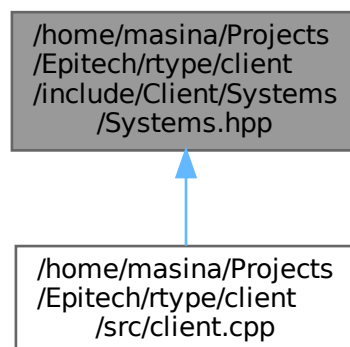
#include "ECS/Component.hpp"
#include "ECS/Registry.hpp"
#include "Interfaces/IAudio.hpp"
#include "Interfaces/IRenderer.hpp"
#include "Engine/Systems.hpp"

```

Include dependency graph for Systems.hpp:



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



Classes

- class [cli::TextSyStem](#)
Class for managing entities and their components.
- class [cli::AudioSystem](#)
Class for managing entities and their components.
- class [cli::SpriteSystem](#)
- class [cli::PixelSystem](#)

Namespaces

- namespace [cli](#)
- namespace [ecs](#)

8.15.1 Detailed Description

This file contains the system definitions.

Definition in file [Systems.hpp](#).

8.16 Systems.hpp

[Go to the documentation of this file.](#)

```

00001 ///
00002 /// @file Systems.hpp
00003 /// @brief This file contains the system definitions
00004 /// @namespace cli
00005 ///
00006
00007 #pragma once
00008
00009 #include "ECS/Component.hpp"
00010 #include "ECS/Registry.hpp"
00011 #include "Interfaces/IAudio.hpp"
00012 #include "Interfaces/IRenderer.hpp"
00013
00014 #include "Engine/Systems.hpp"
00015
00016 namespace cli
00017 {
00018
00019     ///
00020     /// @class TextSyStem
00021     /// @brief Class for managing entities and their components
00022     /// @namespace ecs
00023     ///
00024     class TextSyStem final : public eng::ASystem
00025     {
00026     public:
00027         explicit TextSyStem(eng::IRenderer &renderer) : m_renderer(renderer) {}
00028         ~TextSyStem() override = default;
00029
00030         TextSyStem(const TextSyStem &) = delete;
00031         TextSyStem &operator=(const TextSyStem &) = delete;
00032         TextSyStem(TextSyStem &&) = delete;
00033         TextSyStem &operator=(TextSyStem &&) = delete;
00034
00035         void update(ecs::Registry &registry, float dt) override
00036         {
00037             for (auto &[entity, text] : registry.getAll<ecs::Text>())
00038             {
00039                 const auto *transform = registry.getComponent<ecs::Transform>(entity);
00040                 const auto *color = registry.getComponent<ecs::Color>(entity);
00041
00042                 const float x = (transform != nullptr) ? transform->x : 0.F;
00043                 const float y = (transform != nullptr) ? transform->y : 0.F;
00044             }
00045         }
00046     };
00047 }
```

```

00045
00046         const std::uint8_t r = color ? color->r : 255u;
00047         const std::uint8_t g = color ? color->g : 255u;
00048         const std::uint8_t b = color ? color->b : 255u;
00049         const std::uint8_t a = color ? color->a : 255u;
00050
00051         m_renderer.setTextContent(text.id, text.content);
00052         m_renderer.setTextPosition(text.id, x, y);
00053         m_renderer.setTextColor(text.id, {.r = r, .g = g, .b = b, .a = a});
00054         m_renderer.drawText(text.id);
00055     }
00056 }
00057
00058 private:
00059     eng::IRenderer &m_renderer;
00060 }; // class TextRenderSystem
00061
00062 ///
00063 /// @class AudioSystem
00064 /// @brief Class for managing entities and their components
00065 /// @namespace ecs
00066 ///
00067 class AudioSystem final : public eng::ASystem
00068 {
00069 public:
00070     explicit AudioSystem(eng::IAudio &audio) : m_audio(audio) {}
00071     ~AudioSystem() override = default;
00072
00073     AudioSystem(const AudioSystem &) = delete;
00074     AudioSystem &operator=(const AudioSystem &) = delete;
00075     AudioSystem(AudioSystem &&) = delete;
00076     AudioSystem &operator=(AudioSystem &&) = delete;
00077
00078     void update(ecs::Registry &registry, float dt) override
00079     {
00080         for (auto &[entity, audio] : registry.getAll<ecs::Audio>())
00081         {
00082             m_audio.setVolume(audio.id + std::to_string(entity), audio.volume);
00083             m_audio.setLoop(audio.id + std::to_string(entity), audio.loop);
00084             if (audio.play && m_audio.isPlaying(audio.id + std::to_string(entity)) != eng::Status::Playing)
00085             {
00086                 m_audio.playAudio(audio.id + std::to_string(entity));
00087             }
00088             else if (!audio.play &&
00089                     m_audio.isPlaying(audio.id + std::to_string(entity)) != eng::Status::Stopped)
00090             {
00091                 m_audio.stopAudio(audio.id + std::to_string(entity));
00092             }
00093         }
00094     }
00095
00096 private:
00097     eng::IAudio &m_audio;
00098 }; // class AudioSystem
00099
00100 class SpriteSystem final : public eng::ASystem
00101 {
00102 public:
00103     explicit SpriteSystem(eng::IRenderer &renderer) : m_renderer(renderer) {}
00104     ~SpriteSystem() override = default;
00105
00106     SpriteSystem(const SpriteSystem &) = delete;
00107     SpriteSystem &operator=(const SpriteSystem &) = delete;
00108     SpriteSystem(SpriteSystem &&) = delete;
00109     SpriteSystem &operator=(SpriteSystem &&) = delete;
00110
00111     void update(ecs::Registry &registry, float dt) override
00112     {
00113         for (auto &[entity, sprite] : registry.getAll<ecs::Texture>())
00114         {
00115             const auto *transform = registry.getComponent<ecs::Transform>(entity);
00116             const auto *color = registry.getComponent<ecs::Color>(entity);
00117             const auto *velocity = registry.getComponent<ecs::Velocity>(entity);
00118
00119             const float x = (transform != nullptr) ? transform->x : 0.F;
00120             const float y = (transform != nullptr) ? transform->y : 0.F;
00121
00122             // std::uint8_t r = color ? static_cast<std::uint8_t>(color->r) : 255;
00123             // std::uint8_t g = color ? static_cast<std::uint8_t>(color->g) : 255;
00124             // std::uint8_t b = color ? static_cast<std::uint8_t>(color->b) : 255;
00125             // std::uint8_t a = color ? static_cast<std::uint8_t>(color->a) : 255;
00126
00127             // int xv = velocity ? static_cast<int>(velocity->x) : 0;
00128             // int yv = velocity ? static_cast<int>(velocity->y) : 0;
00129             // x *= xv;
00130             // y *= yv;
00131             m_renderer.setSpriteTexture(sprite.id + std::to_string(entity), sprite.path);

```

```

00132         m_renderer.setSpritePosition(sprite.id + std::to_string(entity), x, y);
00133         // m_renderer.setSpriteColor(sprite.id, {r, g, b, a});
00134
00135         m_renderer.drawSprite(sprite.id + std::to_string(entity));
00136     }
00137 }
00138
00139 private:
00140     eng::IRenderer &m_renderer;
00141 }; // class SpriteSystem
00142
00143 class PixelSystem final : public eng::ASystem
00144 {
00145 public:
00146     explicit PixelSystem(eng::IRenderer &renderer) : m_renderer(renderer) {}
00147     ~PixelSystem() override = default;
00148
00149     explicit PixelSystem(const SpriteSystem &) = delete;
00150     PixelSystem &operator=(const SpriteSystem &) = delete;
00151     explicit PixelSystem(SpriteSystem &&) = delete;
00152     PixelSystem &operator=(SpriteSystem &&) = delete;
00153
00154     void update(ecs::Registry &registry, float dt) override
00155     {
00156         for (const auto &entity : registry.getAll<ecs::Pixel>() | std::views::keys)
00157         {
00158             const auto *color = registry.getComponent<ecs::Color>(entity);
00159             const auto *transform = registry.getComponent<ecs::Transform>(entity);
00160             m_renderer.drawPoint(transform->x, transform->y,
00161                                 {r = color->r, .g = color->g, .b = color->b, .a = color->a});
00162         }
00163     }
00164
00165 private:
00166     eng::IRenderer &m_renderer;
00167 }; // class PixelSystem
00168
00169 } // namespace cli

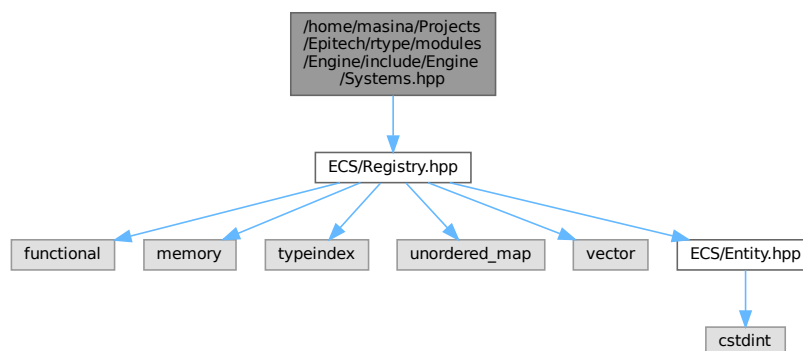
```

8.17 /home/masina/Projects/Epitech/rtype/modules/Engine/include/↵ Engine/Systems.hpp File Reference

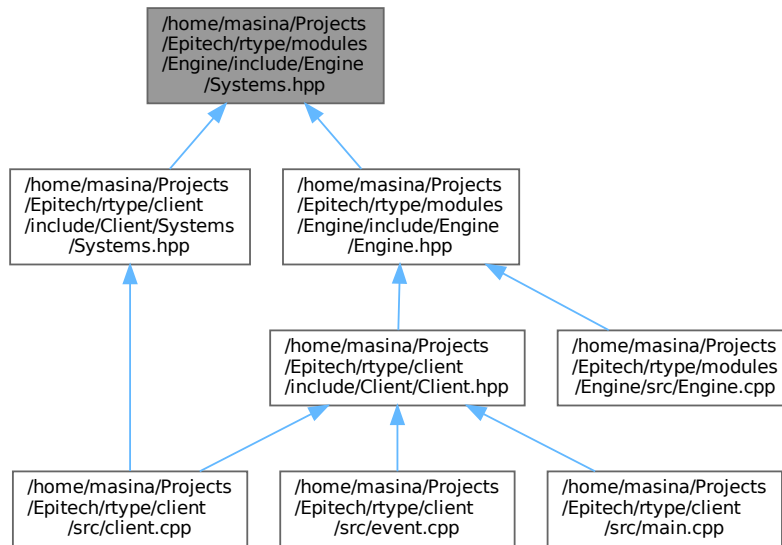
This file contains the system definitions.

#include "ECS/Registry.hpp"

Include dependency graph for Systems.hpp:



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



Classes

- class [eng::ISystem](#)
- class [eng::ASystem](#)

Namespaces

- namespace [eng](#)

8.17.1 Detailed Description

This file contains the system definitions.

Definition in file [Systems.hpp](#).

8.18 Systems.hpp

[Go to the documentation of this file.](#)

```

00001 ///
00002 /// @file Systems.hpp
00003 /// @brief This file contains the system definitions
00004 /// @namespace eng
00005 ///
00006
00007 #pragma once
00008
00009 #include "ECS/Registry.hpp"
00010
00011 namespace eng
00012 {
00013
```

```

00014 class ISystem
00015 {
00016     public:
00017         virtual ~ISystem() = default;
00018         virtual void update(ecs::Registry &registry, float dt) = 0;
00019         virtual bool isEnabled() = 0;
00020         virtual void setEnable(bool enable) = 0;
00021 };
00022
00023 class ASystem : public ISystem
00024 {
00025     public:
00026         bool isEnabled() override { return m_isEnable; }
00027         void setEnable(const bool enable) override { m_isEnable = enable; }
00028
00029     private:
00030         bool m_isEnable = true;
00031 };
00032
00033 } // namespace eng

```

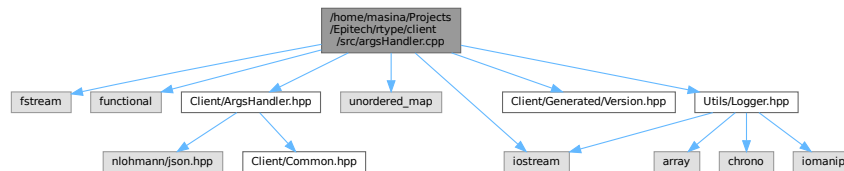
8.19 /home/masina/Projects/Epitech/rtype/client/src/argsHandler.cpp File Reference

```

#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`

8.19.1 Macro Definition Documentation

8.19.1.1 APP_EXTENSION

```
#define APP_EXTENSION ""
```

Definition at line 9 of file `argsHandler.cpp`.

8.19.2 Variable Documentation

8.19.2.1 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().

8.19.2.2 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().

8.20 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 }
```

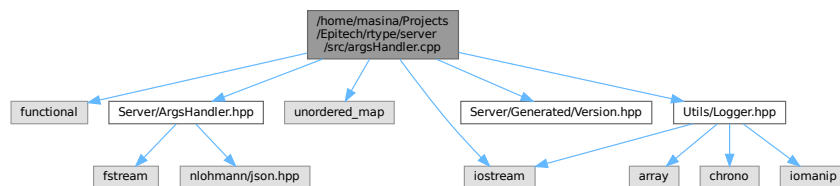
```

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
00086     for (const auto *const opt : {"-c", "--config"})
00087     {
00088         handlers[opt] = [&config](const char *arg)
00089         {
00090             if (!arg)
00091             {
00092                 throw std::runtime_error("Missing config file argument");
00093             }
00094             config = ArgsConfig::fromFile(arg);
00095             utl::Logger::log("Loaded config from file: " + std::string(arg), utl::LogLevel::INFO);
00096             std::cout « "\tWidth: " « config.width « '\n'
00097                     « "\tHeight: " « config.height « '\n'
00098                     « "\tFrameLimit: " « config.frameLimit « "\n"
00099                     « "\tFullscreen: " « (config.fullscreen ? "true" : "false") « '\n';
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 }

```

8.21 /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`

8.21.1 Macro Definition Documentation

8.21.1.1 APP_EXTENSION

```
#define APP_EXTENSION ""
```

Definition at line 9 of file `argsHandler.cpp`.

8.21.2 Variable Documentation

8.21.2.1 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`.

8.21.2.2 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](#).

8.22 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-t--help, -h      Show this help message\n"
00019     "\t-t--version, -v    Show version information\n"
00020     "\t-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
00076 for (const auto *const opt : {"-c", "--config"})
00077 {
00078     handlers[opt] = [&config](const char *arg)
00079     {
00080         if (!arg)
00081         {
00082             throw std::runtime_error("Missing config file argument");
00083         }
00084         config = ArgsConfig::fromFile(arg);
00085         utl::Logger::log("Loaded config from file: " + std::string(arg), utl::LogLevel::INFO);
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 {}; }

```

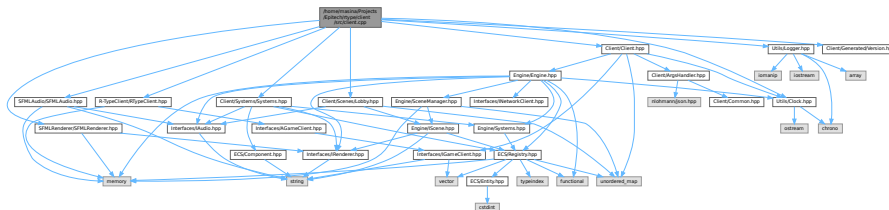
8.23 /home/masina/Projects/Epitech/rtype/client/src/client.cpp File Reference

```

#include "Client/Client.hpp"
#include "Client/Generated/Version.hpp"
#include "Client/Scenes/Lobby.hpp"
#include "Client/Systems/Systems.hpp"
#include "R-TypeClient/RTypeClient.hpp"
#include "SFMLAudio/SFMLAudio.hpp"
#include "SFMLRenderer/SFMLRenderer.hpp"
#include "Utils/Clock.hpp"
#include "Utils/Logger.hpp"

```

Include dependency graph for client.cpp:



Variables

- static constexpr eng::Color DARK = {.r = 0U, .g = 0U, .b = 0U, .a = 255U}

8.23.1 Variable Documentation

8.23.1.1 DARK

`eng::Color` DARK = {`r` = 0U, `.g` = 0U, `.b` = 0U, `.a` = 255U} [static], [constexpr]

Definition at line 11 of file `client.cpp`.

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

8.24 client.cpp

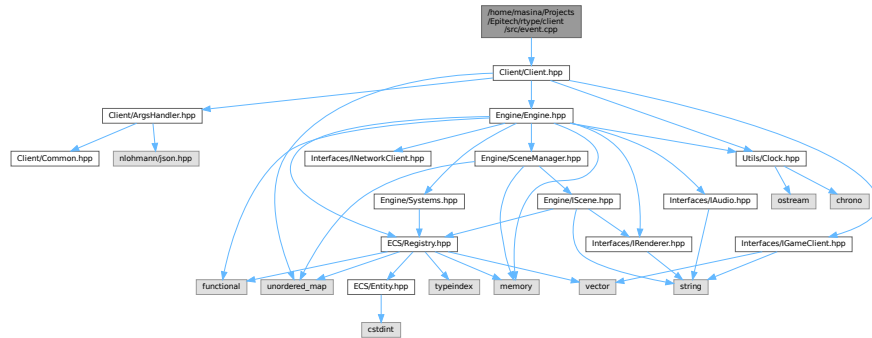
Go to the documentation of this file.

```
00001 #include "Client/Client.hpp"
00002 #include "Client/Generated/Version.hpp"
00003 #include "Client/Scenes/Lobby.hpp"
00004 #include "Client/Systems/Systems.hpp"
00005 #include "R-TypeClient/RTypeClient.hpp"
00006 #include "SFMLAudio/SFMLAudio.hpp"
00007 #include "SFMLRenderer/SFMLRenderer.hpp"
00008 #include "Utils/Clock.hpp"
00009 #include "Utils/Logger.hpp"
00010
00011 static constexpr eng::Color DARK = {r = 0U, .g = 0U, .b = 0U, .a = 255U};
00012
00013 cli::Client::Client(const ArgsConfig &cfg)
00014 {
00015     utl::Logger::log("PROJECT INFO:", utl::LogLevel::INFO);
00016     std::cout << "\tName: " PROJECT_NAME "\n"
00017               << "\tVersion: " PROJECT_VERSION "\n"
00018               << "\tBuild type: " BUILD_TYPE "\n"
00019               << "\tGit tag: " GIT_TAG "\n"
00020               << "\tGit commit hash: " GIT_COMMIT_HASH "\n";
00021
00022     m_engine =
00023         std::make_unique<eng::Engine>([]() { return std::make_unique<eng::SFMLAudio>(); }, []() { return nullptr; },
00024         []() { return std::make_unique<eng::SFMLRenderer>(); });
00025     m_game = std::make_unique<gme::RTypeClient>();
00026
00027     m_engine->getRenderer()->createWindow("R-Type Client", cfg.height, cfg.width, cfg.frameLimit, cfg.fullscreen);
00028
00029     m_engine->addSystem(std::make_unique<AudioSystem>(*m_engine->getAudio()));
00030     m_engine->addSystem(std::make_unique<PixelSystem>(*m_engine->getRenderer()));
00031     m_engine->addSystem(std::make_unique<SpriteSystem>(*m_engine->getRenderer()));
00032     m_engine->addSystem(std::make_unique<TextSystem>(*m_engine->getRenderer()));
00033
00034     auto lobby = std::make_unique<Lobby>(m_engine->getRenderer(), m_engine->getAudio());
00035     const auto lobbyId = lobby->getId();
00036     m_engine->getSceneManager()->addScene(std::move(lobby));
00037     m_engine->getSceneManager()->switchToScene(lobbyId);
00038 }
00039
00040 void cli::Client::run()
00041 {
00042     eng::Event event;
00043
00044     while (m_engine->getState() == eng::State::RUN && m_engine->getRenderer()->windowIsOpen())
00045     {
00046         const float delta = m_engine->getClock()->getDeltaSeconds();
00047
00048         m_engine->getClock()->restart();
00049         m_engine->getSceneManager()->getCurrentScene()->update(delta, m_engine->getRenderer()->getWindowSize());
00050         handleEvents(event);
00051         m_engine->render(m_engine->getSceneManager()->getCurrentScene()->getRegistry(), DARK, delta);
00052     }
00053     m_engine->stop();
00054 }
```

8.25 /home/masina/Projects/Epitech/rtype/client/src/event.cpp File Reference

#include "Client/Client.hpp"

Include dependency graph for event.cpp:



8.26 event.cpp

[Go to the documentation of this file.](#)

```
00001 #include "Client/Client.hpp"
00002
00003 void cli::Client::handleEvents(eng::Event &event)
00004 {
00005     const auto &scene = m_engine->getSceneManager()->getCurrentScene();
00006
00007     while (m_engine->getRenderer()->pollEvent(event))
00008     {
00009         scene->event(event);
00010
00011         switch (event.type)
00012         {
00013             case eng::EventType::Closed:
00014                 m_engine->setState(eng::State::STOP);
00015                 break;
00016
00017             case eng::EventType::KeyPressed:
00018                 if (event.key == eng::Key::Escape)
00019                 {
00020                     m_engine->setState(eng::State::STOP);
00021                 }
00022                 else
00023                 {
00024                     m_keysPressed[event.key] = true;
00025                 }
00026                 break;
00027
00028             case eng::EventType::KeyReleased:
00029                 m_keysPressed[event.key] = false;
00030                 break;
00031
00032             default:
00033                 break;
00034         }
00035     }
00036 }
```

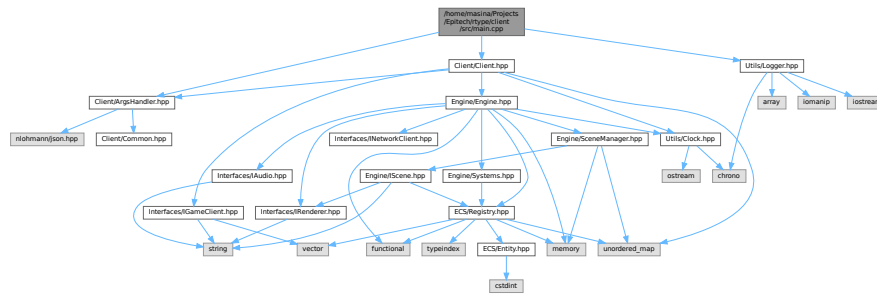
8.27 /home/masina/Projects/Epitech/rtype/client/src/main.cpp File Reference

#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[])

8.27.1 Function Documentation

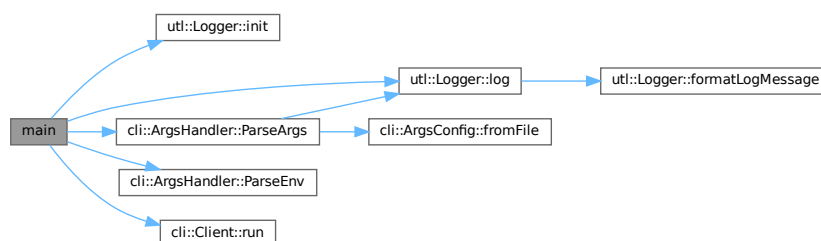
8.27.1.1 main()

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

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

References [cli::ArgsConfig::exit](#), [utl::Logger::init\(\)](#), [utl::Logger::log\(\)](#), [cli::ArgsHandler::ParseArgs\(\)](#), [cli::ArgsHandler::ParseEnv\(\)](#), [cli::Client::run\(\)](#), and [utl::WARNING](#).

Here is the call graph for this function:



8.28 main.cpp

[Go to the documentation of this file.](#)

```

00001 #include "Client/ArgsHandler.hpp"
00002 #include "Client/Client.hpp"
00003 #include "Utils/Logger.hpp"
00004
00005 int main(const int argc, const char *const argv[], const char *const env[])
00006 {
00007     utl::Logger::init();
00008
00009     try
00010     {
00011         const cli::ArgsConfig argsConf = cli::ArgsHandler::ParseArgs(argc, argv);
00012         const cli::EnvConfig envConf = cli::ArgsHandler::ParseEnv(env);
00013         if (argsConf.exit)
00014         {
00015             return EXIT_SUCCESS;
00016         }
00017         cli::Client client(argsConf);
00018         client.run();
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 }

```

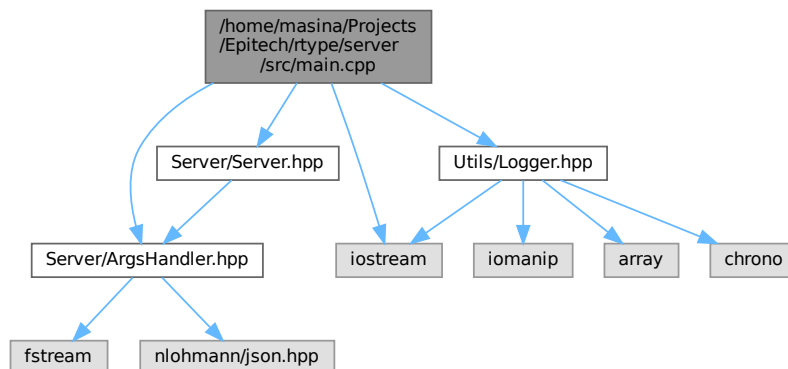
8.29 /home/masina/Projects/Epitech/rtype/server/src/main.cpp File Reference

```

#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[])

8.29.1 Function Documentation

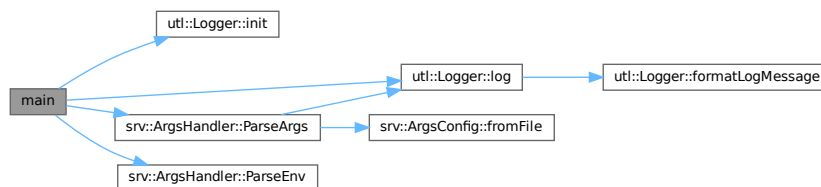
8.29.1.1 main()

```
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:



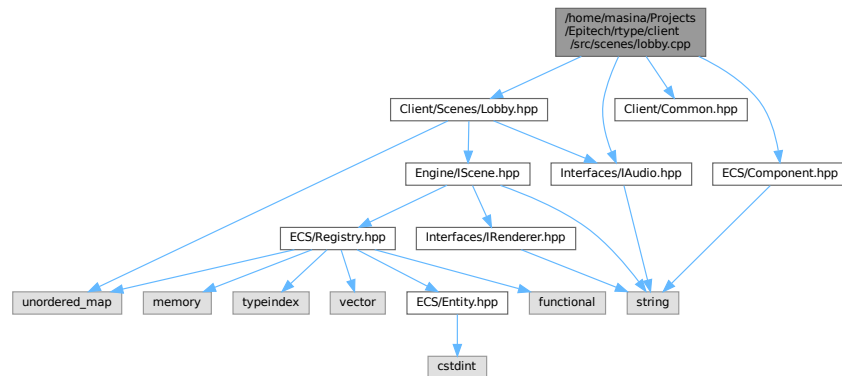
8.30 main.cpp

[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 }
```

8.31 /home/masina/Projects/Epitech/rtype/client/src/scenes/lobby.cpp File Reference

```
#include "Client/Scenes/Lobby.hpp"
#include "Client/Common.hpp"
#include "ECS/Component.hpp"
#include "Interfaces/IAudio.hpp"
Include dependency graph for lobby.cpp:
```



Variables

- static constexpr [eng::Color](#) [WHITE](#) = { .r = 255U, .g = 255U, .b = 255U, .a = 255U }

8.31.1 Variable Documentation

8.31.1.1 WHITE

[eng::Color](#) [WHITE](#) = { .r = 255U, .g = 255U, .b = 255U, .a = 255U } [static], [constexpr]

Definition at line 6 of file [lobby.cpp](#).

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

8.32 lobby.cpp

[Go to the documentation of this file.](#)

```
00001 #include "Client/Scenes/Lobby.hpp"
00002 #include "Client/Common.hpp"
00003 #include "ECS/Component.hpp"
00004 #include "Interfaces/IAudio.hpp"
00005
00006 static constexpr eng::Color WHITE = { .r = 255U, .g = 255U, .b = 255U, .a = 255U };
00007
00008 cli::Lobby::Lobby(const std::unique_ptr<eng::IRenderer> &renderer, const std::unique_ptr<eng::IAudio> &audio)
00009 {
00010     auto &registry = AScene::getRegistry();
00011
00012     registry.onComponentAdded(
00013         [&renderer, &audio, &registry](const ecs::Entity e, const std::type_info &type)
```

```

00014     {
00015         const auto *audioComp = registry.getComponent<ecs::Audio>(e);
00016         const auto *colorComp = registry.getComponent<ecs::Color>(e);
00017         const auto *fontComp = registry.getComponent<ecs::Font>(e);
00018         const auto *rectComp = registry.getComponent<ecs::Rect>(e);
00019         const auto *scaleComp = registry.getComponent<ecs::Scale>(e);
00020         const auto *textComp = registry.getComponent<ecs::Text>(e);
00021         const auto *textureComp = registry.getComponent<ecs::Texture>(e);
00022         const auto *transform = registry.getComponent<ecs::Transform>(e);
00023
00024         if (type == typeid(ecs::Text))
00025         {
00026             if (textComp && transform && fontComp)
00027             {
00028                 renderer->createFont(fontComp->id, fontComp->path);
00029                 renderer->createText(
00030                     {font_name = fontComp->id,
00031                      .color = {.r = colorComp->r, .g = colorComp->g, .b = colorComp->b, .a = colorComp->a},
00032                      .content = textComp->content,
00033                      .size = textComp->font_size,
00034                      .x = transform->x,
00035                      .y = transform->y,
00036                      .name = textComp->id});
00037             }
00038         }
00039         else if (type == typeid(ecs::Texture))
00040         {
00041             const float scale_x = scaleComp ? scaleComp->x : 1.F;
00042             const float scale_y = scaleComp ? scaleComp->y : 1.F;
00043
00044             renderer->createTexture(textureComp->id, textureComp->path);
00045
00046             if (transform && textureComp)
00047             {
00048                 if (rectComp)
00049                 {
00050                     renderer->createSprite(textureComp->id + std::to_string(e), textureComp->id, transform->x,
00051                                             transform->y, scale_x, scale_y, static_cast<int>(rectComp->pos_x),
00052                                             static_cast<int>(rectComp->pos_y), rectComp->size_x, rectComp->size_y);
00053                 }
00054                 else
00055                 {
00056                     renderer->createSprite(textureComp->id + std::to_string(e), textureComp->id, transform->x,
00057                                             transform->y);
00058                 }
00059             }
00060         }
00061         else if (type == typeid(ecs::Audio))
00062         {
00063             if (audioComp)
00064             {
00065                 audio->createAudio(audioComp->path, audioComp->volume, audioComp->loop,
00066                                     audioComp->id + std::to_string(e));
00067             }
00068         }
00069     });
00070
00071     registry.createEntity()
00072         .with<ecs::Audio>("id_audio", Path::Audio::AUDIO_TITLE, 5.F, true, true)
00073         .build();
00074     registry.createEntity()
00075         .with<ecs::Font>("main_font", Path::Font::FONTS_RTYPE)
00076         .with<ecs::Transform>("transform_title", 10.F, 10.F, 0.F)
00077         .with<ecs::Color>("color_title", WHITE.r, WHITE.g, WHITE.b, WHITE.a)
00078         .with<ecs::Text>("id", std::string("RType Client"), 50U)
00079         .build();
00080     m_fpsEntity = registry.createEntity()
00081         .with<ecs::Font>("main_font", Path::Font::FONTS_RTYPE)
00082         .with<ecs::Transform>("transform_fps", 10.F, 70.F, 0.F)
00083         .with<ecs::Color>("color_fps", WHITE.r, WHITE.g, WHITE.b, WHITE.a)
00084         .with<ecs::Text>("id_text", std::string("RType Client"), 20U)
00085         .build();
00086     m_playerEntity = registry.createEntity()
00087         .with<ecs::Transform>("player_transform", 200.F, 100.F, 0.F)
00088         .with<ecs::Velocity>("player_velocity", 500.F, 500.F)
00089         .with<ecs::Rect>("player_rect", 0.F, 0.F, 33, 20)
00090         .with<ecs::Scale>("player_scale", 2.F, 2.F)
00091         .with<ecs::Texture>("player_texture", Path::Texture::TEXTURE_PLAYER)
00092         .with
00093         .build();
00094     for (int i = 0; i < 100; i++)
00095     {
00096         registry.createEntity()
00097             .with<ecs::Pixel>("star_point_" + std::to_string(i))
00098             .with<ecs::Transform>("star_point_transform", 0.F, 0.F, 0.F)
00099             .with<ecs::Velocity>("star_vel", -20.F - static_cast<float>(std::rand() % 30), 0.F)
00100             .with<ecs::Color>("star_color", static_cast<unsigned char>(100U), static_cast<unsigned char>(100U),

```

```

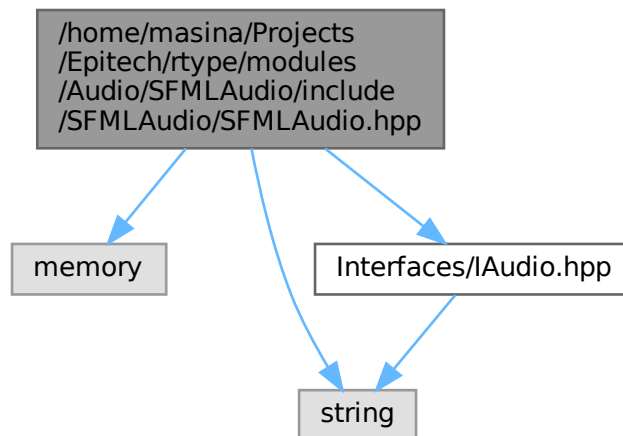
00101         static_cast<unsigned char>(200U), static_cast<unsigned char>(255U))
00102     .build();
00103 }
00104 }
00105
00106 void cli::Lobby::update(const float dt, const eng::WindowSize &size)
00107 {
00108     auto &reg = getRegistry();
00109     auto *playerTransform = reg.getComponent<ecs::Transform>(m_playerEntity);
00110     const auto *playerVelocity = reg.getComponent<ecs::Velocity>(m_playerEntity);
00111     for (auto &[entity, velocity] : reg.getAll<ecs::Velocity>())
00112     {
00113         if (auto *pixel = reg.getComponent<ecs::Pixel>(entity))
00114         {
00115             if (auto *transform = reg.getComponent<ecs::Transform>(entity))
00116             {
00117                 transform->x += velocity.x * dt;
00118                 transform->y += velocity.y * dt;
00119
00120                 if (transform->x < 2.F || transform->y < 2.F)
00121                 {
00122                     transform->x = static_cast<float>(std::rand() % (size.width * 2));
00123                     transform->y = static_cast<float>(std::rand() % size.height);
00124                 }
00125             }
00126         }
00127     }
00128     if (auto *fpsText = reg.getComponent<ecs::Text>(m_fpsEntity))
00129     {
00130         fpsText->content = "FPS " + std::to_string(static_cast<int>(1 / dt));
00131     }
00132     if (m_keysPressed[eng::Key::Up])
00133     {
00134         playerTransform->y -= playerVelocity->y * dt;
00135     }
00136     if (m_keysPressed[eng::Key::Down])
00137     {
00138         playerTransform->y += playerVelocity->y * dt;
00139     }
00140     if (m_keysPressed[eng::Key::Left])
00141     {
00142         playerTransform->x -= playerVelocity->x * dt;
00143     }
00144     if (m_keysPressed[eng::Key::Right])
00145     {
00146         playerTransform->x += playerVelocity->x * dt;
00147     }
00148     playerTransform->x = std::max(playerTransform->x, 0.F);
00149     playerTransform->y = std::max(playerTransform->y, 0.F);
00150     playerTransform->x =
00151         std::min(playerTransform->x, static_cast<float>(size.width - 66.F)); // TODO(bobis33): getTextureSize.x
00152     playerTransform->y =
00153         std::min(playerTransform->y, static_cast<float>(size.height - 40.F)); // TODO(bobis33): getTextureSize.y
00154 }
00155
00156 void cli::Lobby::event(const eng::Event &event)
00157 {
00158     auto &reg = getRegistry();
00159     switch (event.type)
00160     {
00161     case eng::EventType::KeyPressed:
00162         if (event.key == eng::Key::Up)
00163             m_keysPressed[eng::Key::Up] = true;
00164         if (event.key == eng::Key::Down)
00165             m_keysPressed[eng::Key::Down] = true;
00166         if (event.key == eng::Key::Left)
00167             m_keysPressed[eng::Key::Left] = true;
00168         if (event.key == eng::Key::Right)
00169             m_keysPressed[eng::Key::Right] = true;
00170         break;
00171     case eng::EventType::KeyReleased:
00172         if (event.key == eng::Key::Up)
00173             m_keysPressed[eng::Key::Up] = false;
00174         if (event.key == eng::Key::Down)
00175             m_keysPressed[eng::Key::Down] = false;
00176         if (event.key == eng::Key::Left)
00177             m_keysPressed[eng::Key::Left] = false;
00178         if (event.key == eng::Key::Right)
00179             m_keysPressed[eng::Key::Right] = false;
00180         break;
00181     default:
00182         break;
00183     }
00184 }
00185 }
00186 }

```

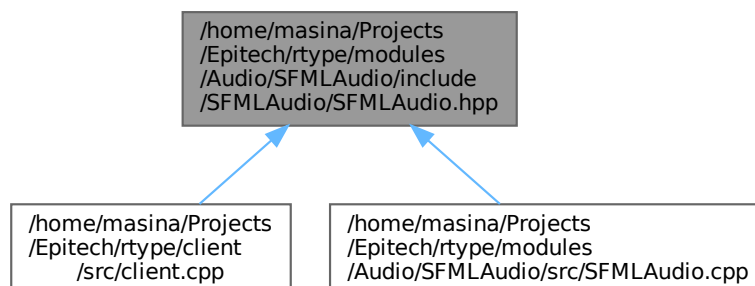
8.33 /home/masina/Projects/Epitech/rtype/modules/Audio/↔ SFMLAudio/include/SFMLAudio/SFMLAudio.hpp File Reference

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](#)

8.33.1 Detailed Description

SFMLAudio class declaration.

Definition in file [SFMLAudio.hpp](#).

8.34 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         void stopAudio(const std::string &name) override;  

00038         Status isPlaying(const std::string &name) override;  

00039  

00040     private:  

00041         struct Impl;  

00042         std::unique_ptr<Impl> pImpl;  

00043     }; // class SFMLAudio  

00044  

00045 } // namespace eng

```

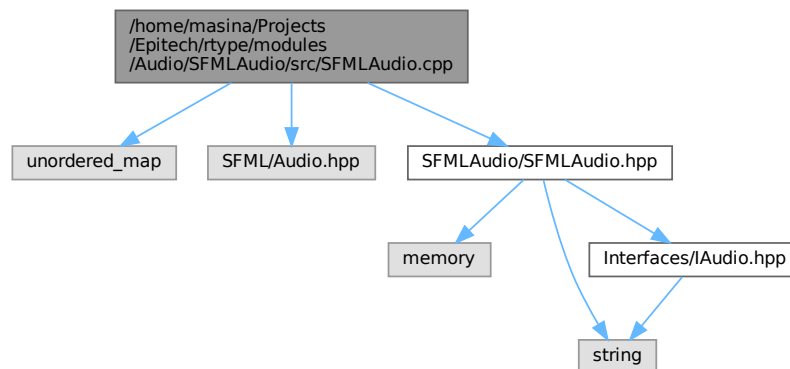
8.35 /home/masina/Projects/Epitech/rtype/modules/Audio/↵ SFMLAudio/src/SFMLAudio.cpp File Reference

```

#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`

8.36 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, const float volume, const 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         if (const auto it = pImpl->musics.find(name); it != pImpl->musics.end())
00033             it->second->play();
00034     }
00035 }

```



```

00034     }
00035
00036 void SFMLAudio::setVolume(const std::string &name, const float volume)
00037 {
00038     if (const auto it = pImpl->musics.find(name); it != pImpl->musics.end())
00039         it->second->setVolume(volume);
00040 }
00041
00042 void SFMLAudio::setLoop(const std::string &name, const bool loop)
00043 {
00044     if (const auto it = pImpl->musics.find(name); it != pImpl->musics.end())
00045         it->second->setLooping(loop);
00046 }
00047
00048 void SFMLAudio::stopAudio(const std::string &name)
00049 {
00050     if (const auto it = pImpl->musics.find(name); it != pImpl->musics.end())
00051     {
00052         it->second->stop();
00053     }
00054 }
00055
00056 Status SFMLAudio::isPlaying(const std::string &name)
00057 {
00058     if (const auto it = pImpl->musics.find(name); it != pImpl->musics.end())
00059     {
00060         switch (it->second->getStatus())
00061         {
00062             case sf::Music::Status::Playing:
00063                 return Status::Playing;
00064             case sf::Music::Status::Paused:
00065                 return Status::Paused;
00066             case sf::Music::Status::Stopped:
00067                 return Status::Stopped;
00068             default:
00069                 return Status::Stopped;
00070         }
00071     }
00072     return Status::Stopped;
00073 }
00074 } // namespace eng

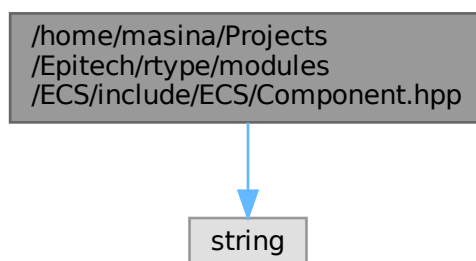
```

8.37 /home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.hpp File Reference

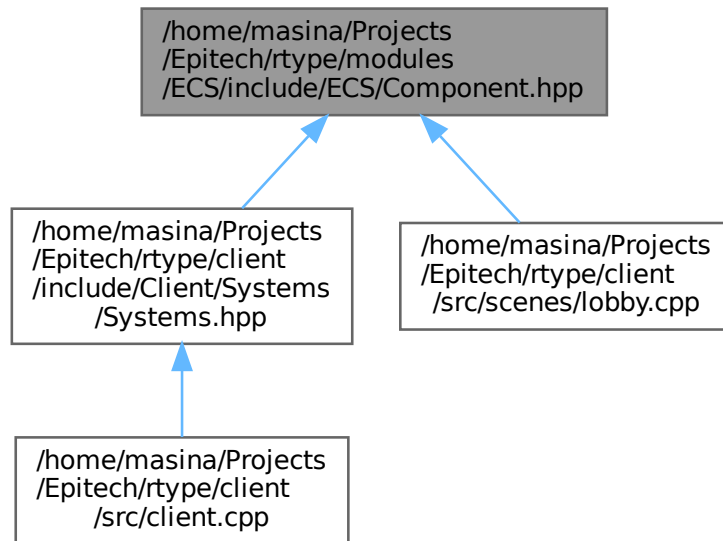
This file contains the component definitions.

#include <string>

Include dependency graph for Component.hpp:



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



Classes

- struct [ecs::IComponent](#)
- struct [ecs::Audio](#)
- struct [ecs::Color](#)
- struct [ecs::Font](#)
- struct [ecs::Mob](#)
- struct [ecs::Player](#)
- struct [ecs::Pixel](#)
- struct [ecs::Rect](#)
- struct [ecs::Scale](#)
- struct [ecs::Text](#)
- struct [ecs::Texture](#)
- struct [ecs::Transform](#)
- struct [ecs::Velocity](#)

Namespaces

- namespace [ecs](#)

8.37.1 Detailed Description

This file contains the component definitions.

Definition in file [Component.hpp](#).

8.38 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 IComponent  

00014     {  

00015         std::string id;  

00016     };  

00017     struct Audio final : IComponent  

00018     {  

00019         std::string path;  

00020         float volume;  

00021         bool loop;  

00022         bool play;  

00023     };  

00024     struct Color final : IComponent  

00025     {  

00026         unsigned char r{};  

00027         unsigned char g{};  

00028         unsigned char b{};  

00029         unsigned char a{};  

00030     };  

00031     struct Font final : IComponent  

00032     {  

00033         std::string path;  

00034     };  

00035     struct Mob final : IComponent  

00036     {  

00037         // bool is_alive{};  

00038     };  

00039     struct Player final : IComponent  

00040     {  

00041         bool is_cli{};  

00042         // bool is_alive{};  

00043     };  

00044     struct Pixel final : IComponent  

00045     {  

00046     };  

00047     struct Rect final : IComponent  

00048     { // TODO(bobis33): remove, only used for texture actually  

00049         float pos_x{}, pos_y{};  

00050         int size_x{}, size_y{};  

00051     };  

00052     struct Scale final : IComponent  

00053     {  

00054         float x{}, y{};  

00055     };  

00056     struct Text final : IComponent  

00057     {  

00058         std::string content;  

00059         unsigned int font_size;  

00060     };  

00061     struct Texture final : IComponent  

00062     {  

00063         std::string path;  

00064         // float rect_pos_x{}, rect_pos_y{};  

00065         // int rect_size_x{}, rect_size_y{};  

00066     };  

00067     struct Transform final : IComponent  

00068     {  

00069         float x{}, y{};  

00070         float rotation{};  

00071     };  

00072     struct Velocity final : IComponent  

00073     {  

00074         float x{}, y{};  

00075     };  

00076 } // namespace ecs

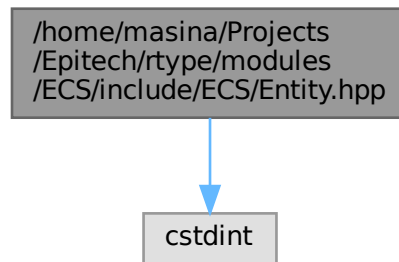
```

8.39 /home/masina/Projects/Epitech/rtype/modules/ECS/include/↵ ECS/Entity.hpp File Reference

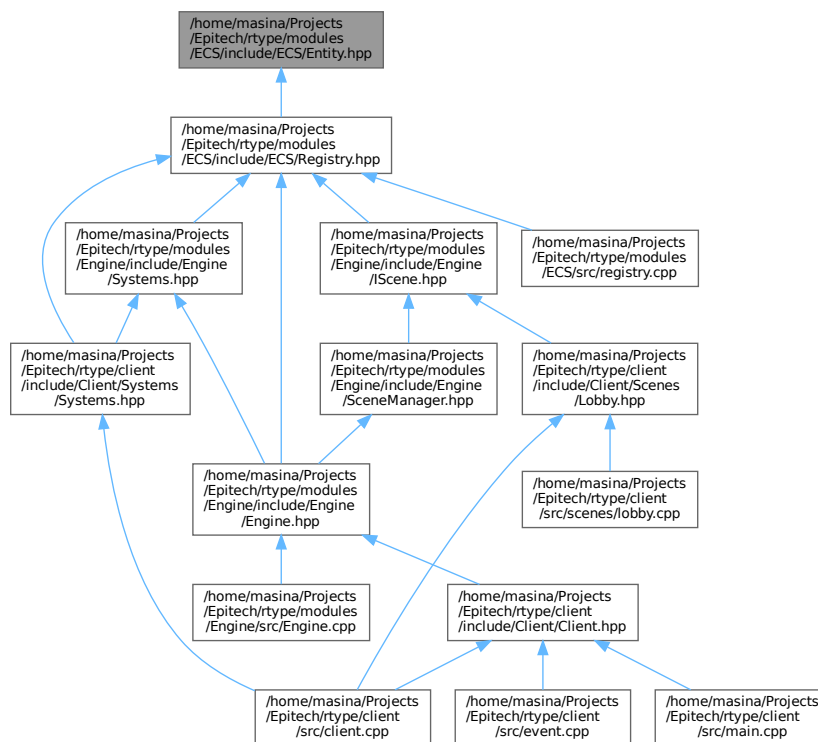
This file contains the entity definitions.

```
#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

8.39.1 Detailed Description

This file contains the entity definitions.

Definition in file [Entity.hpp](#).

8.40 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
```

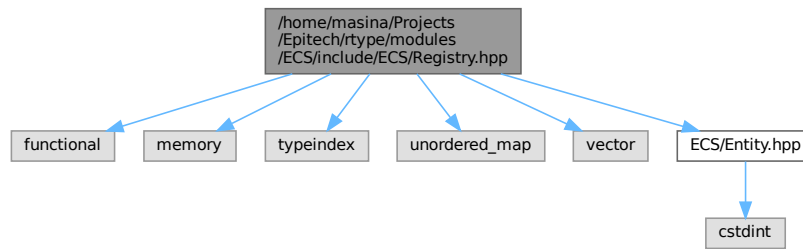
8.41 `/home/masina/Projects/Epitech/rtype/modules/ECS/include/↵`
ECS/Registry.hpp File Reference

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::EntityBuilder](#)
- class [ecs::Registry::IPool](#)
- class [ecs::Registry::Pool< T >](#)

Namespaces

- namespace [ecs](#)

8.41.1 Detailed Description

This file contains the Registry class declaration.

Definition in file [Registry.hpp](#).

8.42 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         class EntityBuilder  

00036         {  

00037         public:  

00038             EntityBuilder(Registry &reg, Entity e) : m_registry(reg), m_entity(e) {}  

00039  

00040             template <typename T, typename... Args> EntityBuilder &with(Args &&...args)  

00041             {  

00042                 m_registry.addComponent<T>(m_entity, std::forward<Args>(args)...);  

00043                 return *this;  

00044             }  

00045  

00046             Entity build() const { return m_entity; }  

00047  

00048         private:  

00049             Registry &m_registry;  

00050             Entity m_entity;  

00051         };  

00052  

00053         EntityBuilder createEntity()  

00054         {  

00055             const Entity entity = ++m_lastEntity;  

00056             m_entities.push_back(entity);  

00057             return EntityBuilder(*this, entity);  

00058         }  

00059  

00060         template <typename T, typename... Args> T &addComponent(Entity e, Args &&...args)  

00061         {  

00062             auto &pool = getPool<T>();  

00063             T &comp = pool.add(e, std::forward<Args>(args)...);  

00064             for (auto &cb : m_onComponentAddedCallbacks)  

00065             {  

00066                 cb(e, typeid(T));  

00067             }  

00068             return comp;  

00069         }  

00070     }

```

```

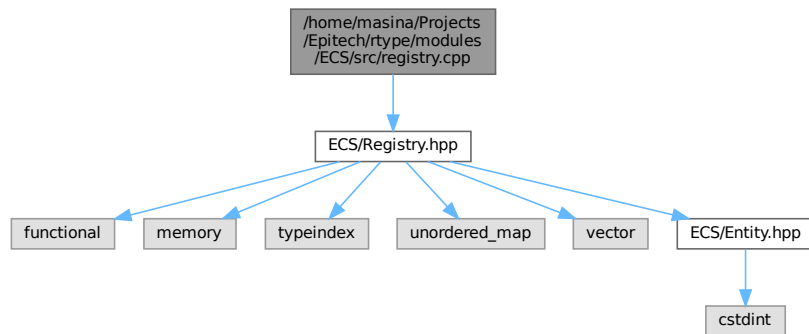
00071     template <typename T> T *GetComponent(Entity e)
00072     {
00073         auto &pool = getPool<T>();
00074         return pool.get(e);
00075     }
00076
00077     template <typename T> std::unordered_map<Entity, T> &getAll() { return getPool<T>().data; }
00078
00079     template <typename T> bool hasComponent(Entity e)
00080     {
00081         auto &pool = getPool<T>();
00082         return pool.has(e);
00083     }
00084
00085     template <typename T> void removeComponent(Entity e)
00086     {
00087         auto &pool = getPool<T>();
00088         pool.remove(e);
00089     }
00090
00091     void onComponentAdded(std::function<void(Entity, const std::type_info &)> cb)
00092     {
00093         m_onComponentAddedCallbacks.push_back(std::move(cb));
00094     }
00095
00096 private:
00097     class IPool
00098     {
00099     public:
00100         virtual ~IPool() = default;
00101         virtual void remove(Entity e) = 0;
00102     };
00103
00104     template <typename T> class Pool final : public IPool
00105     {
00106     public:
00107         std::unordered_map<Entity, T> data;
00108
00109         template <typename... Args> T &add(Entity e, Args &&...args)
00110         {
00111             return data.emplace(e, T{std::forward<Args>(args)...}).first->second;
00112         }
00113
00114         T *get(Entity e)
00115         {
00116             auto it = data.find(e);
00117             if (it != data.end())
00118             {
00119                 return &it->second;
00120             }
00121             return nullptr;
00122         }
00123
00124         bool has(Entity e) { return data.contains(e); }
00125
00126         void remove(Entity e) override { data.erase(e); }
00127     };
00128
00129     template <typename T> Pool<T> &getPool()
00130     {
00131         const std::type_index ti(typeid(T));
00132         if (!m_components.contains(ti))
00133         {
00134             m_components[ti] = std::make_unique<Pool<T>>();
00135         }
00136         return *static_cast<Pool<T>>*>(m_components[ti].get());
00137     }
00138     Entity m_lastEntity = INVALID_ENTITY;
00139     std::vector<Entity> m_entities;
00140     std::unordered_map<std::type_index, std::unique_ptr<IPool>> m_components;
00141     std::vector<std::function<void(Entity, const std::type_info &)>> m_onComponentAddedCallbacks;
00142
00143 }; // class Registry
00144
00145 } // namespace ecs

```


8.43 /home/masina/Projects/Epitech/rtype/modules/↵ ECS/src/registry.cpp File Reference

```
#include "ECS/Registry.hpp"
```

Include dependency graph for registry.cpp:



8.44 registry.cpp

[Go to the documentation of this file.](#)

```
00001 #include "ECS/Registry.hpp"
```

8.45 /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/SceneManager.hpp"
```

```
#include "Engine/Systems.hpp"
```

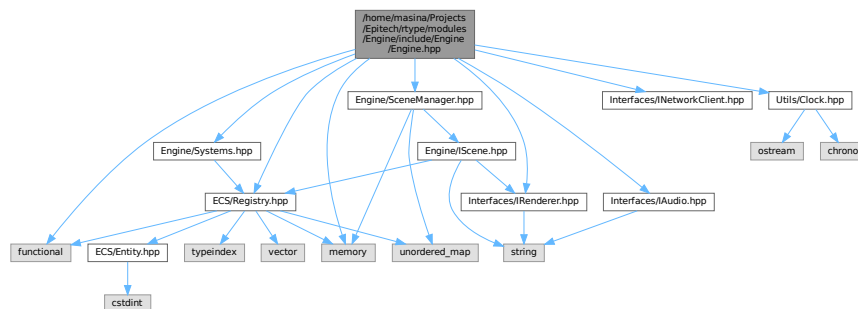
```
#include "Interfaces/IAudio.hpp"
```

```
#include "Interfaces/INetworkClient.hpp"
```

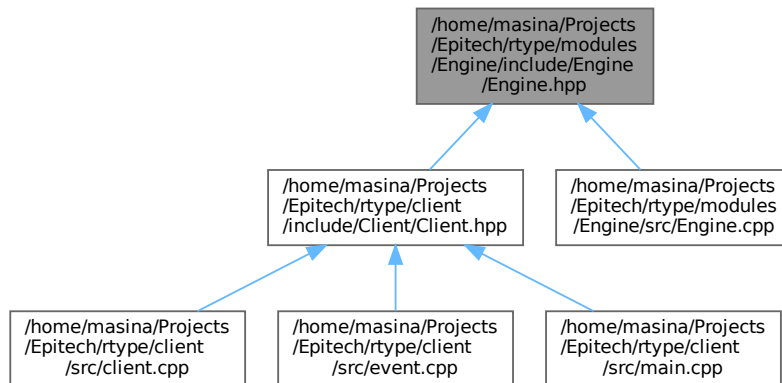
```
#include "Interfaces/IRenderer.hpp"
```

```
#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](#)

Enumerations

- enum [eng::State](#) : unsigned char { [eng::STOP](#) = 0 , [eng::RUN](#) = 1 , [eng::DEFAULT](#) = 2 }

8.45.1 Detailed Description

This file contains the Engine class declaration.

Definition in file [Engine.hpp](#).

8.46 Engine.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  

00012 #include "ECS/Registry.hpp"
  
```

```

00013 #include "Engine/SceneManager.hpp"
00014 #include "Engine/Systems.hpp"
00015 #include "Interfaces/IAudio.hpp"
00016 #include "Interfaces/INetworkClient.hpp"
00017 #include "Interfaces/IRenderer.hpp"
00018 #include "Utils/Clock.hpp"
00019
00020 namespace eng
00021 {
00022
00023     enum State : unsigned char
00024     {
00025         STOP = 0,
00026         RUN = 1,
00027         DEFAULT = 2,
00028     };
00029
00030     ///
00031     /// @class Engine
00032     /// @brief Class for the game engine
00033     /// @namespace eng
00034     ///
00035     class Engine
00036     {
00037
00038     public:
00039         Engine(const std::function<std::unique_ptr<IAudio>()> &audioFactory,
00040             const std::function<std::unique_ptr<INetworkClient>()> &networkFactory,
00041             const std::function<std::unique_ptr<IRenderer>()> &rendererFactory);
00042         ~Engine() = default;
00043
00044         Engine(const Engine &) = delete;
00045         Engine &operator=(const Engine &) = delete;
00046         Engine(Engine &&) = delete;
00047         Engine &operator=(Engine &&) = delete;
00048
00049         std::unique_ptr<IAudio> &getAudio() { return m_audio; }
00050         std::unique_ptr<INetworkClient> &getNetwork() { return m_network; }
00051         std::unique_ptr<IRenderer> &getRenderer() { return m_renderer; }
00052         std::unique_ptr<utl::Clock> &getClock() { return m_clock; }
00053         std::unique_ptr<SceneManager> &getSceneManager() { return m_sceneManager; }
00054         State getState() const { return m_state; }
00055
00056         void addSystem(std::unique_ptr<ISystem> system) { m_systems.emplace_back(std::move(system)); }
00057         void setState(const State newState) { m_state = newState; }
00058
00059         void render(ecs::Registry &registry, Color clearColor, float dt) const;
00060         void stop() const { m_renderer->closeWindow(); }
00061
00062     private:
00063         void updateSystems(ecs::Registry &registry, float dt) const;
00064
00065         State m_state = RUN;
00066         std::unique_ptr<utl::Clock> m_clock;
00067         std::unique_ptr<SceneManager> m_sceneManager;
00068         std::vector<std::unique_ptr<ISystem> > m_systems;
00069         std::unique_ptr<IAudio> m_audio;
00070         std::unique_ptr<INetworkClient> m_network;
00071         std::unique_ptr<IRenderer> m_renderer;
00072     }; // class Engine
00073
00074 } // namespace eng

```

8.47 /home/masina/Projects/Epitech/rtype/modules/Engine/include/↵ Engine/IScene.hpp File Reference

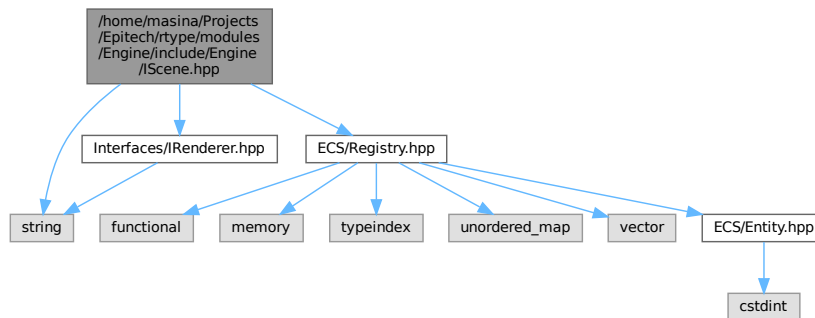
This file contains the IScene class.

```

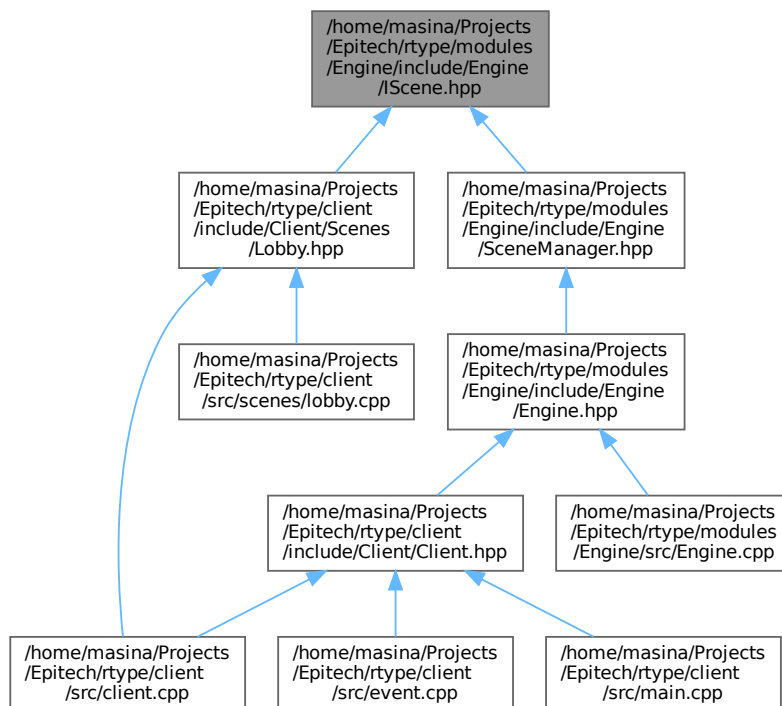
#include <string>
#include "ECS/Registry.hpp"
#include "Interfaces/IRenderer.hpp"

```

Include dependency graph for IScene.hpp:



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



Classes

- class `eng::IScene`
interface class for scene
- class `eng::AScene`
Class for scene.

Namespaces

- namespace [eng](#)

Typedefs

- using [eng::id](#) = unsigned int

8.47.1 Detailed Description

This file contains the IScene class.

Definition in file [IScene.hpp](#).

8.48 IScene.hpp

[Go to the documentation of this file.](#)

```

00001 ///  

00002 ///  

00003 ///  

00004 ///  

00005 ///  

00006 ///  

00007 #pragma once  

00008  

00009 #include <string>  

00010  

00011 #include "ECS/Registry.hpp"  

00012 #include "Interfaces/IRenderer.hpp"  

00013  

00014 namespace eng  

00015 {  

00016  

00017     using id = unsigned int;  

00018  

00019     ///  

00020     ///  

00021     ///  

00022     ///  

00023     ///  

00024     class IScene  

00025     {  

00026     public:  

00027         virtual ~IScene() = default;  

00028  

00029         [[nodiscard]] virtual std::string &getName() = 0;  

00030         [[nodiscard]] virtual id getId() const = 0;  

00031         [[nodiscard]] virtual ecs::Registry &getRegistry() = 0;  

00032  

00033         virtual void setName(const std::string &newName) = 0;  

00034  

00035         virtual void update(float dt, const WindowSize &size) = 0;  

00036         virtual void event(const Event &event) = 0;  

00037  

00038     }; // class IScene  

00039  

00040     ///  

00041     ///  

00042     ///  

00043     ///  

00044     ///  

00045     class AScene : public IScene  

00046     {  

00047     public:  

00048         AScene() : m_id(s_nextId++) {}  

00049         ~AScene() override = default;  

00050  

00051         AScene(const AScene &other) = delete;  

00052         AScene(AScene &&other) = delete;  

00053         AScene &operator=(const AScene &other) = delete;  

00054         AScene &operator=(AScene &&other) = delete;  


```

```

00055
00056 [[nodiscard]] std::string &getName() override { return m_name; }
00057 [[nodiscard]] id getId() const override { return m_id; }
00058 [[nodiscard]] ecs::Registry &getRegistry() override { return m_registry; }
00059
00060 void setName(const std::string &newName) override { m_name = newName; }
00061
00062 private:
00063     std::string m_name = "default_name";
00064     id m_id = 1;
00065     ecs::Registry m_registry;
00066     inline static id s_nextId = 1;
00067 }; // class AScene
00068
00069 } // namespace eng

```

8.49 /home/masina/Projects/Epitech/rtype/modules/Engine/include/↵ Engine/SceneManager.hpp File Reference

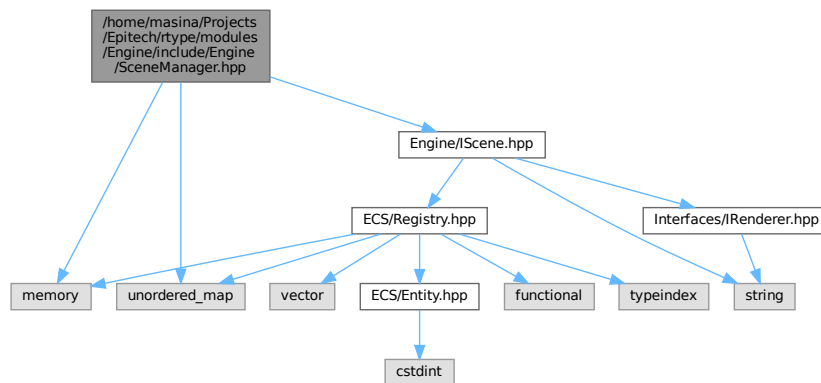
This file contains the SceneManager class declaration.

```

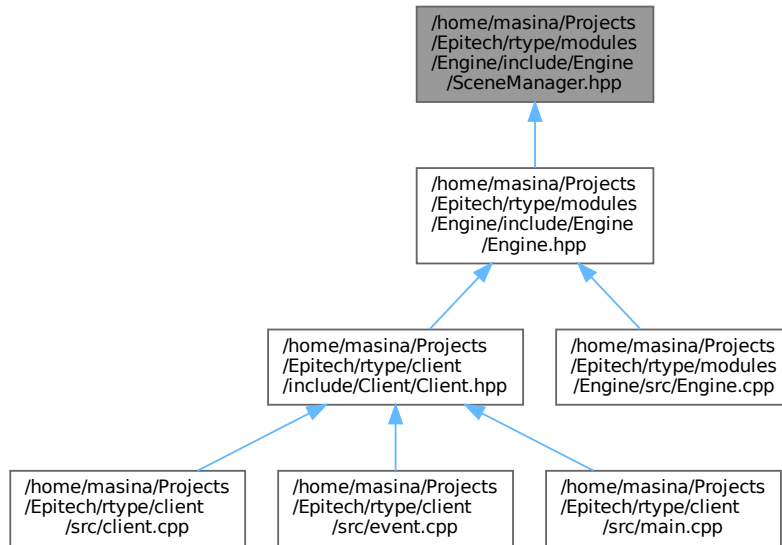
#include <memory>
#include <unordered_map>
#include "Engine/IScene.hpp"

```

Include dependency graph for SceneManager.hpp:



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



Classes

- class [eng::SceneManager](#)
Class for managing scenes.

Namespaces

- namespace [eng](#)

8.49.1 Detailed Description

This file contains the SceneManager class declaration.

Definition in file [SceneManager.hpp](#).

8.50 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/IScene.hpp"
  
```

```

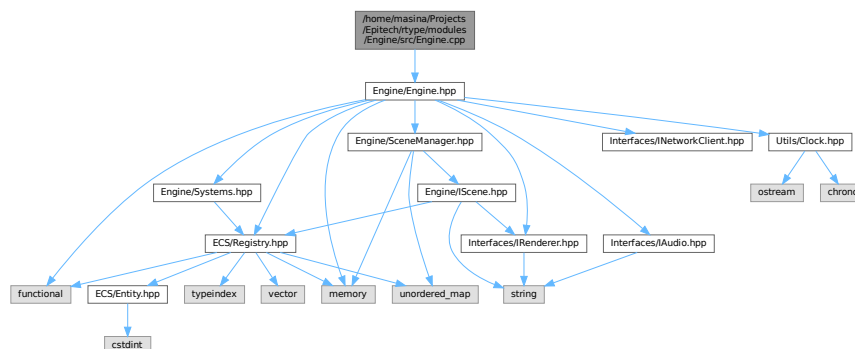
00013
00014 namespace eng
00015 {
00016
00017     ///
00018     /// @class SceneManager
00019     /// @brief Class for managing scenes
00020     /// @namespace eng
00021     ///
00022     class SceneManager
00023     {
00024
00025     public:
00026         SceneManager() = default;
00027         ~SceneManager() = 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<IScene> &getScene(const id sceneId) { return m_scenes.at(sceneId); }
00035         std::unique_ptr<IScene> &getCurrentScene() { return m_scenes.at(m_currentSceneId); }
00036         void switchToScene(const id sceneId) { m_currentSceneId = sceneId; }
00037         void addScene(std::unique_ptr<IScene> scene) { m_scenes[scene->getId()] = std::move(scene); }
00038
00039         // template <typename... EntityDefs>
00040         // IScene &createScene(const std::string &name, const std::function<void(const Event&)> eventHandler, const
00041         // std::function<void(float)> updateHandler,
00042         //     EntityDefs&&... defs) {
00043         //     auto scene = std::make_unique<IScene>();
00044         //     scene->setName(name);
00045         //     scene->setEventHandler(eventHandler);
00046         //     scene->setUpdateHandler(updateHandler);
00047         //
00048         //     Scene &ref = *scene;
00049         //     (defs(ref.getRegistry()), ...);
00050         //
00051         //     m_scenes[ref.getId()] = std::move(scene);
00052         //     return ref;
00053         // }
00054
00055     private:
00056         std::unordered_map<id, std::unique_ptr<IScene> > m_scenes;
00057         id m_currentSceneId = 1;
00058     }; // class SceneManager
00059 } // namespace eng

```

8.51 /home/masina/Projects/Epitech/rtype/modules/Engine/src/↵ Engine.cpp File Reference

#include "Engine/Engine.hpp"

Include dependency graph for Engine.cpp:



8.52 Engine.cpp

[Go to the documentation of this file.](#)

```

00001 #include "Engine/Engine.hpp"
00002
00003 eng::Engine::Engine(const std::function<std::unique_ptr<IAudio>()> &audioFactory,
00004                     const std::function<std::unique_ptr<INetworkClient>()> &networkFactory,
00005                     const std::function<std::unique_ptr<IRenderer>()> &rendererFactory)
00006 : m_clock(std::make_unique<utl::Clock>()), m_sceneManager(std::make_unique<SceneManager>()),
00007   m_audio(audioFactory()), m_network(networkFactory()), m_renderer(rendererFactory())
00008 {
00009 }
00010
00011 void eng::Engine::updateSystems(ecs::Registry &registry, const float dt) const
00012 {
00013     for (const auto &system : m_systems)
00014     {
00015         system->update(registry, dt);
00016     }
00017 }
00018
00019 void eng::Engine::render(ecs::Registry &registry, const Color clearColor, const float dt) const
00020 {
00021     m_renderer->clearWindow(clearColor);
00022     updateSystems(registry, dt);
00023     m_renderer->displayWindow();
00024 }

```

8.53 /home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Client/include/R-TypeClient/LobbyScene.hpp File Reference

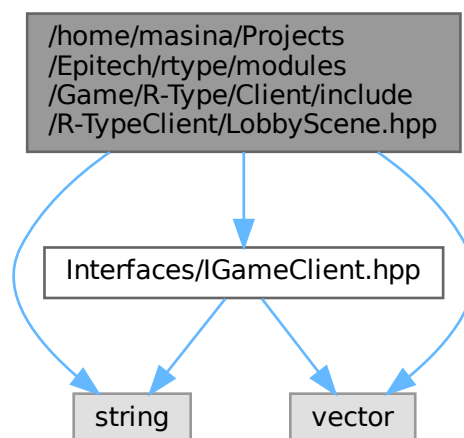
This file contains the lobby scene.

```

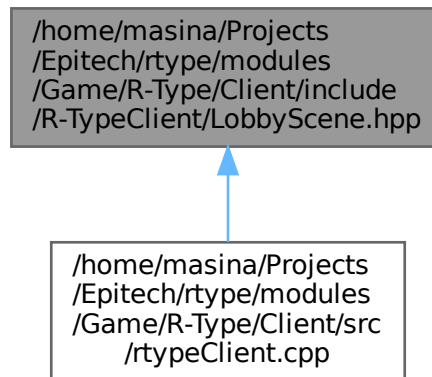
#include <string>
#include <vector>
#include "Interfaces/IGameClient.hpp"

```

Include dependency graph for LobbyScene.hpp:



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



Classes

- class [gme::LobbyScene](#)
Class for the Lobby scene.

Namespaces

- namespace [gme](#)

8.53.1 Detailed Description

This file contains the lobby scene.

Definition in file [LobbyScene.hpp](#).

8.54 LobbyScene.hpp

[Go to the documentation of this file.](#)

```

00001 ///  

00002 ///  

00003 ///  

00004 ///  

00005 ///  

00006 ///  

00007 #pragma once  

00008  

00009 #include <string>  

00010 #include <vector>  

00011  

00012 #include "Interfaces/IGameClient.hpp"  

00013  

00014 namespace gme  

00015 {  

00016  

00017     ///  

  
```

```

00018  /// @class LobbyScene
00019  /// @brief Class for the Lobby scene
00020  /// @namespace gme
00021  ///
00022  class LobbyScene final : public IScene
00023  {
00024  public:
00025      LobbyScene() : m_name("Lobby") {}
00026
00027      [[nodiscard]] const std::string &getName() const override { return m_name; }
00028      [[nodiscard]] const std::vector<Sprite> &getEntities() const override { return m_entities; }
00029
00030      void addEntity(const Sprite &e) { m_entities.push_back(e); }
00031      std::vector<Sprite> &getEntitiesMutable() override { return m_entities; }
00032
00033  private:
00034      std::string m_name;
00035      std::vector<Sprite> m_entities;
00036  }; // class LobbyScene
00037
00038 } // namespace gme

```

8.55 /home/masina/Projects/Epitech/rtype/modules/Game/R-Type/↵ Client/include/R-TypeClient/RTypeClient.hpp File Reference

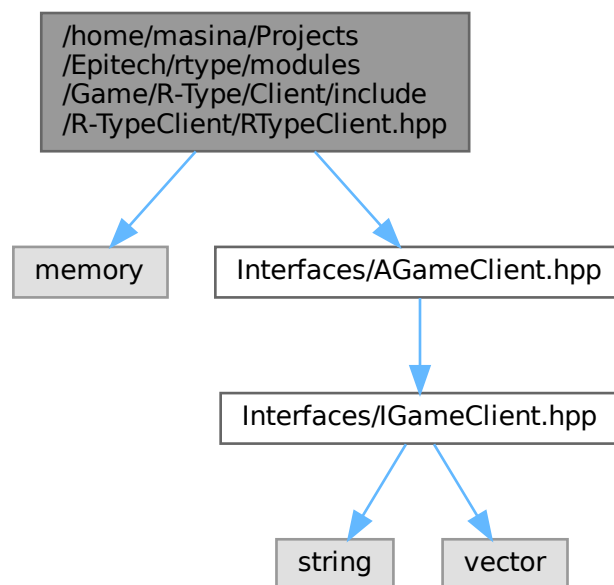
RType client class declaration.

```

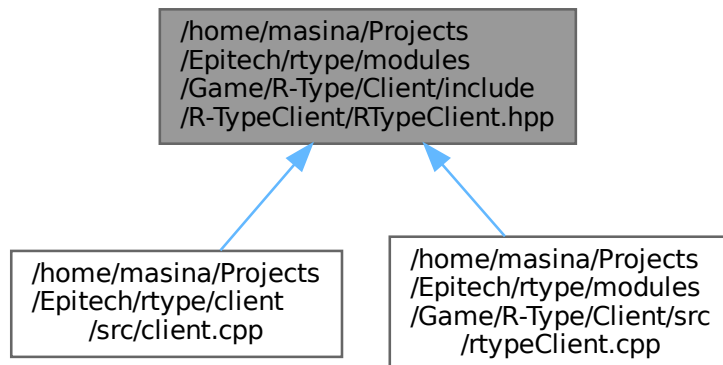
#include <memory>
#include "Interfaces/AGameClient.hpp"

```

Include dependency graph for RTypeClient.hpp:



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



Classes

- class [gme::RTypeClient](#)
Class for the R-Type game.

Namespaces

- namespace [gme](#)

8.55.1 Detailed Description

RType client class declaration.

Definition in file [RTypeClient.hpp](#).

8.56 RTypeClient.hpp

[Go to the documentation of this file.](#)

```

00001 ///  

00002 ///  

00003 ///  

00004 ///  

00005 ///  

00006 ///  

00007 #pragma once  

00008  

00009 #include <memory>  

00010  

00011 #include "Interfaces/AGameClient.hpp"  

00012  

00013 namespace gme  

00014 {  

00015  

00016     ///  

00017     ///  

00018     ///  


```

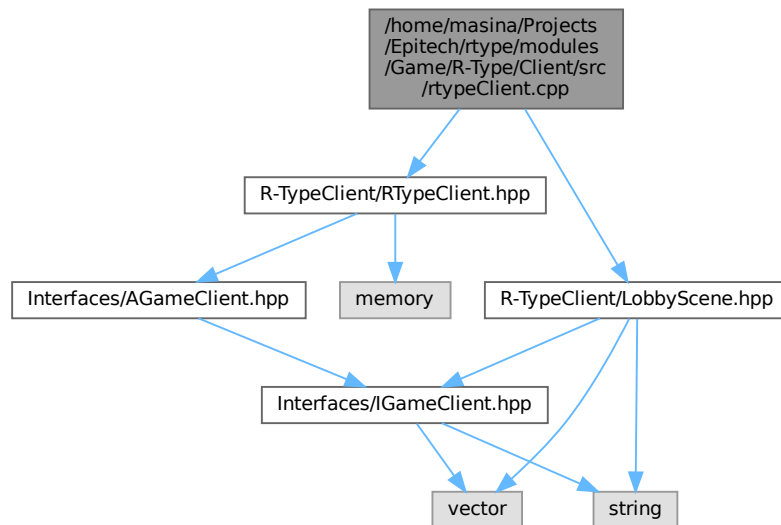
```

00019  ///< @namespace gme
00020  ///<
00021  class RTypeClient final : public AGameClient
00022  {
00023  public:
00024      RTypeClient();
00025      ~RTypeClient() override = default;
00026
00027      RTypeClient(const RTypeClient &) = delete;
00028      RTypeClient &operator=(const RTypeClient &) = delete;
00029      RTypeClient(RTypeClient &&) = delete;
00030      RTypeClient &operator=(RTypeClient &&) = delete;
00031
00032      void update(float deltaTime, unsigned int width, unsigned int height) override;
00033      [[nodiscard]] const IScene &getCurrentScene() const override { return *m_currentScene; }
00034
00035  private:
00036      std::unique_ptr<IScene> m_currentScene;
00037  }; // class RTypeClient
00038
00039 } // namespace gme

```

8.57 /home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Client/src/rtypeClient.cpp File Reference

#include "R-TypeClient/RTypeClient.hpp"
#include "R-TypeClient/LobbyScene.hpp"
Include dependency graph for rtypeClient.cpp:



8.58 rtypeClient.cpp

[Go to the documentation of this file.](#)

```

00001 #include "R-TypeClient/RTypeClient.hpp"
00002 #include "R-TypeClient/LobbyScene.hpp"
00003
00004 gme::RTypeClient::RTypeClient() : m_currentScene(std::make_unique<LobbyScene>()) {
00005     AGameClient::setName("R-Type"); }
00005
00006 void gme::RTypeClient::update(const float deltaTime, const unsigned int width, const unsigned int height) {}

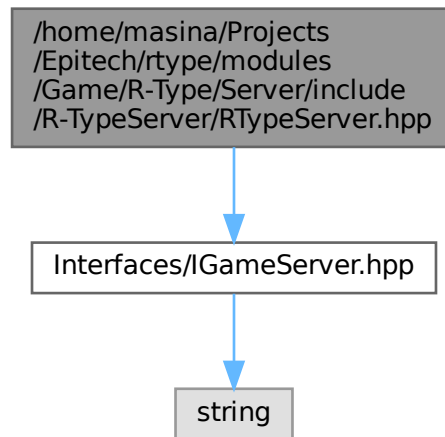
```

8.59 /home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Server/include/R-TypeServer/RTypeServer.hpp File Reference

RType client class declaration.

```
#include "Interfaces/IGameServer.hpp"
```

Include dependency graph for RTypeServer.hpp:



Classes

- class `gme::RTypeServer`
Class for the R-Type game.

Namespaces

- namespace `gme`

8.59.1 Detailed Description

RType client class declaration.

Definition in file [RTypeServer.hpp](#).

8.60 RTypeServer.hpp

[Go to the documentation of this file.](#)

```

00001 ///  

00002 ///  

00003 ///  

00004 ///  

00005 ///  

00006 ///  

00007 #pragma once  

00008 ///  

00009 #include "Interfaces/IGameServer.hpp"  

00010 ///  

00011 namespace gme  

00012 {  

00013     ///  

00014     ///  

00015     ///  

00016     ///  

00017     ///  

00018     ///  

00019     class RTypeServer final : public IGameServer  

00020     {  

00021     public:  

00022         RTypeServer() = default;  

00023         ~RTypeServer() override = default;  

00024         RTypeServer(const RTypeServer &) = delete;  

00025         RTypeServer &operator=(const RTypeServer &) = delete;  

00026         RTypeServer(RTypeServer &&) = delete;  

00027         RTypeServer &operator=(RTypeServer &&) = delete;  

00028     private:  

00029     };  

00030 } // namespace gme

```

8.61 /home/masina/Projects/Epitech/rtype/modules/Game/R-Type/↵ Server/src/rtypeServer.cpp File Reference

8.62 rtypeServer.cpp

[Go to the documentation of this file.](#)

```

00001

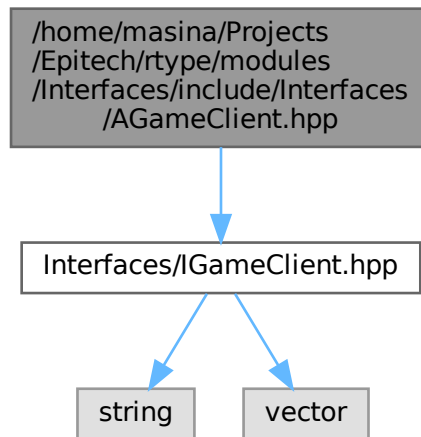
```

8.63 /home/masina/Projects/Epitech/rtype/modules/↵ Interfaces/include/Interfaces/AGameClient.hpp File Reference

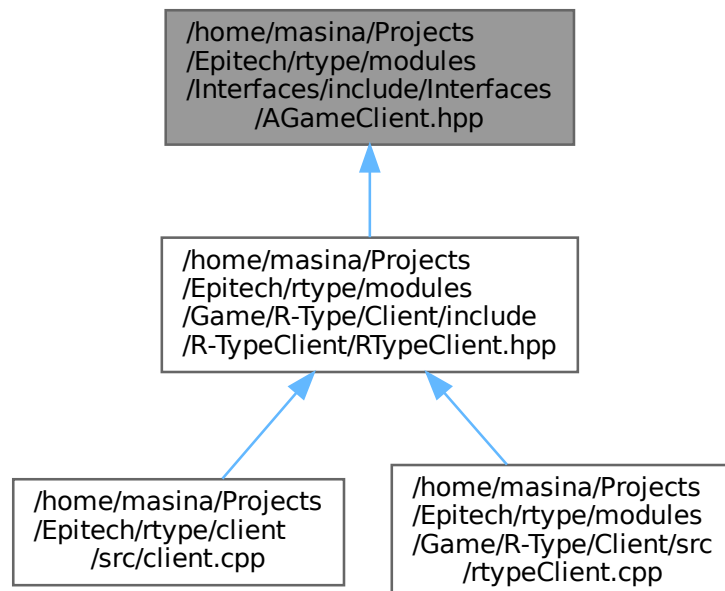
This file contains the game abstract class.

```
#include "Interfaces/IGameClient.hpp"
```

Include dependency graph for AGameClient.hpp:



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



Classes

- class `gme::AGameClient`
Abstraction for the games.

Namespaces

- namespace [gme](#)

8.63.1 Detailed Description

This file contains the game abstract class.

Definition in file [AGameClient.hpp](#).

8.64 AGameClient.hpp

[Go to the documentation of this file.](#)

```

00001 ///  

00002 ///  

00003 ///  

00004 ///  

00005 ///  

00006 ///  

00007 #pragma once  

00008  

00009 #include "Interfaces/IGameClient.hpp"  

00010  

00011 namespace gme  

00012 {  

00013  

00014     ///  

00015     ///  

00016     ///  

00017     ///  

00018     ///  

00019     class AGameClient : public IGameClient  

00020     {  

00021     public:  

00022         ~AGameClient() override = default;  

00023  

00024         [[nodiscard]] std::string &getName() override { return m_name; }  

00025         void setName(const std::string &newName) override { m_name = newName; }  

00026  

00027     private:  

00028         std::string m_name = "default_name";  

00029     }; // class AGameClient  

00030  

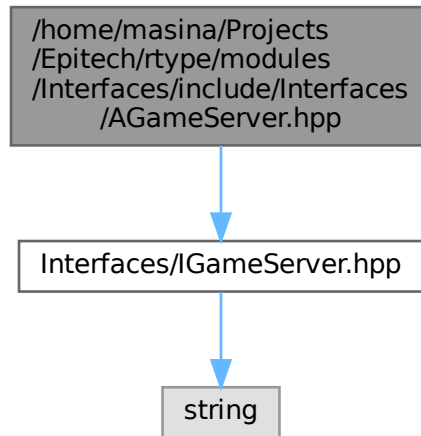
00031 } // namespace gme

```

8.65 /home/masina/Projects/Epitech/rtype/modules/ Interfaces/include/Interfaces/AGameServer.hpp File Reference

This file contains the game abstract class.

```
#include "Interfaces/IGameServer.hpp"
Include dependency graph for AGameServer.hpp:
```



Classes

- class `gme::AGameServer`
Abstraction for the games.

Namespaces

- namespace `gme`

8.65.1 Detailed Description

This file contains the game abstract class.

Definition in file `AGameServer.hpp`.

8.66 AGameServer.hpp

[Go to the documentation of this file.](#)

```
00001 ///
00002 /// @file AGameServer.hpp
00003 /// @brief This file contains the game abstract class
00004 /// @namespace gme
00005 ///
00006
00007 #pragma once
00008
00009 #include "Interfaces/IGameServer.hpp"
00010
00011 namespace gme
00012 {
00013
```

```

00014  ///
00015  /// @class AGameServer
00016  /// @brief Abstraction for the games
00017  /// @namespace gme
00018  ///
00019  class AGameServer : public IGameServer
00020  {
00021  public:
00022      ~AGameServer() override = default;
00023
00024      [[nodiscard]] std::string &getName() override { return m_name; }
00025      void setName(const std::string &newName) override { m_name = newName; }
00026
00027  private:
00028      std::string m_name = "default_name";
00029  }; // class AGameServer
00030
00031 } // namespace gme

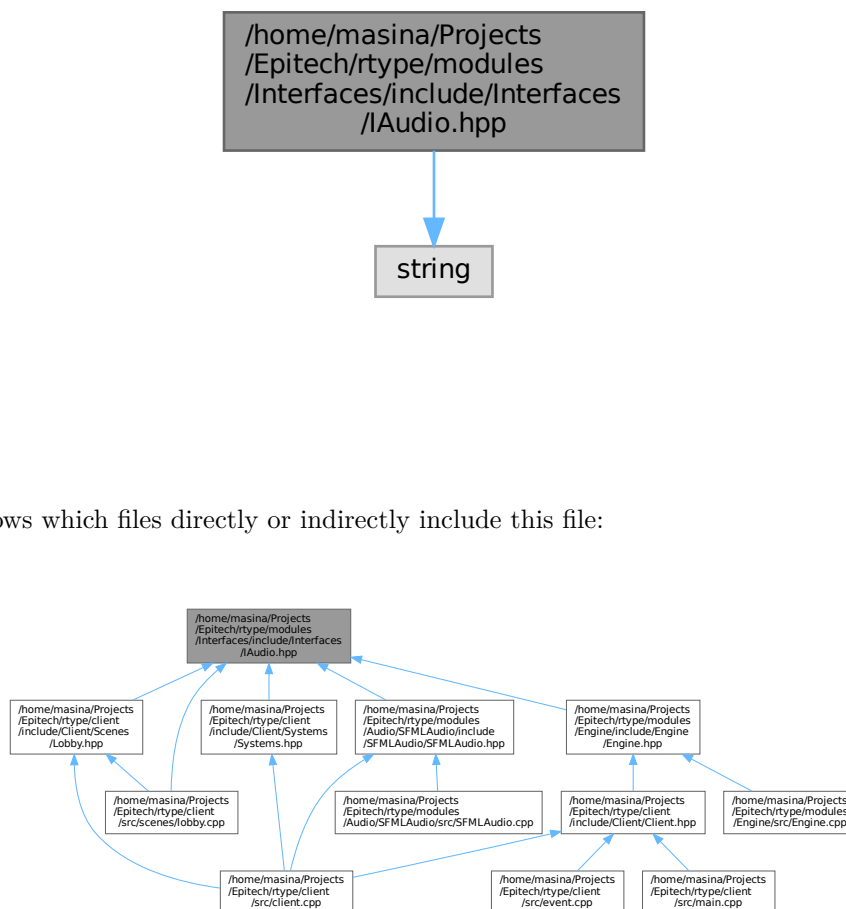
```

8.67 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IAudio.hpp File Reference

This file contains the Audio interface.

```
#include <string>
```

Include dependency graph for IAudio.hpp:



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

Classes

- class [eng::IAudio](#)
Interface for the audio.

Namespaces

- namespace [eng](#)

Enumerations

- enum class [eng::Status](#) { [eng::Stopped](#) , [eng::Paused](#) , [eng::Playing](#) }

8.67.1 Detailed Description

This file contains the Audio interface.

Definition in file [IAudio.hpp](#).

8.68 IAudio.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     enum class Status  

00015     {  

00016         Stopped,  

00017         Paused,  

00018         Playing  

00019     };  

00020  

00021 ///  

00022 ///  

00023 ///  

00024 ///  

00025 ///  

00026     class IAudio  

00027     {  

00028     public:  

00029         virtual ~IAudio() = default;  

00030  

00031         virtual void createAudio(const std::string &path, float volume, bool loop, const std::string &name) = 0;  

00032         virtual void playAudio(const std::string &name) = 0;  

00033  

00034         virtual void setVolume(const std::string &name, float volume) = 0;  

00035         virtual void setLoop(const std::string &name, bool loop) = 0;  

00036         virtual void stopAudio(const std::string &name) = 0;  

00037         virtual Status isPlaying(const std::string &name) = 0;  

00038  

00039     private:  

00040     }; // class IAudio  

00041  

00042 } // namespace eng

```

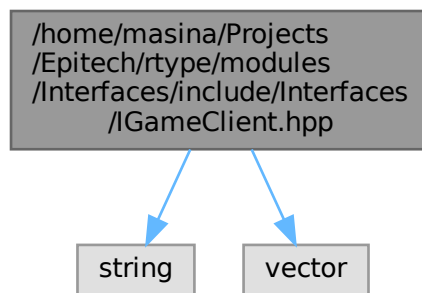
8.69 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IGameClient.hpp File Reference

This file contains the Game interface.

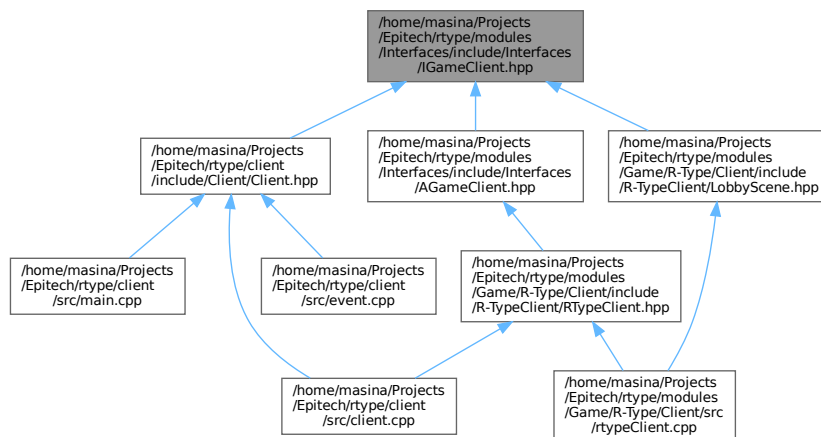
```
#include <string>
```

```
#include <vector>
```

Include dependency graph for IGameClient.hpp:



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



Classes

- struct `gme::Sprite`
- class `gme::IScene`
Interface for scenes.
- class `gme::IGameClient`
Interface for the games.

Namespaces

- namespace [gme](#)

8.69.1 Detailed Description

This file contains the Game interface.

Definition in file [IGameClient.hpp](#).

8.70 IGameClient.hpp

[Go to the documentation of this file.](#)

```

00001 ///
00002 /// @file IGameClient.hpp
00003 /// @brief This file contains the Game interface
00004 /// @namespace gme
00005 ///
00006
00007 #pragma once
00008
00009 #include <string>
00010 #include <vector>
00011
00012 namespace gme
00013 {
00014
00015     struct Sprite
00016     { // TODO(bobis33): should have path to texture, and all necessary data
00017         std::string type;
00018         float pos_x = 0.F, pos_y = 0.F;
00019         float v_x = 0.F, v_y = 0.F;
00020         float scale_x = 1.F, scale_y = 1.F;
00021         unsigned char r = 255u, g = 255u, b = 255u, a = 255u;
00022         std::string texture_path = "";
00023         float text_rect_x = 0.F, text_rect_y = 0.F;
00024         int text_rect_fx = 0, text_rect_fy = 0;
00025         std::string id;
00026     };
00027
00028     ///
00029     /// @class IScene
00030     /// @brief Interface for scenes
00031     /// @namespace gme
00032     ///
00033     class IScene
00034     {
00035     public:
00036         virtual ~IScene() = default;
00037
00038         [[nodiscard]] virtual const std::string &getName() const = 0;
00039         [[nodiscard]] virtual const std::vector<Sprite> &getEntities() const = 0;
00040         [[nodiscard]] virtual std::vector<Sprite> &getEntitiesMutable() = 0;
00041     }; // class IScene
00042
00043     ///
00044     /// @class IGameClient
00045     /// @brief Interface for the games
00046     /// @namespace gme
00047     ///
00048     class IGameClient
00049     {
00050     public:
00051         virtual ~IGameClient() = default;
00052
00053         [[nodiscard]] virtual std::string &getName() = 0;
00054         virtual void setName(const std::string &newName) = 0;
00055
00056         virtual void update(float deltaTime, unsigned int width, unsigned int height) = 0;
00057         [[nodiscard]] virtual const IScene &getCurrentScene() const = 0;
00058
00059     private:
00060     }; // class IGameClient
00061
00062 } // namespace gme

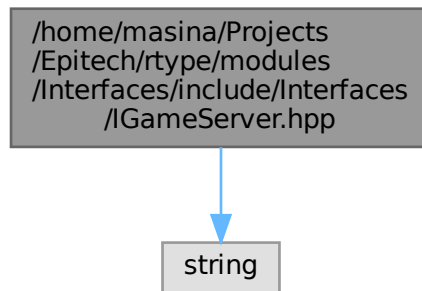
```

8.71 /home/masina/Projects/Epitech/rtype/modules/ Interfaces/include/Interfaces/IGameServer.hpp File Reference

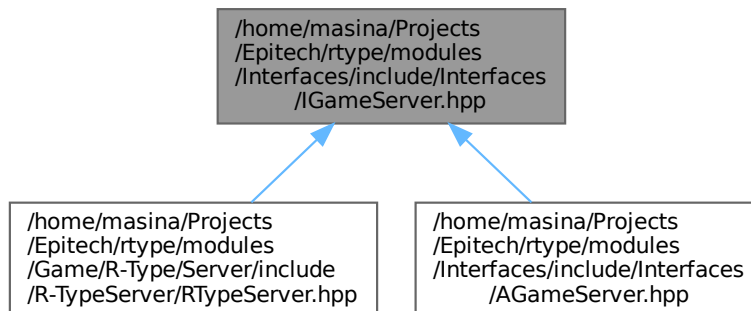
This file contains the Game interface.

```
#include <string>
```

Include dependency graph for IGameServer.hpp:



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



Classes

- class `gme::IGameServer`
Interface for the games.

Namespaces

- namespace `gme`

8.71.1 Detailed Description

This file contains the Game interface.

Definition in file [IGameServer.hpp](#).

8.72 IGameServer.hpp

[Go to the documentation of this file.](#)

```

00001 ///  

00002 ///  

00003 ///  

00004 ///  

00005 ///  

00006 ///  

00007 #pragma once  

00008  

00009 #include <string>  

00010  

00011 namespace gme  

00012 {  

00013  

00014     ///  

00015     ///  

00016     ///  

00017     ///  

00018     ///  

00019     class IGameServer  

00020     {  

00021     public:  

00022         virtual ~IGameServer() = default;  

00023  

00024         [[nodiscard]] virtual std::string &getName();  

00025         virtual void setName(const std::string &newName);  

00026  

00027     private:  

00028     }; // class IGameServer  

00029  

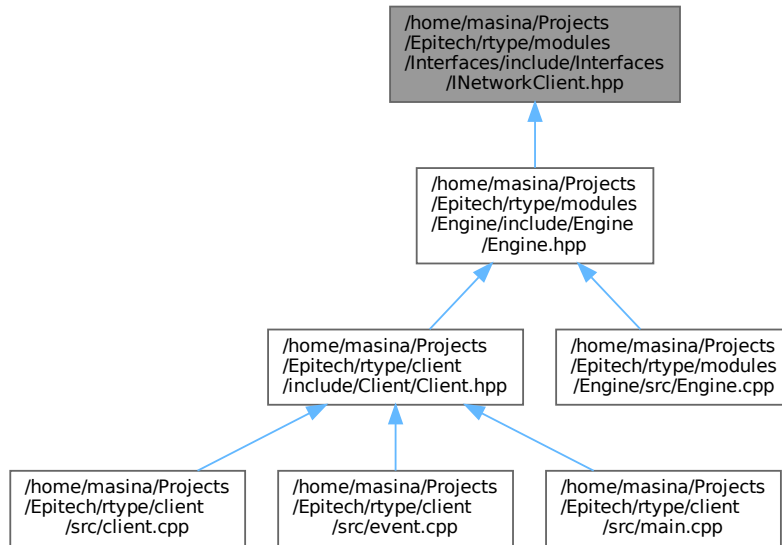
00030 } // namespace gme

```

8.73 /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](#)

8.73.1 Detailed Description

This file contains the client network interface.

Definition in file [INetworkClient.hpp](#).

8.74 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

```

8.75 /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`

8.75.1 Detailed Description

This file contains the server network interface.

Definition in file `INetworkServer.hpp`.

8.76 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

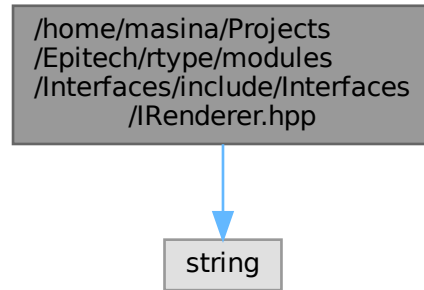
```

8.77 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IRenderer.hpp File Reference

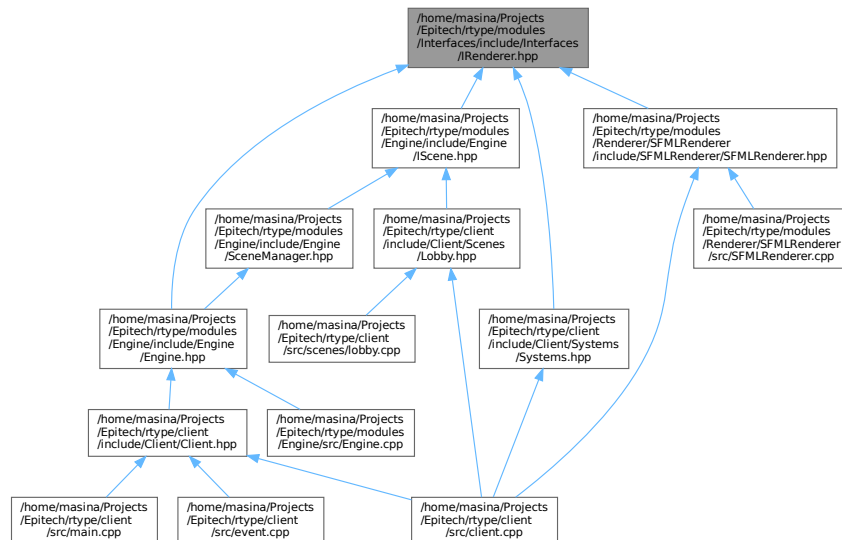
This file contains the IRenderer class declaration.

```
#include <string>
```

Include dependency graph for IRenderer.hpp:



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



Classes

- struct `eng::Color`
- struct `eng::Text`
- struct `eng::Event`
- struct `eng::WindowSize`
- class `eng::IRenderer`

Interface for the renderer.

Namespaces

- namespace [eng](#)

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](#) }

8.77.1 Detailed Description

This file contains the IRenderer class declaration.

Definition in file [IRenderer.hpp](#).

8.78 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 Color  
00014     {  
00015         unsigned char r;  
00016         unsigned char g;  
00017         unsigned char b;  
00018         unsigned char a;  
00019     };  
00020     struct Text  
00021     {  
00022         std::string font_name;  
00023         Color color;  
00024         std::string content;  
00025         unsigned int size;  
00026         float x;  
00027         float y;  
00028         std::string name;  
00029     };  
00030  
00031     enum class Key  
00032     {  
00033         Unknown,  
00034         Escape,  
00035         Space,  
00036         Up,
```

```

00037     Down,
00038     Left,
00039     Right,
00040     A,
00041     B,
00042     C,
00043     D,
00044     E,
00045     F,
00046     G,
00047     H,
00048     I,
00049     J,
00050     K,
00051     L,
00052     M,
00053     N,
00054     O,
00055     P,
00056     Q,
00057     R,
00058     S,
00059     T,
00060     U,
00061     V,
00062     W,
00063     X,
00064     Y,
00065     Z,
00066     Num0,
00067     Num1,
00068     Num2,
00069     Num3,
00070     Num4,
00071     Num5,
00072     Num6,
00073     Num7,
00074     Num8,
00075     Num9
00076 };
00077 enum class EventType
00078 {
00079     Closed,
00080     KeyPressed,
00081     KeyReleased,
00082     None
00083 };
00084
00085 struct Event
00086 {
00087     EventType type = EventType::None;
00088     Key key = Key::Unknown;
00089 };
00090
00091 struct WindowSize
00092 {
00093     unsigned int width;
00094     unsigned int height;
00095 };
00096
00097 ///
00098 /// @class IRenderer
00099 /// @brief Interface for the renderer
00100 /// @namespace eng
00101 ///
00102 class IRenderer
00103 {
00104
00105     public:
00106         virtual ~IRenderer() = default;
00107
00108         virtual void createWindow(const std::string &title, unsigned int height, unsigned int width,
00109                                 unsigned int frameLimit, bool fullscreen) = 0;
00110         [[nodiscard]] virtual bool windowIsOpen() const = 0;
00111         virtual void closeWindow() = 0;
00112         virtual void clearWindow(Color color) = 0;
00113         virtual void displayWindow() = 0;
00114         [[nodiscard]] virtual WindowSize getWindowSize() = 0;
00115
00116         [[nodiscard]] virtual bool pollEvent(Event &event) = 0;
00117         virtual void setFrameLimit(unsigned int frameLimit) = 0;
00118
00119         virtual void createFont(const std::string &name, const std::string &path) = 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, float x, float y) = 0;

```

```

00124     virtual void setTextColor(const std::string &name, Color color) = 0;
00125
00126     virtual void createTexture(const std::string &name, const std::string &path) = 0;
00127     virtual void createSprite(const std::string &name, const std::string &textureName, float x, float y,
00128                             float scale_x = 1, float scale_y = 1, int fx = 0, int fy = 0, int fnx = -1,
00129                             int fny = -1) = 0;
00130     virtual void drawSprite(const std::string &name) = 0;
00131     virtual void setSpritePosition(const std::string &name, float x, float y) = 0;
00132     virtual void setSpriteTexture(const std::string &name, const std::string &path) = 0;
00133     virtual void setSpriteScale(const std::string &name, int x, int y) = 0;
00134     virtual void setSpriteFrame(const std::string &name, int fx, int fy, int fnx, int fny) = 0;
00135
00136     virtual void drawPoint(float x, float y, Color color) = 0;
00137
00138     private:
00139 }; // class IRenderer
00140
00141 } // namespace eng

```

8.79 /home/masina/Projects/Epitech/rtype/modules/Renderer/SFMLRenderer/include/SFMLRenderer/SFMLRenderer.hpp File Reference

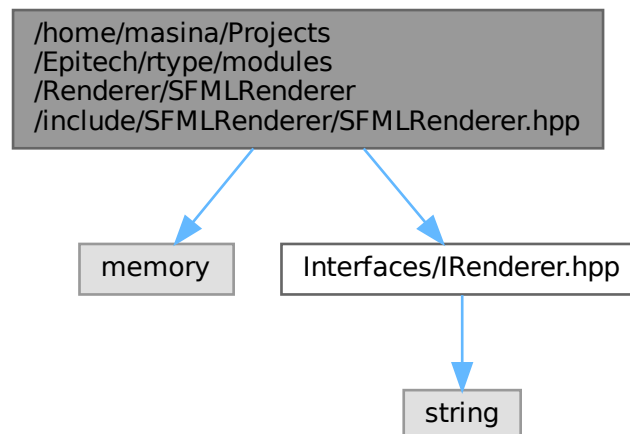
SFMLRenderer class declaration with PImpl.

```

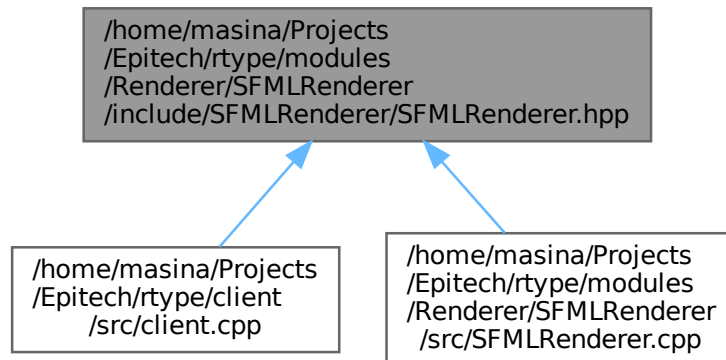
#include <memory>
#include "Interfaces/IRenderer.hpp"

```

Include dependency graph for SFMLRenderer.hpp:



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



Classes

- class [eng::SFMLRendererer](#)
Class for the R-Type game.

Namespaces

- namespace [eng](#)

8.79.1 Detailed Description

SFMLRendererer class declaration with PImpl.

Definition in file [SFMLRendererer.hpp](#).

8.80 SFMLRendererer.hpp

[Go to the documentation of this file.](#)

```

00001 ///  

00002 ///  

00003 ///  

00004 ///  

00005 ///  

00006 ///  

00007 #pragma once  

00008  

00009 #include <memory>  

00010  

00011 #include "Interfaces/IRenderer.hpp"  

00012  

00013 namespace eng  

00014 {  

00015  

00016     ///  

00017     ///  

00018     ///  


```

```

00019  /// @namespace eng
00020  ///
00021  class SFMLRenderer final : public IRenderer
00022  {
00023  public:
00024      SFMLRenderer();
00025      ~SFMLRenderer() override;
00026
00027      SFMLRenderer(const SFMLRenderer &) = delete;
00028      SFMLRenderer &operator=(const SFMLRenderer &) = delete;
00029      SFMLRenderer(SFMLRenderer &&) = delete;
00030      SFMLRenderer &operator=(SFMLRenderer &&) = delete;
00031
00032      void createWindow(const std::string &title, unsigned int height, unsigned int width,
00033                      unsigned int frameLimit, bool fullscreen) override;
00034      bool windowIsOpen() const override;
00035      void closeWindow() override;
00036      void clearWindow(Color color) override;
00037      void displayWindow() override;
00038      WindowSize getWindowSize() override;
00039
00040      bool pollEvent(Event &event) override;
00041      void setFrameLimit(unsigned int frameLimit) override;
00042
00043      void createFont(const std::string &name, const std::string &path) override;
00044      void createText(Text text) override;
00045      void setTextContent(const std::string &name, const std::string &content) override;
00046      void setTextPosition(const std::string &name, float x, float y) override;
00047      void setTextColor(const std::string &name, Color color) override;
00048
00049      void drawText(const std::string &name) override;
00050      void createTexture(const std::string &name, const std::string &path) override;
00051      void createSprite(const std::string &name, const std::string &textureName, float x, float y, float scale_x,
00052                      float scale_y, int fx, int fy, int fnx, int fny) override;
00053      void setSpritePosition(const std::string &name, float x, float y) override;
00054      void setSpriteTexture(const std::string &name, const std::string &path) override;
00055      void setSpriteFrame(const std::string &name, int fx, int fy, int fnx, int fny) override;
00056      void setSpriteScale(const std::string &name, int x, int y) override;
00057      void drawSprite(const std::string &name) override;
00058
00059      void drawPoint(float x, float y, Color color) override;
00060
00061  private:
00062      struct Impl;
00063      std::unique_ptr<Impl> m_impl;
00064  }; // class SFMLRenderer
00065
00066 } // namespace eng

```

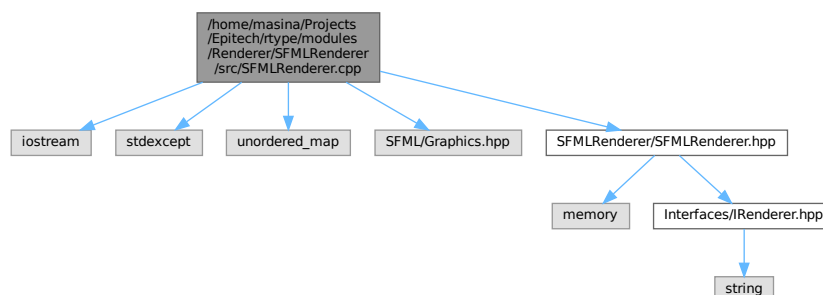
8.81 /home/masina/Projects/Epitech/rtype/modules/Renderer/↵ SFMLRenderer/src/SFMLRenderer.cpp File Reference

```

#include <iostream>
#include <stdexcept>
#include <unordered_map>
#include <SFML/Graphics.hpp>
#include "SFMLRenderer/SFMLRenderer.hpp"

```

Include dependency graph for SFMLRenderer.cpp:



Classes

- struct [eng::SFMLRenderrer::Impl](#)

Functions

- static [eng::Key](#) [scancodeToKey](#) (const sf::Keyboard::Scancode sc)

8.81.1 Function Documentation

8.81.1.1 scancodeToKey()

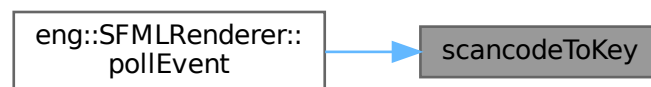
static [eng::Key](#) [scancodeToKey](#) (
 const sf::Keyboard::Scancode sc) [static]

Definition at line 113 of file [SFMLRenderrer.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::SFMLRenderrer::pollEvent\(\)](#).

Here is the caller graph for this function:



8.82 SFMLRenderrer.cpp

[Go to the documentation of this file.](#)

```

00001 #include <iostream>
00002 #include <stdexcept>
00003 #include <unordered_map>
00004
00005 #include <SFML/Graphics.hpp>
00006
00007 #include "SFMLRenderrer/SFMLRenderrer.hpp"
00008
00009 struct eng::SFMLRenderrer::Impl
00010 {
00011     std::unordered_map<std::string, sf::Texture> textures;
00012
00013     sf::RenderWindow window;
00014     std::unordered_map<std::string, sf::Font> fonts;
00015     std::unordered_map<std::string, sf::Text> texts;
00016     std::unordered_map<std::string, sf::Sprite> sprites;
00017 };
  
```

```

00018
00019 eng::SFMLRenderer::SFMLRenderer() : m_impl(std::make_unique<Impl>()) {}
00020
00021 eng::SFMLRenderer::~~SFMLRenderer() = default;
00022
00023 void eng::SFMLRenderer::createWindow(const std::string &title, unsigned int height, unsigned int width,
00024                                     const unsigned int frameLimit, const bool fullscreen)
00025 {
00026     const sf::VideoMode mode = fullscreen ? sf::VideoMode::getDesktopMode() : sf::VideoMode({width, height});
00027     m_impl->window.create(mode, title, fullscreen ? sf::State::Fullscreen : sf::State::Windowed);
00028     m_impl->window.setFramerateLimit(frameLimit);
00029 }
00030
00031 bool eng::SFMLRenderer::windowIsOpen() const { return m_impl->window.isOpen(); }
00032
00033 void eng::SFMLRenderer::closeWindow() { m_impl->window.close(); }
00034
00035 void eng::SFMLRenderer::setFrameLimit(const unsigned int frameLimit) {
00036     m_impl->window.setFramerateLimit(frameLimit); }
00037
00038 void eng::SFMLRenderer::createFont(const std::string &name, const std::string &path)
00039 {
00040     sf::Font sfFont;
00041     if (!sfFont.openFromFile(path))
00042     {
00043         throw std::runtime_error("Failed to load font: " + path);
00044     }
00045     m_impl->fonts.emplace(name, std::move(sfFont));
00046 }
00047
00048 void eng::SFMLRenderer::createText(Text text)
00049 {
00050     const auto &font = m_impl->fonts.at(text.font_name);
00051     sf::Text sfText(font);
00052     sfText.setString(text.content);
00053     sfText.setCharacterSize(text.size);
00054     sfText.setPosition({(text.x), (text.y)});
00055     sfText.setFillColor(sf::Color(text.color.r, text.color.g, text.color.b, text.color.a));
00056     m_impl->texts.emplace(text.name, std::move(sfText));
00057 }
00058
00059 void eng::SFMLRenderer::setTextContent(const std::string &name, const std::string &content)
00060 {
00061     if (const auto it = m_impl->texts.find(name); it != m_impl->texts.end())
00062     {
00063         it->second.setString(content);
00064     }
00065     else
00066     {
00067         throw std::runtime_error("Text not found: " + name);
00068     }
00069 }
00070
00071 void eng::SFMLRenderer::setTextPosition(const std::string &name, const float x, const float y)
00072 {
00073     if (const auto it = m_impl->texts.find(name); it != m_impl->texts.end())
00074     {
00075         it->second.setPosition({x, y});
00076     }
00077     else
00078     {
00079         throw std::runtime_error("Text not found: " + name);
00080     }
00081 }
00082
00083 void eng::SFMLRenderer::setTextColor(const std::string &name, const Color color)
00084 {
00085     if (const auto it = m_impl->texts.find(name); it != m_impl->texts.end())
00086     {
00087         it->second.setFillColor(sf::Color(color.r, color.g, color.b, color.a));
00088     }
00089     else
00090     {
00091         throw std::runtime_error("Text not found: " + name);
00092     }
00093 }
00094
00095 void eng::SFMLRenderer::drawText(const std::string &name)
00096 {
00097     if (const auto it = m_impl->texts.find(name); it != m_impl->texts.end())
00098     {
00099         m_impl->window.draw(it->second);
00100     }
00101     else
00102     {
00103         throw std::runtime_error("Text not found: " + name);
00104     }
00105 }

```

```

00104 }
00105
00106 void eng::SFMLRenderrer::clearWindow(const Color color)
00107 {
00108     m_impl->window.clear(sf::Color(color.r, color.g, color.b, color.a));
00109 }
00110
00111 void eng::SFMLRenderrer::displayWindow() { m_impl->window.display(); }
00112
00113 static eng::Key scancodeToKey(const sf::Keyboard::Scancode sc)
00114 {
00115     using S = sf::Keyboard::Scancode;
00116     switch (sc)
00117     {
00118         case S::Escape:
00119             return eng::Key::Escape;
00120         case S::Space:
00121             return eng::Key::Space;
00122         case S::Up:
00123             return eng::Key::Up;
00124         case S::Down:
00125             return eng::Key::Down;
00126         case S::Left:
00127             return eng::Key::Left;
00128         case S::Right:
00129             return eng::Key::Right;
00130         case S::A:
00131             return eng::Key::A;
00132         case S::B:
00133             return eng::Key::B;
00134         case S::C:
00135             return eng::Key::C;
00136         case S::D:
00137             return eng::Key::D;
00138         case S::E:
00139             return eng::Key::E;
00140         case S::F:
00141             return eng::Key::F;
00142         case S::G:
00143             return eng::Key::G;
00144         case S::H:
00145             return eng::Key::H;
00146         case S::I:
00147             return eng::Key::I;
00148         case S::J:
00149             return eng::Key::J;
00150         case S::K:
00151             return eng::Key::K;
00152         case S::L:
00153             return eng::Key::L;
00154         case S::M:
00155             return eng::Key::M;
00156         case S::N:
00157             return eng::Key::N;
00158         case S::O:
00159             return eng::Key::O;
00160         case S::P:
00161             return eng::Key::P;
00162         case S::Q:
00163             return eng::Key::Q;
00164         case S::R:
00165             return eng::Key::R;
00166         case S::S:
00167             return eng::Key::S;
00168         case S::T:
00169             return eng::Key::T;
00170         case S::U:
00171             return eng::Key::U;
00172         case S::V:
00173             return eng::Key::V;
00174         case S::W:
00175             return eng::Key::W;
00176         case S::X:
00177             return eng::Key::X;
00178         case S::Y:
00179             return eng::Key::Y;
00180         case S::Z:
00181             return eng::Key::Z;
00182         case S::Num0:
00183             return eng::Key::Num0;
00184         case S::Num1:
00185             return eng::Key::Num1;
00186         case S::Num2:
00187             return eng::Key::Num2;
00188         case S::Num3:
00189             return eng::Key::Num3;
00190         case S::Num4:

```

```

00191         return eng::Key::Num4;
00192     case S::Num5:
00193         return eng::Key::Num5;
00194     case S::Num6:
00195         return eng::Key::Num6;
00196     case S::Num7:
00197         return eng::Key::Num7;
00198     case S::Num8:
00199         return eng::Key::Num8;
00200     case S::Num9:
00201         return eng::Key::Num9;
00202     default:
00203         return eng::Key::Unknown;
00204     }
00205 }
00206
00207 bool eng::SFMLRenderer::pollEvent(Event &event)
00208 {
00209     if (const auto eventOpt = m_impl->window.pollEvent())
00210     {
00211         const auto &e = *eventOpt;
00212
00213         if (e.is<sf::Event::Closed>())
00214         {
00215             event.type = EventType::Closed;
00216             return true;
00217         }
00218
00219         if (const auto *const key = e.getIf<sf::Event::KeyPressed>())
00220         {
00221             event.type = EventType::KeyPressed;
00222             std::cout << "Key pressed: " << std::to_string(static_cast<int>(key->scancode)) << '\n';
00223             event.key = scancodeToKey(key->scancode);
00224             return true;
00225         }
00226
00227         if (const auto *const key = e.getIf<sf::Event::KeyReleased>())
00228         {
00229             event.type = EventType::KeyReleased;
00230             std::cout << "Key released: " << std::to_string(static_cast<int>(key->scancode)) << '\n';
00231             event.key = scancodeToKey(key->scancode);
00232             return true;
00233         }
00234
00235         event.type = EventType::None;
00236         return true;
00237     }
00238     return false;
00239 }
00240
00241 void eng::SFMLRenderer::createSprite(const std::string &name, const std::string &textureName, const float x,
00242                                     const float y, float scale_x, float scale_y, int fx, int fy, int fnx, int fny)
00243 {
00244     sf::Sprite sfSprite(m_impl->textures[textureName]);
00245     sfSprite.setPosition({x, y});
00246     sfSprite.setScale({scale_x, scale_y});
00247     if (fnx == -1)
00248     {
00249         fnx = static_cast<int>(m_impl->textures[textureName].getSize().x);
00250     }
00251     if (fny == -1)
00252     {
00253         fny = static_cast<int>(m_impl->textures[textureName].getSize().y);
00254     }
00255     sfSprite.setTextureRect(sf::IntRect({fx, fy}, {fnx, fny}));
00256     m_impl->sprites.emplace(name, std::move(sfSprite));
00257 }
00258
00259 void eng::SFMLRenderer::createTexture(const std::string &name, const std::string &path)
00260 {
00261     if (m_impl->textures.contains(name))
00262     {
00263         return;
00264     }
00265
00266     sf::Texture texture;
00267     if (!texture.loadFromFile(path))
00268     {
00269         throw std::runtime_error("Failed to load texture: " + path);
00270     }
00271     m_impl->textures[name] = std::move(texture);
00272 }
00273
00274 void eng::SFMLRenderer::drawSprite(const std::string &name)
00275 {
00276     if (const auto it = m_impl->sprites.find(name); it != m_impl->sprites.end())
00277 
```

```

00278     {
00279         m_impl->window.draw(it->second);
00280     }
00281     else
00282     {
00283         throw std::runtime_error("Sprite not found: " + name);
00284     }
00285 }
00286
00287 void eng::SFMLRenderer::setSpritePosition(const std::string &name, const float x, const float y)
00288 {
00289     if (const auto it = m_impl->sprites.find(name); it != m_impl->sprites.end())
00290     {
00291         it->second.setPosition({x, y});
00292     }
00293     else
00294     {
00295         throw std::runtime_error("Sprite not found: " + name);
00296     }
00297 }
00298
00299 void eng::SFMLRenderer::setSpriteTexture(const std::string &name, const std::string &path)
00300 {
00301     sf::Texture texture;
00302     if (!texture.loadFromFile(path))
00303     {
00304         throw std::runtime_error("Failed to load texture: " + path);
00305     }
00306     m_impl->textures[name] = std::move(texture);
00307
00308     if (const auto it = m_impl->sprites.find(name); it != m_impl->sprites.end())
00309     {
00310         it->second.setTexture(m_impl->textures[name]);
00311     }
00312     else
00313     {
00314         throw std::runtime_error("Sprite not found: " + name);
00315     }
00316 }
00317 }
00318
00319 void eng::SFMLRenderer::setSpriteFrame(const std::string &name, int fx, int fy, int fnx, int fny)
00320 {
00321     if (const auto it = m_impl->sprites.find(name); it != m_impl->sprites.end())
00322     {
00323         it->second.setTextureRect(sf::IntRect({fx, fy}, {fnx, fny}));
00324     }
00325     else
00326     {
00327         throw std::runtime_error("Sprite not found: " + name);
00328     }
00329 }
00330
00331 void eng::SFMLRenderer::setSpriteScale(const std::string &name, const int x, const int y)
00332 {
00333     if (const auto it = m_impl->sprites.find(name); it != m_impl->sprites.end())
00334     {
00335         it->second.setScale({static_cast<float>(x), static_cast<float>(y)});
00336     }
00337     else
00338     {
00339         throw std::runtime_error("Sprite not found: " + name);
00340     }
00341 }
00342
00343 void eng::SFMLRenderer::drawPoint(const float x, const float y, const Color color)
00344 {
00345     const sf::Vertex point(sf::Vector2f(x, y), sf::Color(color.r, color.g, color.b, color.a));
00346     m_impl->window.draw(&point, 1, sf::PrimitiveType::Points);
00347 }
00348
00349 eng::WindowSize eng::SFMLRenderer::getWindowSize()
00350 {
00351     const sf::Vector2u size = m_impl->window.getSize();
00352     return {.width = size.x, .height = size.y};
00353 }

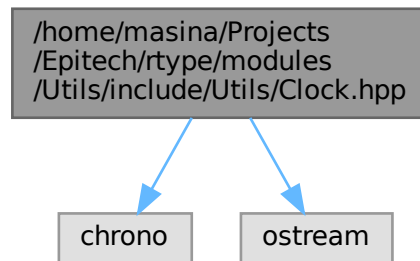
```

8.83 /home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/Clock.hpp File Reference ↵

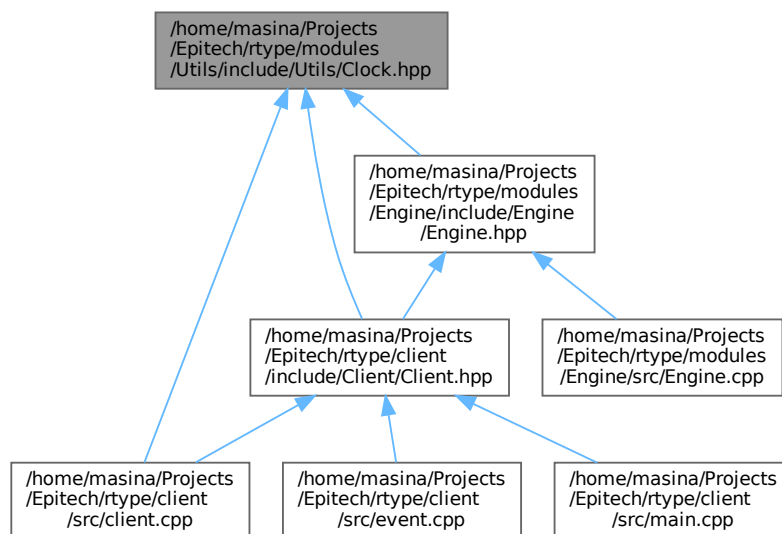
This file contains the Clock class.

```
#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](#)

8.83.1 Detailed Description

This file contains the Clock class.

Definition in file [Clock.hpp](#).

8.84 Clock.hpp

[Go to the documentation of this file.](#)

```

00001 ///  

00002 ///  

00003 ///  

00004 ///  

00005 ///  

00006 ///  

00007 #pragma once  

00008  

00009 #include <chrono>  

00010 #include <ostream>  

00011  

00012 namespace utl  

00013 {  

00014  

00015     ///  

00016     ///  

00017     ///  

00018     ///  

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

```

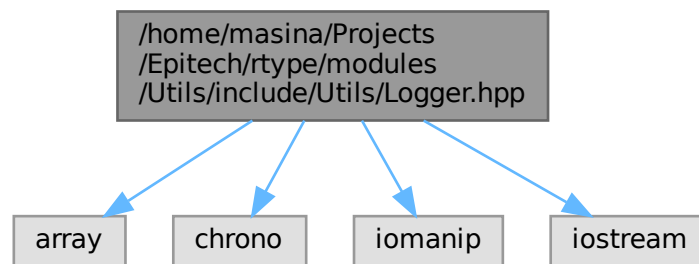
8.85 /home/masina/Projects/Epitech/rtype/modules/Utils/include/↵ Utils/Logger.hpp File Reference

```

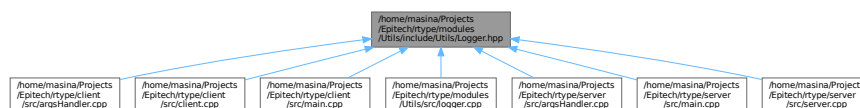
#include <array>
#include <chrono>
#include <iomanip>
#include <iostream>

```

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` }

8.86 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
00020     public:
00021         Logger(const Logger &) = delete;
00022         Logger &operator=(const Logger &) = delete;
00023         Logger(Logger &&) = delete;
00024         Logger &operator=(Logger &&) = delete;
00025
00026         static void init();
00027
00028         template <typename Func> static void logExecutionTime(const std::string &message, Func &&func)
00029         {
00030             const auto start = std::chrono::high_resolution_clock::now();
00031             func();
00032             const auto end = std::chrono::high_resolution_clock::now();
00033             const auto duration = std::chrono::duration<float, std::milli>(end - start).count();
00034
00035             std::cout << getColorForDuration(duration)
00036                     << formatLogMessage(LogLevel::INFO, message + " took " + std::to_string(duration) + " ms")
00037                     << LOG_LEVEL_COLOR[COLOR_RESET];
00038         }
00039
00040         static void log(const std::string &message, const LogLevel &logLevel)
00041         {
00042             std::cout << (logLevel == LogLevel::INFO ? LOG_LEVEL_COLOR[COLOR_INFO] :
LOG_LEVEL_COLOR[COLOR_WARNING])
00043                     << formatLogMessage(logLevel, 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] :
LOG_LEVEL_COLOR[COLOR_ERROR]);
00072     }
00073
00074     [[nodiscard]] static std::string formatLogMessage(LogLevel level, const std::string &message)
00075     {
00076         const auto inTime = std::chrono::system_clock::to_time_t(std::chrono::system_clock::now());
00077         std::ostringstream ss;
00078         ss << "[" << std::put_time(std::localtime(&inTime), "%Y-%m-%d %X") << "] ";
00079         ss << "[" << LOG_LEVEL_STRING[static_cast<uint8_t>(level)] << "] " << message;
00080         return ss.str();
00081     }
00082
00083 }; // class Logger
00084
00085 } // namespace utl

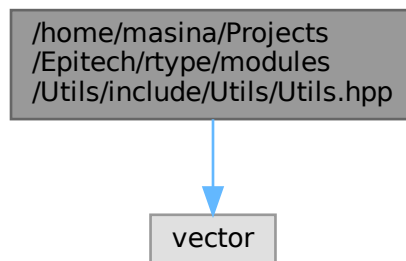
```

8.87 /home/masina/Projects/Epitech/rtype/modules/Utils/include/↵ Utils/Utils.hpp File Reference

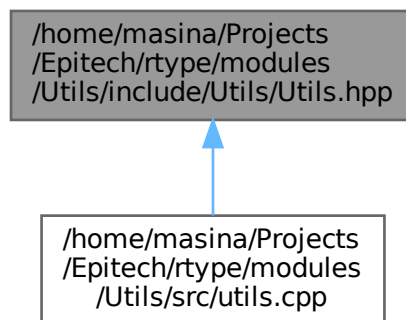
This file contains utility functions.

#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)`

8.87.1 Detailed Description

This file contains utility functions.

Definition in file [Utlis.hpp](#).

8.88 Utlis.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);  

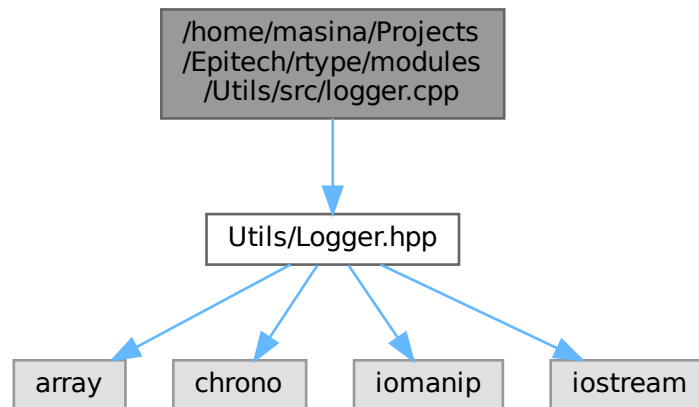
00015  

00016 } // namespace utl
  
```

8.89 /home/masina/Projects/Epitech/rtype/modules/↵ Utils/src/logger.cpp File Reference

```
#include "Utils/Logger.hpp"
```

Include dependency graph for logger.cpp:



8.90 logger.cpp

[Go to the documentation of this file.](#)

```

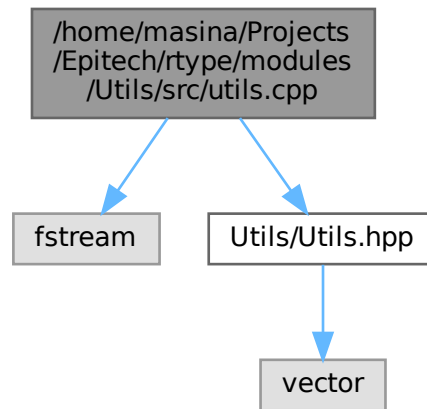
00001 #ifdef _WIN32
00002 #include <windows.h>
00003 #endif
00004
00005 #include "Utils/Logger.hpp"
00006
00007 void utl::Logger::init()
00008 {
00009     #ifdef _WIN32
00010         const HANDLE hOut = GetStdHandle(STD_OUTPUT_HANDLE);
00011         DWORD dwMode = 0;
00012         if (hOut != INVALID_HANDLE_VALUE && GetConsoleMode(hOut, &dwMode))
00013         {
00014             SetConsoleMode(hOut, dwMode | ENABLE_VIRTUAL_TERMINAL_PROCESSING);
00015         }
00016     #endif
00017 }
  
```

8.91 /home/masina/Projects/Epitech/rtype/modules/↵ Utils/src/utls.cpp File Reference

```
#include <fstream>
```

```
#include "Utils/Utils.hpp"
```

Include dependency graph for utils.cpp:



8.92 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 }

```

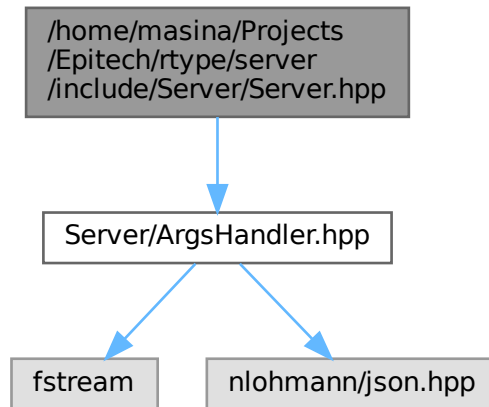
8.93 /home/masina/Projects/Epitech/rtype/README.md File Reference

8.94 /home/masina/Projects/Epitech/rtype/server/include/Server/↵ Server.hpp File Reference

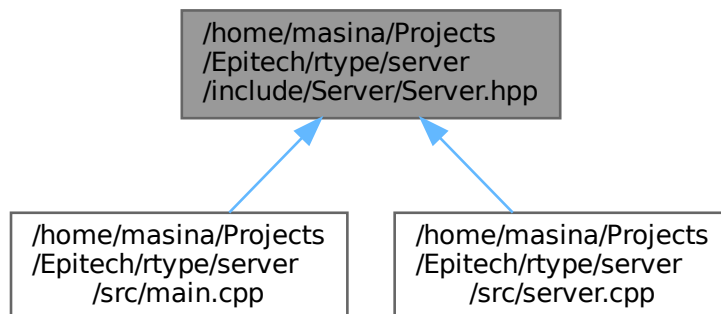
This file contains the Server class declaration.

```
#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](#)

8.94.1 Detailed Description

This file contains the Server class declaration.

Definition in file [Server.hpp](#).

8.95 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

```

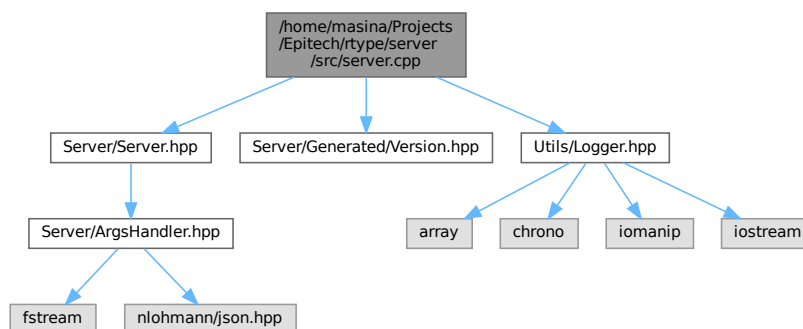
8.96 /home/masina/Projects/Epitech/rtype/server/src/server.cpp File Reference

```

#include "Server/Server.hpp"
#include "Server/Generated/Version.hpp"
#include "Utils/Logger.hpp"

```

Include dependency graph for server.cpp:



8.97 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](#), 293

[/home/masina/Projects/Epitech/rtype/client/include/Client/ArgsHandler.hpp](#), 205, 206

[/home/masina/Projects/Epitech/rtype/client/include/Client/Client.hpp](#), 209, 211

[/home/masina/Projects/Epitech/rtype/client/include/Client/TypeServer.hpp](#), 211, 213

[/home/masina/Projects/Epitech/rtype/client/include/Client/Version.hpp](#), 213, 215

[/home/masina/Projects/Epitech/rtype/client/include/Client/Scene/Project.hpp](#), 217, 218

[/home/masina/Projects/Epitech/rtype/client/include/Client/Scene/Project.hpp](#), 219, 220

[/home/masina/Projects/Epitech/rtype/client/src/argsHandler.cpp](#), 224, 225

[/home/masina/Projects/Epitech/rtype/client/src/client.cpp](#), 229, 230

[/home/masina/Projects/Epitech/rtype/client/src/event.cpp](#), 231

[/home/masina/Projects/Epitech/rtype/client/src/main.cpp](#), 231, 233

[/home/masina/Projects/Epitech/rtype/client/src/scenes/project.cpp](#), 235

[/home/masina/Projects/Epitech/rtype/modules/Audio/SFMLAudio/Project/SFMLAudio.hpp](#), 238, 239

[/home/masina/Projects/Epitech/rtype/modules/Audio/SFMLAudio/Project/SFMLAudio.cpp](#), 239, 240

[/home/masina/Projects/Epitech/rtype/modules/ECS/EntityManager/Project/EntityManager.hpp](#), 241, 243

[/home/masina/Projects/Epitech/rtype/modules/ECS/EntityManager/Project/EntityManager.cpp](#), 244, 245

[/home/masina/Projects/Epitech/rtype/modules/ECS/EntityManager/Project/EntityManager.hpp](#), 245, 247

[/home/masina/Projects/Epitech/rtype/modules/ECS/EntityManager/Project/EntityManager.cpp](#), 249

[/home/masina/Projects/Epitech/rtype/modules/Engine/Engine/Project/Engine.hpp](#), 249, 250

[/home/masina/Projects/Epitech/rtype/modules/Engine/Engine/Project/Engine.cpp](#), 251, 253

[/home/masina/Projects/Epitech/rtype/modules/Engine/Engine/Project/Server/EventManager.hpp](#), 254, 255

[/home/masina/Projects/Epitech/rtype/modules/Engine/Engine/Project/Server/EventManager.cpp](#), 222, 223

[/home/masina/Projects/Epitech/rtype/modules/Engine/Engine/Project/Server/EventManager.hpp](#), 256, 257

[/home/masina/Projects/Epitech/rtype/modules/Game/Renderer/Project/Renderer.hpp](#), 257, 258

[/home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Client/include/R-TypeClient/RTypeClient.hpp](#), 259, 260

[/home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Client/src/rtypeClient.cpp](#), 261

[/home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Server/include/R-TypeServer/RTypeServer.hpp](#), 262, 263

[/home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Server/src/rtypeServer.cpp](#), 263

[/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interface.hpp](#), 263, 265

[/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interface.hpp](#), 265, 266

[/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interface.hpp](#), 267, 268

[/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interface.hpp](#), 269, 270

[/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interface.hpp](#), 271, 272

[/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interface.hpp](#), 272, 273

[/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interface.hpp](#), 274

[/home/masina/Projects/Epitech/rtype/modules/Renderer/SFMLRender/Project/SFMLRender.hpp](#), 275, 276

[/home/masina/Projects/Epitech/rtype/modules/Renderer/SFMLRender/Project/SFMLRender.cpp](#), 278, 279

[/home/masina/Projects/Epitech/rtype/modules/Renderer/SFMLRender/Project/SFMLRender.hpp](#), 280, 281

[/home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils.hpp](#), 285, 287

[/home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils.hpp](#), 288, 289

[/home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils.hpp](#), 290, 291

[/home/masina/Projects/Epitech/rtype/modules/Utils/src/logger.cpp](#), 292

[/home/masina/Projects/Epitech/rtype/modules/Utils/src/Utils.cpp](#), 292, 293

[/home/masina/Projects/Epitech/rtype/server/include/Server/ArgsHandler.hpp](#), 207, 209

[/home/masina/Projects/Epitech/rtype/server/include/Server/EventManager.hpp](#), 215, 217

[/home/masina/Projects/Epitech/rtype/server/include/Server/EventManager.cpp](#), 293, 295

[/home/masina/Projects/Epitech/rtype/server/src/argsHandler.cpp](#), 227, 228

/home/masina/Projects/Epitech/rtype/server/src/main.cpp
 233, 234
 /home/masina/Projects/Epitech/rtype/server/src/server.cpp
 295, 296
 ~AGameClient
 gme::AGameClient, 24
 ~AGameServer
 gme::AGameServer, 27
 ~AScene
 eng::AScene, 40
 ~ArgsHandler
 cli::ArgsHandler, 33
 srv::ArgsHandler, 36
 ~AudioSystem
 cli::AudioSystem, 50
 ~Client
 cli::Client, 53
 ~Clock
 utl::Clock, 58
 ~Engine
 eng::Engine, 67
 ~IAudio
 eng::IAudio, 80
 ~IGameClient
 gme::IGameClient, 86
 ~IGameServer
 gme::IGameServer, 89
 ~INetworkClient
 eng::INetworkClient, 93
 ~INetworkServer
 srv::INetworkServer, 94
 ~IPool
 ecs::Registry::IPool, 95
 ~IRenderer
 eng::IRenderer, 98
 ~IScene
 eng::IScene, 106
 gme::IScene, 109
 ~ISystem
 eng::ISystem, 111
 ~Lobby
 cli::Lobby, 114
 ~Logger
 utl::Logger, 123
 ~PixelSystem
 cli::PixelSystem, 133
 ~RTypeClient
 gme::RTypeClient, 152
 ~RTypeServer
 gme::RTypeServer, 156
 ~Registry
 ecs::Registry, 144
 ~SFMLAudio
 eng::SFMLAudio, 167
 ~SFMLRenderer
 eng::SFMLRenderer, 172
 ~SceneManager
 eng::SceneManager, 161
 main.cpp
 srv::Server, 164
 SpriteSystem
 cli::SpriteSystem, 185
 ~TextSyStem
 cli::TextSyStem, 194
 A
 eng, 17
 a
 ecs::Color, 64
 eng::Color, 65
 gme::Sprite, 180
 add
 ecs::Registry::Pool< T >, 139
 addComponent
 ecs::Registry, 145
 addEntity
 gme::LobbyScene, 119
 addScene
 eng::SceneManager, 161
 addSystem
 eng::Engine, 68
 APP_EXTENSION
 argsHandler.cpp, 224, 227
 ArgsHandler
 cli::ArgsHandler, 33
 srv::ArgsHandler, 36
 argsHandler.cpp
 APP_EXTENSION, 224, 227
 HELP_MESSAGE, 225, 227
 VERSION_MESSAGE, 225, 227
 AScene
 eng::AScene, 40
 AUDIO_BATTLE_THEME
 cli::Path::Audio, 13
 AUDIO_COIN
 cli::Path::Audio, 13
 AUDIO_TITLE
 cli::Path::Audio, 13
 AudioSystem
 cli::AudioSystem, 50
 B
 eng, 17
 b
 ecs::Color, 64
 eng::Color, 65
 gme::Sprite, 180
 build
 ecs::Registry::EntityBuilder, 74
 BUILD_TYPE
 Version.hpp, 214, 216
 C
 eng, 17
 clearWindow
 eng::IRenderer, 98
 eng::SFMLRenderer, 173

- cli, [11](#)
 - json, [11](#)
- cli::ArgsConfig, [28](#)
 - exit, [29](#)
 - frameLimit, [29](#)
 - fromFile, [28](#)
 - fullscreen, [29](#)
 - height, [29](#)
 - width, [29](#)
- cli::ArgsHandler, [32](#)
 - ~ArgsHandler, [33](#)
 - ArgsHandler, [33](#)
 - operator=, [33](#)
 - ParseArgs, [33](#)
 - ParseEnv, [34](#)
- cli::AudioSystem, [47](#)
 - ~AudioSystem, [50](#)
 - AudioSystem, [50](#)
 - m_audio, [51](#)
 - operator=, [51](#)
 - update, [51](#)
- cli::Client, [52](#)
 - ~Client, [53](#)
 - Client, [53](#), [54](#)
 - handleEvents, [54](#)
 - lobbyScene, [54](#)
 - m_engine, [55](#)
 - m_game, [55](#)
 - m_keysPressed, [55](#)
 - operator=, [54](#), [55](#)
 - run, [55](#)
- cli::Config, [12](#)
- cli::Config::Audio, [12](#)
 - DEFAULT_AUDIO_MUTED, [12](#)
 - DEFAULT_AUDIO_VOLUME, [12](#)
- cli::Config::Window, [12](#)
 - DEFAULT_WINDOW_FRAME_LIMIT, [12](#)
 - DEFAULT_WINDOW_FULLSCREEN, [12](#)
 - DEFAULT_WINDOW_HEIGHT, [13](#)
 - DEFAULT_WINDOW_WIDTH, [13](#)
- cli::EnvConfig, [75](#)
- cli::Lobby, [112](#)
 - ~Lobby, [114](#)
 - event, [115](#)
 - Lobby, [114](#), [115](#)
 - m_fpsEntity, [116](#)
 - m_keysPressed, [116](#)
 - m_playerEntity, [116](#)
 - operator=, [115](#)
 - update, [115](#)
- cli::Path, [13](#)
- cli::Path::Audio, [13](#)
 - AUDIO_BATTLE_THEME, [13](#)
 - AUDIO_COIN, [13](#)
 - AUDIO_TITLE, [13](#)
- cli::Path::Font, [14](#)
 - FONTS_RTYPE, [14](#)
- cli::Path::Texture, [14](#)
 - TEXTURE_PLAYER, [14](#)
- cli::PixelSystem, [131](#)
 - ~PixelSystem, [133](#)
 - m_renderer, [134](#)
 - operator=, [133](#)
 - PixelSystem, [133](#)
 - update, [134](#)
- cli::SpriteSystem, [182](#)
 - ~SpriteSystem, [185](#)
 - m_renderer, [186](#)
 - operator=, [185](#)
 - SpriteSystem, [185](#)
 - update, [186](#)
- cli::TextSyStem, [191](#)
 - ~TextSyStem, [194](#)
 - m_renderer, [195](#)
 - operator=, [195](#)
 - TextSyStem, [194](#)
 - update, [195](#)
- Client
 - cli::Client, [53](#), [54](#)
- client.cpp
 - DARK, [230](#)
- Clock
 - utl::Clock, [58](#)
- Closed
 - eng, [17](#)
- closeWindow
 - eng::IRenderer, [98](#)
 - eng::SFMLRenderer, [173](#)
- color
 - eng::Text, [190](#)
- COLOR_ERROR
 - utl::Logger, [122](#)
- COLOR_INFO
 - utl::Logger, [122](#)
- COLOR_RESET
 - utl::Logger, [122](#)
- COLOR_WARNING
 - utl::Logger, [122](#)
- ColorIndex
 - utl::Logger, [122](#)
- content
 - ecs::Text, [189](#)
 - eng::Text, [190](#)
- createAudio
 - eng::IAudio, [80](#)
 - eng::SFMLAudio, [168](#)
- createEntity
 - ecs::Registry, [145](#)
- createFont
 - eng::IRenderer, [98](#)
 - eng::SFMLRenderer, [173](#)
- createSprite
 - eng::IRenderer, [98](#)
 - eng::SFMLRenderer, [173](#)
- createText
 - eng::IRenderer, [99](#)

- eng::SFMLRenderer, 174
- createTexture
 - eng::IRenderer, 99
 - eng::SFMLRenderer, 174
- createWindow
 - eng::IRenderer, 99
 - eng::SFMLRenderer, 174
- D
 - eng, 17
- DARK
 - client.cpp, 230
- data
 - ecs::Registry::Pool< T >, 140
- DEFAULT
 - eng, 18
- DEFAULT_AUDIO_MUTED
 - cli::Config::Audio, 12
- DEFAULT_AUDIO_VOLUME
 - cli::Config::Audio, 12
- DEFAULT_WINDOW_FRAME_LIMIT
 - cli::Config::Window, 12
- DEFAULT_WINDOW_FULLSCREEN
 - cli::Config::Window, 12
- DEFAULT_WINDOW_HEIGHT
 - cli::Config::Window, 13
- DEFAULT_WINDOW_WIDTH
 - cli::Config::Window, 13
- displayWindow
 - eng::IRenderer, 99
 - eng::SFMLRenderer, 174
- Down
 - eng, 17
- drawPoint
 - eng::IRenderer, 99
 - eng::SFMLRenderer, 175
- drawSprite
 - eng::IRenderer, 100
 - eng::SFMLRenderer, 175
- drawText
 - eng::IRenderer, 100
 - eng::SFMLRenderer, 175
- Duration
 - utl::Clock, 57
- E
 - eng, 17
- ecs, 15
 - Entity, 15
 - INVALID_ENTITY, 15
- ecs::Audio, 45
 - loop, 47
 - path, 47
 - play, 47
 - volume, 47
- ecs::Color, 62
 - a, 64
 - b, 64
 - g, 64
 - r, 64
- ecs::Font, 77
 - path, 79
- ecs::IComponent, 83
 - id, 84
- ecs::Mob, 127
- ecs::Pixel, 129
- ecs::Player, 135
 - is_cli, 137
- ecs::Rect, 140
 - pos_x, 142
 - pos_y, 142
 - size_x, 142
 - size_y, 142
- ecs::Registry, 142
 - ~Registry, 144
 - addComponent, 145
 - createEntity, 145
 - getAll, 145
 - getComponent, 146
 - getPool, 147
 - hasComponent, 147
 - m_components, 149
 - m_entities, 149
 - m_lastEntity, 149
 - m_onComponentAddedCallbacks, 149
 - onComponentAdded, 148
 - operator=, 148
 - Registry, 144
 - removeComponent, 148
- ecs::Registry::EntityBuilder, 72
 - build, 74
 - EntityBuilder, 74
 - m_entity, 74
 - m_registry, 74
 - with, 74
- ecs::Registry::IPool, 94
 - ~IPool, 95
 - remove, 95
- ecs::Registry::Pool< T >, 137
 - add, 139
 - data, 140
 - get, 139
 - has, 139
 - remove, 139
- ecs::Scale, 157
 - x, 159
 - y, 159
- ecs::Text, 187
 - content, 189
 - font_size, 189
- ecs::Texture, 196
 - path, 198
- ecs::Transform, 198
 - rotation, 200
 - x, 200
 - y, 200
- ecs::Velocity, 200

- x, [202](#)
- y, [202](#)
- eng, [16](#)
 - A, [17](#)
 - B, [17](#)
 - C, [17](#)
 - Closed, [17](#)
 - D, [17](#)
 - DEFAULT, [18](#)
 - Down, [17](#)
 - E, [17](#)
 - Escape, [17](#)
 - EventType, [17](#)
 - F, [17](#)
 - G, [17](#)
 - H, [17](#)
 - I, [17](#)
 - id, [17](#)
 - J, [17](#)
 - K, [17](#)
 - Key, [17](#)
 - KeyPressed, [17](#)
 - KeyReleased, [17](#)
 - L, [17](#)
 - Left, [17](#)
 - M, [17](#)
 - N, [18](#)
 - None, [17](#)
 - Num0, [18](#)
 - Num1, [18](#)
 - Num2, [18](#)
 - Num3, [18](#)
 - Num4, [18](#)
 - Num5, [18](#)
 - Num6, [18](#)
 - Num7, [18](#)
 - Num8, [18](#)
 - Num9, [18](#)
 - O, [18](#)
 - P, [18](#)
 - Paused, [18](#)
 - Playing, [18](#)
 - Q, [18](#)
 - R, [18](#)
 - Right, [17](#)
 - RUN, [18](#)
 - S, [18](#)
 - Space, [17](#)
 - State, [18](#)
 - Status, [18](#)
 - STOP, [18](#)
 - Stopped, [18](#)
 - T, [18](#)
 - U, [18](#)
 - Unknown, [17](#)
 - Up, [17](#)
 - V, [18](#)
 - W, [18](#)
 - X, [18](#)
 - Y, [18](#)
 - Z, [18](#)
- eng::AScene, [38](#)
 - ~AScene, [40](#)
 - AScene, [40](#)
 - getId, [41](#)
 - getName, [41](#)
 - getRegistry, [41](#)
 - m_id, [42](#)
 - m_name, [42](#)
 - m_registry, [42](#)
 - operator=, [41](#)
 - s_nextId, [42](#)
 - setName, [41](#)
- eng::ASystem, [43](#)
 - isEnabled, [44](#)
 - m_isEnabled, [44](#)
 - setEnabled, [44](#)
- eng::Color, [64](#)
 - a, [65](#)
 - b, [65](#)
 - g, [65](#)
 - r, [65](#)
- eng::Engine, [66](#)
 - ~Engine, [67](#)
 - addSystem, [68](#)
 - Engine, [67](#), [68](#)
 - getAudio, [68](#)
 - getClock, [68](#)
 - getNetwork, [69](#)
 - getRenderer, [69](#)
 - getSceneManager, [69](#)
 - getState, [69](#)
 - m_audio, [71](#)
 - m_clock, [71](#)
 - m_network, [71](#)
 - m_renderer, [71](#)
 - m_sceneManager, [71](#)
 - m_state, [72](#)
 - m_systems, [72](#)
 - operator=, [70](#)
 - render, [70](#)
 - setState, [70](#)
 - stop, [70](#)
 - updateSystems, [71](#)
- eng::Event, [76](#)
 - key, [76](#)
 - type, [76](#)
- eng::IAudio, [79](#)
 - ~IAudio, [80](#)
 - createAudio, [80](#)
 - isPlaying, [80](#)
 - playAudio, [81](#)
 - setLoop, [81](#)
 - setVolume, [82](#)
 - stopAudio, [82](#)
- eng::INetworkClient, [92](#)

- ~INetworkClient, 93
- eng::IRenderer, 95
 - ~IRenderer, 98
 - clearWindow, 98
 - closeWindow, 98
 - createFont, 98
 - createSprite, 98
 - createText, 99
 - createTexture, 99
 - createWindow, 99
 - displayWindow, 99
 - drawPoint, 99
 - drawSprite, 100
 - drawText, 100
 - getWindowSize, 101
 - pollEvent, 101
 - setFrameLimit, 101
 - setSpriteFrame, 101
 - setSpritePosition, 101
 - setSpriteScale, 102
 - setSpriteTexture, 102
 - setTextColor, 102
 - setTextContent, 103
 - setTextPosition, 103
 - windowIsOpen, 104
- eng::IScene, 104
 - ~IScene, 106
 - event, 106
 - getId, 106
 - getName, 107
 - getRegistry, 107
 - setName, 107
 - update, 107
- eng::ISystem, 110
 - ~ISystem, 111
 - isEnabled, 111
 - setEnabled, 111
 - update, 111
- eng::SceneManager, 159
 - ~SceneManager, 161
 - addScene, 161
 - getCurrentScene, 161
 - getScene, 162
 - m_currentSceneId, 162
 - m_scenes, 162
 - operator=, 162
 - SceneManager, 161
 - switchToScene, 162
- eng::SFMLAudio, 165
 - ~SFMLAudio, 167
 - createAudio, 168
 - isPlaying, 168
 - operator=, 168
 - pImpl, 169
 - playAudio, 168
 - setLoop, 168
 - setVolume, 169
 - SFMLAudio, 167
 - stopAudio, 169
- eng::SFMLAudio::Impl, 90
 - musics, 91
- eng::SFMLRenderer, 170
 - ~SFMLRenderer, 172
 - clearWindow, 173
 - closeWindow, 173
 - createFont, 173
 - createSprite, 173
 - createText, 174
 - createTexture, 174
 - createWindow, 174
 - displayWindow, 174
 - drawPoint, 175
 - drawSprite, 175
 - drawText, 175
 - getWindowSize, 175
 - m_impl, 178
 - operator=, 175, 176
 - pollEvent, 176
 - setFrameLimit, 176
 - setSpriteFrame, 176
 - setSpritePosition, 176
 - setSpriteScale, 177
 - setSpriteTexture, 177
 - setTextColor, 177
 - setTextContent, 177
 - setTextPosition, 177
 - SFMLRenderer, 172, 173
 - windowIsOpen, 178
- eng::SFMLRenderer::Impl, 91
 - fonts, 92
 - sprites, 92
 - texts, 92
 - textures, 92
 - window, 92
- eng::Text, 189
 - color, 190
 - content, 190
 - font_name, 190
 - name, 190
 - size, 190
 - x, 191
 - y, 191
- eng::WindowSize, 202
 - height, 203
 - width, 203
- Engine
 - eng::Engine, 67, 68
- Entity
 - ecs, 15
- EntityBuilder
 - ecs::Registry::EntityBuilder, 74
- Escape
 - eng, 17
- event
 - cli::Lobby, 115
 - eng::IScene, 106

EventType
 eng, [17](#)
 exit
 cli::ArgsConfig, [29](#)
 srv::ArgsConfig, [31](#)

 F
 eng, [17](#)
 font_name
 eng::Text, [190](#)
 font_size
 ecs::Text, [189](#)
 fonts
 eng::SFMLRenderer::Impl, [92](#)
 FONTS_RTYPE
 cli::Path::Font, [14](#)
 formatLogMessage
 utl::Logger, [123](#)
 frameLimit
 cli::ArgsConfig, [29](#)
 fromFile
 cli::ArgsConfig, [28](#)
 srv::ArgsConfig, [31](#)
 fullscreen
 cli::ArgsConfig, [29](#)

 G
 eng, [17](#)

 g
 ecs::Color, [64](#)
 eng::Color, [65](#)
 gme::Sprite, [180](#)

 get
 ecs::Registry::Pool< T >, [139](#)
 getAll
 ecs::Registry, [145](#)
 getAudio
 eng::Engine, [68](#)
 getClock
 eng::Engine, [68](#)
 getColorForDuration
 utl::Logger, [123](#)
 getComponent
 ecs::Registry, [146](#)
 getCurrentScene
 eng::SceneManager, [161](#)
 gme::IGameClient, [86](#)
 gme::RTypeClient, [153](#)
 getDeltaSeconds
 utl::Clock, [58](#)
 getElapsed
 utl::Clock, [58](#)
 getEntities
 gme::IScene, [109](#)
 gme::LobbyScene, [119](#)
 getEntitiesMutable
 gme::IScene, [109](#)
 gme::LobbyScene, [119](#)
 getId
 eng::AScene, [41](#)
 eng::IScene, [106](#)
 getName
 eng::AScene, [41](#)
 eng::IScene, [107](#)
 gme::AGameClient, [24](#)
 gme::AGameServer, [27](#)
 gme::IGameClient, [86](#)
 gme::IGameServer, [89](#)
 gme::IScene, [109](#)
 gme::LobbyScene, [120](#)
 getNetwork
 eng::Engine, [69](#)
 getPool
 ecs::Registry, [147](#)
 getRegistry
 eng::AScene, [41](#)
 eng::IScene, [107](#)
 getRenderer
 eng::Engine, [69](#)
 getScene
 eng::SceneManager, [162](#)
 getSceneManager
 eng::Engine, [69](#)
 getState
 eng::Engine, [69](#)
 getWindowSize
 eng::IRenderer, [101](#)
 eng::SFMLRenderer, [175](#)
 GIT_COMMIT_HASH
 Version.hpp, [214](#), [216](#)
 GIT_TAG
 Version.hpp, [214](#), [216](#)
 gme, [19](#)
 gme::AGameClient, [21](#)
 ~AGameClient, [24](#)
 getName, [24](#)
 m_name, [25](#)
 setName, [24](#)
 gme::AGameServer, [25](#)
 ~AGameServer, [27](#)
 getName, [27](#)
 m_name, [27](#)
 setName, [27](#)
 gme::IGameClient, [84](#)
 ~IGameClient, [86](#)
 getCurrentScene, [86](#)
 getName, [86](#)
 setName, [87](#)
 update, [87](#)
 gme::IGameServer, [87](#)
 ~IGameServer, [89](#)
 getName, [89](#)
 setName, [89](#)
 gme::IScene, [108](#)
 ~IScene, [109](#)
 getEntities, [109](#)
 getEntitiesMutable, [109](#)

- getName, [109](#)
- gme::LobbyScene, [116](#)
 - addEntity, [119](#)
 - getEntities, [119](#)
 - getEntitiesMutable, [119](#)
 - getName, [120](#)
 - LobbyScene, [119](#)
 - m_entities, [120](#)
 - m_name, [120](#)
- gme::RTypeClient, [150](#)
 - ~RTypeClient, [152](#)
 - getCurrentScene, [153](#)
 - m_currentScene, [153](#)
 - operator=, [153](#)
 - RTypeClient, [152](#)
 - update, [153](#)
- gme::RTypeServer, [154](#)
 - ~RTypeServer, [156](#)
 - operator=, [156](#)
 - RTypeServer, [156](#)
- gme::Sprite, [178](#)
 - a, [180](#)
 - b, [180](#)
 - g, [180](#)
 - id, [180](#)
 - pos_x, [180](#)
 - pos_y, [180](#)
 - r, [181](#)
 - scale_x, [181](#)
 - scale_y, [181](#)
 - text_rect_fx, [181](#)
 - text_rect_fy, [181](#)
 - text_rect_x, [181](#)
 - text_rect_y, [181](#)
 - texture_path, [181](#)
 - type, [182](#)
 - v_x, [182](#)
 - v_y, [182](#)
- H
 - eng, [17](#)
- handleEvents
 - cli::Client, [54](#)
- has
 - ecs::Registry::Pool< T >, [139](#)
- hasComponent
 - ecs::Registry, [147](#)
- height
 - cli::ArgsConfig, [29](#)
 - eng::WindowSize, [203](#)
- HELP_MESSAGE
 - argsHandler.cpp, [225](#), [227](#)
- host
 - srv::ArgsConfig, [31](#)
- I
 - eng, [17](#)
- id
 - ecs::IComponent, [84](#)
 - eng, [17](#)
 - gme::Sprite, [180](#)
- INFO
 - utl, [20](#)
- init
 - utl::Logger, [124](#)
- INVALID_ENTITY
 - ecs, [15](#)
- is_cli
 - ecs::Player, [137](#)
- isEnabled
 - eng::ASystem, [44](#)
 - eng::ISystem, [111](#)
- isPlaying
 - eng::IAudio, [80](#)
 - eng::SFMLAudio, [168](#)
- J
 - eng, [17](#)
- json
 - cli, [11](#)
 - srv, [19](#)
- K
 - eng, [17](#)
- Key
 - eng, [17](#)
- key
 - eng::Event, [76](#)
- KeyPressed
 - eng, [17](#)
- KeyReleased
 - eng, [17](#)
- L
 - eng, [17](#)
- Left
 - eng, [17](#)
- Lobby
 - cli::Lobby, [114](#), [115](#)
- lobby.cpp
 - WHITE, [235](#)
- LobbyScene
 - gme::LobbyScene, [119](#)
- lobbyScene
 - cli::Client, [54](#)
- log
 - utl::Logger, [124](#)
- LOG_LEVEL_COLOR
 - utl::Logger, [126](#)
- LOG_LEVEL_STRING
 - utl::Logger, [126](#)
- logExecutionTime
 - utl::Logger, [125](#)
- Logger
 - utl::Logger, [122](#), [123](#)
- LogLevel
 - utl, [20](#)
- loop

- ecs::Audio, 47
- M
 - eng, 17
- m_audio
 - cli::AudioSystem, 51
 - eng::Engine, 71
- m_clock
 - eng::Engine, 71
- m_components
 - ecs::Registry, 149
- m_currentScene
 - gme::RTypeClient, 153
- m_currentSceneId
 - eng::SceneManager, 162
- m_engine
 - cli::Client, 55
- m_entities
 - ecs::Registry, 149
 - gme::LobbyScene, 120
- m_entity
 - ecs::Registry::EntityBuilder, 74
- m_fpsEntity
 - cli::Lobby, 116
- m_game
 - cli::Client, 55
- m_id
 - eng::AScene, 42
- m_impl
 - eng::SFMLRenderer, 178
- m_isEnable
 - eng::ASystem, 44
- m_isPaused
 - utl::Clock, 61
- m_keysPressed
 - cli::Client, 55
 - cli::Lobby, 116
- m_lastEntity
 - ecs::Registry, 149
- m_name
 - eng::AScene, 42
 - gme::AGameClient, 25
 - gme::AGameServer, 27
 - gme::LobbyScene, 120
- m_network
 - eng::Engine, 71
- m_onComponentAddedCallbacks
 - ecs::Registry, 149
- m_pausedDuration
 - utl::Clock, 61
- m_pausedTime
 - utl::Clock, 61
- m_playerEntity
 - cli::Lobby, 116
- m_registry
 - ecs::Registry::EntityBuilder, 74
 - eng::AScene, 42
- m_renderer
 - cli::PixelSystem, 134
 - cli::SpriteSystem, 186
 - cli::TextSyStem, 195
 - eng::Engine, 71
- m_sceneManager
 - eng::Engine, 71
- m_scenes
 - eng::SceneManager, 162
- m_start
 - utl::Clock, 62
- m_state
 - eng::Engine, 72
- m_systems
 - eng::Engine, 72
- main
 - main.cpp, 232, 234
- main.cpp
 - main, 232, 234
- musics
 - eng::SFMLAudio::Impl, 91
- N
 - eng, 18
- name
 - eng::Text, 190
- None
 - eng, 17
- now
 - utl::Clock, 59
- Num0
 - eng, 18
- Num1
 - eng, 18
- Num2
 - eng, 18
- Num3
 - eng, 18
- Num4
 - eng, 18
- Num5
 - eng, 18
- Num6
 - eng, 18
- Num7
 - eng, 18
- Num8
 - eng, 18
- Num9
 - eng, 18
- O
 - eng, 18
- onComponentAdded
 - ecs::Registry, 148
- operator<<
 - utl::Clock, 61
- operator=
 - cli::ArgsHandler, 33
 - cli::AudioSystem, 51
 - cli::Client, 54, 55

- cli::Lobby, 115
 - cli::PixelSystem, 133
 - cli::SpriteSystem, 185
 - cli::TextSyStem, 195
 - ecs::Registry, 148
 - eng::AScene, 41
 - eng::Engine, 70
 - eng::SceneManager, 162
 - eng::SFMLAudio, 168
 - eng::SFMLRenderer, 175, 176
 - gme::RTypeClient, 153
 - gme::RTypeServer, 156
 - srv::ArgsHandler, 36
 - srv::Server, 164
 - utl::Clock, 59, 60
 - utl::Logger, 126
- P
- eng, 18
- ParseArgs
- cli::ArgsHandler, 33
 - srv::ArgsHandler, 36
- ParseEnv
- cli::ArgsHandler, 34
 - srv::ArgsHandler, 37
- path
- ecs::Audio, 47
 - ecs::Font, 79
 - ecs::Texture, 198
- pause
- utl::Clock, 60
- Paused
- eng, 18
- pImpl
- eng::SFMLAudio, 169
- PixelSystem
- cli::PixelSystem, 133
- play
- ecs::Audio, 47
- playAudio
- eng::IAudio, 81
 - eng::SFMLAudio, 168
- Playing
- eng, 18
- pollEvent
- eng::IRenderer, 101
 - eng::SFMLRenderer, 176
- port
- srv::ArgsConfig, 31
- pos_x
- ecs::Rect, 142
 - gme::Sprite, 180
- pos_y
- ecs::Rect, 142
 - gme::Sprite, 180
- PROJECT_NAME
- Version.hpp, 214, 216
- PROJECT_VERSION
- Version.hpp, 214, 216
- PROJECT_VERSION_MAJOR
- Version.hpp, 214, 216
- PROJECT_VERSION_MINOR
- Version.hpp, 215, 216
- PROJECT_VERSION_PATCH
- Version.hpp, 215, 217
- Q
- eng, 18
- R
- eng, 18
- r
- ecs::Color, 64
 - eng::Color, 65
 - gme::Sprite, 181
- r-type, 1
- readFile
- utl, 20
- Registry
- ecs::Registry, 144
- remove
- ecs::Registry::IPool, 95
 - ecs::Registry::Pool< T >, 139
- removeComponent
- ecs::Registry, 148
- render
- eng::Engine, 70
- restart
- utl::Clock, 60
- resume
- utl::Clock, 60
- Right
- eng, 17
- rotation
- ecs::Transform, 200
- RTypeClient
- gme::RTypeClient, 152
- RTypeServer
- gme::RTypeServer, 156
- RUN
- eng, 18
- run
- cli::Client, 55
- S
- eng, 18
- s_nextId
- eng::AScene, 42
- scale_x
- gme::Sprite, 181
- scale_y
- gme::Sprite, 181
- scancodeToKey
- SFMLRenderer.cpp, 281
- SceneManager
- eng::SceneManager, 161
- Server
- srv::Server, 164

- setEnabled
 - eng::ASystem, [44](#)
 - eng::ISystem, [111](#)
- setFrameLimit
 - eng::IRenderer, [101](#)
 - eng::SFMLRenderer, [176](#)
- setLoop
 - eng::IAudio, [81](#)
 - eng::SFMLAudio, [168](#)
- setName
 - eng::AScene, [41](#)
 - eng::IScene, [107](#)
 - gme::AGameClient, [24](#)
 - gme::AGameServer, [27](#)
 - gme::IGameClient, [87](#)
 - gme::IGameServer, [89](#)
- setSpriteFrame
 - eng::IRenderer, [101](#)
 - eng::SFMLRenderer, [176](#)
- setSpritePosition
 - eng::IRenderer, [101](#)
 - eng::SFMLRenderer, [176](#)
- setSpriteScale
 - eng::IRenderer, [102](#)
 - eng::SFMLRenderer, [177](#)
- setSpriteTexture
 - eng::IRenderer, [102](#)
 - eng::SFMLRenderer, [177](#)
- setState
 - eng::Engine, [70](#)
- setTextColor
 - eng::IRenderer, [102](#)
 - eng::SFMLRenderer, [177](#)
- setTextContent
 - eng::IRenderer, [103](#)
 - eng::SFMLRenderer, [177](#)
- setTextPosition
 - eng::IRenderer, [103](#)
 - eng::SFMLRenderer, [177](#)
- setVolume
 - eng::IAudio, [82](#)
 - eng::SFMLAudio, [169](#)
- SFMLAudio
 - eng::SFMLAudio, [167](#)
- SFMLRenderer
 - eng::SFMLRenderer, [172](#), [173](#)
- SFMLRenderer.cpp
 - scanCodeToKey, [281](#)
- size
 - eng::Text, [190](#)
- size_x
 - ecs::Rect, [142](#)
- size_y
 - ecs::Rect, [142](#)
- Space
 - eng, [17](#)
- sprites
 - eng::SFMLRenderer::Impl, [92](#)
- SpriteSystem
 - cli::SpriteSystem, [185](#)
- srv, [19](#)
 - json, [19](#)
- srv::ArgsConfig, [30](#)
 - exit, [31](#)
 - fromFile, [31](#)
 - host, [31](#)
 - port, [31](#)
- srv::ArgsHandler, [35](#)
 - ~ArgsHandler, [36](#)
 - ArgsHandler, [36](#)
 - operator=, [36](#)
 - ParseArgs, [36](#)
 - ParseEnv, [37](#)
- srv::EnvConfig, [75](#)
- srv::INetworkServer, [93](#)
 - ~INetworkServer, [94](#)
- srv::Server, [163](#)
 - ~Server, [164](#)
 - operator=, [164](#)
 - Server, [164](#)
- State
 - eng, [18](#)
- Status
 - eng, [18](#)
- STOP
 - eng, [18](#)
- stop
 - eng::Engine, [70](#)
- stopAudio
 - eng::IAudio, [82](#)
 - eng::SFMLAudio, [169](#)
- Stopped
 - eng, [18](#)
- switchToScene
 - eng::SceneManager, [162](#)
- T
 - eng, [18](#)
- text_rect_fx
 - gme::Sprite, [181](#)
- text_rect_fy
 - gme::Sprite, [181](#)
- text_rect_x
 - gme::Sprite, [181](#)
- text_rect_y
 - gme::Sprite, [181](#)
- texts
 - eng::SFMLRenderer::Impl, [92](#)
- TextSyStem
 - cli::TextSyStem, [194](#)
- texture_path
 - gme::Sprite, [181](#)
- TEXTURE_PLAYER
 - cli::Path::Texture, [14](#)
- textures
 - eng::SFMLRenderer::Impl, [92](#)
- TimePoint

- utl::Clock, 57
- type
 - eng::Event, 76
 - gme::Sprite, 182
- U
 - eng, 18
- Unknown
 - eng, 17
- Up
 - eng, 17
- update
 - cli::AudioSystem, 51
 - cli::Lobby, 115
 - cli::PixelSystem, 134
 - cli::SpriteSystem, 186
 - cli::TextSyStem, 195
 - eng::IScene, 107
 - eng::ISystem, 111
 - gme::IGameClient, 87
 - gme::RTypeClient, 153
- updateSystems
 - eng::Engine, 71
- utl, 20
 - INFO, 20
 - LogLevel, 20
 - readFile, 20
 - WARNING, 20
- utl::Clock, 56
 - ~Clock, 58
 - Clock, 58
 - Duration, 57
 - getDeltaSeconds, 58
 - getElapsed, 58
 - m_isPaused, 61
 - m_pausedDuration, 61
 - m_pausedTime, 61
 - m_start, 62
 - now, 59
 - operator<<, 61
 - operator=, 59, 60
 - pause, 60
 - restart, 60
 - resume, 60
 - TimePoint, 57
- utl::Logger, 121
 - ~Logger, 123
 - COLOR_ERROR, 122
 - COLOR_INFO, 122
 - COLOR_RESET, 122
 - COLOR_WARNING, 122
 - ColorIndex, 122
 - formatLogMessage, 123
 - getColorForDuration, 123
 - init, 124
 - log, 124
 - LOG_LEVEL_COLOR, 126
 - LOG_LEVEL_STRING, 126
 - logExecutionTime, 125
 - Logger, 122, 123
 - operator=, 126
- V
 - eng, 18
- v__x
 - gme::Sprite, 182
- v__y
 - gme::Sprite, 182
- Version.hpp
 - BUILD_TYPE, 214, 216
 - GIT_COMMIT_HASH, 214, 216
 - GIT_TAG, 214, 216
 - PROJECT_NAME, 214, 216
 - PROJECT_VERSION, 214, 216
 - PROJECT_VERSION_MAJOR, 214, 216
 - PROJECT_VERSION_MINOR, 215, 216
 - PROJECT_VERSION_PATCH, 215, 217
 - VERSION_MESSAGE
 - argsHandler.cpp, 225, 227
- volume
 - ecs::Audio, 47
- W
 - eng, 18
- WARNING
 - utl, 20
- WHITE
 - lobby.cpp, 235
- width
 - cli::ArgsConfig, 29
 - eng::WindowSize, 203
- window
 - eng::SFMLRenderer::Impl, 92
- windowIsOpen
 - eng::IRenderer, 104
 - eng::SFMLRenderer, 178
- with
 - ecs::Registry::EntityBuilder, 74
- X
 - eng, 18
- x
 - ecs::Scale, 159
 - ecs::Transform, 200
 - ecs::Velocity, 202
 - eng::Text, 191
- Y
 - eng, 18
- y
 - ecs::Scale, 159
 - ecs::Transform, 200
 - ecs::Velocity, 202
 - eng::Text, 191
- Z
 - eng, 18