

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	9
4.1 Class List	9
5 File Index	13
5.1 File List	13
6 Namespace Documentation	17
6.1 cli Namespace Reference	17
6.1.1 Typedef Documentation	18
6.1.1.1 json	18
6.2 cli::Config Namespace Reference	18
6.3 cli::Config::Audio Namespace Reference	18
6.3.1 Variable Documentation	18
6.3.1.1 DEFAULT_AUDIO_MUTED	18
6.3.1.2 DEFAULT_AUDIO_VOLUME	19
6.4 cli::Config::Network Namespace Reference	19
6.4.1 Variable Documentation	19
6.4.1.1 DEFAULT_NETWORK_HOST	19
6.4.1.2 DEFAULT_NETWORK_PORT	19
6.5 cli::Config::Window Namespace Reference	19
6.5.1 Variable Documentation	19
6.5.1.1 DEFAULT_WINDOW_FRAME_LIMIT	19
6.5.1.2 DEFAULT_WINDOW_FULLSCREEN	20
6.5.1.3 DEFAULT_WINDOW_HEIGHT	20
6.5.1.4 DEFAULT_WINDOW_WIDTH	20
6.6 cli::GameConfig Namespace Reference	20
6.7 cli::GameConfig::Animation Namespace Reference	20

6.7.1 Variable Documentation	20
6.7.1.1 FRAME_DURATION	20
6.8 cli::GameConfig::Asteroid Namespace Reference	21
6.9 cli::GameConfig::Asteroid::Large Namespace Reference	21
6.9.1 Variable Documentation	21
6.9.1.1 HEALTH	21
6.9.1.2 ROTATION_SPEED	21
6.9.1.3 SCALE	21
6.9.1.4 SPAWN_RATE	21
6.9.1.5 SPEED	22
6.9.1.6 SPRITE_HEIGHT	22
6.9.1.7 SPRITE_WIDTH	22
6.10 cli::GameConfig::Asteroid::Medium Namespace Reference	22
6.10.1 Variable Documentation	22
6.10.1.1 HEALTH	22
6.10.1.2 ROTATION_SPEED	22
6.10.1.3 SCALE	23
6.10.1.4 SPAWN_RATE	23
6.10.1.5 SPEED	23
6.10.1.6 SPRITE_HEIGHT	23
6.10.1.7 SPRITE_WIDTH	23
6.11 cli::GameConfig::Asteroid::Small Namespace Reference	23
6.11.1 Variable Documentation	24
6.11.1.1 ANIMATION_DURATION	24
6.11.1.2 ANIMATION_FRAMES	24
6.11.1.3 FRAMES_PER_ROW	24
6.11.1.4 HEALTH	24
6.11.1.5 ROTATION_SPEED	24
6.11.1.6 SCALE	24
6.11.1.7 SPAWN_RATE	25
6.11.1.8 SPEED	25
6.11.1.9 SPRITE_HEIGHT	25
6.11.1.10 SPRITE_WIDTH	25
6.12 cli::GameConfig::Beam Namespace Reference	25
6.12.1 Variable Documentation	25
6.12.1.1 BAR_HEIGHT	25
6.12.1.2 BAR_WIDTH	26
6.12.1.3 BAR_X	26
6.12.1.4 BAR_Y	26
6.12.1.5 CHARGE_RATE	26
6.12.1.6 MAX_CHARGE	26
6.13 cli::GameConfig::Enemy Namespace Reference	26
6.14 cli::GameConfig::Enemy::Easy Namespace Reference	27

6.14.1 Variable Documentation	27
6.14.1.1 ANIMATION_DURATION	27
6.14.1.2 ANIMATION_FRAMES	27
6.14.1.3 DAMAGE	27
6.14.1.4 FRAMES_PER_ROW	27
6.14.1.5 HEALTH	28
6.14.1.6 SCALE	28
6.14.1.7 SHOOT_COOLDOWN	28
6.14.1.8 SPAWN_RATE	28
6.14.1.9 SPEED	28
6.14.1.10 SPRITE_HEIGHT	28
6.14.1.11 SPRITE_WIDTH	29
6.15 cli::GameConfig::Explosion Namespace Reference	29
6.15.1 Variable Documentation	29
6.15.1.1 ANIMATION_DURATION	29
6.15.1.2 ANIMATION_FRAMES	29
6.15.1.3 FRAMES_PER_ROW	29
6.15.1.4 LIFETIME	30
6.15.1.5 SCALE	30
6.15.1.6 SPRITE_HEIGHT	30
6.15.1.7 SPRITE_WIDTH	30
6.16 cli::GameConfig::Hitbox Namespace Reference	30
6.16.1 Variable Documentation	30
6.16.1.1 ASTEROID_LARGE_RADIUS	30
6.16.1.2 ASTEROID_MEDIUM_RADIUS	31
6.16.1.3 ASTEROID_SMALL_RADIUS	31
6.16.1.4 ENEMY_RADIUS	31
6.16.1.5 PLAYER_RADIUS	31
6.16.1.6 PROJECTILE_BASIC_RADIUS	31
6.16.1.7 PROJECTILE_SUPERCHARGED_RADIUS	31
6.17 cli::GameConfig::LoadingAnimation Namespace Reference	32
6.17.1 Variable Documentation	32
6.17.1.1 ANIMATION_DURATION	32
6.17.1.2 ANIMATION_FRAMES	32
6.17.1.3 OFFSET_X	32
6.17.1.4 OFFSET_Y	32
6.17.1.5 SPRITE_HEIGHT	32
6.17.1.6 SPRITE_WIDTH	32
6.18 cli::GameConfig::Player Namespace Reference	33
6.18.1 Variable Documentation	33
6.18.1.1 DIAGONAL_SPEED_MULTIPLIER	33
6.18.1.2 FRAMES_PER_ROW	33
6.18.1.3 SCALE	33

6.18.1.4 SPEED	33
6.18.1.5 SPRITE_HEIGHT	34
6.18.1.6 SPRITE_WIDTH	34
6.18.1.7 TOTAL_FRAMES	34
6.19 cli::GameConfig::Projectile Namespace Reference	34
6.20 cli::GameConfig::Projectile::Basic Namespace Reference	34
6.20.1 Variable Documentation	34
6.20.1.1 DAMAGE	34
6.20.1.2 FIRE_COOLDOWN	35
6.20.1.3 LIFETIME	35
6.20.1.4 SCALE	35
6.20.1.5 SPEED	35
6.20.1.6 SPRITE_HEIGHT	35
6.20.1.7 SPRITE_WIDTH	35
6.21 cli::GameConfig::Projectile::Supercharged Namespace Reference	35
6.21.1 Variable Documentation	36
6.21.1.1 ANIMATION_DURATION	36
6.21.1.2 ANIMATION_FRAMES	36
6.21.1.3 CHARGE_TIME	36
6.21.1.4 DAMAGE	36
6.21.1.5 FIRE_COOLDOWN	36
6.21.1.6 LIFETIME	36
6.21.1.7 SCALE	36
6.21.1.8 SPEED	37
6.21.1.9 SPRITE_HEIGHT	37
6.21.1.10 SPRITE_WIDTH	37
6.22 cli::GameConfig::Screen Namespace Reference	37
6.22.1 Variable Documentation	37
6.22.1.1 HEIGHT	37
6.22.1.2 MAX_Y	37
6.22.1.3 MIN_Y	38
6.22.1.4 REMOVE_MAX_Y	38
6.22.1.5 REMOVE_MIN_Y	38
6.22.1.6 REMOVE_X	38
6.22.1.7 SPAWN_X	38
6.22.1.8 WIDTH	38
6.23 cli::Path Namespace Reference	39
6.24 cli::Path::Audio Namespace Reference	39
6.24.1 Variable Documentation	39
6.24.1.1 AUDIO_BATTLE_THEME	39
6.24.1.2 AUDIO_COIN	39
6.24.1.3 AUDIO_TITLE	39
6.25 cli::Path::Font Namespace Reference	39

6.25.1 Variable Documentation	40
6.25.1.1 FONTS_RTYPE	40
6.26 cli::Path::Plugin Namespace Reference	40
6.26.1 Variable Documentation	40
6.26.1.1 PLUGIN_AUDIO_SFML	40
6.26.1.2 PLUGIN_NETWORK_ASIO_CLIENT	40
6.26.1.3 PLUGIN_RENDERER_SFML	40
6.27 cli::Path::Texture Namespace Reference	41
6.27.1 Variable Documentation	41
6.27.1.1 TEXTURE_ASTEROID	41
6.27.1.2 TEXTURE_ENEMY_EASY	41
6.27.1.3 TEXTURE_EXPLOSION	41
6.27.1.4 TEXTURE_PLAYER	41
6.27.1.5 TEXTURE_SHOOT	42
6.27.1.6 TEXTURE_SHOOT_CHARGED	42
6.27.1.7 TEXTURE_SHOOT_LOADING	42
6.28 cli::SpriteRect Namespace Reference	42
6.28.1 Function Documentation	42
6.28.1.1 asteroidRect()	42
6.28.1.2 enemyRect()	43
6.29 cli::SpriteRect::SpriteSheet Namespace Reference	43
6.30 ecs Namespace Reference	43
6.30.1 Typedef Documentation	44
6.30.1.1 Entity	44
6.30.2 Variable Documentation	44
6.30.2.1 INVALID_ENTITY	44
6.31 eng Namespace Reference	44
6.31.1 Typedef Documentation	45
6.31.1.1 id	45
6.31.2 Enumeration Type Documentation	45
6.31.2.1 EventType	45
6.31.2.2 Key	46
6.31.2.3 State	47
6.31.2.4 Status	47
6.32 gme Namespace Reference	47
6.33 rnp Namespace Reference	48
6.33.1 Enumeration Type Documentation	49
6.33.1.1 DisconnectReason	49
6.33.1.2 EntityType	49
6.33.1.3 ErrorCode	50
6.33.1.4 EventType	50
6.33.1.5 PacketFlags	50
6.33.1.6 PacketType	51

6.33.2 Function Documentation	51
6.33.2.1 deserializeEvents()	51
6.33.2.2 deserializeHeader()	52
6.33.2.3 serialize()	52
6.33.2.4 serializeEvents()	52
6.33.2.5 serializeHeader()	53
6.33.3 Variable Documentation	53
6.33.3.1 MAX_PAYLOAD	53
6.33.3.2 PROTOCOL_VERSION	53
6.34 srv Namespace Reference	53
6.34.1 Typedef Documentation	54
6.34.1.1 id	54
6.34.1.2 json	54
6.34.2 Variable Documentation	54
6.34.2.1 MAX_CLIENTS	54
6.34.2.2 MAX_IP_LENGTH	55
6.34.2.3 MAX_LEN_RECV_BUFFER	55
6.35 srv::Config Namespace Reference	55
6.36 srv::Config::Network Namespace Reference	55
6.36.1 Variable Documentation	55
6.36.1.1 DEFAULT_MAX_CLIENT	55
6.36.1.2 DEFAULT_NETWORK_HOST	55
6.36.1.3 DEFAULT_NETWORK_PORT	55
6.37 srv::Game Namespace Reference	56
6.37.1 Variable Documentation	56
6.37.1.1 DEFAULT_TICK_RATE	56
6.37.1.2 DEFAULT_UPDATE_INTERVAL	56
6.38 srv::Path Namespace Reference	56
6.39 srv::Path::Plugin Namespace Reference	56
6.39.1 Variable Documentation	56
6.39.1.1 PLUGINS_NETWORK_ASIO_SERVER	56
6.40 utl Namespace Reference	57
6.40.1 Typedef Documentation	57
6.40.1.1 EntryPointFn	57
6.40.1.2 LibHandle	57
6.40.2 Enumeration Type Documentation	57
6.40.2.1 LogLevel	57
6.40.2.2 PluginType	58
6.40.3 Function Documentation	58
6.40.3.1 getEnvMap()	58
6.40.3.2 readFile()	58
7 Class Documentation	59

7.1 gme::AGameClient Class Reference	59
7.1.1 Detailed Description	62
7.1.2 Constructor & Destructor Documentation	62
7.1.2.1 ~AGameClient()	62
7.1.3 Member Function Documentation	62
7.1.3.1 getName()	62
7.1.3.2 setName()	62
7.1.4 Member Data Documentation	63
7.1.4.1 m_name	63
7.2 gme::AGameServer Class Reference	63
7.2.1 Detailed Description	64
7.2.2 Constructor & Destructor Documentation	65
7.2.2.1 ~AGameServer()	65
7.2.3 Member Function Documentation	65
7.2.3.1 getName()	65
7.2.3.2 setName()	65
7.2.4 Member Data Documentation	65
7.2.4.1 m_name	65
7.3 ecs::Animation Struct Reference	66
7.3.1 Detailed Description	68
7.3.2 Member Data Documentation	68
7.3.2.1 current_frame	68
7.3.2.2 current_time	68
7.3.2.3 frame_duration	68
7.3.2.4 frame_height	68
7.3.2.5 frame_width	68
7.3.2.6 frames_per_row	68
7.3.2.7 total_frames	69
7.4 cli::AnimationSystem Class Reference	69
7.4.1 Detailed Description	71
7.4.2 Constructor & Destructor Documentation	71
7.4.2.1 AnimationSystem() [1/3]	71
7.4.2.2 ~AnimationSystem()	71
7.4.2.3 AnimationSystem() [2/3]	71
7.4.2.4 AnimationSystem() [3/3]	71
7.4.3 Member Function Documentation	71
7.4.3.1 operator=() [1/2]	71
7.4.3.2 operator=() [2/2]	71
7.4.3.3 update()	72
7.5 cli::ArgsConfig Struct Reference	72
7.5.1 Detailed Description	74
7.5.2 Member Function Documentation	74
7.5.2.1 fromFile()	74

7.5.3 Member Data Documentation	74
7.5.3.1 audio_lib_path	74
7.5.3.2 exit	74
7.5.3.3 frameLimit	74
7.5.3.4 fullscreen	75
7.5.3.5 height	75
7.5.3.6 host	75
7.5.3.7 network_lib_path	75
7.5.3.8 port	75
7.5.3.9 renderer_lib_path	75
7.5.3.10 width	76
7.6 srv::ArgsConfig Struct Reference	76
7.6.1 Detailed Description	77
7.6.2 Member Function Documentation	77
7.6.2.1 fromFile()	77
7.6.3 Member Data Documentation	77
7.6.3.1 exit	77
7.6.3.2 host	77
7.6.3.3 network_lib_path	78
7.6.3.4 port	78
7.7 cli::ArgsHandler Class Reference	78
7.7.1 Detailed Description	79
7.7.2 Constructor & Destructor Documentation	79
7.7.2.1 ArgsHandler() [1/3]	79
7.7.2.2 ~ArgsHandler()	79
7.7.2.3 ArgsHandler() [2/3]	79
7.7.2.4 ArgsHandler() [3/3]	79
7.7.3 Member Function Documentation	79
7.7.3.1 operator=() [1/2]	79
7.7.3.2 operator=() [2/2]	80
7.7.3.3 ParseArgs()	80
7.7.3.4 ParseEnv()	80
7.8 srv::ArgsHandler Class Reference	81
7.8.1 Detailed Description	82
7.8.2 Constructor & Destructor Documentation	82
7.8.2.1 ArgsHandler() [1/3]	82
7.8.2.2 ~ArgsHandler()	82
7.8.2.3 ArgsHandler() [2/3]	82
7.8.2.4 ArgsHandler() [3/3]	82
7.8.3 Member Function Documentation	82
7.8.3.1 operator=() [1/2]	82
7.8.3.2 operator=() [2/2]	82
7.8.3.3 ParseArgs()	83

7.8.3.4 ParseEnv()	83
7.9 eng::AScene Class Reference	84
7.9.1 Detailed Description	86
7.9.2 Constructor & Destructor Documentation	86
7.9.2.1 AScene() [1/3]	86
7.9.2.2 ~AScene()	86
7.9.2.3 AScene() [2/3]	86
7.9.2.4 AScene() [3/3]	86
7.9.3 Member Function Documentation	87
7.9.3.1 getId()	87
7.9.3.2 getName()	87
7.9.3.3 getRegistry()	87
7.9.3.4 operator=() [1/2]	87
7.9.3.5 operator=() [2/2]	87
7.9.3.6 setName()	87
7.9.4 Member Data Documentation	88
7.9.4.1 m_id	88
7.9.4.2 m_name	88
7.9.4.3 m_registry	88
7.9.4.4 s_nextId	88
7.10 srv::AScene Class Reference	89
7.10.1 Detailed Description	91
7.10.2 Constructor & Destructor Documentation	91
7.10.2.1 AScene() [1/3]	91
7.10.2.2 ~AScene()	91
7.10.2.3 AScene() [2/3]	91
7.10.2.4 AScene() [3/3]	91
7.10.3 Member Function Documentation	92
7.10.3.1 getId()	92
7.10.3.2 getName()	92
7.10.3.3 getRegistry()	92
7.10.3.4 operator=() [1/2]	92
7.10.3.5 operator=() [2/2]	92
7.10.3.6 setName()	92
7.10.4 Member Data Documentation	93
7.10.4.1 m_id	93
7.10.4.2 m_name	93
7.10.4.3 m_registry	93
7.10.4.4 s_nextId	93
7.11 cli::SpriteRect::SpriteSheet::Asteroid Struct Reference	93
7.11.1 Detailed Description	94
7.11.2 Member Data Documentation	94
7.11.2.1 frameH	94

7.11.2.2 frameW	94
7.11.2.3 marginX	94
7.11.2.4 marginY	94
7.11.2.5 spacingX	95
7.11.2.6 spacingY	95
7.11.2.7 totalFrames	95
7.12 ecs::Asteroid Struct Reference	95
7.12.1 Detailed Description	97
7.12.2 Member Enumeration Documentation	97
7.12.2.1 Size	97
7.12.3 Member Data Documentation	97
7.12.3.1 health	97
7.12.3.2 rotation_speed	97
7.12.3.3 size	97
7.13 cli::AsteroidSystem Class Reference	98
7.13.1 Detailed Description	100
7.13.2 Constructor & Destructor Documentation	100
7.13.2.1 AsteroidSystem() [1/3]	100
7.13.2.2 ~AsteroidSystem()	100
7.13.2.3 AsteroidSystem() [2/3]	100
7.13.2.4 AsteroidSystem() [3/3]	100
7.13.3 Member Function Documentation	100
7.13.3.1 isEnabled()	100
7.13.3.2 operator=() [1/2]	101
7.13.3.3 operator=() [2/2]	101
7.13.3.4 setEnable()	101
7.13.3.5 update()	101
7.13.4 Member Data Documentation	102
7.13.4.1 m_renderer	102
7.14 eng::ASystem Class Reference	102
7.14.1 Detailed Description	103
7.14.2 Member Function Documentation	103
7.14.2.1 isEnabled()	103
7.14.2.2 setEnable()	103
7.14.3 Member Data Documentation	103
7.14.3.1 m_isEnabled	103
7.15 ecs::Audio Struct Reference	104
7.15.1 Detailed Description	105
7.15.2 Member Data Documentation	106
7.15.2.1 loop	106
7.15.2.2 path	106
7.15.2.3 play	106
7.15.2.4 volume	106

7.16 cli::AudioSystem Class Reference	106
7.16.1 Detailed Description	109
7.16.2 Constructor & Destructor Documentation	109
7.16.2.1 AudioSystem() [1/3]	109
7.16.2.2 ~AudioSystem()	109
7.16.2.3 AudioSystem() [2/3]	109
7.16.2.4 AudioSystem() [3/3]	109
7.16.3 Member Function Documentation	110
7.16.3.1 operator=() [1/2]	110
7.16.3.2 operator=() [2/2]	110
7.16.3.3 update()	110
7.16.4 Member Data Documentation	110
7.16.4.1 m_audio	110
7.17 ecs::BeamCharge Struct Reference	111
7.17.1 Detailed Description	112
7.17.2 Member Data Documentation	113
7.17.2.1 current_charge	113
7.17.2.2 max_charge	113
7.18 cli::BeamSystem Class Reference	113
7.18.1 Detailed Description	116
7.18.2 Constructor & Destructor Documentation	116
7.18.2.1 BeamSystem() [1/3]	116
7.18.2.2 ~BeamSystem()	116
7.18.2.3 BeamSystem() [2/3]	116
7.18.2.4 BeamSystem() [3/3]	116
7.18.3 Member Function Documentation	116
7.18.3.1 operator=() [1/2]	116
7.18.3.2 operator=() [2/2]	117
7.18.3.3 update()	117
7.18.4 Member Data Documentation	117
7.18.4.1 m_renderer	117
7.19 cli::Client Class Reference	118
7.19.1 Detailed Description	119
7.19.2 Constructor & Destructor Documentation	119
7.19.2.1 Client() [1/3]	119
7.19.2.2 ~Client()	119
7.19.2.3 Client() [2/3]	120
7.19.2.4 Client() [3/3]	120
7.19.3 Member Function Documentation	120
7.19.3.1 handleEvents()	120
7.19.3.2 operator=() [1/2]	120
7.19.3.3 operator=() [2/2]	120
7.19.3.4 run()	121

7.19.4 Member Data Documentation	121
7.19.4.1 m_engine	121
7.19.4.2 m_game	121
7.19.4.3 m_keysPressed	121
7.19.4.4 m_pluginLoader	122
7.20 utl::Clock Class Reference	122
7.20.1 Detailed Description	123
7.20.2 Member Typedef Documentation	123
7.20.2.1 Duration	123
7.20.2.2 TimePoint	124
7.20.3 Constructor & Destructor Documentation	124
7.20.3.1 Clock() [1/3]	124
7.20.3.2 ~Clock()	124
7.20.3.3 Clock() [2/3]	124
7.20.3.4 Clock() [3/3]	124
7.20.4 Member Function Documentation	124
7.20.4.1 getDeltaSeconds()	124
7.20.4.2 getElapsed()	125
7.20.4.3 now()	125
7.20.4.4 operator=() [1/2]	126
7.20.4.5 operator=() [2/2]	126
7.20.4.6 pause()	126
7.20.4.7 restart()	126
7.20.4.8 resume()	127
7.20.5 Friends And Related Symbol Documentation	127
7.20.5.1 operator<<	127
7.20.6 Member Data Documentation	127
7.20.6.1 m_isPaused	127
7.20.6.2 m_pausedDuration	127
7.20.6.3 m_pausedTime	128
7.20.6.4 m_start	128
7.21 cli::CollisionSystem Class Reference	128
7.21.1 Detailed Description	131
7.21.2 Constructor & Destructor Documentation	131
7.21.2.1 CollisionSystem() [1/3]	131
7.21.2.2 ~CollisionSystem()	131
7.21.2.3 CollisionSystem() [2/3]	131
7.21.2.4 CollisionSystem() [3/3]	132
7.21.3 Member Function Documentation	132
7.21.3.1 checkCircularCollision()	132
7.21.3.2 createExplosion()	132
7.21.3.3 isEnabled()	133
7.21.3.4 operator=() [1/2]	133

7.21.3.5 operator==() [2/2]	133
7.21.3.6 removeAsteroid()	134
7.21.3.7 removeEnemy()	134
7.21.3.8 removeProjectile()	135
7.21.3.9 setEnable()	135
7.21.3.10 update()	136
7.21.4 Member Data Documentation	136
7.21.4.1 m_renderer	136
7.22 ecs::Color Struct Reference	136
7.22.1 Detailed Description	138
7.22.2 Member Data Documentation	139
7.22.2.1 a	139
7.22.2.2 b	139
7.22.2.3 g	139
7.22.2.4 r	139
7.23 eng::Color Struct Reference	139
7.23.1 Detailed Description	140
7.23.2 Member Data Documentation	140
7.23.2.1 a	140
7.23.2.2 b	140
7.23.2.3 g	140
7.23.2.4 r	140
7.24 cli::SpriteRect::SpriteSheet::Enemy Struct Reference	141
7.24.1 Detailed Description	141
7.24.2 Member Data Documentation	141
7.24.2.1 frameH	141
7.24.2.2 frameW	142
7.24.2.3 marginX	142
7.24.2.4 marginY	142
7.24.2.5 spacingX	142
7.24.2.6 spacingY	142
7.24.2.7 totalFrames	142
7.25 ecs::Enemy Struct Reference	143
7.25.1 Detailed Description	145
7.25.2 Member Data Documentation	145
7.25.2.1 damage	145
7.25.2.2 health	145
7.25.2.3 last_shot_time	145
7.25.2.4 max_health	145
7.25.2.5 shoot_cooldown	145
7.25.2.6 speed	145
7.26 cli::EnemySystem Class Reference	146
7.26.1 Detailed Description	148

7.26.2 Constructor & Destructor Documentation	148
7.26.2.1 EnemySystem() [1/3]	148
7.26.2.2 ~EnemySystem()	148
7.26.2.3 EnemySystem() [2/3]	148
7.26.2.4 EnemySystem() [3/3]	148
7.26.3 Member Function Documentation	148
7.26.3.1 isEnabled()	148
7.26.3.2 operator=() [1/2]	149
7.26.3.3 operator=() [2/2]	149
7.26.3.4 setEnabled()	149
7.26.3.5 update()	149
7.26.4 Member Data Documentation	150
7.26.4.1 m_renderer	150
7.27 eng::Engine Class Reference	150
7.27.1 Detailed Description	152
7.27.2 Constructor & Destructor Documentation	152
7.27.2.1 Engine() [1/3]	152
7.27.2.2 ~Engine()	152
7.27.2.3 Engine() [2/3]	153
7.27.2.4 Engine() [3/3]	153
7.27.3 Member Function Documentation	153
7.27.3.1 addSystem()	153
7.27.3.2 getAudio()	153
7.27.3.3 getClock()	154
7.27.3.4 getNetwork()	154
7.27.3.5 getRenderer()	154
7.27.3.6 getSceneManager()	154
7.27.3.7 getState()	155
7.27.3.8 operator=() [1/2]	155
7.27.3.9 operator=() [2/2]	155
7.27.3.10 render()	155
7.27.3.11 setState()	155
7.27.3.12 stop()	156
7.27.3.13 updateSystems()	156
7.27.4 Member Data Documentation	156
7.27.4.1 m_audio	156
7.27.4.2 m_clock	156
7.27.4.3 m_network	156
7.27.4.4 m_renderer	156
7.27.4.5 m_sceneManager	157
7.27.4.6 m_state	157
7.27.4.7 m_systems	157
7.28 ecs::Registry::EntityBuilder Class Reference	157

7.28.1 Detailed Description	159
7.28.2 Constructor & Destructor Documentation	159
7.28.2.1 EntityBuilder()	159
7.28.3 Member Function Documentation	159
7.28.3.1 build()	159
7.28.3.2 with()	159
7.28.4 Member Data Documentation	160
7.28.4.1 m_entity	160
7.28.4.2 m_registry	160
7.29 rnp:: EntityState Struct Reference	160
7.29.1 Detailed Description	161
7.29.2 Member Data Documentation	161
7.29.2.1 id	161
7.29.2.2 stateFlags	161
7.29.2.3 type	161
7.29.2.4 vx	161
7.29.2.5 vy	161
7.29.2.6 x	162
7.29.2.7 y	162
7.30 cli::EnvConfig Struct Reference	162
7.30.1 Detailed Description	162
7.31 srv::EnvConfig Struct Reference	163
7.31.1 Detailed Description	163
7.32 eng::Event Struct Reference	163
7.32.1 Detailed Description	164
7.32.2 Member Data Documentation	164
7.32.2.1 key	164
7.32.2.2 type	164
7.33 rnp::EventRecord Struct Reference	164
7.33.1 Detailed Description	165
7.33.2 Member Data Documentation	165
7.33.2.1 data	165
7.33.2.2 entityId	165
7.33.2.3 type	165
7.34 ecs::Explosion Struct Reference	166
7.34.1 Detailed Description	168
7.34.2 Member Data Documentation	168
7.34.2.1 current_frame	168
7.34.2.2 current_lifetime	168
7.34.2.3 current_time	168
7.34.2.4 frame_duration	168
7.34.2.5 frame_height	168
7.34.2.6 frame_width	169

7.34.2.7 frames_per_row	169
7.34.2.8 lifetime	169
7.34.2.9 total_frames	169
7.35 cli::ExplosionSystem Class Reference	169
7.35.1 Detailed Description	172
7.35.2 Constructor & Destructor Documentation	172
7.35.2.1 ExplosionSystem() [1/3]	172
7.35.2.2 ~ExplosionSystem()	172
7.35.2.3 ExplosionSystem() [2/3]	172
7.35.2.4 ExplosionSystem() [3/3]	172
7.35.3 Member Function Documentation	173
7.35.3.1 isEnabled()	173
7.35.3.2 operator=() [1/2]	173
7.35.3.3 operator=() [2/2]	173
7.35.3.4 removeExplosion()	173
7.35.3.5 setEnable()	174
7.35.3.6 update()	174
7.35.4 Member Data Documentation	174
7.35.4.1 m_renderer	174
7.36 ecs::Font Struct Reference	175
7.36.1 Detailed Description	176
7.36.2 Member Data Documentation	177
7.36.2.1 path	177
7.37 rnp::FragmentHeader Struct Reference	177
7.37.1 Detailed Description	177
7.37.2 Member Data Documentation	178
7.37.2.1 fragCount	178
7.37.2.2 fragId	178
7.37.2.3 fragIndex	178
7.38 cli::Game Class Reference	178
7.38.1 Detailed Description	181
7.38.2 Constructor & Destructor Documentation	181
7.38.2.1 Game() [1/3]	181
7.38.2.2 ~Game()	182
7.38.2.3 Game() [2/3]	182
7.38.2.4 Game() [3/3]	182
7.38.3 Member Function Documentation	182
7.38.3.1 event()	182
7.38.3.2 operator=() [1/2]	182
7.38.3.3 operator=() [2/2]	182
7.38.3.4 update()	182
7.38.4 Member Data Documentation	183
7.38.4.1 m_asteroidCounterEntity	183

7.38.4.2 m_audio	183
7.38.4.3 m_enemyCounterEntity	183
7.38.4.4 m_fpsEntity	183
7.38.4.5 m_keysPressed	183
7.38.4.6 m_playerEntity	183
7.39 ecs::Hitbox Struct Reference	184
7.39.1 Detailed Description	185
7.39.2 Member Data Documentation	186
7.39.2.1 radius	186
7.40 eng::IAudio Class Reference	186
7.40.1 Detailed Description	187
7.40.2 Member Function Documentation	188
7.40.2.1 createAudio()	188
7.40.2.2 isPlaying()	188
7.40.2.3 playAudio()	188
7.40.2.4 setLoop()	189
7.40.2.5 setVolume()	189
7.40.2.6 stopAudio()	189
7.41 ecs::IComponent Struct Reference	190
7.41.1 Detailed Description	190
7.41.2 Member Data Documentation	191
7.41.2.1 id	191
7.42 gme::IGameClient Class Reference	191
7.42.1 Detailed Description	193
7.42.2 Constructor & Destructor Documentation	193
7.42.2.1 ~IGameClient()	193
7.42.3 Member Function Documentation	193
7.42.3.1 getCurrentScene()	193
7.42.3.2 getName()	194
7.42.3.3 setName()	194
7.42.3.4 update()	194
7.43 gme::IGameServer Class Reference	194
7.43.1 Detailed Description	196
7.43.2 Constructor & Destructor Documentation	196
7.43.2.1 ~IGameServer()	196
7.43.3 Member Function Documentation	196
7.43.3.1 getName()	196
7.43.3.2 setName()	196
7.44 eng::INetworkClient Class Reference	196
7.44.1 Detailed Description	199
7.44.2 Member Typedef Documentation	199
7.44.2.1 PacketHandler	199
7.44.3 Constructor & Destructor Documentation	199

7.44.3.1 ~INetworkClient()	199
7.44.4 Member Function Documentation	199
7.44.4.1 connect()	199
7.44.4.2 disconnect()	200
7.44.4.3 getServerTickRate()	200
7.44.4.4 getSessionId()	200
7.44.4.5 sendAck()	200
7.44.4.6 sendConnect()	200
7.44.4.7 sendConnectWithCaps()	200
7.44.4.8 sendDisconnect() [1/2]	200
7.44.4.9 sendDisconnect() [2/2]	200
7.44.4.10 sendPing() [1/2]	200
7.44.4.11 sendPing() [2/2]	201
7.44.4.12 sendPlayerInput()	201
7.44.4.13 sendPlayerInputAsEvent()	201
7.44.4.14 setEventsHandler()	201
7.44.4.15 setPacketHandler()	201
7.45 srv::INetworkServer Class Reference	201
7.45.1 Detailed Description	203
7.45.2 Constructor & Destructor Documentation	204
7.45.2.1 ~INetworkServer()	204
7.45.3 Member Function Documentation	204
7.45.3.1 init()	204
7.45.3.2 setServerCapabilities()	204
7.45.3.3 setTickRate()	204
7.45.3.4 start()	204
7.45.3.5 stop()	204
7.46 utl::IPlugin Interface Reference	205
7.46.1 Detailed Description	206
7.46.2 Constructor & Destructor Documentation	206
7.46.2.1 ~IPlugin()	206
7.46.3 Member Function Documentation	206
7.46.3.1 getName()	206
7.46.3.2 getType()	206
7.47 ecs::Registry::IPool Class Reference	206
7.47.1 Detailed Description	207
7.47.2 Constructor & Destructor Documentation	207
7.47.2.1 ~IPool()	207
7.47.3 Member Function Documentation	207
7.47.3.1 remove()	207
7.48 eng::IRenderer Class Reference	208
7.48.1 Detailed Description	210
7.48.2 Member Function Documentation	210

7.48.2.1 clearWindow()	210
7.48.2.2 closeWindow()	210
7.48.2.3 createFont()	210
7.48.2.4 createSprite()	211
7.48.2.5 createText()	211
7.48.2.6 createTexture()	211
7.48.2.7 createWindow()	211
7.48.2.8 displayWindow()	212
7.48.2.9 drawPoint()	212
7.48.2.10 drawSprite()	212
7.48.2.11 drawText()	213
7.48.2.12 getWindowSize()	213
7.48.2.13 pollEvent()	213
7.48.2.14 setFrameLimit()	213
7.48.2.15 setSpriteFrame()	213
7.48.2.16 setSpritePosition()	214
7.48.2.17 setSpriteScale()	214
7.48.2.18 setSpriteTexture()	214
7.48.2.19 setTextColor()	215
7.48.2.20 setTextContent()	215
7.48.2.21 setTextPosition()	215
7.48.2.22 windowIsOpen()	216
7.49 eng::IScene Class Reference	216
7.49.1 Detailed Description	218
7.49.2 Constructor & Destructor Documentation	218
7.49.2.1 ~IScene()	218
7.49.3 Member Function Documentation	218
7.49.3.1 event()	218
7.49.3.2 getId()	219
7.49.3.3 getName()	219
7.49.3.4 getRegistry()	219
7.49.3.5 setName()	219
7.49.3.6 update()	219
7.50 gme::IScene Class Reference	220
7.50.1 Detailed Description	221
7.50.2 Constructor & Destructor Documentation	221
7.50.2.1 ~IScene()	221
7.50.3 Member Function Documentation	221
7.50.3.1 getEntities()	221
7.50.3.2 getEntitiesMutable()	221
7.50.3.3 getName()	221
7.51 srv::IScene Class Reference	222
7.51.1 Detailed Description	223

7.51.2 Constructor & Destructor Documentation	223
7.51.2.1 ~IScene()	223
7.51.3 Member Function Documentation	223
7.51.3.1 getId()	223
7.51.3.2 getName()	224
7.51.3.3 getRegistry()	224
7.51.3.4 setName()	224
7.51.3.5 update()	224
7.52 eng::ISystem Class Reference	224
7.52.1 Detailed Description	225
7.52.2 Constructor & Destructor Documentation	225
7.52.2.1 ~ISystem()	225
7.52.3 Member Function Documentation	225
7.52.3.1 isEnabled()	225
7.52.3.2 setEnabled()	225
7.52.3.3 update()	226
7.53 ecs::LoadingAnimation Struct Reference	226
7.53.1 Detailed Description	228
7.53.2 Member Data Documentation	228
7.53.2.1 current_frame	228
7.53.2.2 current_time	228
7.53.2.3 frame_duration	228
7.53.2.4 frame_height	228
7.53.2.5 frame_width	228
7.53.2.6 frames_per_row	228
7.53.2.7 total_frames	229
7.54 cli::LoadingAnimationSystem Class Reference	229
7.54.1 Detailed Description	231
7.54.2 Constructor & Destructor Documentation	231
7.54.2.1 LoadingAnimationSystem() [1/3]	231
7.54.2.2 ~LoadingAnimationSystem()	231
7.54.2.3 LoadingAnimationSystem() [2/3]	231
7.54.2.4 LoadingAnimationSystem() [3/3]	231
7.54.3 Member Function Documentation	231
7.54.3.1 operator=() [1/2]	231
7.54.3.2 operator=() [2/2]	232
7.54.3.3 update()	232
7.54.4 Member Data Documentation	232
7.54.4.1 m_renderer	232
7.55 cli::Lobby Class Reference	233
7.55.1 Detailed Description	235
7.55.2 Constructor & Destructor Documentation	235
7.55.2.1 Lobby() [1/3]	235

7.55.2.2 ~Lobby()	236
7.55.2.3 Lobby() [2/3]	236
7.55.2.4 Lobby() [3/3]	236
7.55.3 Member Function Documentation	236
7.55.3.1 event()	236
7.55.3.2 operator=() [1/2]	236
7.55.3.3 operator=() [2/2]	236
7.55.3.4 update()	236
7.55.4 Member Data Documentation	237
7.55.4.1 m_audio	237
7.55.4.2 m_fpsEntity	237
7.55.4.3 m_keysPressed	237
7.55.4.4 m_menuOptions	237
7.55.4.5 m_selectedIndex	237
7.55.4.6 onOptionSelected	237
7.56 gme::LobbyScene Class Reference	238
7.56.1 Detailed Description	240
7.56.2 Constructor & Destructor Documentation	240
7.56.2.1 LobbyScene()	240
7.56.3 Member Function Documentation	240
7.56.3.1 addEntity()	240
7.56.3.2 getEntities()	240
7.56.3.3 getEntitiesMutable()	241
7.56.3.4 getName()	241
7.56.4 Member Data Documentation	241
7.56.4.1 m_entities	241
7.56.4.2 m_name	241
7.57 utl::Logger Class Reference	242
7.57.1 Detailed Description	243
7.57.2 Member Enumeration Documentation	243
7.57.2.1 ColorIndex	243
7.57.3 Constructor & Destructor Documentation	243
7.57.3.1 Logger() [1/3]	243
7.57.3.2 Logger() [2/3]	244
7.57.3.3 Logger() [3/3]	244
7.57.3.4 ~Logger()	244
7.57.4 Member Function Documentation	244
7.57.4.1 formatLogMessage()	244
7.57.4.2 getColorForDuration()	245
7.57.4.3 init()	245
7.57.4.4 log()	246
7.57.4.5 logExecutionTime()	247
7.57.4.6 operator=() [1/2]	247

7.57.4.7 operator==() [2/2]	247
7.57.5 Member Data Documentation	247
7.57.5.1 LOG_LEVEL_COLOR	247
7.57.5.2 LOG_LEVEL_STRING	248
7.58 ecs::Mob Struct Reference	248
7.58.1 Detailed Description	249
7.59 rnp::PacketAck Struct Reference	250
7.59.1 Detailed Description	250
7.59.2 Member Data Documentation	250
7.59.2.1 ackBits	250
7.59.2.2 cumulativeAck	250
7.60 rnp::PacketConnect Struct Reference	251
7.60.1 Detailed Description	251
7.60.2 Member Data Documentation	251
7.60.2.1 clientCaps	251
7.60.2.2 nameLen	251
7.60.2.3 playerName	252
7.61 rnp::PacketConnectAccept Struct Reference	252
7.61.1 Detailed Description	252
7.61.2 Member Data Documentation	253
7.61.2.1 mtuPayloadBytes	253
7.61.2.2 serverCaps	253
7.61.2.3 sessionId	253
7.61.2.4 tickRateHz	253
7.62 rnp::PacketDisconnect Struct Reference	253
7.62.1 Detailed Description	254
7.62.2 Member Data Documentation	254
7.62.2.1 reasonCode	254
7.63 rnp::PacketError Struct Reference	254
7.63.1 Detailed Description	255
7.63.2 Member Data Documentation	255
7.63.2.1 description	255
7.63.2.2 errorCode	255
7.63.2.3 msgLen	255
7.64 rnp::PacketHeader Struct Reference	256
7.64.1 Detailed Description	256
7.64.2 Member Data Documentation	256
7.64.2.1 flags	256
7.64.2.2 length	257
7.64.2.3 reserved	257
7.64.2.4 sequence	257
7.64.2.5 sessionId	257
7.64.2.6 type	257

7.65 rnp::PacketPingPong Struct Reference	258
7.65.1 Detailed Description	258
7.65.2 Member Data Documentation	258
7.65.2.1 nonce	258
7.65.2.2 sendTimeMs	258
7.66 rnp::PacketWorldState Struct Reference	259
7.66.1 Detailed Description	259
7.66.2 Member Data Documentation	260
7.66.2.1 entities	260
7.66.2.2 entityCount	260
7.66.2.3 serverTick	260
7.67 ecs::Pixel Struct Reference	260
7.67.1 Detailed Description	261
7.68 cli::PixelSystem Class Reference	262
7.68.1 Detailed Description	264
7.68.2 Constructor & Destructor Documentation	264
7.68.2.1 PixelSystem() [1/3]	264
7.68.2.2 ~PixelSystem()	264
7.68.2.3 PixelSystem() [2/3]	264
7.68.2.4 PixelSystem() [3/3]	264
7.68.3 Member Function Documentation	264
7.68.3.1 operator=() [1/2]	264
7.68.3.2 operator=() [2/2]	265
7.68.3.3 update()	265
7.68.4 Member Data Documentation	265
7.68.4.1 m_renderer	265
7.69 ecs::Player Struct Reference	266
7.69.1 Detailed Description	267
7.69.2 Member Data Documentation	268
7.69.2.1 is_cli	268
7.70 cli::PlayerDirectionSystem Class Reference	268
7.70.1 Detailed Description	270
7.70.2 Constructor & Destructor Documentation	270
7.70.2.1 PlayerDirectionSystem() [1/3]	270
7.70.2.2 ~PlayerDirectionSystem()	270
7.70.2.3 PlayerDirectionSystem() [2/3]	270
7.70.2.4 PlayerDirectionSystem() [3/3]	270
7.70.3 Member Function Documentation	270
7.70.3.1 operator=() [1/2]	270
7.70.3.2 operator=() [2/2]	270
7.70.3.3 update()	271
7.71 utl::PluginLoader Class Reference	271
7.71.1 Detailed Description	273

7.71.2 Constructor & Destructor Documentation	273
7.71.2.1 PluginLoader() [1/3]	273
7.71.2.2 ~PluginLoader()	273
7.71.2.3 PluginLoader() [2/3]	273
7.71.2.4 PluginLoader() [3/3]	273
7.71.3 Member Function Documentation	274
7.71.3.1 getEntryPoint()	274
7.71.3.2 loadLibrary()	274
7.71.3.3 loadPlugin()	274
7.71.3.4 operator=() [1/2]	275
7.71.3.5 operator=() [2/2]	276
7.71.4 Member Data Documentation	276
7.71.4.1 m_handles	276
7.71.4.2 m_mutex	276
7.71.4.3 m_plugins	276
7.72 ecs::Registry::Pool< T > Class Template Reference	277
7.72.1 Detailed Description	279
7.72.2 Member Function Documentation	279
7.72.2.1 add()	279
7.72.2.2 get()	279
7.72.2.3 has()	279
7.72.2.4 remove()	279
7.72.3 Member Data Documentation	280
7.72.3.1 data	280
7.73 ecs::Projectile Struct Reference	280
7.73.1 Detailed Description	282
7.73.2 Member Enumeration Documentation	282
7.73.2.1 Type	282
7.73.3 Member Data Documentation	282
7.73.3.1 current_lifetime	282
7.73.3.2 damage	282
7.73.3.3 lifetime	282
7.73.3.4 type	282
7.74 cli::ProjectileManager Class Reference	283
7.74.1 Detailed Description	284
7.74.2 Constructor & Destructor Documentation	284
7.74.2.1 ProjectileManager() [1/3]	284
7.74.2.2 ~ProjectileManager()	284
7.74.2.3 ProjectileManager() [2/3]	284
7.74.2.4 ProjectileManager() [3/3]	284
7.74.3 Member Function Documentation	284
7.74.3.1 createBasicProjectile()	284
7.74.3.2 createProjectile()	285

7.74.3.3 createSuperchargedProjectile()	286
7.74.3.4 operator=() [1/2]	287
7.74.3.5 operator=() [2/2]	287
7.75 cli::ProjectileSystem Class Reference	288
7.75.1 Detailed Description	290
7.75.2 Constructor & Destructor Documentation	290
7.75.2.1 ProjectileSystem() [1/3]	290
7.75.2.2 ~ProjectileSystem()	290
7.75.2.3 ProjectileSystem() [2/3]	290
7.75.2.4 ProjectileSystem() [3/3]	290
7.75.3 Member Function Documentation	290
7.75.3.1 operator=() [1/2]	290
7.75.3.2 operator=() [2/2]	290
7.75.3.3 update()	291
7.76 cli::SpriteRect::Rect Struct Reference	291
7.76.1 Detailed Description	292
7.76.2 Member Data Documentation	292
7.76.2.1 h	292
7.76.2.2 w	292
7.76.2.3 x	292
7.76.2.4 y	292
7.77 ecs::Rect Struct Reference	293
7.77.1 Detailed Description	294
7.77.2 Member Data Documentation	295
7.77.2.1 pos_x	295
7.77.2.2 pos_y	295
7.77.2.3 size_x	295
7.77.2.4 size_y	295
7.78 ecs::Registry Class Reference	295
7.78.1 Detailed Description	297
7.78.2 Constructor & Destructor Documentation	297
7.78.2.1 Registry() [1/3]	297
7.78.2.2 ~Registry()	297
7.78.2.3 Registry() [2/3]	297
7.78.2.4 Registry() [3/3]	297
7.78.3 Member Function Documentation	298
7.78.3.1 addComponent()	298
7.78.3.2 createEntity()	298
7.78.3.3 getAll()	299
7.78.3.4 getComponent()	300
7.78.3.5 getPool()	302
7.78.3.6 hasComponent()	302
7.78.3.7 onComponentAdded()	303

7.78.3.8 operator=() [1/2]	303
7.78.3.9 operator=() [2/2]	304
7.78.3.10 removeComponent()	304
7.78.4 Member Data Documentation	305
7.78.4.1 m_components	305
7.78.4.2 m_entities	305
7.78.4.3 m_lastEntity	305
7.78.4.4 m_onComponentAddedCallbacks	305
7.79 gme::RTypClient Class Reference	306
7.79.1 Detailed Description	308
7.79.2 Constructor & Destructor Documentation	308
7.79.2.1 RTypeClient() [1/3]	308
7.79.2.2 ~RTypeClient()	308
7.79.2.3 RTypeClient() [2/3]	308
7.79.2.4 RTypeClient() [3/3]	309
7.79.3 Member Function Documentation	309
7.79.3.1 getCurrentScene()	309
7.79.3.2 operator=() [1/2]	309
7.79.3.3 operator=() [2/2]	309
7.79.3.4 update()	309
7.79.4 Member Data Documentation	309
7.79.4.1 m_currentScene	309
7.80 gme::RTypServer Class Reference	310
7.80.1 Detailed Description	311
7.80.2 Constructor & Destructor Documentation	312
7.80.2.1 RTypeServer() [1/3]	312
7.80.2.2 ~RTypeServer()	312
7.80.2.3 RTypeServer() [2/3]	312
7.80.2.4 RTypeServer() [3/3]	312
7.80.3 Member Function Documentation	312
7.80.3.1 operator=() [1/2]	312
7.80.3.2 operator=() [2/2]	312
7.81 ecs::Scale Struct Reference	313
7.81.1 Detailed Description	314
7.81.2 Member Data Documentation	315
7.81.2.1 x	315
7.81.2.2 y	315
7.82 eng::SceneManager Class Reference	315
7.82.1 Detailed Description	317
7.82.2 Constructor & Destructor Documentation	317
7.82.2.1 SceneManager() [1/3]	317
7.82.2.2 ~SceneManager()	317
7.82.2.3 SceneManager() [2/3]	317

7.82.2.4 SceneManager() [3/3]	317
7.82.3 Member Function Documentation	317
7.82.3.1 addScene()	317
7.82.3.2 getCurrentScene()	318
7.82.3.3 getScene()	318
7.82.3.4 operator=() [1/2]	318
7.82.3.5 operator=() [2/2]	318
7.82.3.6 switchToScene()	318
7.82.4 Member Data Documentation	318
7.82.4.1 m_currentSceneId	318
7.82.4.2 m_scenes	319
7.83 srv::SceneManager Class Reference	319
7.83.1 Detailed Description	321
7.83.2 Constructor & Destructor Documentation	321
7.83.2.1 SceneManager() [1/3]	321
7.83.2.2 ~SceneManager()	321
7.83.2.3 SceneManager() [2/3]	321
7.83.2.4 SceneManager() [3/3]	321
7.83.3 Member Function Documentation	321
7.83.3.1 addScene()	321
7.83.3.2 getCurrentScene()	322
7.83.3.3 getScene()	322
7.83.3.4 operator=() [1/2]	322
7.83.3.5 operator=() [2/2]	322
7.83.3.6 switchToScene()	322
7.83.4 Member Data Documentation	322
7.83.4.1 m_currentSceneId	322
7.83.4.2 m_scenes	323
7.84 srv::Server Class Reference	323
7.84.1 Detailed Description	325
7.84.2 Constructor & Destructor Documentation	325
7.84.2.1 Server() [1/3]	325
7.84.2.2 ~Server()	325
7.84.2.3 Server() [2/3]	325
7.84.2.4 Server() [3/3]	325
7.84.3 Member Function Documentation	326
7.84.3.1 operator=() [1/2]	326
7.84.3.2 operator=() [2/2]	326
7.84.3.3 run()	326
7.84.4 Member Data Documentation	326
7.84.4.1 m_network	326
7.84.4.2 m_pluginLoader	326
7.85 cli::Settings Class Reference	327

7.85.1 Detailed Description	329
7.85.2 Constructor & Destructor Documentation	329
7.85.2.1 Settings() [1/3]	329
7.85.2.2 ~Settings()	329
7.85.2.3 Settings() [2/3]	330
7.85.2.4 Settings() [3/3]	330
7.85.3 Member Function Documentation	330
7.85.3.1 event()	330
7.85.3.2 operator=() [1/2]	330
7.85.3.3 operator=() [2/2]	330
7.85.3.4 update()	330
7.85.4 Member Data Documentation	331
7.85.4.1 m_audio	331
7.85.4.2 m_keysPressed	331
7.85.4.3 onLeave	331
7.86 utl::SharedLib Struct Reference	331
7.86.1 Detailed Description	332
7.86.2 Constructor & Destructor Documentation	332
7.86.2.1 SharedLib() [1/3]	332
7.86.2.2 ~SharedLib()	332
7.86.2.3 SharedLib() [2/3]	333
7.86.2.4 SharedLib() [3/3]	333
7.86.3 Member Function Documentation	333
7.86.3.1 close()	333
7.86.3.2 operator bool()	333
7.86.3.3 operator=() [1/2]	333
7.86.3.4 operator=() [2/2]	334
7.86.4 Member Data Documentation	334
7.86.4.1 handle	334
7.87 cli::SpawnSystem Class Reference	334
7.87.1 Detailed Description	337
7.87.2 Constructor & Destructor Documentation	337
7.87.2.1 SpawnSystem() [1/3]	337
7.87.2.2 ~SpawnSystem()	337
7.87.2.3 SpawnSystem() [2/3]	337
7.87.2.4 SpawnSystem() [3/3]	338
7.87.3 Member Function Documentation	338
7.87.3.1 isEnabled()	338
7.87.3.2 operator=() [1/2]	338
7.87.3.3 operator=() [2/2]	338
7.87.3.4 setEnable()	338
7.87.3.5 spawnAsteroid()	339
7.87.3.6 spawnEnemy()	339

7.87.3.7 spawnWave()	340
7.87.3.8 update()	341
7.87.4 Member Data Documentation	341
7.87.4.1 m_asteroidSpawnTimer	341
7.87.4.2 m_enemySpawnTimer	341
7.87.4.3 m_renderer	341
7.87.4.4 m_waveTimer	342
7.88 gme::Sprite Struct Reference	342
7.88.1 Detailed Description	344
7.88.2 Member Data Documentation	344
7.88.2.1 a	344
7.88.2.2 b	344
7.88.2.3 g	344
7.88.2.4 id	344
7.88.2.5 pos_x	344
7.88.2.6 pos_y	345
7.88.2.7 r	345
7.88.2.8 scale_x	345
7.88.2.9 scale_y	345
7.88.2.10 text_rect_fx	345
7.88.2.11 text_rect_fy	345
7.88.2.12 text_rect_x	345
7.88.2.13 text_rect_y	345
7.88.2.14 texture_path	346
7.88.2.15 type	346
7.88.2.16 v_x	346
7.88.2.17 v_y	346
7.89 cli::SpriteSystem Class Reference	346
7.89.1 Detailed Description	349
7.89.2 Constructor & Destructor Documentation	349
7.89.2.1 SpriteSystem() [1/3]	349
7.89.2.2 ~SpriteSystem()	349
7.89.2.3 SpriteSystem() [2/3]	349
7.89.2.4 SpriteSystem() [3/3]	349
7.89.3 Member Function Documentation	349
7.89.3.1 operator=() [1/2]	349
7.89.3.2 operator=() [2/2]	350
7.89.3.3 update()	350
7.89.4 Member Data Documentation	350
7.89.4.1 m_renderer	350
7.90 ecs::Text Struct Reference	351
7.90.1 Detailed Description	352
7.90.2 Member Data Documentation	353

7.90.2.1 content	353
7.90.2.2 font_size	353
7.91 eng::Text Struct Reference	353
7.91.1 Detailed Description	354
7.91.2 Member Data Documentation	354
7.91.2.1 color	354
7.91.2.2 content	354
7.91.2.3 font_name	354
7.91.2.4 name	354
7.91.2.5 size	354
7.91.2.6 x	355
7.91.2.7 y	355
7.92 cli::TextSystem Class Reference	355
7.92.1 Detailed Description	358
7.92.2 Constructor & Destructor Documentation	358
7.92.2.1 TextSystem() [1/3]	358
7.92.2.2 ~TextSystem()	358
7.92.2.3 TextSystem() [2/3]	358
7.92.2.4 TextSystem() [3/3]	358
7.92.3 Member Function Documentation	359
7.92.3.1 operator=() [1/2]	359
7.92.3.2 operator=() [2/2]	359
7.92.3.3 update()	359
7.92.4 Member Data Documentation	359
7.92.4.1 m_renderer	359
7.93 ecs::Texture Struct Reference	360
7.93.1 Detailed Description	361
7.93.2 Member Data Documentation	362
7.93.2.1 path	362
7.94 ecs::Transform Struct Reference	362
7.94.1 Detailed Description	363
7.94.2 Member Data Documentation	364
7.94.2.1 rotation	364
7.94.2.2 x	364
7.94.2.3 y	364
7.95 ecs::Velocity Struct Reference	364
7.95.1 Detailed Description	365
7.95.2 Member Data Documentation	366
7.95.2.1 x	366
7.95.2.2 y	366
7.96 cli::WeaponSystem Class Reference	366
7.96.1 Detailed Description	369
7.96.2 Constructor & Destructor Documentation	369

7.96.2.1 WeaponSystem() [1/3]	369
7.96.2.2 ~WeaponSystem()	370
7.96.2.3 WeaponSystem() [2/3]	370
7.96.2.4 WeaponSystem() [3/3]	370
7.96.3 Member Function Documentation	370
7.96.3.1 hideLoadingAnimation()	370
7.96.3.2 operator=() [1/2]	371
7.96.3.3 operator=() [2/2]	371
7.96.3.4 reset()	371
7.96.3.5 showLoadingAnimation()	371
7.96.3.6 tryFireBasic()	372
7.96.3.7 tryFireSupercharged()	373
7.96.3.8 update() [1/2]	374
7.96.3.9 update() [2/2]	374
7.96.4 Member Data Documentation	375
7.96.4.1 m_fireCooldown	375
7.96.4.2 m_isCharging	375
7.97 eng:WindowSize Struct Reference	376
7.97.1 Detailed Description	376
7.97.2 Member Data Documentation	376
7.97.2.1 height	376
7.97.2.2 width	376
8 File Documentation	377
8.1 /home/masina/Projects/Epitech/rtype/client/include/Client/ArgsHandler.hpp File Reference	377
8.1.1 Detailed Description	378
8.2 ArgsHandler.hpp	378
8.3 /home/masina/Projects/Epitech/rtype/server/include/Server/ArgsHandler.hpp File Reference	379
8.3.1 Detailed Description	381
8.4 ArgsHandler.hpp	381
8.5 /home/masina/Projects/Epitech/rtype/client/include/Client/Client.hpp File Reference	381
8.5.1 Detailed Description	382
8.6 Client.hpp	383
8.7 /home/masina/Projects/Epitech/rtype/client/include/Client/Common.hpp File Reference	383
8.7.1 Detailed Description	385
8.8 Common.hpp	385
8.9 /home/masina/Projects/Epitech/rtype/server/include/Server/Common.hpp File Reference	386
8.9.1 Detailed Description	387
8.10 Common.hpp	388
8.11 /home/masina/Projects/Epitech/rtype/client/include/Client/GameConfig.hpp File Reference	388
8.11.1 Detailed Description	391

8.12 GameConfig.hpp	391
8.13 /home/masina/Projects/Epitech/rtype/client/include/Client/Generated/Version.hpp File Reference	393
8.13.1 Macro Definition Documentation	393
8.13.1.1 BUILD_TYPE	393
8.13.1.2 GIT_COMMIT_HASH	394
8.13.1.3 GIT_TAG	394
8.13.1.4 PROJECT_NAME	394
8.13.1.5 PROJECT_VERSION	394
8.13.1.6 PROJECT_VERSION_MAJOR	394
8.13.1.7 PROJECT_VERSION_MINOR	394
8.13.1.8 PROJECT_VERSION_PATCH	394
8.14 Version.hpp	395
8.15 /home/masina/Projects/Epitech/rtype/server/include/Server/Generated/Version.hpp File Reference	395
8.15.1 Macro Definition Documentation	396
8.15.1.1 BUILD_TYPE	396
8.15.1.2 GIT_COMMIT_HASH	396
8.15.1.3 GIT_TAG	396
8.15.1.4 PROJECT_NAME	396
8.15.1.5 PROJECT_VERSION	396
8.15.1.6 PROJECT_VERSION_MAJOR	396
8.15.1.7 PROJECT_VERSION_MINOR	396
8.15.1.8 PROJECT_VERSION_PATCH	397
8.16 Version.hpp	397
8.17 /home/masina/Projects/Epitech/rtype/client/include/Client/ProjectileManager.hpp File Reference	397
8.17.1 Detailed Description	398
8.18 ProjectileManager.hpp	398
8.19 /home/masina/Projects/Epitech/rtype/client/include/Client/Scenes/Game.hpp File Reference	399
8.19.1 Detailed Description	400
8.20 Game.hpp	401
8.21 /home/masina/Projects/Epitech/rtype/client/include/Client/Scenes/Lobby.hpp File Reference	401
8.21.1 Detailed Description	402
8.22 Lobby.hpp	403
8.23 /home/masina/Projects/Epitech/rtype/client/include/Client/Scenes/Settings.hpp File Reference	403
8.23.1 Detailed Description	404
8.24 Settings.hpp	404
8.25 /home/masina/Projects/Epitech/rtype/client/include/Client/SpriteRect.hpp File Reference	405
8.25.1 Detailed Description	405
8.26 SpriteRect.hpp	406

8.27 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Animation.hpp File Reference	406
8.28 Animation.hpp	408
8.29 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Asteroid.hpp File Reference	408
8.30 Asteroid.hpp	410
8.31 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Audio.hpp File Reference	411
8.32 Audio.hpp	412
8.33 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Beam.hpp File Reference	413
8.34 Beam.hpp	414
8.35 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Collision.hpp File Reference	416
8.36 Collision.hpp	417
8.37 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Enemy.hpp File Reference	419
8.38 Enemy.hpp	420
8.39 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Explosion.hpp File Reference	422
8.40 Explosion.hpp	423
8.41 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems>LoadingAnimation.hpp File Reference	424
8.42 LoadingAnimation.hpp	425
8.43 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Pixel.hpp File Ref- erence	426
8.44 Pixel.hpp	427
8.45 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/PlayerDirection.hpp File Reference	428
8.45.1 Macro Definition Documentation	429
8.45.1.1 M_PI	429
8.46 PlayerDirection.hpp	430
8.47 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Projectile.hpp File Reference	431
8.48 Projectile.hpp	432
8.49 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Spawn.hpp File Reference	433
8.50 Spawn.hpp	434
8.51 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Sprite.hpp File Reference	435
8.52 Sprite.hpp	437
8.53 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Systems.hpp File Reference	437
8.53.1 Detailed Description	438
8.54 Systems.hpp	438
8.55 /home/masina/Projects/Epitech/rtype/server/include/Server/Systems/Systems.hpp File Reference	439

8.55.1 Detailed Description	439
8.56 Systems.hpp	440
8.57 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Text.hpp File Reference	440
8.58 Text.hpp	441
8.59 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Weapon.hpp File Reference	442
8.60 Weapon.hpp	443
8.61 /home/masina/Projects/Epitech/rtype/client/src/argsHandler.cpp File Reference	444
8.61.1 Macro Definition Documentation	445
8.61.1.1 APP_EXTENSION	445
8.61.2 Variable Documentation	445
8.61.2.1 HELP_MESSAGE	445
8.61.2.2 VERSION_MESSAGE	446
8.62 argsHandler.cpp	446
8.63 /home/masina/Projects/Epitech/rtype/server/src/argsHandler.cpp File Reference	448
8.63.1 Macro Definition Documentation	448
8.63.1.1 APP_EXTENSION	448
8.63.2 Variable Documentation	448
8.63.2.1 HELP_MESSAGE	448
8.63.2.2 VERSION_MESSAGE	449
8.64 argsHandler.cpp	449
8.65 /home/masina/Projects/Epitech/rtype/client/src/client.cpp File Reference	450
8.65.1 Variable Documentation	451
8.65.1.1 DARK	451
8.66 client.cpp	451
8.67 /home/masina/Projects/Epitech/rtype/client/src/event.cpp File Reference	452
8.68 event.cpp	452
8.69 /home/masina/Projects/Epitech/rtype/client/src/main.cpp File Reference	453
8.69.1 Function Documentation	453
8.69.1.1 main()	453
8.70 main.cpp	454
8.71 /home/masina/Projects/Epitech/rtype/server/src/main.cpp File Reference	454
8.71.1 Function Documentation	455
8.71.1.1 main()	455
8.72 main.cpp	455
8.73 /home/masina/Projects/Epitech/rtype/client/src/ProjectileManager.cpp File Reference	456
8.73.1 Detailed Description	456
8.74 ProjectileManager.cpp	456
8.75 /home/masina/Projects/Epitech/rtype/client/src/scenes/game.cpp File Reference	457
8.75.1 Variable Documentation	458
8.75.1.1 BLUE	458
8.75.1.2 BLUE_SECOND	458
8.75.1.3 GREEN	458

8.75.1.4 PURPLE	458
8.75.1.5 WHITE	458
8.75.1.6 WHITE_TRANS	459
8.75.1.7 YELLOW	459
8.76 game.cpp	459
8.77 /home/masina/Projects/Epitech/rtype/client/src/scenes/lobby.cpp File Reference . . .	463
8.77.1 Variable Documentation	464
8.77.1.1 WHITE	464
8.78 lobby.cpp	464
8.79 /home/masina/Projects/Epitech/rtype/client/src/scenes/settings.cpp File Reference . . .	467
8.79.1 Variable Documentation	467
8.79.1.1 WHITE	467
8.80 settings.cpp	467
8.81 /home/masina/Projects/Epitech/rtype/client/src/systems/weapon.cpp File Reference . . .	469
8.82 weapon.cpp	469
8.83 /home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.hpp File Reference	472
8.83.1 Detailed Description	473
8.84 Component.hpp	473
8.85 /home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Entity.hpp File Reference	475
8.85.1 Detailed Description	476
8.86 Entity.hpp	476
8.87 /home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Interfaces/ISystems.hpp File Reference	476
8.88 ISystems.hpp	477
8.89 /home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Registry.hpp File Reference	478
8.89.1 Detailed Description	478
8.90 Registry.hpp	479
8.91 /home/masina/Projects/Epitech/rtype/modules/ECS/src/registry.cpp File Reference . . .	481
8.92 registry.cpp	481
8.93 /home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/Engine.hpp File Reference	481
8.93.1 Detailed Description	482
8.94 Engine.hpp	482
8.95 /home/masina/Projects/Epitech/rtype/modules/Engine/src/Engine.cpp File Reference . . .	484
8.96 Engine.cpp	484
8.97 /home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Client/include/R-Type-Client/LobbyScene.hpp File Reference	484
8.97.1 Detailed Description	486
8.98 LobbyScene.hpp	486
8.99 /home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Client/include/R-Type-Client/RTypeClient.hpp File Reference	486
8.99.1 Detailed Description	488

8.100 RTypeClient.hpp	488
8.101 /home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Client/src/rtype← Client.cpp File Reference	488
8.102 rtypeClient.cpp	489
8.103 /home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Server/include/R← TypeServer/RTypeServer.hpp File Reference	489
8.103.1 Detailed Description	490
8.104 RTypeServer.hpp	490
8.105 /home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Server/src/rtype← Server.cpp File Reference	491
8.106 rtypeServer.cpp	491
8.107 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/AGame← Client.hpp File Reference	491
8.107.1 Detailed Description	492
8.108 AGameClient.hpp	492
8.109 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/AGame← Server.hpp File Reference	493
8.109.1 Detailed Description	494
8.110 AGameServer.hpp	494
8.111 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IAudio.hpp File Reference	494
8.111.1 Detailed Description	496
8.112 IAudio.hpp	496
8.113 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IGame← Client.hpp File Reference	496
8.113.1 Detailed Description	498
8.114 IGameClient.hpp	498
8.115 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IGame← Server.hpp File Reference	499
8.115.1 Detailed Description	500
8.116 IGameServer.hpp	500
8.117 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/INetwork← Client.hpp File Reference	500
8.117.1 Detailed Description	501
8.118 INetworkClient.hpp	501
8.119 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/INetwork← Server.hpp File Reference	502
8.119.1 Detailed Description	504
8.120 INetworkServer.hpp	504
8.121 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IRenderer.hpp File Reference	504
8.121.1 Detailed Description	506
8.122 IRenderer.hpp	506
8.123 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/Protocol/← Protocol.hpp File Reference	508
8.123.1 Detailed Description	511

8.124 Protocol.hpp	511
8.125 /home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/Clock.hpp File Reference	515
8.125.1 Detailed Description	516
8.126 Clock.hpp	516
8.127 /home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/Interfaces/← IPlugin.hpp File Reference	517
8.127.1 Detailed Description	518
8.128 IPlugin.hpp	519
8.129 /home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/Logger.hpp File Reference	519
8.129.1 Detailed Description	520
8.130 Logger.hpp	520
8.131 /home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/PluginLoader.hpp File Reference	521
8.131.1 Detailed Description	523
8.131.2 Macro Definition Documentation	523
8.131.2.1 PLUGINS_EXTENSION	523
8.132 PluginLoader.hpp	523
8.133 /home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/Utils.hpp File Reference	525
8.133.1 Detailed Description	526
8.134 Utils.hpp	526
8.135 /home/masina/Projects/Epitech/rtype/modules/Utils/src/logger.cpp File Reference	527
8.136 logger.cpp	527
8.137 /home/masina/Projects/Epitech/rtype/modules/Utils/src/utils.cpp File Reference	527
8.138 utils.cpp	528
8.139 /home/masina/Projects/Epitech/rtype/README.md File Reference	529
8.140 /home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/Interfaces/← IScene.hpp File Reference	529
8.140.1 Detailed Description	530
8.141 IScene.hpp	531
8.142 /home/masina/Projects/Epitech/rtype/server/include/Server/Interfaces/IScene.hpp File Reference	531
8.142.1 Detailed Description	533
8.143 IScene.hpp	533
8.144 /home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/SceneManager.hpp File Reference	534
8.144.1 Detailed Description	535
8.145 SceneManager.hpp	535
8.146 /home/masina/Projects/Epitech/rtype/server/include/Server/SceneManager.hpp File Reference	536
8.146.1 Detailed Description	536
8.147 SceneManager.hpp	537
8.148 /home/masina/Projects/Epitech/rtype/server/include/Server/Scenes/Level_0.hpp File Reference	537

8.148.1 Detailed Description	538
8.149 Level_0.hpp	538
8.150 /home/masina/Projects/Epitech/rtype/server/include/Server/Scenes/Level_1.hpp File Reference	538
8.150.1 Detailed Description	538
8.151 Level_1.hpp	538
8.152 /home/masina/Projects/Epitech/rtype/server/include/Server/Server.hpp File Reference	539
8.152.1 Detailed Description	539
8.153 Server.hpp	540
8.154 /home/masina/Projects/Epitech/rtype/server/src/server.cpp File Reference	540
8.155 server.cpp	541
Index	543

Chapter 1

r-type

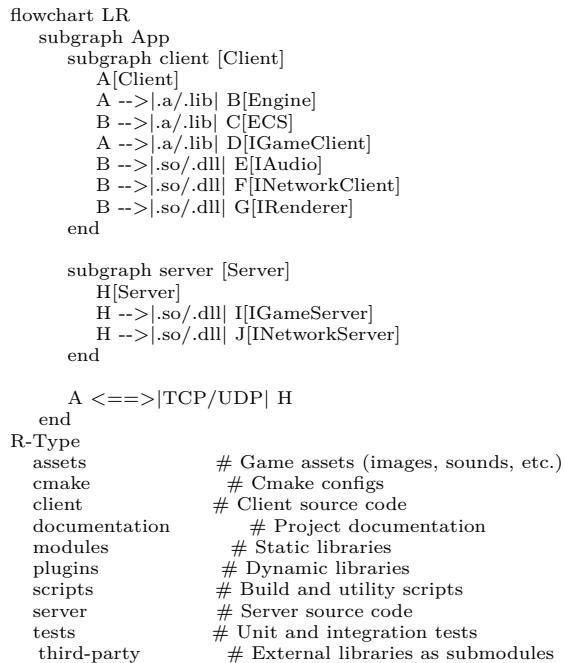
1.1 R-Type

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



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/bin/r-type_client ## client
./cmake-build-release/bin/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	17
cli::Config	18
cli::Config::Audio	18
cli::Config::Network	19
cli::Config::Window	19
cli::GameConfig	20
cli::GameConfig::Animation	20
cli::GameConfig::Asteroid	21
cli::GameConfig::Asteroid::Large	21
cli::GameConfig::Asteroid::Medium	22
cli::GameConfig::Asteroid::Small	23
cli::GameConfig::Beam	25
cli::GameConfig::Enemy	26
cli::GameConfig::Enemy::Easy	27
cli::GameConfig::Explosion	29
cli::GameConfig::Hitbox	30
cli::GameConfig::LoadingAnimation	32
cli::GameConfig::Player	33
cli::GameConfig::Projectile	34
cli::GameConfig::Projectile::Basic	34
cli::GameConfig::Projectile::Supercharged	35
cli::GameConfig::Screen	37
cli::Path	39
cli::Path::Audio	39
cli::Path::Font	39
cli::Path::Plugin	40
cli::Path::Texture	41
cli::SpriteRect	42
cli::SpriteRect::SpriteSheet	43
ecs	43
eng	44
gme	47
rnp	48
srv	53
srv::Config	55

srv::Config::Network	55
srv::Game	56
srv::Path	56
srv::Path::Plugin	56
utl	57

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

cli::ArgsConfig	72
srv::ArgsConfig	76
cli::ArgsHandler	78
srv::ArgsHandler	81
cli::SpriteRect::SpriteSheet::Asteroid	93
cli::Client	118
utl::Clock	122
eng::Color	139
cli::SpriteRect::SpriteSheet::Enemy	141
eng::Engine	150
ecs::Registry::EntityBuilder	157
rnp:: EntityState	160
cli::EnvConfig	162
srv::EnvConfig	163
eng::Event	163
rnp::EventRecord	164
rnp::FragmentHeader	177
ecs::IComponent	190
ecs::Animation	66
ecs::Asteroid	95
ecs::Audio	104
ecs::BeamCharge	111
ecs::Color	136
ecs::Enemy	143
ecs::Explosion	166
ecs::Font	175
ecs::Hitbox	184
ecs::LoadingAnimation	226
ecs::Mob	248
ecs::Pixel	260
ecs::Player	266
ecs::Projectile	280
ecs::Rect	293
ecs::Scale	313
ecs::Text	351

ecs::Texture	360
ecs::Transform	362
ecs::Velocity	364
gme::IGameClient	191
gme::AGameClient	59
gme::RTYPEClient	306
gme::IGameServer	194
gme::AGameServer	63
gme::RTYPEServer	310
utl::IPlugin	205
eng::IAudio	186
eng::INetworkClient	196
eng::IRenderer	208
srv::INetworkServer	201
ecs::Registry::IPool	206
ecs::Registry::Pool< T >	277
eng::IScene	216
eng::AScene	84
cli::Game	178
cli::Lobby	233
cli::Settings	327
gme::IScene	220
gme::LobbyScene	238
srv::IScene	222
srv::AScene	89
eng::ISystem	224
cli::AsteroidSystem	98
cli::CollisionSystem	128
cli::EnemySystem	146
cli::ExplosionSystem	169
cli::SpawnSystem	334
eng::ASystem	102
cli::AnimationSystem	69
cli::AudioSystem	106
cli::BeamSystem	113
cli::LoadingAnimationSystem	229
cli::PixelSystem	262
cli::PlayerDirectionSystem	268
cli::ProjectileSystem	288
cli::SpriteSystem	346
cli::TextSystem	355
cli::WeaponSystem	366
utl::Logger	242
rnp::PacketAck	250
rnp::PacketConnect	251
rnp::PacketConnectAccept	252
rnp::PacketDisconnect	253
rnp::PacketError	254
rnp::PacketHeader	256
rnp::PacketPingPong	258
rnp::PacketWorldState	259
utl::PluginLoader	271
cli::ProjectileManager	283
cli::SpriteRect::Rect	291
ecs::Registry	295
eng::SceneManager	315

srv::SceneManager	319
srv::Server	323
utl::SharedLib	331
gme::Sprite	342
eng::Text	353
eng::WindowSize	376

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	59
gme::AGameServer	Abstraction for the games	63
ecs::Animation	66
cli::AnimationSystem	69
cli::ArgsConfig	72
srv::ArgsConfig	76
cli::ArgsHandler	Class to handle command line arguments	78
srv::ArgsHandler	Class to handle command line arguments	81
eng::AScene	Class for scene	84
srv::AScene	Class for scene	89
cli::SpriteRect::SpriteSheet::Asteroid	93
ecs::Asteroid	95
cli::AsteroidSystem	98
eng::ASystem	102
ecs::Audio	104
cli::AudioSystem	Class for managing entities and their components	106
ecs::BeamCharge	111
cli::BeamSystem	113
cli::Client	Class for the client	118
utl::Clock	Class for clock	122
cli::CollisionSystem	128
ecs::Color	136
eng::Color	139
cli::SpriteRect::SpriteSheet::Enemy	141
ecs::Enemy	143
cli::EnemySystem	146

eng::Engine	
Class for the game engine	150
ecs::Registry::EntityBuilder	
.	157
rnp:: EntityState	
Entity state for WORLD_STATE packet	160
cli::EnvConfig	
.	162
srv::EnvConfig	
.	163
eng::Event	
.	163
rnp::EventRecord	
Event record for ENTITY_EVENT packets (TLV format)	164
ecs::Explosion	
.	166
cli::ExplosionSystem	
.	169
ecs::Font	
.	175
rnp::FragmentHeader	
Fragmentation header (when FRAG flag is set)	177
cli::Game	
Game scene	178
ecs::Hitbox	
.	184
eng::IAudio	
Interface for the audio	186
ecs::IComponent	
.	190
gme::IGameClient	
Interface for the games	191
gme::IGameServer	
Interface for the games	194
eng::INetworkClient	
Interface for the client network	196
srv::INetworkServer	
Interface for the server network	201
utl::IPPlugin	
Interface for plugins	205
ecs::Registry::IPool	
.	206
eng::IRenderer	
Interface for the renderer	208
eng::IScene	
Interface class for scene	216
gme::IScene	
Interface for scenes	220
srv::IScene	
Interface class for scene	222
eng::ISystem	
.	224
ecs::LoadingAnimation	
.	226
cli::LoadingAnimationSystem	
.	229
cli::Lobby	
Lobby scene	233
gme::LobbyScene	
Class for the Lobby scene	238
utl::Logger	
Class for logging	242
ecs::Mob	
.	248
rnp::PacketAck	
ACK packet payload	250
rnp::PacketConnect	
CONNECT packet payload	251
rnp::PacketConnectAccept	
CONNECT_ACCEPT packet payload	252
rnp::PacketDisconnect	
DISCONNECT packet payload	253

rnp::PacketError	ERROR packet payload	254
rnp::PacketHeader	Packet header according to RNP specification (Big Endian) Total size: 16 bytes	256
rnp::PacketPingPong	PING/PONG packet payload	258
rnp::PacketWorldState	WORLD_STATE packet payload	259
ecs::Pixel	260
cli::PixelSystem	262
ecs::Player	266
cli::PlayerDirectionSystem	268
utl::PluginLoader	Modern, type-safe plugin loader	271
ecs::Registry::Pool< T >	277
ecs::Projectile	280
cli::ProjectileManager	Handles projectile creation and management	283
cli::ProjectileSystem	288
cli::SpriteRect::Rect	291
ecs::Rect	293
ecs::Registry	Class for managing entities and their components	295
gme::RTypeClient	Class for the R-Type game	306
gme::RTypeServer	Class for the R-Type game	310
ecs::Scale	313
eng::SceneManager	Class for managing scenes	315
srv::SceneManager	Class for managing scenes	319
srv::Server	Class for the server	323
cli::Settings	Settings scene	327
utl::SharedLib	Handle to a dynamic library with RAII	331
cli::SpawnSystem	334
gme::Sprite	342
cli::SpriteSystem	346
ecs::Text	351
eng::Text	353
cli::TextSystem	Class for managing entities and their components	355
ecs::Texture	360
ecs::Transform	362
ecs::Velocity	364
cli::WeaponSystem	Manages weapon firing and charging	366
eng::WindowSize	376

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	377
This file contains the ArgsHandler class declaration	
/home/masina/Projects/Epitech/rtype/client/include/Client/Client.hpp	381
This file contains the Client class declaration	
/home/masina/Projects/Epitech/rtype/client/include/Client/Common.hpp	383
This file contains common definitions and constants	
/home/masina/Projects/Epitech/rtype/client/include/Client/GameConfig.hpp	388
Configuration constants for the game	
/home/masina/Projects/Epitech/rtype/client/include/Client/ProjectileManager.hpp	397
Manages projectile creation and configuration	
/home/masina/Projects/Epitech/rtype/client/include/Client/SpriteRect.hpp	405
Helper functions for pixel-perfect sprite rectangles	
/home/masina/Projects/Epitech/rtype/client/include/Client/Generated/Version.hpp	393
/home/masina/Projects/Epitech/rtype/client/include/Client/Scenes/Game.hpp	399
This file contains the Game scene	
/home/masina/Projects/Epitech/rtype/client/include/Client/Scenes/Lobby.hpp	401
This file contains the lobby scene	
/home/masina/Projects/Epitech/rtype/client/include/Client/Scenes/Settings.hpp	403
This file contains the settings scene	
/home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Animation.hpp	406
/home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Asteroid.hpp	408
/home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Audio.hpp	411
/home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Beam.hpp	413
/home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Collision.hpp	416
/home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Enemy.hpp	419
/home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Explosion.hpp	422
/home/masina/Projects/Epitech/rtype/client/include/Client/Systems>LoadingAnimation.hpp	424
/home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Pixel.hpp	426
/home/masina/Projects/Epitech/rtype/client/include/Client/Systems/PlayerDirection.hpp	428
/home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Projectile.hpp	431
/home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Spawn.hpp	433
/home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Sprite.hpp	435
/home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Systems.hpp	437
This file contains the system definitions	
/home/masina/Projects/Epitech/rtype/client/include/Client/Text.hpp	440

/home/masina/Projects/Epitech/rtype/client/include/Client/Systems/ Weapon.hpp	442
/home/masina/Projects/Epitech/rtype/client/src/ argsHandler.cpp	444
/home/masina/Projects/Epitech/rtype/client/src/ client.cpp	450
/home/masina/Projects/Epitech/rtype/client/src/ event.cpp	452
/home/masina/Projects/Epitech/rtype/client/src/ main.cpp	453
/home/masina/Projects/Epitech/rtype/client/src/ ProjectileManager.cpp Implementation of ProjectileManager	456
/home/masina/Projects/Epitech/rtype/client/src/scenes/ game.cpp	457
/home/masina/Projects/Epitech/rtype/client/src/scenes/ lobby.cpp	463
/home/masina/Projects/Epitech/rtype/client/src/scenes/ settings.cpp	467
/home/masina/Projects/Epitech/rtype/client/src/systems/ weapon.cpp	469
/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/ Component.hpp This file contains the component definitions	472
/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/ Entity.hpp This file contains the entity definitions	475
/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/ Registry.hpp This file contains the Registry class declaration	478
/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Interfaces/ ISystems.hpp	476
/home/masina/Projects/Epitech/rtype/modules/ECS/src/ registry.cpp	481
/home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/ Engine.hpp This file contains the Engine class declaration	481
/home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/ SceneManager.hpp This file contains the SceneManager class declaration	534
/home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/Interfaces/ IScene.hpp This file contains the IScene class	529
/home/masina/Projects/Epitech/rtype/modules/Engine/src/ Engine.cpp	484
/home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Client/include/R-Type↔ Client/ LobbyScene.hpp This file contains the lobby scene	484
/home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Client/include/R-Type↔ Client/ RTTypeClient.hpp RTType client class declaration	486
/home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Client/src/ rtypeClient.cpp	488
/home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Server/include/R-Type↔ Server/ RTTypeServer.hpp RTType client class declaration	489
/home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Server/src/ rtypeServer.cpp	491
/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/ AGameClient.hpp This file contains the game abstract class	491
/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/ AGameServer.hpp This file contains the game abstract class	493
/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/ IAudio.hpp This file contains the Audio interface	494
/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/ IGameClient.hpp This file contains the Game interface	496
/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/ IGameServer.hpp This file contains the Game interface	499
/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/ INetworkClient.hpp This file contains the client network interface	500
/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/ INetworkServer.hpp This file contains the server network interface	502
/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/ IRenderer.hpp This file contains the IRenderer class declaration	504
/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/Protocol/ Protocol.hpp This file contains the network protocol	508
/home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/ Clock.hpp This file contains the Clock class	515

/home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/ Logger.hpp	519
This file contains the Logger class	
/home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/ PluginLoader.hpp	521
Modern, cross-platform plugin loader	
/home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/ Utils.hpp	525
This file contains utility functions	
/home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/ Interfaces/IPlugin.hpp	517
This file contains the plugin interface	
/home/masina/Projects/Epitech/rtype/modules/src/logger.cpp	527
/home/masina/Projects/Epitech/rtype/modules/src/utils.cpp	527
/home/masina/Projects/Epitech/rtype/server/include/Server/ ArgsHandler.hpp	379
This file contains the ArgsHandler class declaration	
/home/masina/Projects/Epitech/rtype/server/include/Server/ Common.hpp	386
This file contains common definitions and constants	
/home/masina/Projects/Epitech/rtype/server/include/Server/ SceneManager.hpp	536
This file contains the SceneManager class declaration	
/home/masina/Projects/Epitech/rtype/server/include/Server/ Server.hpp	539
This file contains the Server class declaration	
/home/masina/Projects/Epitech/rtype/server/include/Server/Generated/ Version.hpp	395
/home/masina/Projects/Epitech/rtype/server/include/Server/ Interfaces/IScene.hpp	531
This file contains the IScene class	
/home/masina/Projects/Epitech/rtype/server/include/Server/Scenes/ Level_0.hpp	537
This file contains the level_0 definitions	
/home/masina/Projects/Epitech/rtype/server/include/Server/Scenes/ Level_1.hpp	538
This file contains the level_0 definitions	
/home/masina/Projects/Epitech/rtype/server/include/Server/ Systems.hpp	439
This file contains the system definitions	
/home/masina/Projects/Epitech/rtype/server/src/ argsHandler.cpp	448
/home/masina/Projects/Epitech/rtype/server/src/ main.cpp	454
/home/masina/Projects/Epitech/rtype/server/src/ server.cpp	540

Chapter 6

Namespace Documentation

6.1 cli Namespace Reference

Namespaces

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

Classes

- class [AnimationSystem](#)
- struct [ArgsConfig](#)
- class [ArgsHandler](#)

Class to handle command line arguments.
- class [AsteroidSystem](#)
- class [AudioSystem](#)

Class for managing entities and their components.
- class [BeamSystem](#)
- class [Client](#)

Class for the client.
- class [CollisionSystem](#)
- class [EnemySystem](#)
- struct [EnvConfig](#)
- class [ExplosionSystem](#)
- class [Game](#)

[Game](#) scene.
- class [LoadingAnimationSystem](#)
- class [Lobby](#)

[Lobby](#) scene.
- class [PixelSystem](#)
- class [PlayerDirectionSystem](#)
- class [ProjectileManager](#)

Handles projectile creation and management.
- class [ProjectileSystem](#)

- class [Settings](#)
 [Settings](#) scene.
- class [SpawnSystem](#)
- class [SpriteSystem](#)
- class [TextSystem](#)
 Class for managing entities and their components.
- class [WeaponSystem](#)
 Manages weapon firing and charging.

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 [Network](#)
- 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 25 of file [Common.hpp](#).

6.3.1.2 DEFAULT_AUDIO_VOLUME

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

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

6.4 cli::Config::Network Namespace Reference

Variables

- constexpr auto **DEFAULT_NETWORK_HOST** = "127.0.0.1"
- constexpr auto **DEFAULT_NETWORK_PORT** = 2560

6.4.1 Variable Documentation

6.4.1.1 DEFAULT_NETWORK_HOST

auto cli::Config::Network::DEFAULT_NETWORK_HOST = "127.0.0.1" [inline], [constexpr]

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

6.4.1.2 DEFAULT_NETWORK_PORT

auto cli::Config::Network::DEFAULT_NETWORK_PORT = 2560 [inline], [constexpr]

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

6.5 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

6.5.1 Variable Documentation

6.5.1.1 DEFAULT_WINDOW_FRAME_LIMIT

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

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

6.5.1.2 DEFAULT_WINDOW_FULLSCREEN

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

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

6.5.1.3 DEFAULT_WINDOW_HEIGHT

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

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

6.5.1.4 DEFAULT_WINDOW_WIDTH

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

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

6.6 cli::GameConfig Namespace Reference

Namespaces

- namespace [Animation](#)
- namespace [Asteroid](#)
- namespace [Beam](#)
- namespace [Enemy](#)
- namespace [Explosion](#)
- namespace [Hitbox](#)
- namespace [LoadingAnimation](#)
- namespace [Player](#)
- namespace [Projectile](#)
- namespace [Screen](#)

6.7 cli::GameConfig::Animation Namespace Reference

Variables

- constexpr float [FRAME_DURATION](#) = 0.1f

6.7.1 Variable Documentation

6.7.1.1 FRAME_DURATION

```
float cli::GameConfig::Animation::FRAME_DURATION = 0.1f [inline], [constexpr]
```

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

6.8 cli::GameConfig::Asteroid Namespace Reference

Namespaces

- namespace **Large**
- namespace **Medium**
- namespace **Small**

6.9 cli::GameConfig::Asteroid::Large Namespace Reference

Variables

- constexpr float **HEALTH** = 80.0f
- constexpr float **SPEED** = 40.0f
- constexpr float **SPRITE_WIDTH** = 64.0f
- constexpr float **SPRITE_HEIGHT** = 64.0f
- constexpr float **SCALE** = 1.5f
- constexpr float **ROTATION_SPEED** = 30.0f
- constexpr float **SPAWN_RATE** = 4.0f

6.9.1 Variable Documentation

6.9.1.1 HEALTH

```
float cli::GameConfig::Asteroid::Large::HEALTH = 80.0f [inline], [constexpr]
```

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

6.9.1.2 ROTATION_SPEED

```
float cli::GameConfig::Asteroid::Large::ROTATION_SPEED = 30.0f [inline], [constexpr]
```

Definition at line 144 of file [GameConfig.hpp](#).

6.9.1.3 SCALE

```
float cli::GameConfig::Asteroid::Large::SCALE = 1.5f [inline], [constexpr]
```

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

6.9.1.4 SPAWN_RATE

```
float cli::GameConfig::Asteroid::Large::SPAWN_RATE = 4.0f [inline], [constexpr]
```

Definition at line 145 of file [GameConfig.hpp](#).

6.9.1.5 SPEED

```
float cli::GameConfig::Asteroid::Large::SPEED = 40.0f [inline], [constexpr]
```

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

6.9.1.6 SPRITE_HEIGHT

```
float cli::GameConfig::Asteroid::Large::SPRITE_HEIGHT = 64.0f [inline], [constexpr]
```

Definition at line 142 of file [GameConfig.hpp](#).

6.9.1.7 SPRITE_WIDTH

```
float cli::GameConfig::Asteroid::Large::SPRITE_WIDTH = 64.0f [inline], [constexpr]
```

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

6.10 cli::GameConfig::Asteroid::Medium Namespace Reference

Variables

- constexpr float **HEALTH** = 40.0f
- constexpr float **SPEED** = 60.0f
- constexpr float **SPRITE_WIDTH** = 32.0f
- constexpr float **SPRITE_HEIGHT** = 32.0f
- constexpr float **SCALE** = 1.0f
- constexpr float **ROTATION_SPEED** = 60.0f
- constexpr float **SPAWN_RATE** = 2.0f

6.10.1 Variable Documentation

6.10.1.1 HEALTH

```
float cli::GameConfig::Asteroid::Medium::HEALTH = 40.0f [inline], [constexpr]
```

Definition at line 128 of file [GameConfig.hpp](#).

6.10.1.2 ROTATION_SPEED

```
float cli::GameConfig::Asteroid::Medium::ROTATION_SPEED = 60.0f [inline], [constexpr]
```

Definition at line 133 of file [GameConfig.hpp](#).

6.10.1.3 SCALE

```
float cli::GameConfig::Asteroid::Medium::SCALE = 1.0f [inline], [constexpr]
```

Definition at line 132 of file [GameConfig.hpp](#).

6.10.1.4 SPAWN_RATE

```
float cli::GameConfig::Asteroid::Medium::SPAWN_RATE = 2.0f [inline], [constexpr]
```

Definition at line 134 of file [GameConfig.hpp](#).

6.10.1.5 SPEED

```
float cli::GameConfig::Asteroid::Medium::SPEED = 60.0f [inline], [constexpr]
```

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

6.10.1.6 SPRITE_HEIGHT

```
float cli::GameConfig::Asteroid::Medium::SPRITE_HEIGHT = 32.0f [inline], [constexpr]
```

Definition at line 131 of file [GameConfig.hpp](#).

6.10.1.7 SPRITE_WIDTH

```
float cli::GameConfig::Asteroid::Medium::SPRITE_WIDTH = 32.0f [inline], [constexpr]
```

Definition at line 130 of file [GameConfig.hpp](#).

6.11 cli::GameConfig::Asteroid::Small Namespace Reference

Variables

- constexpr float **HEALTH** = 20.0f
- constexpr float **SPEED** = 80.0f
- constexpr float **SPRITE_WIDTH** = 18.0f
- constexpr float **SPRITE_HEIGHT** = 18.0f
- constexpr float **SCALE** = 2.0f
- constexpr float **ROTATION_SPEED** = 90.0f
- constexpr float **SPAWN_RATE** = 1.0f
- constexpr int **ANIMATION_FRAMES** = 11
- constexpr float **ANIMATION_DURATION** = 0.5f
- constexpr int **FRAMES_PER_ROW** = 11

6.11.1 Variable Documentation

6.11.1.1 ANIMATION_DURATION

```
float cli::GameConfig::Asteroid::Small::ANIMATION_DURATION = 0.5f [inline], [constexpr]
```

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

Referenced by [cli::SpawnSystem::spawnAsteroid\(\)](#).

6.11.1.2 ANIMATION_FRAMES

```
int cli::GameConfig::Asteroid::Small::ANIMATION_FRAMES = 11 [inline], [constexpr]
```

Definition at line 121 of file [GameConfig.hpp](#).

Referenced by [cli::SpawnSystem::spawnAsteroid\(\)](#).

6.11.1.3 FRAMES_PER_ROW

```
int cli::GameConfig::Asteroid::Small::FRAMES_PER_ROW = 11 [inline], [constexpr]
```

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

Referenced by [cli::SpawnSystem::spawnAsteroid\(\)](#).

6.11.1.4 HEALTH

```
float cli::GameConfig::Asteroid::Small::HEALTH = 20.0f [inline], [constexpr]
```

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

Referenced by [cli::SpawnSystem::spawnAsteroid\(\)](#).

6.11.1.5 ROTATION_SPEED

```
float cli::GameConfig::Asteroid::Small::ROTATION_SPEED = 90.0f [inline], [constexpr]
```

Definition at line 119 of file [GameConfig.hpp](#).

Referenced by [cli::SpawnSystem::spawnAsteroid\(\)](#).

6.11.1.6 SCALE

```
float cli::GameConfig::Asteroid::Small::SCALE = 2.0f [inline], [constexpr]
```

Definition at line 118 of file [GameConfig.hpp](#).

Referenced by [cli::SpawnSystem::spawnAsteroid\(\)](#).

6.11.1.7 SPAWN_RATE

float cli::GameConfig::Asteroid::Small::SPAWN_RATE = 1.0f [inline], [constexpr]

Definition at line 120 of file [GameConfig.hpp](#).

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

6.11.1.8 SPEED

float cli::GameConfig::Asteroid::Small::SPEED = 80.0f [inline], [constexpr]

Definition at line 115 of file [GameConfig.hpp](#).

Referenced by [cli::SpawnSystem::spawnAsteroid\(\)](#).

6.11.1.9 SPRITE_HEIGHT

float cli::GameConfig::Asteroid::Small::SPRITE_HEIGHT = 18.0f [inline], [constexpr]

Definition at line 117 of file [GameConfig.hpp](#).

Referenced by [cli::SpawnSystem::spawnAsteroid\(\)](#).

6.11.1.10 SPRITE_WIDTH

float cli::GameConfig::Asteroid::Small::SPRITE_WIDTH = 18.0f [inline], [constexpr]

Definition at line 116 of file [GameConfig.hpp](#).

Referenced by [cli::SpawnSystem::spawnAsteroid\(\)](#), and [cli::AsteroidSystem::update\(\)](#).

6.12 cli::GameConfig::Beam Namespace Reference

Variables

- constexpr float **MAX_CHARGE** = 1.0f
- constexpr float **CHARGE_RATE** = 1.0f
- constexpr float **BAR_WIDTH** = 120.0f
- constexpr float **BAR_HEIGHT** = 12.0f
- constexpr float **BAR_X** = 10.0f
- constexpr float **BAR_Y** = 10.0f

6.12.1 Variable Documentation

6.12.1.1 BAR_HEIGHT

float cli::GameConfig::Beam::BAR_HEIGHT = 12.0f [inline], [constexpr]

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

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

6.12.1.2 BAR_WIDTH

```
float cli::GameConfig::Beam::BAR_WIDTH = 120.0f [inline], [constexpr]
```

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

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

6.12.1.3 BAR_X

```
float cli::GameConfig::Beam::BAR_X = 10.0f [inline], [constexpr]
```

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

6.12.1.4 BAR_Y

```
float cli::GameConfig::Beam::BAR_Y = 10.0f [inline], [constexpr]
```

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

6.12.1.5 CHARGE_RATE

```
float cli::GameConfig::Beam::CHARGE_RATE = 1.0f [inline], [constexpr]
```

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

6.12.1.6 MAX_CHARGE

```
float cli::GameConfig::Beam::MAX_CHARGE = 1.0f [inline], [constexpr]
```

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

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

6.13 cli::GameConfig::Enemy Namespace Reference

Namespaces

- namespace [Easy](#)

6.14 cli::GameConfig::Enemy::Easy Namespace Reference

Variables

- `constexpr float HEALTH = 1.0f`
- `constexpr float DAMAGE = 5.0f`
- `constexpr float SPEED = 80.0f`
- `constexpr float SPRITE_WIDTH = 32.0f`
- `constexpr float SPRITE_HEIGHT = 32.0f`
- `constexpr float SCALE = 2.0f`
- `constexpr float SHOOT_COOLDOWN = 2.0f`
- `constexpr float SPAWN_RATE = 2.0f`
- `constexpr int ANIMATION_FRAMES = 4`
- `constexpr float ANIMATION_DURATION = 0.5f`
- `constexpr int FRAMES_PER_ROW = 4`

6.14.1 Variable Documentation

6.14.1.1 ANIMATION_DURATION

```
float cli::GameConfig::Enemy::Easy::ANIMATION_DURATION = 0.5f [inline], [constexpr]
```

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

Referenced by [cli::SpawnSystem::spawnEnemy\(\)](#), and [cli::SpawnSystem::spawnWave\(\)](#).

6.14.1.2 ANIMATION_FRAMES

```
int cli::GameConfig::Enemy::Easy::ANIMATION_FRAMES = 4 [inline], [constexpr]
```

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

Referenced by [cli::SpawnSystem::spawnEnemy\(\)](#), and [cli::SpawnSystem::spawnWave\(\)](#).

6.14.1.3 DAMAGE

```
float cli::GameConfig::Enemy::Easy::DAMAGE = 5.0f [inline], [constexpr]
```

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

Referenced by [cli::SpawnSystem::spawnEnemy\(\)](#), and [cli::SpawnSystem::spawnWave\(\)](#).

6.14.1.4 FRAMES_PER_ROW

```
int cli::GameConfig::Enemy::Easy::FRAMES_PER_ROW = 4 [inline], [constexpr]
```

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

Referenced by [cli::SpawnSystem::spawnEnemy\(\)](#), and [cli::SpawnSystem::spawnWave\(\)](#).

6.14.1.5 HEALTH

```
float cli::GameConfig::Enemy::Easy::HEALTH = 1.0f [inline], [constexpr]
```

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

Referenced by [cli::SpawnSystem::spawnEnemy\(\)](#), and [cli::SpawnSystem::spawnWave\(\)](#).

6.14.1.6 SCALE

```
float cli::GameConfig::Enemy::Easy::SCALE = 2.0f [inline], [constexpr]
```

Definition at line 92 of file [GameConfig.hpp](#).

Referenced by [cli::SpawnSystem::spawnEnemy\(\)](#), and [cli::SpawnSystem::spawnWave\(\)](#).

6.14.1.7 SHOOT_COOLDOWN

```
float cli::GameConfig::Enemy::Easy::SHOOT_COOLDOWN = 2.0f [inline], [constexpr]
```

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

Referenced by [cli::SpawnSystem::spawnEnemy\(\)](#), and [cli::SpawnSystem::spawnWave\(\)](#).

6.14.1.8 SPAWN_RATE

```
float cli::GameConfig::Enemy::Easy::SPAWN_RATE = 2.0f [inline], [constexpr]
```

Definition at line 94 of file [GameConfig.hpp](#).

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

6.14.1.9 SPEED

```
float cli::GameConfig::Enemy::Easy::SPEED = 80.0f [inline], [constexpr]
```

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

Referenced by [cli::SpawnSystem::spawnEnemy\(\)](#), and [cli::SpawnSystem::spawnWave\(\)](#).

6.14.1.10 SPRITE_HEIGHT

```
float cli::GameConfig::Enemy::Easy::SPRITE_HEIGHT = 32.0f [inline], [constexpr]
```

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

Referenced by [cli::SpawnSystem::spawnEnemy\(\)](#), and [cli::SpawnSystem::spawnWave\(\)](#).

6.14.1.11 SPRITE_WIDTH

```
float cli::GameConfig::Enemy::Easy::SPRITE_WIDTH = 32.0f [inline], [constexpr]
```

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

Referenced by [cli::SpawnSystem::spawnEnemy\(\)](#), [cli::SpawnSystem::spawnWave\(\)](#), and [cli::EnemySystem::update\(\)](#).

6.15 cli::GameConfig::Explosion Namespace Reference

Variables

- `constexpr float SPRITE_WIDTH = 32.0f`
- `constexpr float SPRITE_HEIGHT = 32.0f`
- `constexpr int ANIMATION_FRAMES = 4`
- `constexpr float ANIMATION_DURATION = 0.1f`
- `constexpr int FRAMES_PER_ROW = 4`
- `constexpr float LIFETIME = 0.4f`
- `constexpr float SCALE = 2.0f`

6.15.1 Variable Documentation

6.15.1.1 ANIMATION_DURATION

```
float cli::GameConfig::Explosion::ANIMATION_DURATION = 0.1f [inline], [constexpr]
```

Definition at line 105 of file [GameConfig.hpp](#).

Referenced by [cli::CollisionSystem::createExplosion\(\)](#).

6.15.1.2 ANIMATION_FRAMES

```
int cli::GameConfig::Explosion::ANIMATION_FRAMES = 4 [inline], [constexpr]
```

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

Referenced by [cli::CollisionSystem::createExplosion\(\)](#).

6.15.1.3 FRAMES_PER_ROW

```
int cli::GameConfig::Explosion::FRAMES_PER_ROW = 4 [inline], [constexpr]
```

Definition at line 106 of file [GameConfig.hpp](#).

Referenced by [cli::CollisionSystem::createExplosion\(\)](#).

6.15.1.4 LIFETIME

```
float cli::GameConfig::Explosion::LIFETIME = 0.4f [inline], [constexpr]
```

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

Referenced by [cli::CollisionSystem::createExplosion\(\)](#).

6.15.1.5 SCALE

```
float cli::GameConfig::Explosion::SCALE = 2.0f [inline], [constexpr]
```

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

Referenced by [cli::CollisionSystem::createExplosion\(\)](#).

6.15.1.6 SPRITE_HEIGHT

```
float cli::GameConfig::Explosion::SPRITE_HEIGHT = 32.0f [inline], [constexpr]
```

Definition at line 103 of file [GameConfig.hpp](#).

Referenced by [cli::CollisionSystem::createExplosion\(\)](#).

6.15.1.7 SPRITE_WIDTH

```
float cli::GameConfig::Explosion::SPRITE_WIDTH = 32.0f [inline], [constexpr]
```

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

Referenced by [cli::CollisionSystem::createExplosion\(\)](#).

6.16 cli::GameConfig::Hitbox Namespace Reference

Variables

- `constexpr float PLAYER_RADIUS = 20.0f`
- `constexpr float ENEMY_RADIUS = 15.0f`
- `constexpr float PROJECTILE_BASIC_RADIUS = 5.0f`
- `constexpr float PROJECTILE_SUPERCHARGED_RADIUS = 8.0f`
- `constexpr float ASTEROID_SMALL_RADIUS = 25.0f`
- `constexpr float ASTEROID_MEDIUM_RADIUS = 40.0f`
- `constexpr float ASTEROID_LARGE_RADIUS = 60.0f`

6.16.1 Variable Documentation

6.16.1.1 ASTEROID_LARGE_RADIUS

```
float cli::GameConfig::Hitbox::ASTEROID_LARGE_RADIUS = 60.0f [inline], [constexpr]
```

Definition at line 156 of file [GameConfig.hpp](#).

6.16.1.2 ASTEROID_MEDIUM_RADIUS

```
float cli::GameConfig::Hitbox::ASTEROID_MEDIUM_RADIUS = 40.0f [inline], [constexpr]
```

Definition at line 155 of file [GameConfig.hpp](#).

6.16.1.3 ASTEROID_SMALL_RADIUS

```
float cli::GameConfig::Hitbox::ASTEROID_SMALL_RADIUS = 25.0f [inline], [constexpr]
```

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

Referenced by [cli::SpawnSystem::spawnAsteroid\(\)](#).

6.16.1.4 ENEMY_RADIUS

```
float cli::GameConfig::Hitbox::ENEMY_RADIUS = 15.0f [inline], [constexpr]
```

Definition at line 151 of file [GameConfig.hpp](#).

Referenced by [cli::SpawnSystem::spawnEnemy\(\)](#), and [cli::SpawnSystem::spawnWave\(\)](#).

6.16.1.5 PLAYER_RADIUS

```
float cli::GameConfig::Hitbox::PLAYER_RADIUS = 20.0f [inline], [constexpr]
```

Definition at line 150 of file [GameConfig.hpp](#).

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

6.16.1.6 PROJECTILE_BASIC_RADIUS

```
float cli::GameConfig::Hitbox::PROJECTILE_BASIC_RADIUS = 5.0f [inline], [constexpr]
```

Definition at line 152 of file [GameConfig.hpp](#).

Referenced by [cli::ProjectileManager::createProjectile\(\)](#).

6.16.1.7 PROJECTILE_SUPERCHARGED_RADIUS

```
float cli::GameConfig::Hitbox::PROJECTILE_SUPERCHARGED_RADIUS = 8.0f [inline], [constexpr]
```

Definition at line 153 of file [GameConfig.hpp](#).

Referenced by [cli::ProjectileManager::createProjectile\(\)](#).

6.17 cli::GameConfig::LoadingAnimation Namespace Reference

Variables

- constexpr float `SPRITE_WIDTH` = 29.0f
- constexpr float `SPRITE_HEIGHT` = 24.0f
- constexpr int `ANIMATION_FRAMES` = 4
- constexpr float `ANIMATION_DURATION` = 0.15f
- constexpr float `OFFSET_X` = 40.0f
- constexpr float `OFFSET_Y` = 0.0f

6.17.1 Variable Documentation

6.17.1.1 ANIMATION_DURATION

float cli::GameConfig::LoadingAnimation::ANIMATION_DURATION = 0.15f [inline], [constexpr]

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

6.17.1.2 ANIMATION_FRAMES

int cli::GameConfig::LoadingAnimation::ANIMATION_FRAMES = 4 [inline], [constexpr]

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

6.17.1.3 OFFSET_X

float cli::GameConfig::LoadingAnimation::OFFSET_X = 40.0f [inline], [constexpr]

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

6.17.1.4 OFFSET_Y

float cli::GameConfig::LoadingAnimation::OFFSET_Y = 0.0f [inline], [constexpr]

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

6.17.1.5 SPRITE_HEIGHT

float cli::GameConfig::LoadingAnimation::SPRITE_HEIGHT = 24.0f [inline], [constexpr]

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

6.17.1.6 SPRITE_WIDTH

float cli::GameConfig::LoadingAnimation::SPRITE_WIDTH = 29.0f [inline], [constexpr]

Definition at line 76 of file [GameConfig.hpp](#).

6.18 cli::GameConfig::Player Namespace Reference

Variables

- constexpr float **SPEED** = 500.0f
- constexpr float **DIAGONAL_SPEED_MULTIPLIER** = 0.707f
- constexpr float **SPRITE_WIDTH** = 33.0f
- constexpr float **SPRITE_HEIGHT** = 17.0f
- constexpr float **SCALE** = 2.0f
- constexpr int **FRAMES_PER_ROW** = 5
- constexpr int **TOTAL_FRAMES** = 5

6.18.1 Variable Documentation

6.18.1.1 **DIAGONAL_SPEED_MULTIPLIER**

```
float cli::GameConfig::Player::DIAGONAL_SPEED_MULTIPLIER = 0.707f [inline], [constexpr]
```

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

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

6.18.1.2 **FRAMES_PER_ROW**

```
int cli::GameConfig::Player::FRAMES_PER_ROW = 5 [inline], [constexpr]
```

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

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

6.18.1.3 **SCALE**

```
float cli::GameConfig::Player::SCALE = 2.0f [inline], [constexpr]
```

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

Referenced by [cli::Game::Game\(\)](#), and [cli::Game::update\(\)](#).

6.18.1.4 **SPEED**

```
float cli::GameConfig::Player::SPEED = 500.0f [inline], [constexpr]
```

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

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

6.18.1.5 SPRITE_HEIGHT

```
float cli::GameConfig::Player::SPRITE_HEIGHT = 17.0f [inline], [constexpr]
```

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

Referenced by [cli::Game::Game\(\)](#), [cli::Game::update\(\)](#), [cli::PlayerDirectionSystem::update\(\)](#), and [cli::WeaponSystem::update\(\)](#).

6.18.1.6 SPRITE_WIDTH

```
float cli::GameConfig::Player::SPRITE_WIDTH = 33.0f [inline], [constexpr]
```

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

Referenced by [cli::Game::Game\(\)](#), [cli::BeamSystem::update\(\)](#), [cli::Game::update\(\)](#), [cli::PlayerDirectionSystem::update\(\)](#), and [cli::WeaponSystem::update\(\)](#).

6.18.1.7 TOTAL_FRAMES

```
int cli::GameConfig::Player::TOTAL_FRAMES = 5 [inline], [constexpr]
```

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

6.19 cli::GameConfig::Projectile Namespace Reference

Namespaces

- namespace [Basic](#)
- namespace [Supercharged](#)

6.20 cli::GameConfig::Projectile::Basic Namespace Reference

Variables

- constexpr float [SPEED](#) = 800.0f
- constexpr float [DAMAGE](#) = 10.0f
- constexpr float [LIFETIME](#) = 3.0f
- constexpr float [SCALE](#) = 1.0f
- constexpr float [FIRE_COOLDOWN](#) = 0.3f
- constexpr float [SPRITE_WIDTH](#) = 16.0f
- constexpr float [SPRITE_HEIGHT](#) = 8.0f

6.20.1 Variable Documentation

6.20.1.1 DAMAGE

```
float cli::GameConfig::Projectile::Basic::DAMAGE = 10.0f [inline], [constexpr]
```

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

6.20.1.2 FIRE_COOLDOWN

```
float cli::GameConfig::Projectile::Basic::FIRE_COOLDOWN = 0.3f [inline], [constexpr]
```

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

6.20.1.3 LIFETIME

```
float cli::GameConfig::Projectile::Basic::LIFETIME = 3.0f [inline], [constexpr]
```

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

6.20.1.4 SCALE

```
float cli::GameConfig::Projectile::Basic::SCALE = 1.0f [inline], [constexpr]
```

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

6.20.1.5 SPEED

```
float cli::GameConfig::Projectile::Basic::SPEED = 800.0f [inline], [constexpr]
```

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

6.20.1.6 SPRITE_HEIGHT

```
float cli::GameConfig::Projectile::Basic::SPRITE_HEIGHT = 8.0f [inline], [constexpr]
```

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

6.20.1.7 SPRITE_WIDTH

```
float cli::GameConfig::Projectile::Basic::SPRITE_WIDTH = 16.0f [inline], [constexpr]
```

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

6.21 cli::GameConfig::Projectile::Supercharged Namespace Reference

Variables

- constexpr float **SPEED** = 1200.0f
- constexpr float **DAMAGE** = 25.0f
- constexpr float **LIFETIME** = 5.0f
- constexpr float **SCALE** = 1.5f
- constexpr float **FIRE_COOLDOWN** = 0.2f
- constexpr float **CHARGE_TIME** = 0.5f
- constexpr float **SPRITE_WIDTH** = 29.0f
- constexpr float **SPRITE_HEIGHT** = 24.0f
- constexpr int **ANIMATION_FRAMES** = 4
- constexpr float **ANIMATION_DURATION** = 0.15f

6.21.1 Variable Documentation

6.21.1.1 ANIMATION_DURATION

```
float cli::GameConfig::Projectile::Supercharged::ANIMATION_DURATION = 0.15f [inline], [constexpr]
```

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

6.21.1.2 ANIMATION_FRAMES

```
int cli::GameConfig::Projectile::Supercharged::ANIMATION_FRAMES = 4 [inline], [constexpr]
```

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

6.21.1.3 CHARGE_TIME

```
float cli::GameConfig::Projectile::Supercharged::CHARGE_TIME = 0.5f [inline], [constexpr]
```

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

6.21.1.4 DAMAGE

```
float cli::GameConfig::Projectile::Supercharged::DAMAGE = 25.0f [inline], [constexpr]
```

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

6.21.1.5 FIRE_COOLDOWN

```
float cli::GameConfig::Projectile::Supercharged::FIRE_COOLDOWN = 0.2f [inline], [constexpr]
```

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

6.21.1.6 LIFETIME

```
float cli::GameConfig::Projectile::Supercharged::LIFETIME = 5.0f [inline], [constexpr]
```

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

6.21.1.7 SCALE

```
float cli::GameConfig::Projectile::Supercharged::SCALE = 1.5f [inline], [constexpr]
```

Definition at line 52 of file [GameConfig.hpp](#).

6.21.1.8 SPEED

```
float cli::GameConfig::Projectile::Supercharged::SPEED = 1200.0f [inline], [constexpr]
```

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

6.21.1.9 SPRITE_HEIGHT

```
float cli::GameConfig::Projectile::Supercharged::SPRITE_HEIGHT = 24.0f [inline], [constexpr]
```

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

6.21.1.10 SPRITE_WIDTH

```
float cli::GameConfig::Projectile::Supercharged::SPRITE_WIDTH = 29.0f [inline], [constexpr]
```

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

6.22 cli::GameConfig::Screen Namespace Reference

Variables

- constexpr int **WIDTH** = 1920
- constexpr int **HEIGHT** = 1080
- constexpr float **SPAWN_X** = 1950.0f
- constexpr float **MIN_Y** = 50.0f
- constexpr float **MAX_Y** = 1030.0f
- constexpr float **REMOVE_X** = -100.0f
- constexpr float **REMOVE_MIN_Y** = -50.0f
- constexpr float **REMOVE_MAX_Y** = 1130.0f

6.22.1 Variable Documentation

6.22.1.1 HEIGHT

```
int cli::GameConfig::Screen::HEIGHT = 1080 [inline], [constexpr]
```

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

6.22.1.2 MAX_Y

```
float cli::GameConfig::Screen::MAX_Y = 1030.0f [inline], [constexpr]
```

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

Referenced by [cli::SpawnSystem::spawnAsteroid\(\)](#), [cli::SpawnSystem::spawnEnemy\(\)](#), and [cli::SpawnSystem::spawnWave\(\)](#)

6.22.1.3 MIN_Y

```
float cli::GameConfig::Screen::MIN_Y = 50.0f [inline], [constexpr]
```

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

Referenced by [cli::SpawnSystem::spawnAsteroid\(\)](#), [cli::SpawnSystem::spawnEnemy\(\)](#), and [cli::SpawnSystem::spawnWave\(\)](#).

6.22.1.4 REMOVE_MAX_Y

```
float cli::GameConfig::Screen::REMOVE_MAX_Y = 1130.0f [inline], [constexpr]
```

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

Referenced by [cli::AsteroidSystem::update\(\)](#), and [cli::EnemySystem::update\(\)](#).

6.22.1.5 REMOVE_MIN_Y

```
float cli::GameConfig::Screen::REMOVE_MIN_Y = -50.0f [inline], [constexpr]
```

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

Referenced by [cli::AsteroidSystem::update\(\)](#), and [cli::EnemySystem::update\(\)](#).

6.22.1.6 REMOVE_X

```
float cli::GameConfig::Screen::REMOVE_X = -100.0f [inline], [constexpr]
```

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

Referenced by [cli::AsteroidSystem::update\(\)](#), and [cli::EnemySystem::update\(\)](#).

6.22.1.7 SPAWN_X

```
float cli::GameConfig::Screen::SPAWN_X = 1950.0f [inline], [constexpr]
```

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

Referenced by [cli::SpawnSystem::spawnAsteroid\(\)](#), [cli::SpawnSystem::spawnEnemy\(\)](#), and [cli::SpawnSystem::spawnWave\(\)](#).

6.22.1.8 WIDTH

```
int cli::GameConfig::Screen::WIDTH = 1920 [inline], [constexpr]
```

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

6.23 cli::Path Namespace Reference

Namespaces

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

6.24 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"

6.24.1 Variable Documentation

6.24.1.1 AUDIO_BATTLE_THEME

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

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

6.24.1.2 AUDIO_COIN

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

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

6.24.1.3 AUDIO_TITLE

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

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

Referenced by [cli::Game::Game\(\)](#), [cli::Lobby::Lobby\(\)](#), and [cli::Settings::Settings\(\)](#).

6.25 cli::Path::Font Namespace Reference

Variables

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

6.25.1 Variable Documentation

6.25.1.1 FONTS_RTYPE

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

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

Referenced by [cli::Game::Game\(\)](#), [cli::Lobby::Lobby\(\)](#), and [cli::Settings::Settings\(\)](#).

6.26 cli::Path::Plugin Namespace Reference

Variables

- auto [PLUGIN_AUDIO_SFML](#)
- auto [PLUGIN_NETWORK_ASIO_CLIENT](#)
- auto [PLUGIN_RENDERER_SFML](#)

6.26.1 Variable Documentation

6.26.1.1 PLUGIN_AUDIO_SFML

```
auto cli::Path::Plugin::PLUGIN_AUDIO_SFML [inline]
```

Initial value:

```
= std::filesystem::path(PLUGINS_DIR) / ("audio_sfml" + std::string(PLUGINS_EXTENSION))
```

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

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

6.26.1.2 PLUGIN_NETWORK_ASIO_CLIENT

```
auto cli::Path::Plugin::PLUGIN_NETWORK_ASIO_CLIENT [inline]
```

Initial value:

```
= std::filesystem::path(PLUGINS_DIR) / ("network_asio_client" + std::string(PLUGINS_EXTENSION))
```

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

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

6.26.1.3 PLUGIN_RENDERER_SFML

```
auto cli::Path::Plugin::PLUGIN_RENDERER_SFML [inline]
```

Initial value:

```
= std::filesystem::path(PLUGINS_DIR) / ("renderer_sfml" + std::string(PLUGINS_EXTENSION))
```

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

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

6.27 cli::Path::Texture Namespace Reference

Variables

- `constexpr auto TEXTURE_PLAYER = "assets/sprites/r-typesheet42.gif"`
- `constexpr auto TEXTURE_SHOOT = "assets/sprites/shoot.gif"`
- `constexpr auto TEXTURE_SHOOT_CHARGED = "assets/sprites/shootcharged.gif"`
- `constexpr auto TEXTURE_SHOOT_LOADING = "assets/sprites/shootchargedloading.gif"`
- `constexpr auto TEXTURE_ENEMY_EASY = "assets/sprites/r-typesheet5.gif"`
- `constexpr auto TEXTURE_ASTEROID = "assets/sprites/r-typesheet3.gif"`
- `constexpr auto TEXTURE_EXPLOSION = "assets/sprites/r-typesheet44.gif"`

6.27.1 Variable Documentation

6.27.1.1 TEXTURE_ASTEROID

`auto cli::Path::Texture::TEXTURE_ASTEROID = "assets/sprites/r-typesheet3.gif" [inline], [constexpr]`

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

Referenced by [cli::SpawnSystem::spawnAsteroid\(\)](#).

6.27.1.2 TEXTURE_ENEMY_EASY

`auto cli::Path::Texture::TEXTURE_ENEMY_EASY = "assets/sprites/r-typesheet5.gif" [inline], [constexpr]`

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

Referenced by [cli::SpawnSystem::spawnEnemy\(\)](#), and [cli::SpawnSystem::spawnWave\(\)](#).

6.27.1.3 TEXTURE_EXPLOSION

`auto cli::Path::Texture::TEXTURE_EXPLOSION = "assets/sprites/r-typesheet44.gif" [inline], [constexpr]`

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

Referenced by [cli::CollisionSystem::createExplosion\(\)](#).

6.27.1.4 TEXTURE_PLAYER

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

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

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

6.27.1.5 TEXTURE_SHOOT

```
auto cli::Path::Texture::TEXTURE_SHOOT = "assets/sprites/shoot.gif" [inline], [constexpr]
```

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

Referenced by [cli::ProjectileManager::createProjectile\(\)](#).

6.27.1.6 TEXTURE_SHOOT_CHARGED

```
auto cli::Path::Texture::TEXTURE_SHOOT_CHARGED = "assets/sprites/shootcharged.gif" [inline], [constexpr]
```

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

Referenced by [cli::ProjectileManager::createProjectile\(\)](#).

6.27.1.7 TEXTURE_SHOOT_LOADING

```
auto cli::Path::Texture::TEXTURE_SHOOT_LOADING = "assets/sprites/shootchargedloading.gif" [inline], [constexpr]
```

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

Referenced by [cli::WeaponSystem::showLoadingAnimation\(\)](#).

6.28 cli::SpriteRect Namespace Reference

Namespaces

- namespace [SpriteSheet](#)

Classes

- struct [Rect](#)

Functions

- [Rect asteroidRect \(int col, int row=0\)](#)
- [Rect enemyRect \(int col, int row=0\)](#)

6.28.1 Function Documentation

6.28.1.1 asteroidRect()

```
Rect cli::SpriteRect:::asteroidRect (
    int col,
    int row = 0) [inline]
```

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

References [cli::SpriteRect::SpriteSheet::Asteroid::frameH](#), [cli::SpriteRect::SpriteSheet::Asteroid::frameW](#), [cli::SpriteRect::SpriteSheet::Asteroid::marginX](#), [cli::SpriteRect::SpriteSheet::Asteroid::marginY](#), [cli::SpriteRect::SpriteSheet::Asteroid::spacingY](#).
and [cli::SpriteRect::SpriteSheet::Asteroid::spacingY](#).

6.28.1.2 enemyRect()

```
Rect cli::SpriteRect::enemyRect (
    int col,
    int row = 0) [inline]
```

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

References [cli::SpriteRect::SpriteSheet::Enemy::frameH](#), [cli::SpriteRect::SpriteSheet::Enemy::frameW](#), [cli::SpriteRect::SpriteSheet::Enemy::marginX](#), [cli::SpriteRect::SpriteSheet::Enemy::marginY](#), [cli::SpriteRect::SpriteSheet::Enemy::spacingY](#). and [cli::SpriteRect::SpriteSheet::Enemy::spacingY](#).

6.29 cli::SpriteRect::SpriteSheet Namespace Reference

Classes

- [struct Asteroid](#)
- [struct Enemy](#)

6.30 ecs Namespace Reference

Classes

- [struct Animation](#)
- [struct Asteroid](#)
- [struct Audio](#)
- [struct BeamCharge](#)
- [struct Color](#)
- [struct Enemy](#)
- [struct Explosion](#)
- [struct Font](#)
- [struct Hitbox](#)
- [struct IComponent](#)
- [struct LoadingAnimation](#)
- [struct Mob](#)
- [struct Pixel](#)
- [struct Player](#)
- [struct Projectile](#)
- [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`

6.30.1 Typedef Documentation

6.30.1.1 Entity

```
using ecs::Entity = std::uint32_t
```

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

6.30.2 Variable Documentation

6.30.2.1 INVALID_ENTITY

```
Entity ecs::INVALID_ENTITY = 0 [constexpr]
```

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

6.31 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.
- 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** , **Enter** , **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** }

6.31.1 Typedef Documentation

6.31.1.1 id

```
using eng::id = unsigned int
```

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

6.31.2 Enumeration Type Documentation

6.31.2.1 EventType

```
enum class eng::EventType [strong]
```

Enumerator

Closed	
KeyPressed	
KeyReleased	
None	

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

Enumerator

6.31.2.2 Key

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

Enumerator

Unknown	
Escape	
Enter	
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	

Enumerator

Num8	
Num9	

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

6.31.2.3 State

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

Enumerator

STOP	
RUN	
DEFAULT	

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

6.31.2.4 Status

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

Enumerator

Stopped	
Paused	
Playing	

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

6.32 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 [RTypClient](#)
Class for the R-Type game.
- class [RTypServer](#)
Class for the R-Type game.
- struct [Sprite](#)

6.33 rnp Namespace Reference

Classes

- struct **EntityState**
Entity state for WORLD_STATE packet.
- struct **EventRecord**
Event record for ENTITY_EVENT packets (TLV format)
- struct **FragmentHeader**
Fragmentation header (when FRAG flag is set)
- struct **PacketAck**
ACK packet payload.
- struct **PacketConnect**
CONNECT packet payload.
- struct **PacketConnectAccept**
CONNECT_ACCEPT packet payload.
- struct **PacketDisconnect**
DISCONNECT packet payload.
- struct **PacketError**
ERROR packet payload.
- struct **PacketHeader**
Packet header according to RNP specification (Big Endian) Total size: 16 bytes.
- struct **PacketPingPong**
PING/PONG packet payload.
- struct **PacketWorldState**
WORLD_STATE packet payload.

Enumerations

- enum class **PacketType** : std::uint8_t {
CONNECT = 0x01 , DISCONNECT = 0x02 , WORLD_STATE = 0x03 , PING = 0x04 ,
PONG = 0x05 , PACKET_ERROR = 0x06 , ACK = 0x07 , ENTITY_EVENT = 0x08 ,
CONNECT_ACCEPT = 0x09 , PLAYER_INPUT = 0x03 }
Packet types according to RNP specification.
- enum class **PacketFlags** : std::uint16_t {
NONE = 0x0000 , ACK_REQ = 0x0001 , RELIABLE = 0x0002 , FRAG = 0x0004 ,
COMPRESSED = 0x0008 }
Packet flags for reliability and fragmentation.
- enum class **DisconnectReason** : std::uint16_t {
UNSPECIFIED = 0 , CLIENT_REQUEST = 1 , TIMEOUT = 2 , PROTOCOL_ERROR = 3 ,
SERVER_SHUTDOWN = 4 , SERVER_FULL = 5 , BANNED = 6 }
Disconnect reason codes.
- enum class **ErrorCode** : std::uint16_t { INVALID_PAYLOAD = 1 , UNAUTHORIZED_SESSION = 2 , RATE_LIMITED = 3 , INTERNAL_ERROR = 4 }
Error codes.
- enum class **EventType** : std::uint8_t { SPAWN = 0x01 , DESPAWN = 0x02 , DAMAGE = 0x03 , SCORE = 0x04 ,
POWERUP = 0x05 , INPUT = 0x06 , CUSTOM = 0xFF }
Event types for ENTITY_EVENT packets.
- enum class **EntityType** : std::uint16_t {
PLAYER = 0x01 , ENEMY = 0x02 , PROJECTILE = 0x03 , POWERUP = 0x04 ,
OBSTACLE = 0x05 }
Entity types for world state.

Functions

- `std::vector< std::uint8_t > serializeEvents (const std::vector< EventRecord > &events)`
Serialize events in ENTITY_EVENT format (TLV with entity_id) Format per event: type(1) | entity_id(4, BE) | data_len(1) | data(data_len)
- `std::vector< EventRecord > deserializeEvents (const std::uint8_t *payload, const std::size_t length)`
Deserialize ENTITY_EVENT payload into event records Format per event: type(1) | entity_id(4, BE) | data_len(1) | data(data_len)
- `std::vector< uint8_t > serializeHeader (const PacketHeader &header)`
Serialize packet header (Big Endian as per RNP spec)
- `std::vector< uint8_t > serialize (const PacketHeader &header, const std::uint8_t *payload=nullptr)`
Serialize packet with header and optional payload (Big Endian)
- `PacketHeader deserializeHeader (const std::uint8_t *data, const std::size_t size)`
Deserialize packet header (Big Endian)

Variables

- `constexpr std::uint8_t PROTOCOL_VERSION = 1`
- `constexpr std::size_t MAX_PAYLOAD = 512`

6.33.1 Enumeration Type Documentation

6.33.1.1 DisconnectReason

enum class `rnp::DisconnectReason` : `std::uint16_t` [strong]

Disconnect reason codes.

Enumerator

UNSPECIFIED	
CLIENT_REQUEST	
TIMEOUT	
PROTOCOL_ERROR	
SERVER_SHUTDOWN	
SERVER_FULL	
BANNED	

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

6.33.1.2 EntityType

enum class `rnp::EntityType` : `std::uint16_t` [strong]

Entity types for world state.

Enumerator

PLAYER	
ENEMY	
PROJECTILE	
POWERUP	
OBSTACLE	

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

6.33.1.3 ErrorCode

enum class [rnp::ErrorCode](#) : std::uint16_t [strong]

Error codes.

Enumerator

INVALID_PAYLOAD	
UNAUTHORIZED_SESSION	
RATE_LIMITED	
INTERNAL_ERROR	

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

6.33.1.4 EventType

enum class [rnp::EventType](#) : std::uint8_t [strong]

Event types for ENTITY_EVENT packets.

Enumerator

SPAWN	
DESPAWN	
DAMAGE	
SCORE	
POWERUP	
INPUT	
CUSTOM	

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

6.33.1.5 PacketFlags

enum class [rnp::PacketFlags](#) : std::uint16_t [strong]

Packet flags for reliability and fragmentation.

Enumerator

NONE	
ACK_REQ	
RELIABLE	
FRAG	
COMPRESSED	

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

6.33.1.6 PacketType

```
enum class rnp::PacketType : std::uint8_t [strong]
```

Packet types according to RNP specification.

Enumerator

CONNECT	
DISCONNECT	
WORLD_STATE	
PING	
PONG	
PACKET_ERROR	
ACK	
ENTITY_EVENT	
CONNECT_ACCEPT	
PLAYER_INPUT	

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

6.33.2 Function Documentation

6.33.2.1 deserializeEvents()

```
std::vector< EventRecord > rnp::deserializeEvents (
    const std::uint8_t * payload,
    const std::size_t length) [inline]
```

Deserialize ENTITY_EVENT payload into event records Format per event: type(1) | entity_id(4, BE) | data_len(1) | data(data_len)

Definition at line 253 of file [Protocol.hpp](#).

References [rmp::EventRecord::data](#).

6.33.2.2 deserializeHeader()

```
PacketHeader rnp::deserializeHeader (
    const uint8_t * data,
    const std::size_t size) [inline]
```

Deserialize packet header (Big Endian)

Definition at line 345 of file [Protocol.hpp](#).

References [rnp::PacketHeader::flags](#), [rnp::PacketHeader::length](#), [rnp::PacketHeader::reserved](#), [rnp::PacketHeader::sequence](#), [rnp::PacketHeader::sessionId](#), and [rnp::PacketHeader::type](#).

6.33.2.3 serialize()

```
std::vector< uint8_t > rnp::serialize (
    const PacketHeader & header,
    const uint8_t * payload = nullptr) [inline]
```

Serialize packet with header and optional payload (Big Endian)

Definition at line 330 of file [Protocol.hpp](#).

References [rmp::PacketHeader::length](#), and [serializeHeader\(\)](#).

Here is the call graph for this function:



6.33.2.4 serializeEvents()

```
std::vector< std::uint8_t > rnp::serializeEvents (
    const std::vector< EventRecord > & events) [inline]
```

Serialize events in ENTITY_EVENT format (TLV with entity_id) Format per event: type(1) | entity_id(4, BE) | data_len(1) | data(data_len)

Definition at line 217 of file [Protocol.hpp](#).

References [MAX_PAYLOAD](#).

6.33.2.5 serializeHeader()

```
std::vector< uint8_t > rnp::serializeHeader (const PacketHeader & header) [inline]
```

Serialize packet header (Big Endian as per RNP spec)

Definition at line 292 of file [Protocol.hpp](#).

References [rnp::PacketHeader::flags](#), [rnp::PacketHeader::length](#), [rnp::PacketHeader::reserved](#), [rnp::PacketHeader::sequence](#), [rnp::PacketHeader::sessionId](#), and [rnp::PacketHeader::type](#).

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

Here is the caller graph for this function:



6.33.3 Variable Documentation

6.33.3.1 MAX_PAYLOAD

```
std::size_t rnp::MAX_PAYLOAD = 512 [inline], [constexpr]
```

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

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

6.33.3.2 PROTOCOL_VERSION

```
std::uint8_t rnp::PROTOCOL_VERSION = 1 [inline], [constexpr]
```

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

6.34 srv Namespace Reference

Namespaces

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

Classes

- struct [ArgsConfig](#)
- class [ArgsHandler](#)
Class to handle command line arguments.
- class [AScene](#)
Class for scene.
- struct [EnvConfig](#)
- class [INetworkServer](#)
Interface for the server network.
- class [IScene](#)
interface class for scene
- class [SceneManager](#)
Class for managing scenes.
- class [Server](#)
Class for the server.

Typedefs

- using [json](#) = nlohmann::json
- using [id](#) = unsigned int

Variables

- constexpr size_t [MAX_CLIENTS](#) = 16
- constexpr size_t [MAX_IP_LENGTH](#) = 8
- constexpr size_t [MAX_LEN_RECV_BUFFER](#) = 1024

6.34.1 Typedef Documentation

6.34.1.1 [id](#)

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

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

6.34.1.2 [json](#)

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

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

6.34.2 Variable Documentation

6.34.2.1 [MAX_CLIENTS](#)

size_t [srv::MAX_CLIENTS](#) = 16 [constexpr]

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

6.34.2.2 MAX_IP_LENGTH

```
size_t srv::MAX_IP_LENGTH = 8 [constexpr]
```

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

6.34.2.3 MAX_LEN_RECV_BUFFER

```
size_t srv::MAX_LEN_RECV_BUFFER = 1024 [constexpr]
```

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

6.35 srv::Config Namespace Reference

Namespaces

- namespace [Network](#)

6.36 srv::Config::Network Namespace Reference

Variables

- constexpr auto [DEFAULT_NETWORK_HOST](#) = "0.0.0.0"
- constexpr auto [DEFAULT_NETWORK_PORT](#) = 2560
- constexpr auto [DEFAULT_MAX_CLIENT](#) = 4

6.36.1 Variable Documentation

6.36.1.1 DEFAULT_MAX_CLIENT

```
auto srv::Config::Network::DEFAULT_MAX_CLIENT = 4 [inline], [constexpr]
```

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

6.36.1.2 DEFAULT_NETWORK_HOST

```
auto srv::Config::Network::DEFAULT_NETWORK_HOST = "0.0.0.0" [inline], [constexpr]
```

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

6.36.1.3 DEFAULT_NETWORK_PORT

```
auto srv::Config::Network::DEFAULT_NETWORK_PORT = 2560 [inline], [constexpr]
```

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

6.37 srv::Game Namespace Reference

Variables

- `constexpr auto DEFAULT_TICK_RATE = 60`
- `constexpr auto DEFAULT_UPDATE_INTERVAL = 1 / 20.F`

6.37.1 Variable Documentation

6.37.1.1 DEFAULT_TICK_RATE

`auto srv::Game::DEFAULT_TICK_RATE = 60 [inline], [constexpr]`

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

6.37.1.2 DEFAULT_UPDATE_INTERVAL

`auto srv::Game::DEFAULT_UPDATE_INTERVAL = 1 / 20.F [inline], [constexpr]`

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

6.38 srv::Path Namespace Reference

Namespaces

- namespace [Plugin](#)

6.39 srv::Path::Plugin Namespace Reference

Variables

- `auto PLUGINS_NETWORK_ASIO_SERVER`

6.39.1 Variable Documentation

6.39.1.1 PLUGINS_NETWORK_ASIO_SERVER

`auto srv::Path::Plugin::PLUGINS_NETWORK_ASIO_SERVER [inline]`

Initial value:

```
= std::filesystem::path(PLUGINS_DIR) / ("network_asio_server" + std::string(PLUGINS_EXTENSION))
```

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

6.40 utl Namespace Reference

Classes

- class [Clock](#)
Class for clock.
- interface [IPlugin](#)
Interface for plugins.
- class [Logger](#)
Class for logging.
- class [PluginLoader](#)
Modern, type-safe plugin loader.
- struct [SharedLib](#)
Handle to a dynamic library with RAII.

Typedefs

- using [LibHandle](#)
- using [EntryPointFn = IPlugin *\(*\)\(\)](#)

Enumerations

- enum class [PluginType](#) : uint8_t {
AUDIO = 0 , NETWORK_CLIENT = 1 , NETWORK_SERVER = 1 , RENDERER = 2 ,
UNKNOWN = 255 }
- enum class [LogLevel](#) : uint8_t { [INFO](#) , [WARNING](#) }

Functions

- std::vector< char > [readFile](#) (const std::string &filename)
- std::unordered_map< std::string, std::string > [getEnvMap](#) (const char *const *env)

6.40.1 Typedef Documentation

6.40.1.1 EntryPointFn

using [utl::EntryPointFn = IPlugin *\(*\)\(\)](#)

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

6.40.1.2 LibHandle

using [utl::LibHandle](#)

Initial value:

```
void *
```

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

6.40.2 Enumeration Type Documentation

6.40.2.1 LogLevel

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

Enumerator

INFO	
WARNING	

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

6.40.2.2 PluginType

```
enum class utl::PluginType : uint8_t [strong]
```

Enumerator

AUDIO	
NETWORK_CLIENT	
NETWORK_SERVER	
RENDERER	
UNKNOWN	

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

6.40.3 Function Documentation

6.40.3.1 getEnvMap()

```
std::unordered_map< std::string, std::string > utl::getEnvMap (
    const char *const * env) [nodiscard]
```

Definition at line 30 of file [utils.cpp](#).

6.40.3.2 readFile()

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

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

Chapter 7

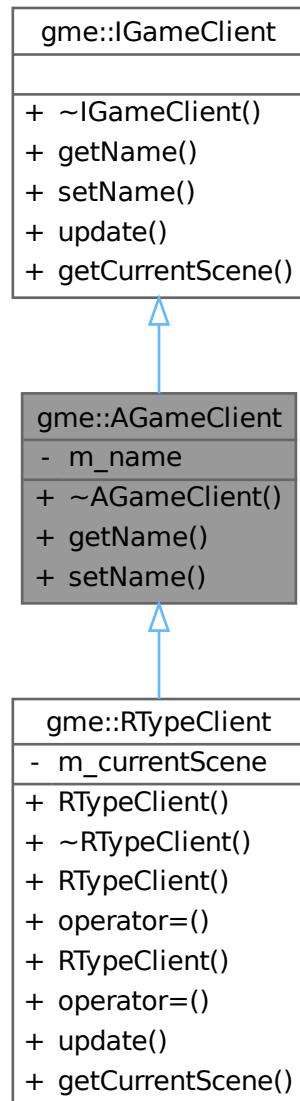
Class Documentation

7.1 gme::AGameClient Class Reference

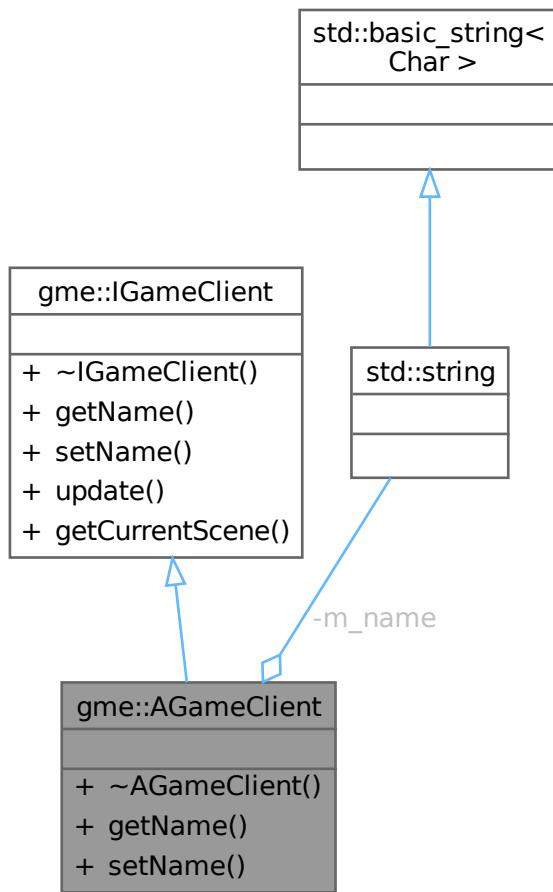
Abstraction for the games.

```
#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:

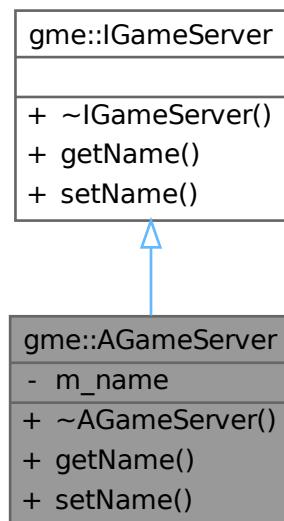
- /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/[AGameClient.hpp](#)

7.2 gme::AGameServer Class Reference

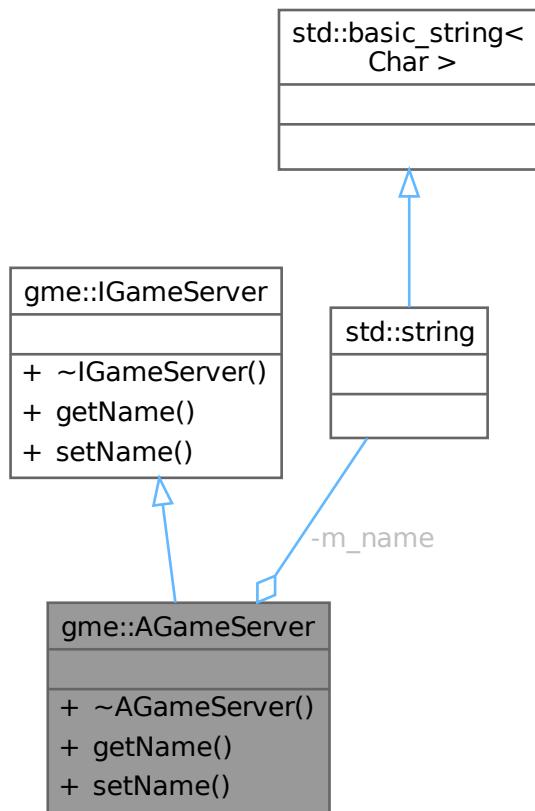
Abstraction for the games.

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

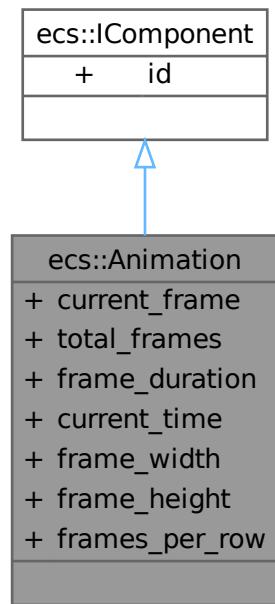
The documentation for this class was generated from the following file:

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

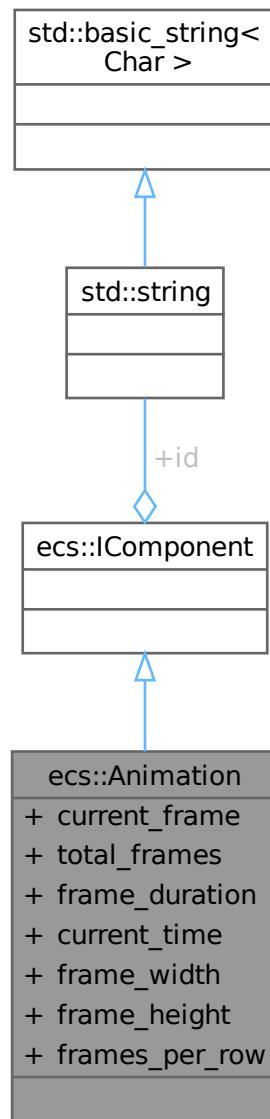
7.3 ecs::Animation Struct Reference

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

Inheritance diagram for ecs::Animation:



Collaboration diagram for ecs::Animation:



Public Attributes

- int `current_frame` {}
- int `total_frames` {}
- float `frame_duration` {}
- float `current_time` {}
- int `frame_width` {}
- int `frame_height` {}
- int `frames_per_row` {}

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

- std::string [id](#)

7.3.1 Detailed Description

Definition at line [76](#) of file [Component.hpp](#).

7.3.2 Member Data Documentation

7.3.2.1 current_frame

```
int ecs::Animation::current_frame {}
```

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

7.3.2.2 current_time

```
float ecs::Animation::current_time {}
```

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

7.3.2.3 frame_duration

```
float ecs::Animation::frame_duration {}
```

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

7.3.2.4 frame_height

```
int ecs::Animation::frame_height {}
```

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

7.3.2.5 frame_width

```
int ecs::Animation::frame_width {}
```

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

7.3.2.6 frames_per_row

```
int ecs::Animation::frames_per_row {}
```

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

7.3.2.7 total_frames

```
int ecs::Animation::total_frames {}
```

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

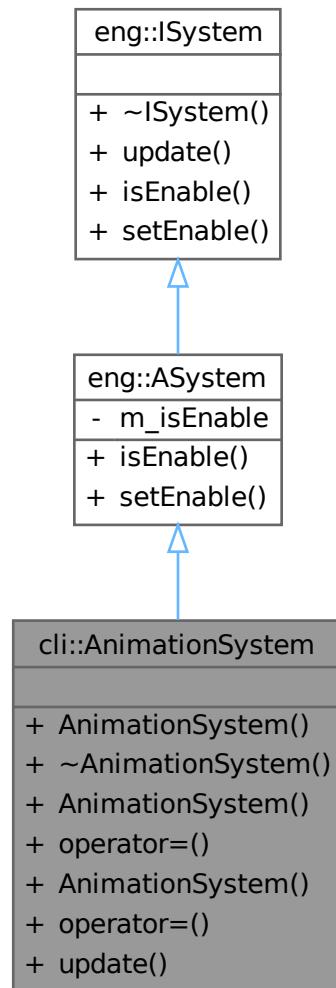
The documentation for this struct was generated from the following file:

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

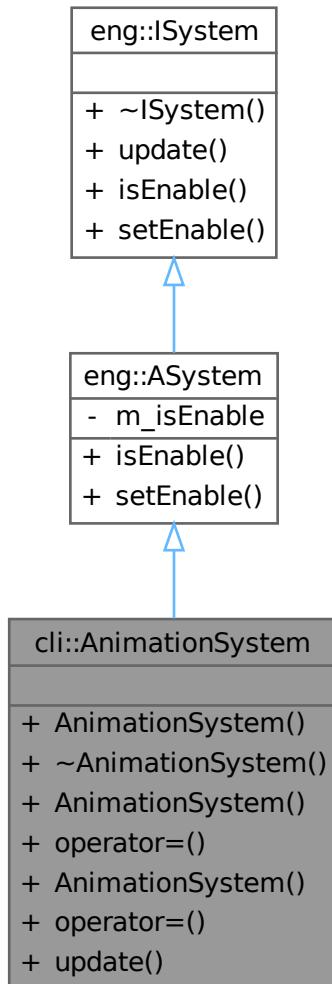
7.4 cli::AnimationSystem Class Reference

```
#include <Animation.hpp>
```

Inheritance diagram for cli::AnimationSystem:



Collaboration diagram for cli::AnimationSystem:



Public Member Functions

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

Public Member Functions inherited from `eng::ASystem`

- `bool isEnabled () override`
- `void setEnable (const bool enable) override`

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

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

7.4.1 Detailed Description

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

7.4.2 Constructor & Destructor Documentation

7.4.2.1 AnimationSystem() [1/3]

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

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

7.4.2.2 ~AnimationSystem()

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

7.4.2.3 AnimationSystem() [2/3]

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

7.4.2.4 AnimationSystem() [3/3]

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

7.4.3 Member Function Documentation

7.4.3.1 operator=() [1/2]

```
AnimationSystem & cli::AnimationSystem::operator= (
    AnimationSystem && ) [delete]
```

7.4.3.2 operator=() [2/2]

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

7.4.3.3 update()

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

Implements [eng::ISystem](#).

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

References [ecs::Registry::getAll\(\)](#), and [ecs::Registry::getComponent\(\)](#).

Here is the call graph for this function:



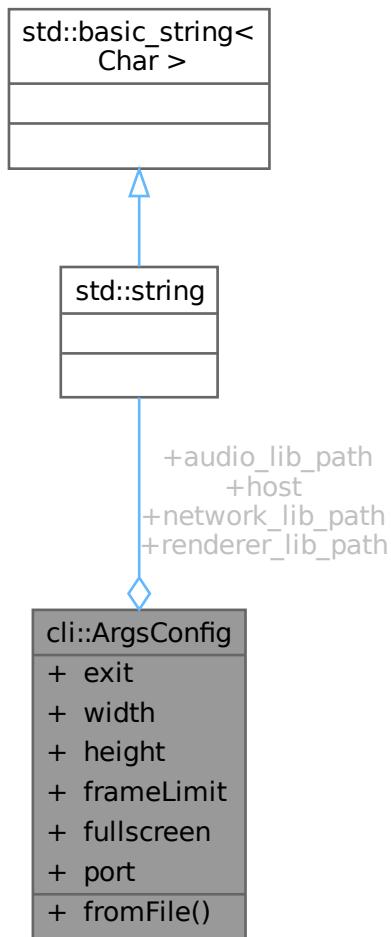
The documentation for this class was generated from the following file:

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

7.5 cli::ArgsConfig Struct Reference

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

Collaboration diagram for cli::ArgsConfig:



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
- std::string `host` = Config::Network::DEFAULT_NETWORK_HOST
- unsigned int `port` = Config::Network::DEFAULT_NETWORK_PORT
- std::string `audio_lib_path` = Path::Plugin::PLUGIN_AUDIO_SFML.string()
- std::string `network_lib_path` = Path::Plugin::PLUGIN_NETWORK_ASIO_CLIENT.string()
- std::string `renderer_lib_path` = Path::Plugin::PLUGIN_RENDERER_SFML.string()

7.5.1 Detailed Description

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

7.5.2 Member Function Documentation

7.5.2.1 fromFile()

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

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

References [audio_lib_path](#), [frameLimit](#), [fullscreen](#), [height](#), [host](#), [network_lib_path](#), [port](#), [renderer_lib_path](#), and [width](#).

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

Here is the caller graph for this function:



7.5.3 Member Data Documentation

7.5.3.1 audio_lib_path

```
std::string cli::ArgsConfig::audio_lib_path = Path::Plugin::PLUGIN_AUDIO_SFML.string()
```

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

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

7.5.3.2 exit

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

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

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

7.5.3.3 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\(\)](#).

7.5.3.4 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\(\)](#).

7.5.3.5 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\(\)](#).

7.5.3.6 host

```
std::string cli::ArgsConfig::host = Config::Network::DEFAULT_NETWORK_HOST
```

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

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

7.5.3.7 network_lib_path

```
std::string cli::ArgsConfig::network_lib_path = Path::Plugin::PLUGIN_NETWORK_ASIO_CLIENT.string()
```

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

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

7.5.3.8 port

```
unsigned int cli::ArgsConfig::port = Config::Network::DEFAULT_NETWORK_PORT
```

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

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

7.5.3.9 renderer_lib_path

```
std::string cli::ArgsConfig::renderer_lib_path = Path::Plugin::PLUGIN_RENDERER_SFML.string()
```

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

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

7.5.3.10 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\(\)](#).

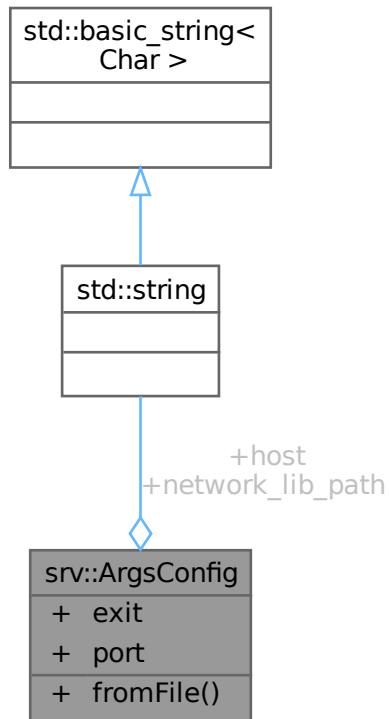
The documentation for this struct was generated from the following files:

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

7.6 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` = Config::Network::DEFAULT_NETWORK_HOST
- uint16_t `port` = Config::Network::DEFAULT_NETWORK_PORT
- std::string `network_lib_path`

7.6.1 Detailed Description

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

7.6.2 Member Function Documentation

7.6.2.1 fromFile()

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

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

References `host`, `network_lib_path`, and `port`.

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

Here is the caller graph for this function:



7.6.3 Member Data Documentation

7.6.3.1 exit

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

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

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

7.6.3.2 host

```
std::string srv::ArgsConfig::host = Config::Network::DEFAULT_NETWORK_HOST
```

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

Referenced by [FromFile\(\)](#), and [srv::Server::Server\(\)](#).

7.6.3.3 network_lib_path

std::string srv::ArgsConfig::network_lib_path

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

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

7.6.3.4 port

uint16_t srv::ArgsConfig::port = [Config::Network::DEFAULT_NETWORK_PORT](#)

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

Referenced by [FromFile\(\)](#), and [srv::Server::Server\(\)](#).

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

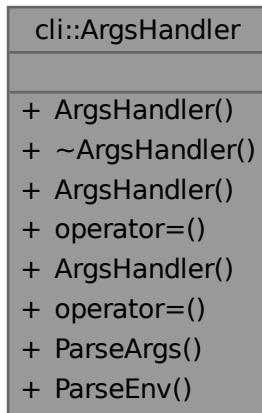
- /home/masina/Projects/Epitech/rtype/server/include/Server/[ArgsHandler.hpp](#)
- /home/masina/Projects/Epitech/rtype/server/src/[argsHandler.cpp](#)

7.7 cli::ArgsHandler Class Reference

Class to handle command line arguments.

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

Collaboration diagram for cli::ArgsHandler:



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[])`

7.7.1 Detailed Description

Class to handle command line arguments.

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

7.7.2 Constructor & Destructor Documentation

7.7.2.1 ArgsHandler() [1/3]

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

7.7.2.2 ~ArgsHandler()

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

7.7.2.3 ArgsHandler() [2/3]

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

7.7.2.4 ArgsHandler() [3/3]

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

7.7.3 Member Function Documentation

7.7.3.1 operator=() [1/2]

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

7.7.3.2 operator=() [2/2]

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

7.7.3.3 ParseArgs()

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

Definition at line 83 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:



7.7.3.4 ParseEnv()

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

Definition at line 139 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:

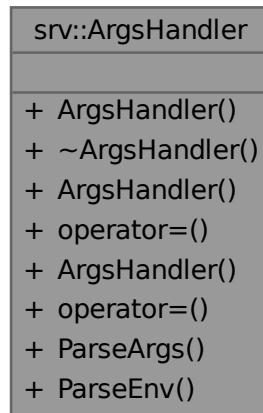
- /home/masina/Projects/Epitech/rtype/client/include/Client/[ArgsHandler.hpp](#)
- /home/masina/Projects/Epitech/rtype/client/src/[argsHandler.cpp](#)

7.8 srv::ArgsHandler Class Reference

Class to handle command line arguments.

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

Collaboration diagram for srv::ArgsHandler:



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[])

7.8.1 Detailed Description

Class to handle command line arguments.

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

7.8.2 Constructor & Destructor Documentation

7.8.2.1 ArgsHandler() [1/3]

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

7.8.2.2 ~ArgsHandler()

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

7.8.2.3 ArgsHandler() [2/3]

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

7.8.2.4 ArgsHandler() [3/3]

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

7.8.3 Member Function Documentation

7.8.3.1 operator=() [1/2]

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

7.8.3.2 operator=() [2/2]

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

7.8.3.3 ParseArgs()

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

Definition at line 53 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:



7.8.3.4 ParseEnv()

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

Definition at line 106 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:

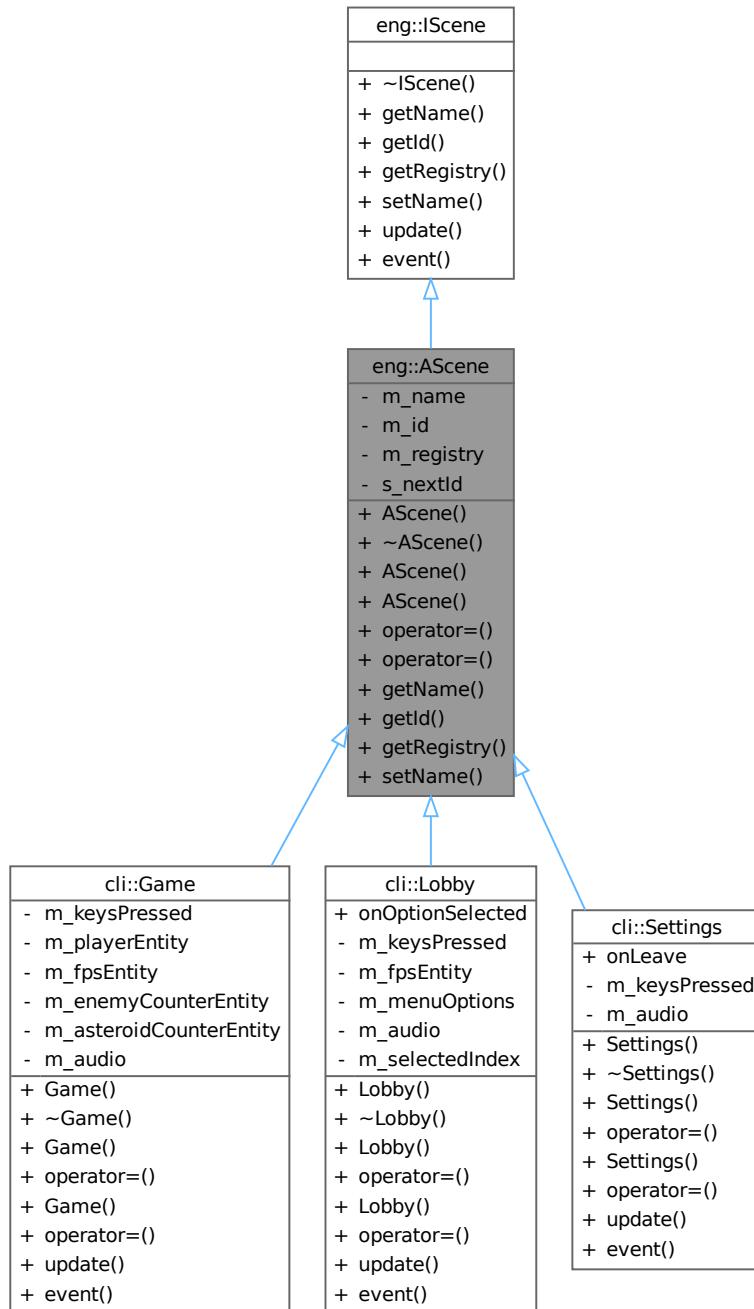
- /home/masina/Projects/Epitech/rtype/server/include/Server/[ArgsHandler.hpp](#)
- /home/masina/Projects/Epitech/rtype/server/src/[argsHandler.cpp](#)

7.9 eng::AScene Class Reference

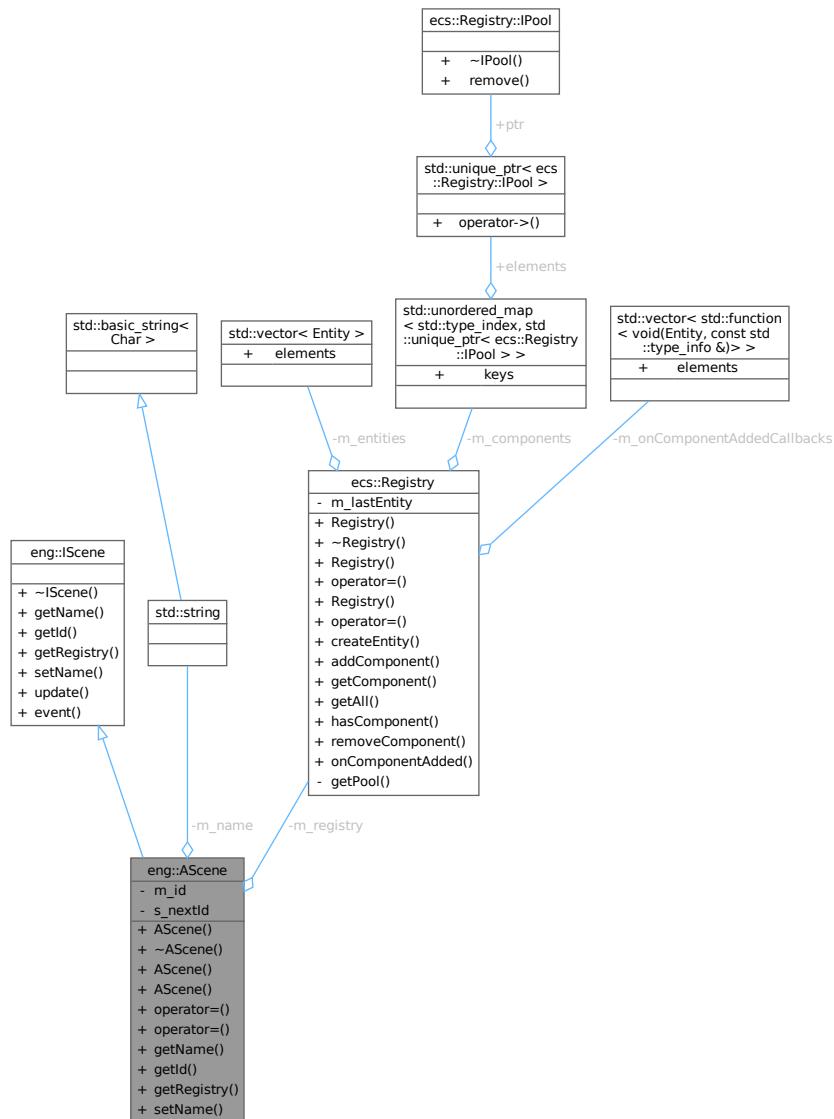
Class for scene.

```
#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

7.9.1 Detailed Description

Class for scene.

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

7.9.2 Constructor & Destructor Documentation

7.9.2.1 AScene() [1/3]

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

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

7.9.2.2 ~AScene()

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

7.9.2.3 AScene() [2/3]

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

7.9.2.4 AScene() [3/3]

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

7.9.3 Member Function Documentation

7.9.3.1 getId()

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

Implements [eng::IScene](#).

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

References [m_id](#).

7.9.3.2 getName()

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

Implements [eng::IScene](#).

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

References [m_name](#).

7.9.3.3 getRegistry()

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

Implements [eng::IScene](#).

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

References [m_registry](#).

7.9.3.4 operator=() [1/2]

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

7.9.3.5 operator=() [2/2]

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

7.9.3.6 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](#).

7.9.4 Member Data Documentation

7.9.4.1 m_id

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

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

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

7.9.4.2 m_name

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

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

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

7.9.4.3 m_registry

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

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

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

7.9.4.4 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:

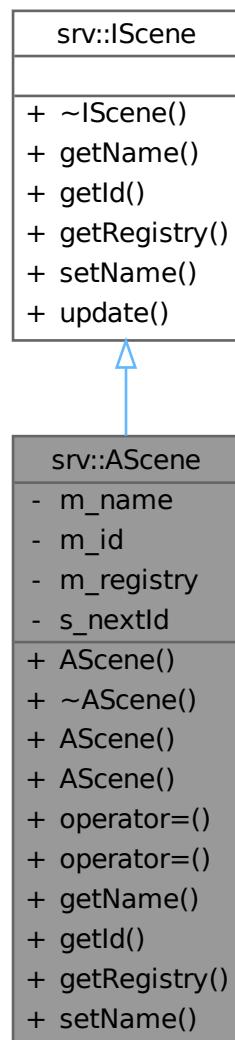
- /home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/Interfaces/[IScene.hpp](#)

7.10 srv::AScene Class Reference

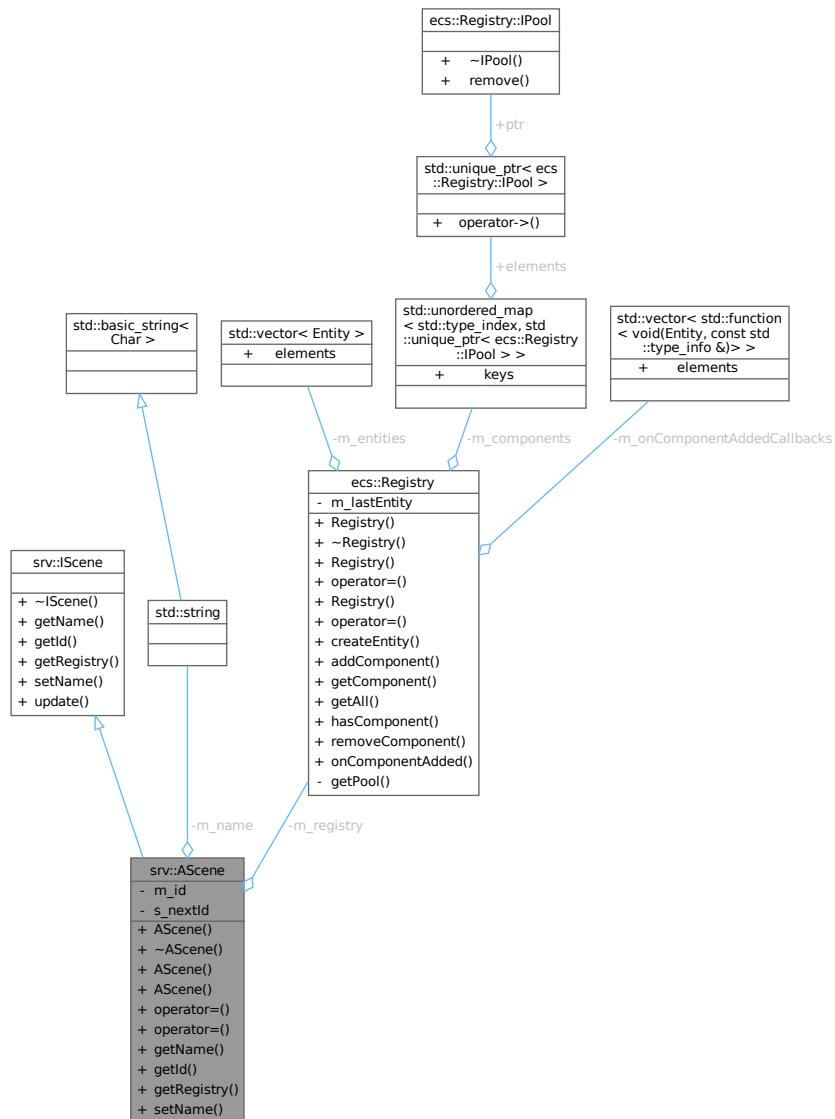
Class for scene.

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

Inheritance diagram for srv::AScene:



Collaboration diagram for `srv::AScene`:



Public Member Functions

- `AScene ()`
- `~AScene ()` override=default
- `AScene (const ASScene &other)=delete`
- `AScene (AScene &&other)=delete`
- `AScene & operator= (const ASScene &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 [srv::IScene](#)

- virtual [~IScene](#) ()=default
- virtual void [update](#) (float dt)=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

7.10.1 Detailed Description

Class for scene.

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

7.10.2 Constructor & Destructor Documentation

7.10.2.1 AScene() [1/3]

`srv::AScene::AScene () [inline]`

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

7.10.2.2 ~AScene()

`srv::AScene::~AScene () [override], [default]`

7.10.2.3 AScene() [2/3]

```
srv::AScene::AScene (
    const AScene & other) [delete]
```

7.10.2.4 AScene() [3/3]

```
srv::AScene::AScene (
    AScene && other) [delete]
```

7.10.3 Member Function Documentation

7.10.3.1 getId()

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

Implements [srv::IScene](#).

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

References [m_id](#).

7.10.3.2 getName()

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

Implements [srv::IScene](#).

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

References [m_name](#).

7.10.3.3 getRegistry()

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

Implements [srv::IScene](#).

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

References [m_registry](#).

7.10.3.4 operator=() [1/2]

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

7.10.3.5 operator=() [2/2]

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

7.10.3.6 setName()

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

Implements [srv::IScene](#).

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

References [m_name](#).

7.10.4 Member Data Documentation

7.10.4.1 m_id

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

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

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

7.10.4.2 m_name

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

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

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

7.10.4.3 m_registry

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

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

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

7.10.4.4 s_nextId

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

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

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

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

7.11 cli::SpriteRect::SpriteSheet::Asteroid Struct Reference

#include <SpriteRect.hpp>

Collaboration diagram for cli::SpriteRect::SpriteSheet::Asteroid:

cli::SpriteRect::SpriteSheet ::Asteroid	
+	frameW
+	frameH
+	marginX
+	marginY
+	spacingX
+	spacingY
+	totalFrames

Static Public Attributes

- static constexpr int `frameW` = 16
- static constexpr int `frameH` = 16
- static constexpr int `marginX` = 1
- static constexpr int `marginY` = 1
- static constexpr int `spacingX` = 1
- static constexpr int `spacingY` = 0
- static constexpr int `totalFrames` = 12

7.11.1 Detailed Description

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

7.11.2 Member Data Documentation

7.11.2.1 frameH

```
int cli::SpriteRect::SpriteSheet::Asteroid::frameH = 16 [static], [constexpr]
```

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

Referenced by [cli::SpriteRect::asteroidRect\(\)](#).

7.11.2.2 frameW

```
int cli::SpriteRect::SpriteSheet::Asteroid::frameW = 16 [static], [constexpr]
```

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

Referenced by [cli::SpriteRect::asteroidRect\(\)](#).

7.11.2.3 marginX

```
int cli::SpriteRect::SpriteSheet::Asteroid::marginX = 1 [static], [constexpr]
```

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

Referenced by [cli::SpriteRect::asteroidRect\(\)](#).

7.11.2.4 marginY

```
int cli::SpriteRect::SpriteSheet::Asteroid::marginY = 1 [static], [constexpr]
```

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

Referenced by [cli::SpriteRect::asteroidRect\(\)](#).

7.11.2.5 spacingX

```
int cli::SpriteRect::SpriteSheet::Asteroid::spacingX = 1 [static], [constexpr]
```

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

Referenced by [cli::SpriteRect::asteroidRect\(\)](#).

7.11.2.6 spacingY

```
int cli::SpriteRect::SpriteSheet::Asteroid::spacingY = 0 [static], [constexpr]
```

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

Referenced by [cli::SpriteRect::asteroidRect\(\)](#).

7.11.2.7 totalFrames

```
int cli::SpriteRect::SpriteSheet::Asteroid::totalFrames = 12 [static], [constexpr]
```

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

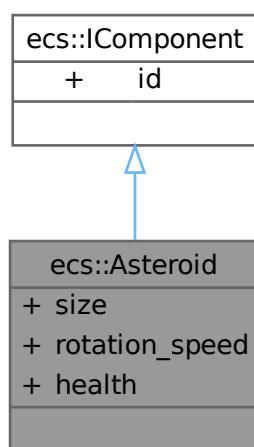
The documentation for this struct was generated from the following file:

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

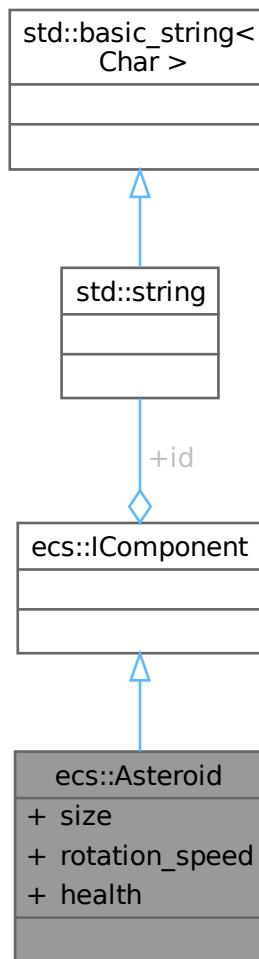
7.12 ecs::Asteroid Struct Reference

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

Inheritance diagram for ecs::Asteroid:



Collaboration diagram for `ecs::Asteroid`:



Public Types

- enum `Size` { `SMALL` , `MEDIUM` , `LARGE` }

Public Attributes

- `Size size`
- float `rotation_speed`
- float `health`

Public Attributes inherited from `ecs::IComponent`

- `std::string id`

7.12.1 Detailed Description

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

7.12.2 Member Enumeration Documentation

7.12.2.1 Size

enum `ecs::Asteroid::Size`

Enumerator

SMALL	
MEDIUM	
LARGE	

Definition at line 128 of file [Component.hpp](#).

7.12.3 Member Data Documentation

7.12.3.1 health

`float ecs::Asteroid::health`

Definition at line 136 of file [Component.hpp](#).

7.12.3.2 rotation_speed

`float ecs::Asteroid::rotation_speed`

Definition at line 135 of file [Component.hpp](#).

7.12.3.3 size

`Size ecs::Asteroid::size`

Definition at line 134 of file [Component.hpp](#).

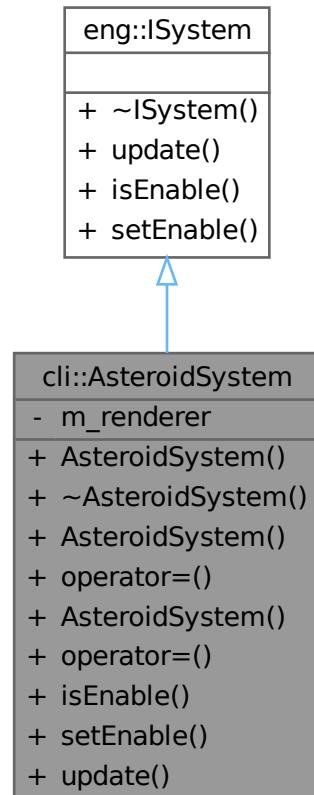
The documentation for this struct was generated from the following file:

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

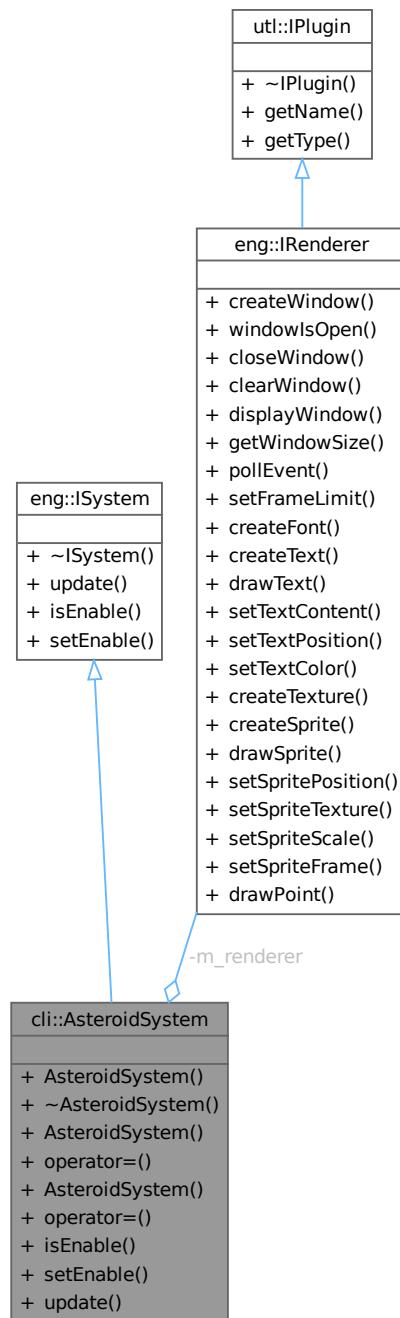
7.13 cli::AsteroidSystem Class Reference

```
#include <Asteroid.hpp>
```

Inheritance diagram for cli::AsteroidSystem:



Collaboration diagram for cli::AsteroidSystem:



Public Member Functions

- `AsteroidSystem (eng::IRenderer &renderer)`
- `~AsteroidSystem ()` override=default
- `AsteroidSystem (const AsteroidSystem &z)=delete`
- `AsteroidSystem & operator= (const AsteroidSystem &)=delete`
- `AsteroidSystem (AsteroidSystem &&)=delete`

- `AsteroidSystem & operator= (AsteroidSystem &&) = delete`
- `bool isEnabled () override`
- `void setEnable (bool enable) override`
- `void update (ecs::Registry ®istry, float dt) override`

Public Member Functions inherited from `eng::ISystem`

- `virtual ~ISystem () = default`

Private Attributes

- `eng::IRenderer & m_renderer`

7.13.1 Detailed Description

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

7.13.2 Constructor & Destructor Documentation

7.13.2.1 AsteroidSystem() [1/3]

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

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

7.13.2.2 ~AsteroidSystem()

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

7.13.2.3 AsteroidSystem() [2/3]

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

7.13.2.4 AsteroidSystem() [3/3]

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

7.13.3 Member Function Documentation

7.13.3.1 isEnabled()

```
bool cli::AsteroidSystem::isEnabled () [inline], [override], [virtual]
```

Implements `eng::ISystem`.

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

7.13.3.2 operator=() [1/2]

```
AsteroidSystem & cli::AsteroidSystem::operator= (
    AsteroidSystem &&) [delete]
```

7.13.3.3 operator=() [2/2]

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

7.13.3.4 setEnable()

```
void cli::AsteroidSystem::setEnable (
    bool enable) [inline], [override], [virtual]
```

Implements [eng::ISystem](#).

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

7.13.3.5 update()

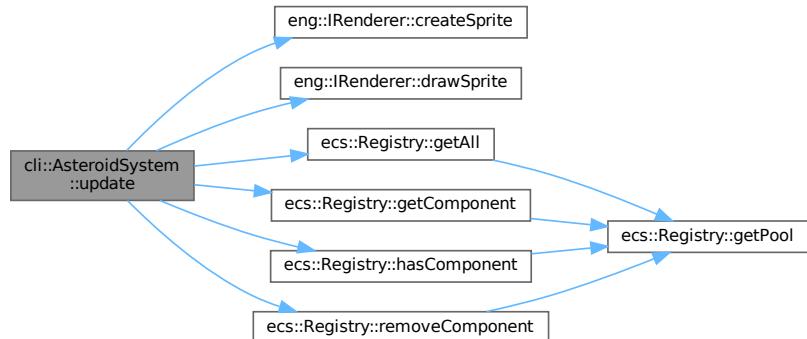
```
void cli::AsteroidSystem::update (
    ecs::Registry & registry,
    float dt) [inline], [override], [virtual]
```

Implements [eng::ISystem](#).

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

References [eng::IRenderer::createSprite\(\)](#), [eng::IRenderer::drawSprite\(\)](#), [ecs::Registry::getAll\(\)](#), [ecs::Registry::getComponent\(\)](#), [ecs::Registry::hasComponent\(\)](#), [ecs::IComponent::id](#), [m_renderer](#), [cli::GameConfig::Screen::REMOVE_MAX_Y](#), [cli::GameConfig::Screen::REMOVE_MIN_Y](#), [cli::GameConfig::Screen::REMOVE_X](#), [ecs::Registry::removeComponent\(\)](#), [cli::GameConfig::Asteroid::Small::SPRITE_WIDTH](#), [ecs::Transform::x](#), [ecs::Velocity::x](#), and [ecs::Transform::y](#).

Here is the call graph for this function:



7.13.4 Member Data Documentation

7.13.4.1 m_renderer

eng::IRenderer& cli::AsteroidSystem::m_renderer [private]

Definition at line 101 of file Asteroid.hpp.

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

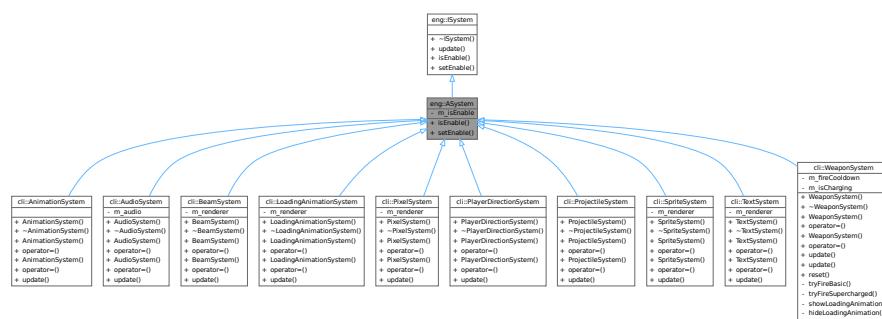
The documentation for this class was generated from the following file:

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

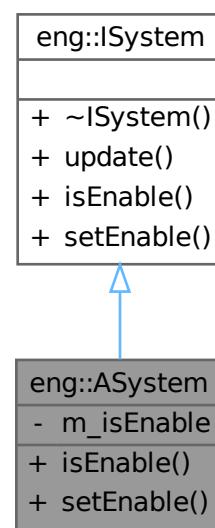
7.14 eng::ASystem Class Reference

```
#include <ISystems.hpp>
```

Inheritance diagram for eng::ASystem:



Collaboration diagram for eng::ASystem:



Public Member Functions

- bool `isEnabled ()` override
- void `setEnable (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`

7.14.1 Detailed Description

Definition at line 23 of file `ISystems.hpp`.

7.14.2 Member Function Documentation

7.14.2.1 `isEnabled()`

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

Implements `eng::ISystem`.

Definition at line 26 of file `ISystems.hpp`.

References `m_isEnable`.

7.14.2.2 `setEnable()`

void eng::ASystem::`setEnable (const bool enable)` [inline], [override], [virtual]

Implements `eng::ISystem`.

Definition at line 27 of file `ISystems.hpp`.

References `m_isEnable`.

7.14.3 Member Data Documentation

7.14.3.1 `m_isEnable`

bool eng::ASystem::`m_isEnable = true` [private]

Definition at line 30 of file `ISystems.hpp`.

Referenced by `isEnabled()`, and `setEnable()`.

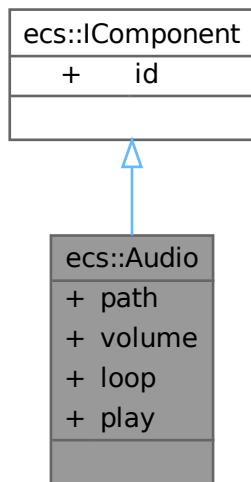
The documentation for this class was generated from the following file:

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

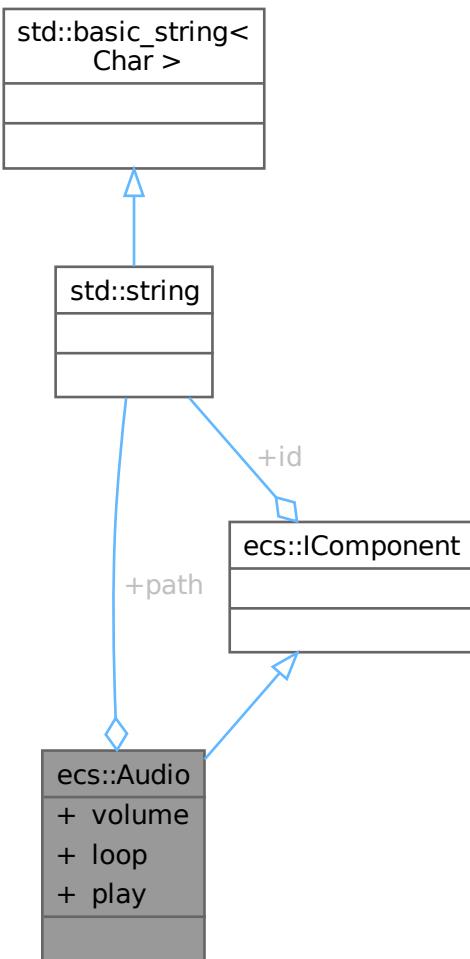
7.15 ecs::Audio Struct Reference

```
#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`

7.15.1 Detailed Description

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

7.15.2 Member Data Documentation

7.15.2.1 loop

```
bool ecs::Audio::loop
```

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

7.15.2.2 path

```
std::string ecs::Audio::path
```

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

7.15.2.3 play

```
bool ecs::Audio::play
```

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

7.15.2.4 volume

```
float ecs::Audio::volume
```

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

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

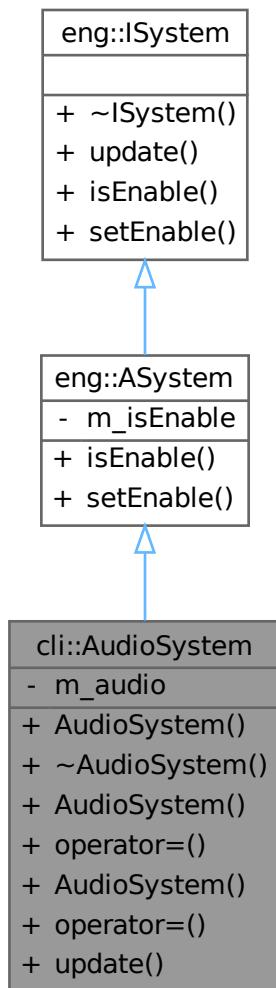
- /home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/[Component.hpp](#)

7.16 cli::AudioSystem Class Reference

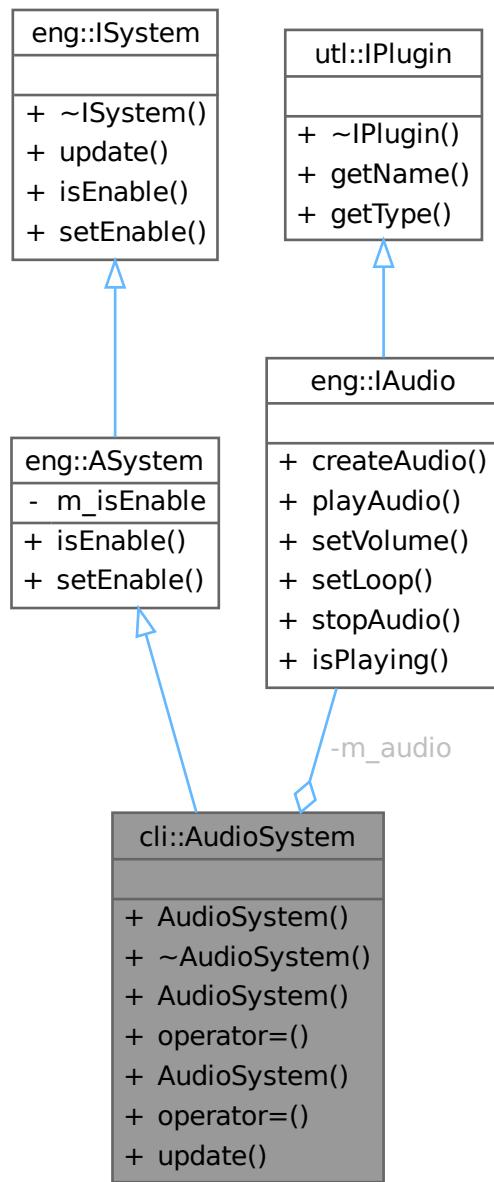
Class for managing entities and their components.

```
#include <Audio.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) override`

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

- bool [isEnable \(\)](#) override
- void [setEnable \(const bool enable\)](#) override

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

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

Private Attributes

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

7.16.1 Detailed Description

Class for managing entities and their components.

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

7.16.2 Constructor & Destructor Documentation

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

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

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

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

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

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

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

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

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

7.16.3 Member Function Documentation

7.16.3.1 operator=() [1/2]

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

7.16.3.2 operator=() [2/2]

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

7.16.3.3 update()

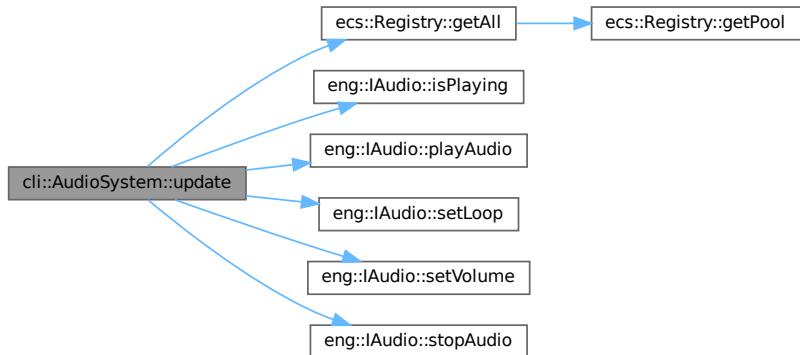
```
void cli::AudioSystem::update (
    ecs::Registry & registry,
    float ) [inline], [override], [virtual]
```

Implements [eng::ISystem](#).

Definition at line [32](#) of file [Audio.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:



7.16.4 Member Data Documentation

7.16.4.1 m_audio

```
eng::IAudio& cli::AudioSystem::m_audio [private]
```

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

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

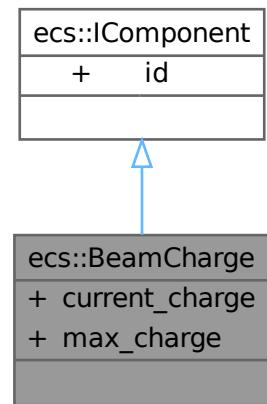
The documentation for this class was generated from the following file:

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

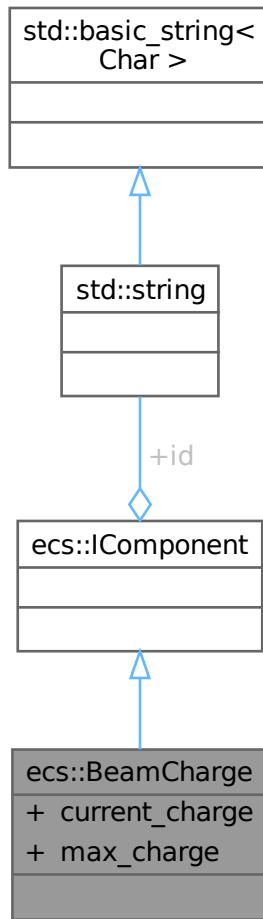
7.17 ecs::BeamCharge Struct Reference

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

Inheritance diagram for ecs::BeamCharge:



Collaboration diagram for ecs::BeamCharge:



Public Attributes

- float `current_charge`
- float `max_charge`

Public Attributes inherited from `ecs::IComponent`

- `std::string id`

7.17.1 Detailed Description

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

7.17.2 Member Data Documentation

7.17.2.1 current_charge

```
float ecs::BeamCharge::current_charge
```

Definition at line 101 of file [Component.hpp](#).

7.17.2.2 max_charge

```
float ecs::BeamCharge::max_charge
```

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

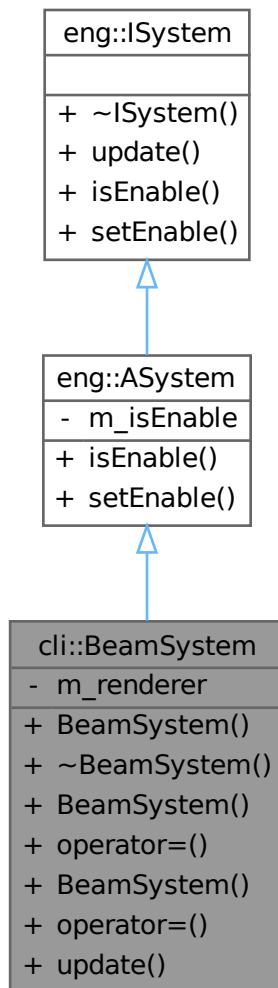
The documentation for this struct was generated from the following file:

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

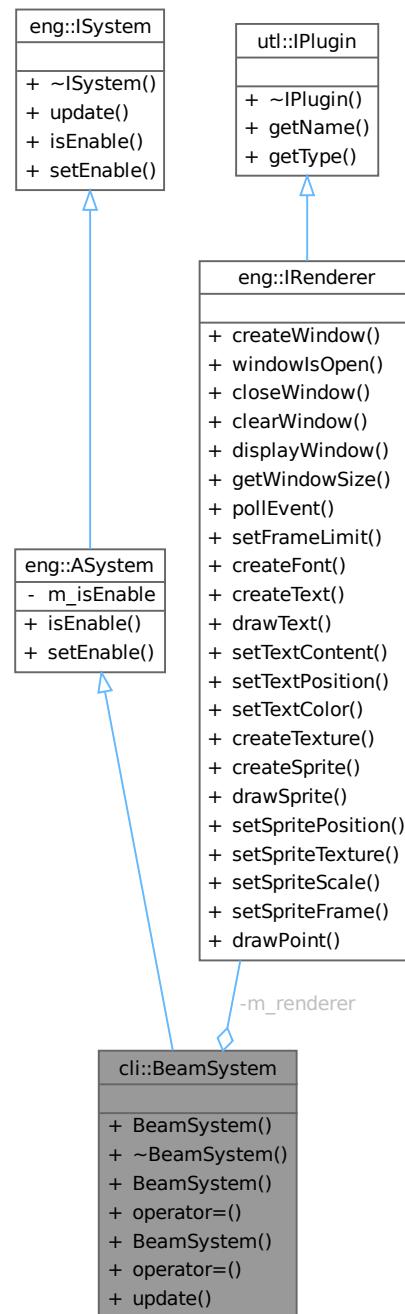
7.18 cli::BeamSystem Class Reference

```
#include <Beam.hpp>
```

Inheritance diagram for cli::BeamSystem:



Collaboration diagram for cli::BeamSystem:



Public Member Functions

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

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

- bool [isEnable \(\)](#) override
- void [setEnable \(const bool enable\)](#) override

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

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

Private Attributes

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

7.18.1 Detailed Description

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

7.18.2 Constructor & Destructor Documentation

7.18.2.1 BeamSystem() [1/3]

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

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

7.18.2.2 ~BeamSystem()

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

7.18.2.3 BeamSystem() [2/3]

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

7.18.2.4 BeamSystem() [3/3]

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

7.18.3 Member Function Documentation

7.18.3.1 operator=() [1/2]

```
BeamSystem & cli::BeamSystem::operator= (
    BeamSystem && ) [delete]
```

7.18.3.2 operator=() [2/2]

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

7.18.3.3 update()

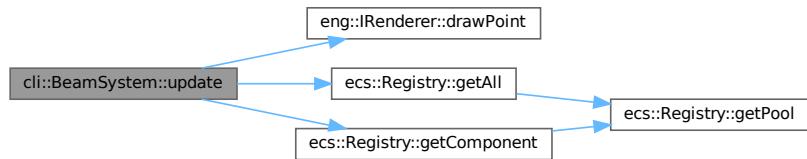
```
void cli::BeamSystem::update (
    ecs::Registry & registry,
    float )  [inline], [override], [virtual]
```

Implements [eng::ISystem](#).

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

References [cli::GameConfig::Beam::BAR_HEIGHT](#), [cli::GameConfig::Beam::BAR_WIDTH](#), [eng::IRenderer::drawPoint\(\)](#), [ecs::Registry::getAll\(\)](#), [ecs::Registry::getComponent\(\)](#), [m_renderer](#), [eng::Color::r](#), [cli::GameConfig::Player::SPRITE_WIDTH](#) and [ecs::Transform::x](#).

Here is the call graph for this function:



7.18.4 Member Data Documentation

7.18.4.1 m_renderer

```
eng::IRenderer& cli::BeamSystem::m_renderer  [private]
```

Definition at line 98 of file [Beam.hpp](#).

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

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

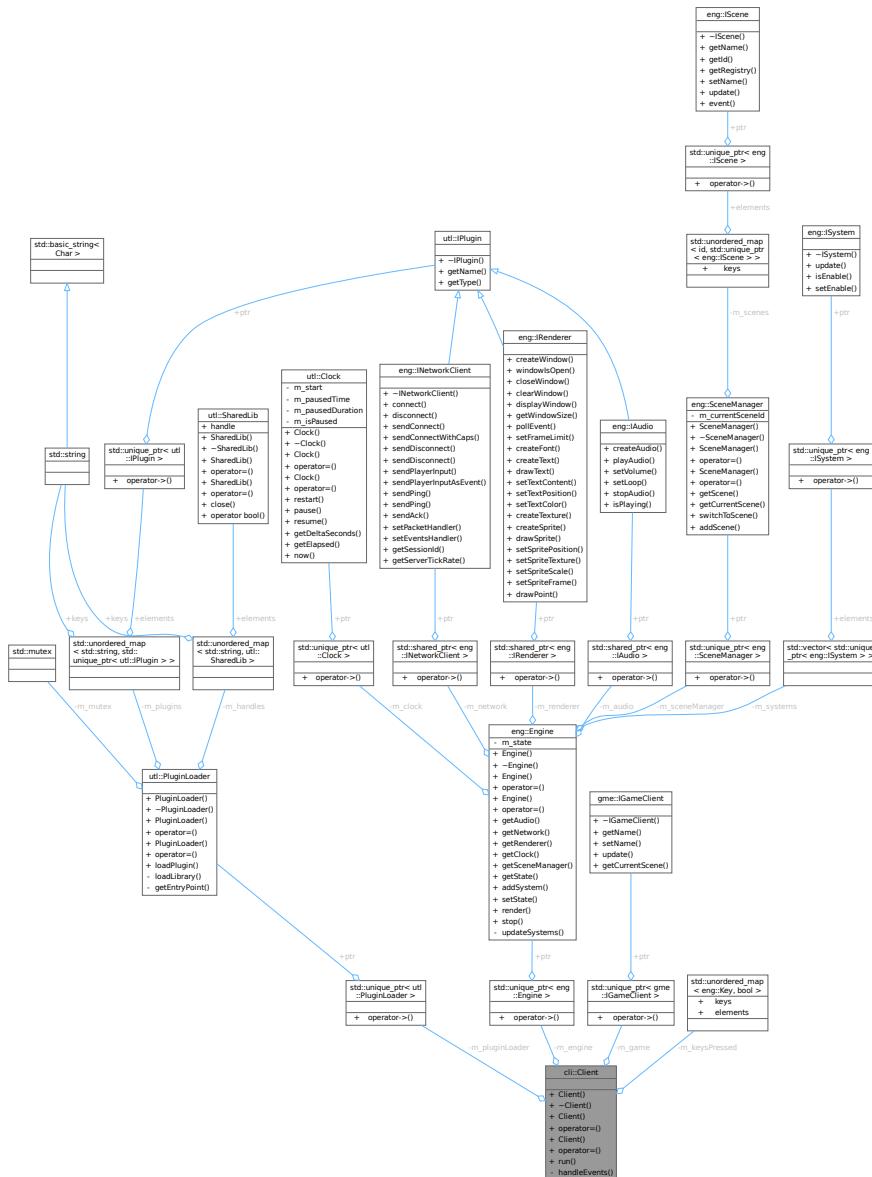
- /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/[Beam.hpp](#)

7.19 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)`

Private Attributes

- std::unique_ptr< `utl::PluginLoader` > `m_pluginLoader`
- std::unique_ptr< `eng::Engine` > `m_engine`
- std::unique_ptr< `gme::IGameClient` > `m_game`
- std::unordered_map< `eng::Key`, bool > `m_keysPressed`

7.19.1 Detailed Description

Class for the client.

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

7.19.2 Constructor & Destructor Documentation

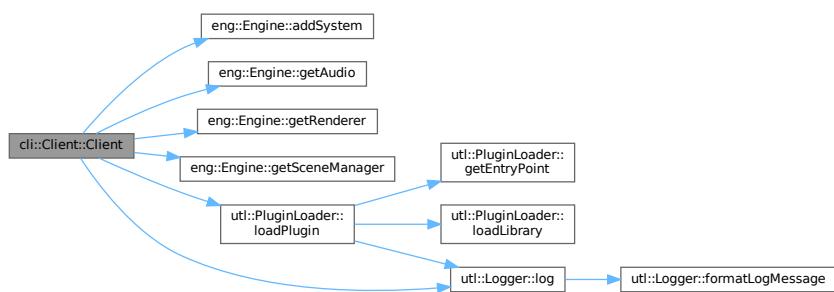
7.19.2.1 Client() [1/3]

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

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

References `eng::Engine::addSystem()`, `cli::ArgsConfig::audio_lib_path`, `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::PluginLoader::loadPlugin()`, `utl::Logger::log()`, `m_engine`, `m_pluginLoader`, `cli::ArgsConfig::network_lib_path`, `cli::Path::Plugin::PLUGIN_AUDIO`, `cli::Path::Plugin::PLUGIN_NETWORK_ASIO_CLIENT`, `cli::Path::Plugin::PLUGIN_RENDERER_SFML`, `PROJECT_NAME`, `PROJECT_VERSION`, `cli::ArgsConfig::renderer_lib_path`, and `cli::ArgsConfig::width`.

Here is the call graph for this function:



7.19.2.2 ~Client()

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

7.19.2.3 Client() [2/3]

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

7.19.2.4 Client() [3/3]

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

7.19.3 Member Function Documentation

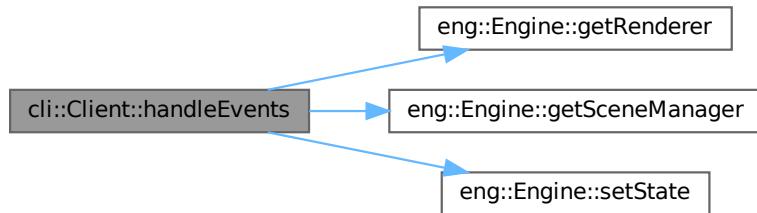
7.19.3.1 handleEvents()

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

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

References [eng::Closed](#), [eng::Engine::getRenderer\(\)](#), [eng::Engine::getSceneManager\(\)](#), [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:



7.19.3.2 operator=() [1/2]

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

7.19.3.3 operator=() [2/2]

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

7.19.3.4 run()

```
void cli::Client::run ()
```

Definition at line [76](#) of file [client.cpp](#).

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

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

Here is the caller graph for this function:



7.19.4 Member Data Documentation

7.19.4.1 m_engine

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

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

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

7.19.4.2 m_game

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

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

7.19.4.3 m_keysPressed

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

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

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

7.19.4.4 m_pluginLoader

`std::unique_ptr<utl::PluginLoader> cli::Client::m_pluginLoader [private]`

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

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

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](#)

7.20 utl::Clock Class Reference

Class for clock.

#include <[Clock.hpp](#)>

Collaboration diagram for utl::Clock:

utl::Clock	
-	m_start
-	m_pausedTime
-	m_pausedDuration
-	m_isPaused
+	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)

7.20.1 Detailed Description

Class for clock.

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

7.20.2 Member Typedef Documentation

7.20.2.1 Duration

```
using utl::Clock::Duration = std::chrono::high_resolution_clock::duration [private]
```

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

7.20.2.2 TimePoint

```
using utl::Clock::TimePoint = std::chrono::time_point<std::chrono::high_resolution_clock>
```

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

7.20.3 Constructor & Destructor Documentation

7.20.3.1 Clock() [1/3]

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

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

7.20.3.2 ~Clock()

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

7.20.3.3 Clock() [2/3]

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

7.20.3.4 Clock() [3/3]

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

7.20.4 Member Function Documentation

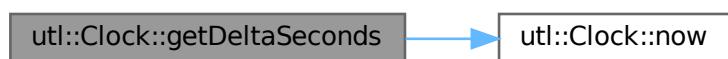
7.20.4.1 getDeltaSeconds()

```
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:



7.20.4.2 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:



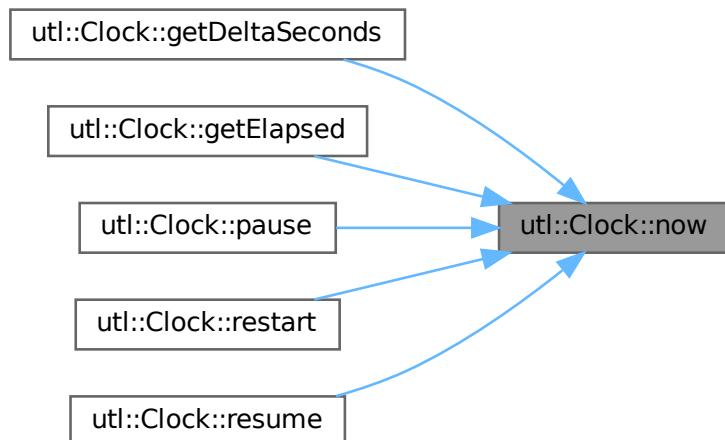
7.20.4.3 now()

```
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:



7.20.4.4 operator=() [1/2]

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

7.20.4.5 operator=() [2/2]

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

7.20.4.6 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:



7.20.4.7 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:



7.20.4.8 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:



7.20.5 Friends And Related Symbol Documentation

7.20.5.1 operator<<

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

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

7.20.6 Member Data Documentation

7.20.6.1 m_isPaused

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

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

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

7.20.6.2 m_pausedDuration

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

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

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

7.20.6.3 m_pausedTime

`TimePoint` `utl::Clock::m_pausedTime` [private]

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

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

7.20.6.4 m_start

`TimePoint` `utl::Clock::m_start` [private]

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

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

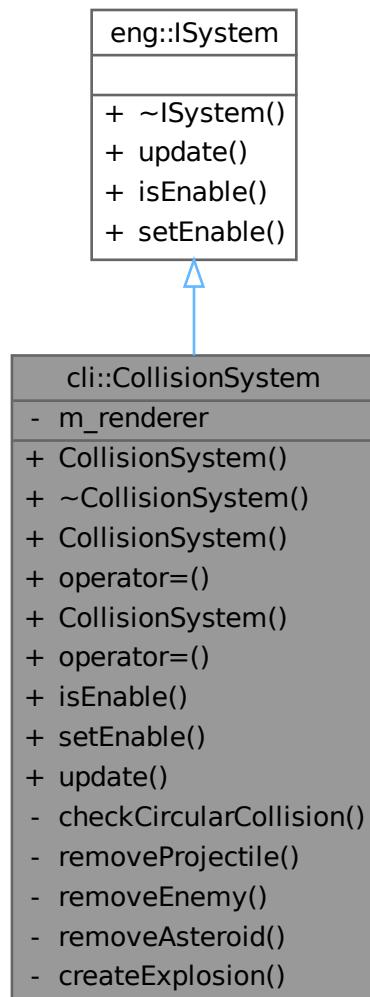
The documentation for this class was generated from the following file:

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

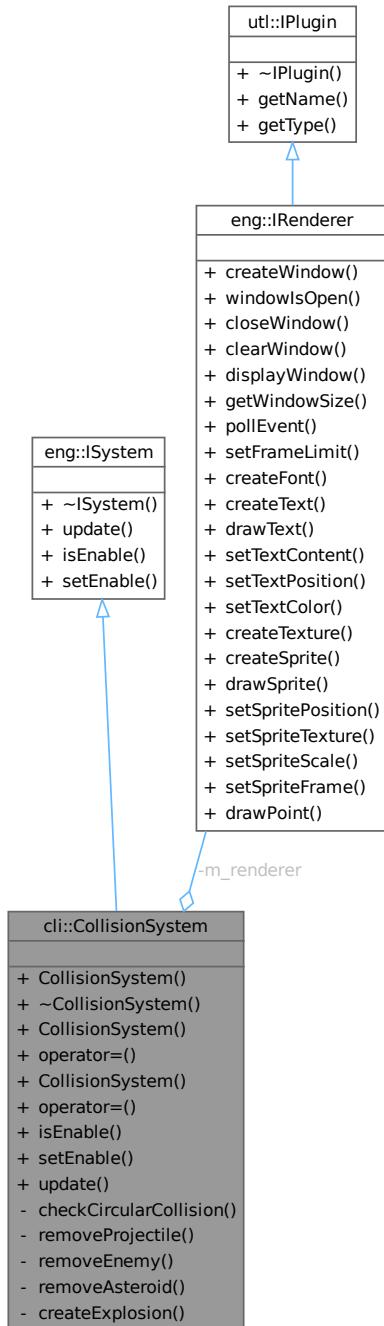
7.21 cli::CollisionSystem Class Reference

#include <Collision.hpp>

Inheritance diagram for cli::CollisionSystem:



Collaboration diagram for cli::CollisionSystem:



Public Member Functions

- `CollisionSystem (eng::IRenderer &renderer)`
- `~CollisionSystem () override=default`
- `CollisionSystem (const CollisionSystem &)=delete`
- `CollisionSystem & operator= (const CollisionSystem &)=delete`
- `CollisionSystem (CollisionSystem &&)=delete`

- `CollisionSystem & operator= (CollisionSystem &&) = delete`
- `bool isEnabled () override`
- `void setEnable (bool enable) override`
- `void update (ecs::Registry ®istry, float dt) override`

Public Member Functions inherited from `eng::ISystem`

- `virtual ~ISystem () = default`

Private Member Functions

- `bool checkCircularCollision (const ecs::Transform &transform1, const ecs::Hitbox &hitbox1, const ecs::Transform &transform2, const ecs::Hitbox &hitbox2)`
- `void removeProjectile (ecs::Registry ®istry, ecs::Entity entity)`
- `void removeEnemy (ecs::Registry ®istry, ecs::Entity entity)`
- `void removeAsteroid (ecs::Registry ®istry, ecs::Entity entity)`
- `void createExplosion (ecs::Registry ®istry, float x, float y)`

Private Attributes

- `eng::IRenderer & m_renderer`

7.21.1 Detailed Description

Definition at line 20 of file `Collision.hpp`.

7.21.2 Constructor & Destructor Documentation

7.21.2.1 CollisionSystem() [1/3]

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

Definition at line 23 of file `Collision.hpp`.

7.21.2.2 ~CollisionSystem()

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

7.21.2.3 CollisionSystem() [2/3]

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

7.21.2.4 CollisionSystem() [3/3]

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

7.21.3 Member Function Documentation

7.21.3.1 checkCircularCollision()

```
bool cli::CollisionSystem::checkCircularCollision (
    const ecs::Transform & transform1,
    const ecs::Hitbox & hitbox1,
    const ecs::Transform & transform2,
    const ecs::Hitbox & hitbox2) [inline], [private]
```

Definition at line 119 of file [Collision.hpp](#).

References [ecs::Hitbox::radius](#), [ecs::Transform::x](#), and [ecs::Transform::y](#).

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

Here is the caller graph for this function:



7.21.3.2 createExplosion()

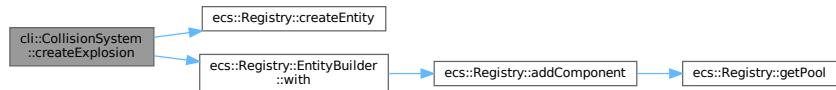
```
void cli::CollisionSystem::createExplosion (
    ecs::Registry & registry,
    float x,
    float y) [inline], [private]
```

Definition at line 188 of file [Collision.hpp](#).

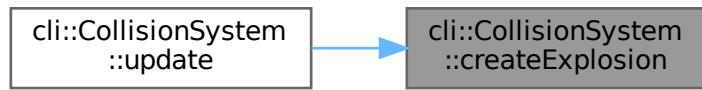
References [cli::GameConfig::Explosion::ANIMATION_DURATION](#), [cli::GameConfig::Explosion::ANIMATION_FRAME](#), [ecs::Registry::createEntity\(\)](#), [cli::GameConfig::Explosion::FRAMES_PER_ROW](#), [cli::GameConfig::Explosion::LIFETIME](#), [cli::GameConfig::Explosion::SCALE](#), [cli::GameConfig::Explosion::SPRITE_HEIGHT](#), [cli::GameConfig::Explosion::SPRITE](#), [cli::Path::Texture::TEXTURE_EXPLOSION](#), and [ecs::Registry::EntityBuilder::with\(\)](#).

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

Here is the call graph for this function:



Here is the caller graph for this function:



7.21.3.3 isEnabled()

`bool cli::CollisionSystem::isEnabled () [inline], [override], [virtual]`

Implements [eng::ISystem](#).

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

7.21.3.4 operator=() [1/2]

```
CollisionSystem & cli::CollisionSystem::operator= (
    CollisionSystem &&) [delete]
```

7.21.3.5 operator=() [2/2]

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

7.21.3.6 removeAsteroid()

```
void cli::CollisionSystem::removeAsteroid (
    ecs::Registry & registry,
    ecs::Entity entity) [inline], [private]
```

Definition at line 170 of file [Collision.hpp](#).

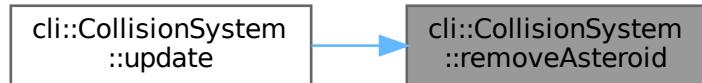
References [ecs::Registry::hasComponent\(\)](#), and [ecs::Registry::removeComponent\(\)](#).

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

Here is the call graph for this function:



Here is the caller graph for this function:



7.21.3.7 removeEnemy()

```
void cli::CollisionSystem::removeEnemy (
    ecs::Registry & registry,
    ecs::Entity entity) [inline], [private]
```

Definition at line 150 of file [Collision.hpp](#).

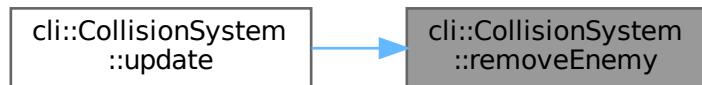
References [ecs::Registry::hasComponent\(\)](#), and [ecs::Registry::removeComponent\(\)](#).

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

Here is the call graph for this function:



Here is the caller graph for this function:



7.21.3.8 removeProjectile()

```
void cli::CollisionSystem::removeProjectile (
    ecs::Registry & registry,
    ecs::Entity entity) [inline], [private]
```

Definition at line 130 of file [Collision.hpp](#).

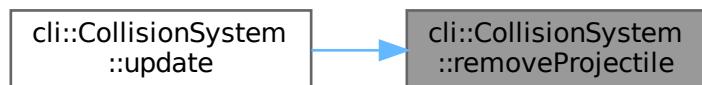
References [ecs::Registry::hasComponent\(\)](#), and [ecs::Registry::removeComponent\(\)](#).

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

Here is the call graph for this function:



Here is the caller graph for this function:



7.21.3.9 setEnable()

```
void cli::CollisionSystem::setEnable (
    bool enable) [inline], [override], [virtual]
```

Implements [eng::ISystem](#).

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

7.21.3.10 update()

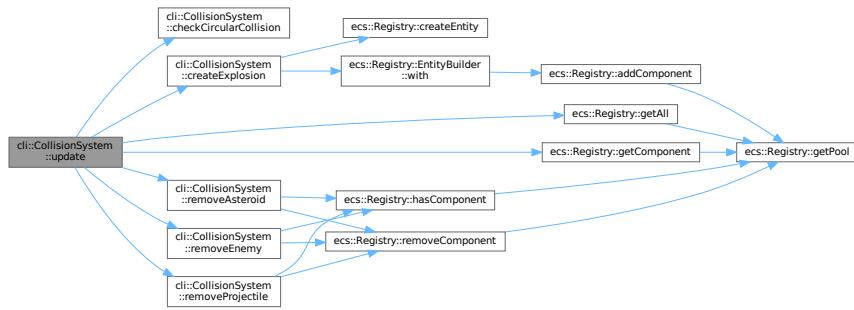
```
void cli::CollisionSystem::update (
    ecs::Registry & registry,
    float dt) [inline], [override], [virtual]
```

Implements [eng::ISystem](#).

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

References [checkCircularCollision\(\)](#), [createExplosion\(\)](#), [ecs::Registry::getAll\(\)](#), [ecs::Registry::getComponent\(\)](#), [removeAsteroid\(\)](#), [removeEnemy\(\)](#), and [removeProjectile\(\)](#).

Here is the call graph for this function:



7.21.4 Member Data Documentation

7.21.4.1 m_renderer

```
eng::IRenderer& cli::CollisionSystem::m_renderer [private]
```

Definition at line 117 of file [Collision.hpp](#).

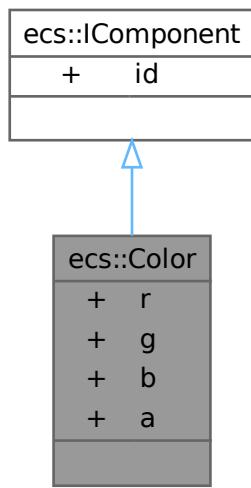
The documentation for this class was generated from the following file:

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

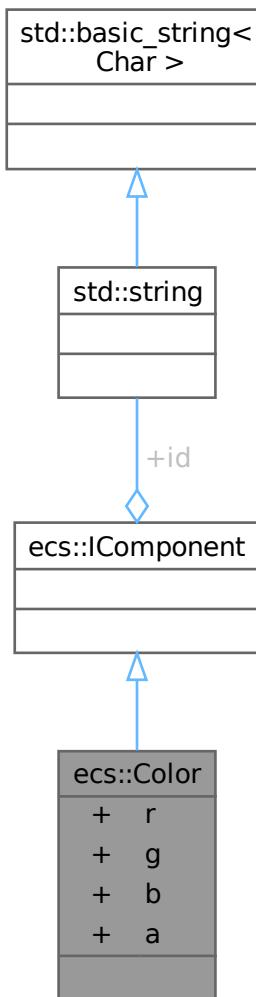
7.22 ecs::Color Struct Reference

```
#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`

7.22.1 Detailed Description

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

7.22.2 Member Data Documentation

7.22.2.1 a

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

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

7.22.2.2 b

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

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

7.22.2.3 g

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

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

7.22.2.4 r

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

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

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

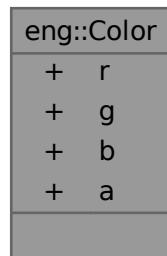
The documentation for this struct was generated from the following file:

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

7.23 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`

7.23.1 Detailed Description

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

7.23.2 Member Data Documentation

7.23.2.1 `a`

unsigned char eng::Color::`a`

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

Referenced by [cli::Game::Game\(\)](#), [cli::Lobby::Lobby\(\)](#), and [cli::Settings::Settings\(\)](#).

7.23.2.2 `b`

unsigned char eng::Color::`b`

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

Referenced by [cli::Game::Game\(\)](#), [cli::Lobby::Lobby\(\)](#), and [cli::Settings::Settings\(\)](#).

7.23.2.3 `g`

unsigned char eng::Color::`g`

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

Referenced by [cli::Game::Game\(\)](#), [cli::Lobby::Lobby\(\)](#), and [cli::Settings::Settings\(\)](#).

7.23.2.4 `r`

unsigned char eng::Color::`r`

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

Referenced by [cli::Game::Game\(\)](#), [cli::Lobby::Lobby\(\)](#), [cli::Settings::Settings\(\)](#), and [cli::BeamSystem::update\(\)](#).

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

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

7.24 cli::SpriteRect::SpriteSheet::Enemy Struct Reference

```
#include <SpriteRect.hpp>
```

Collaboration diagram for cli::SpriteRect::SpriteSheet::Enemy:

cli::SpriteRect::SpriteSheet ::Enemy	
+ frameW	
+ frameH	
+ marginX	
+ marginY	
+ spacingX	
+ spacingY	
+ totalFrames	

Static Public Attributes

- static constexpr int `frameW` = 16
- static constexpr int `frameH` = 16
- static constexpr int `marginX` = 1
- static constexpr int `marginY` = 1
- static constexpr int `spacingX` = 1
- static constexpr int `spacingY` = 0
- static constexpr int `totalFrames` = 8

7.24.1 Detailed Description

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

7.24.2 Member Data Documentation

7.24.2.1 frameH

```
int cli::SpriteRect::SpriteSheet::Enemy::frameH = 16 [static], [constexpr]
```

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

Referenced by [cli::SpriteRect::enemyRect\(\)](#).

7.24.2.2 frameW

```
int cli::SpriteRect::SpriteSheet::Enemy::frameW = 16 [static], [constexpr]
```

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

Referenced by [cli::SpriteRect::enemyRect\(\)](#).

7.24.2.3 marginX

```
int cli::SpriteRect::SpriteSheet::Enemy::marginX = 1 [static], [constexpr]
```

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

Referenced by [cli::SpriteRect::enemyRect\(\)](#).

7.24.2.4 marginY

```
int cli::SpriteRect::SpriteSheet::Enemy::marginY = 1 [static], [constexpr]
```

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

Referenced by [cli::SpriteRect::enemyRect\(\)](#).

7.24.2.5 spacingX

```
int cli::SpriteRect::SpriteSheet::Enemy::spacingX = 1 [static], [constexpr]
```

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

Referenced by [cli::SpriteRect::enemyRect\(\)](#).

7.24.2.6 spacingY

```
int cli::SpriteRect::SpriteSheet::Enemy::spacingY = 0 [static], [constexpr]
```

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

Referenced by [cli::SpriteRect::enemyRect\(\)](#).

7.24.2.7 totalFrames

```
int cli::SpriteRect::SpriteSheet::Enemy::totalFrames = 8 [static], [constexpr]
```

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

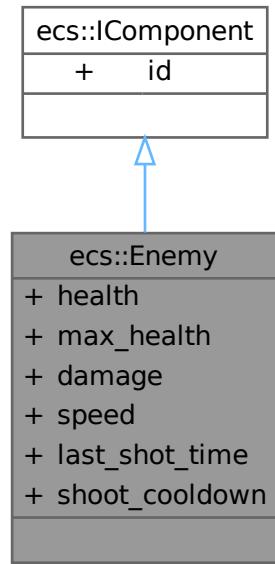
The documentation for this struct was generated from the following file:

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

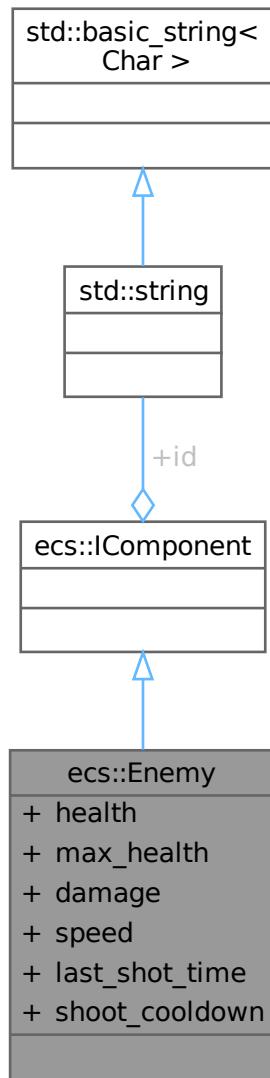
7.25 ecs::Enemy Struct Reference

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

Inheritance diagram for ecs::Enemy:



Collaboration diagram for ecs::Enemy:



Public Attributes

- float `health`
- float `max_health`
- float `damage`
- float `speed`
- float `last_shot_time`
- float `shoot_cooldown`

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

- std::string `id`

7.25.1 Detailed Description

Definition at line 116 of file [Component.hpp](#).

7.25.2 Member Data Documentation

7.25.2.1 damage

```
float ecs::Enemy::damage
```

Definition at line 120 of file [Component.hpp](#).

7.25.2.2 health

```
float ecs::Enemy::health
```

Definition at line 118 of file [Component.hpp](#).

7.25.2.3 last_shot_time

```
float ecs::Enemy::last_shot_time
```

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

7.25.2.4 max_health

```
float ecs::Enemy::max_health
```

Definition at line 119 of file [Component.hpp](#).

7.25.2.5 shoot_cooldown

```
float ecs::Enemy::shoot_cooldown
```

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

7.25.2.6 speed

```
float ecs::Enemy::speed
```

Definition at line 121 of file [Component.hpp](#).

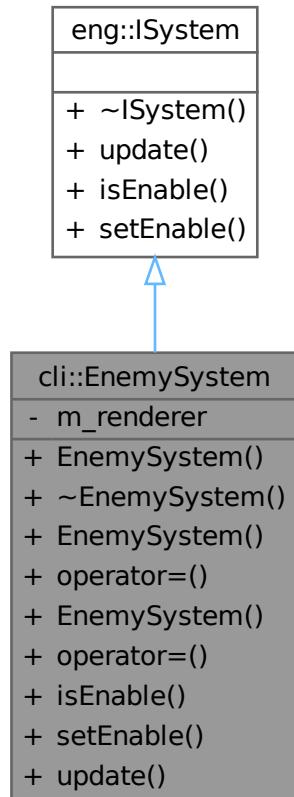
The documentation for this struct was generated from the following file:

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

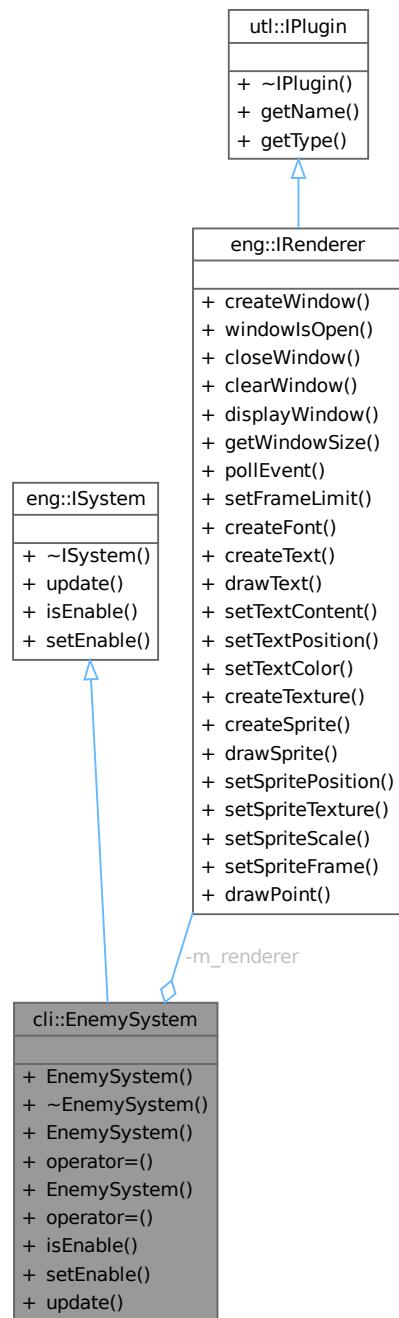
7.26 cli::EnemySystem Class Reference

```
#include <Enemy.hpp>
```

Inheritance diagram for cli::EnemySystem:



Collaboration diagram for cli::EnemySystem:



Public Member Functions

- `EnemySystem (eng::IRenderer &renderer)`
- `~EnemySystem ()` override=default
- `EnemySystem (const EnemySystem &)=delete`
- `EnemySystem & operator= (const EnemySystem &)=delete`
- `EnemySystem (EnemySystem &&)=delete`

- `EnemySystem & operator= (EnemySystem &&) = delete`
- `bool isEnabled () override`
- `void setEnable (bool enable) override`
- `void update (ecs::Registry ®istry, float dt) override`

Public Member Functions inherited from `eng::ISystem`

- `virtual ~ISystem () = default`

Private Attributes

- `eng::IRenderer & m_renderer`

7.26.1 Detailed Description

Definition at line 20 of file `Enemy.hpp`.

7.26.2 Constructor & Destructor Documentation

7.26.2.1 `EnemySystem()` [1/3]

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

Definition at line 23 of file `Enemy.hpp`.

7.26.2.2 `~EnemySystem()`

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

7.26.2.3 `EnemySystem()` [2/3]

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

7.26.2.4 `EnemySystem()` [3/3]

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

7.26.3 Member Function Documentation

7.26.3.1 `isEnabled()`

```
bool cli::EnemySystem::isEnabled () [inline], [override], [virtual]
```

Implements `eng::ISystem`.

Definition at line 31 of file `Enemy.hpp`.

7.26.3.2 operator=() [1/2]

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

7.26.3.3 operator=() [2/2]

```
EnemySystem & cli::EnemySystem::operator= (
    EnemySystem && ) [delete]
```

7.26.3.4 setEnable()

```
void cli::EnemySystem::setEnable (
    bool enable) [inline], [override], [virtual]
```

Implements [eng::ISystem](#).

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

7.26.3.5 update()

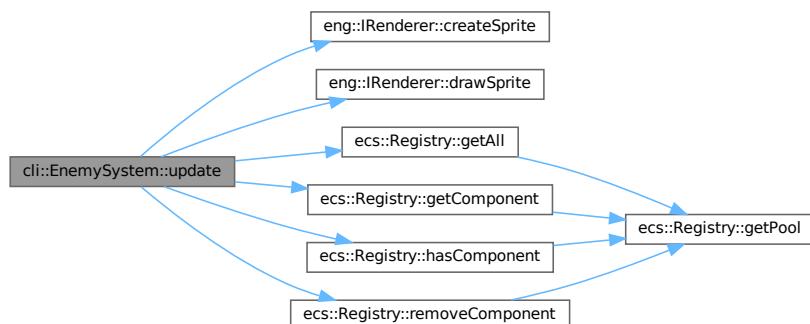
```
void cli::EnemySystem::update (
    ecs::Registry & registry,
    float dt) [inline], [override], [virtual]
```

Implements [eng::ISystem](#).

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

References [eng::IRenderer::createSprite\(\)](#), [eng::IRenderer::drawSprite\(\)](#), [ecs::Registry::getAll\(\)](#), [ecs::Registry::getComponent\(\)](#), [ecs::Registry::hasComponent\(\)](#), [ecs::IComponent::id](#), [m_renderer](#), [cli::GameConfig::Screen::REMOVE_MAX_Y](#), [cli::GameConfig::Screen::REMOVE_MIN_Y](#), [cli::GameConfig::Screen::REMOVE_X](#), [ecs::Registry::removeComponent\(\)](#), [cli::GameConfig::Enemy::Easy::SPRITE_WIDTH](#), [ecs::Transform::x](#), and [ecs::Transform::y](#).

Here is the call graph for this function:



7.26.4 Member Data Documentation

7.26.4.1 m_renderer

`eng::IRenderer& cli::EnemySystem::m_renderer [private]`

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

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

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

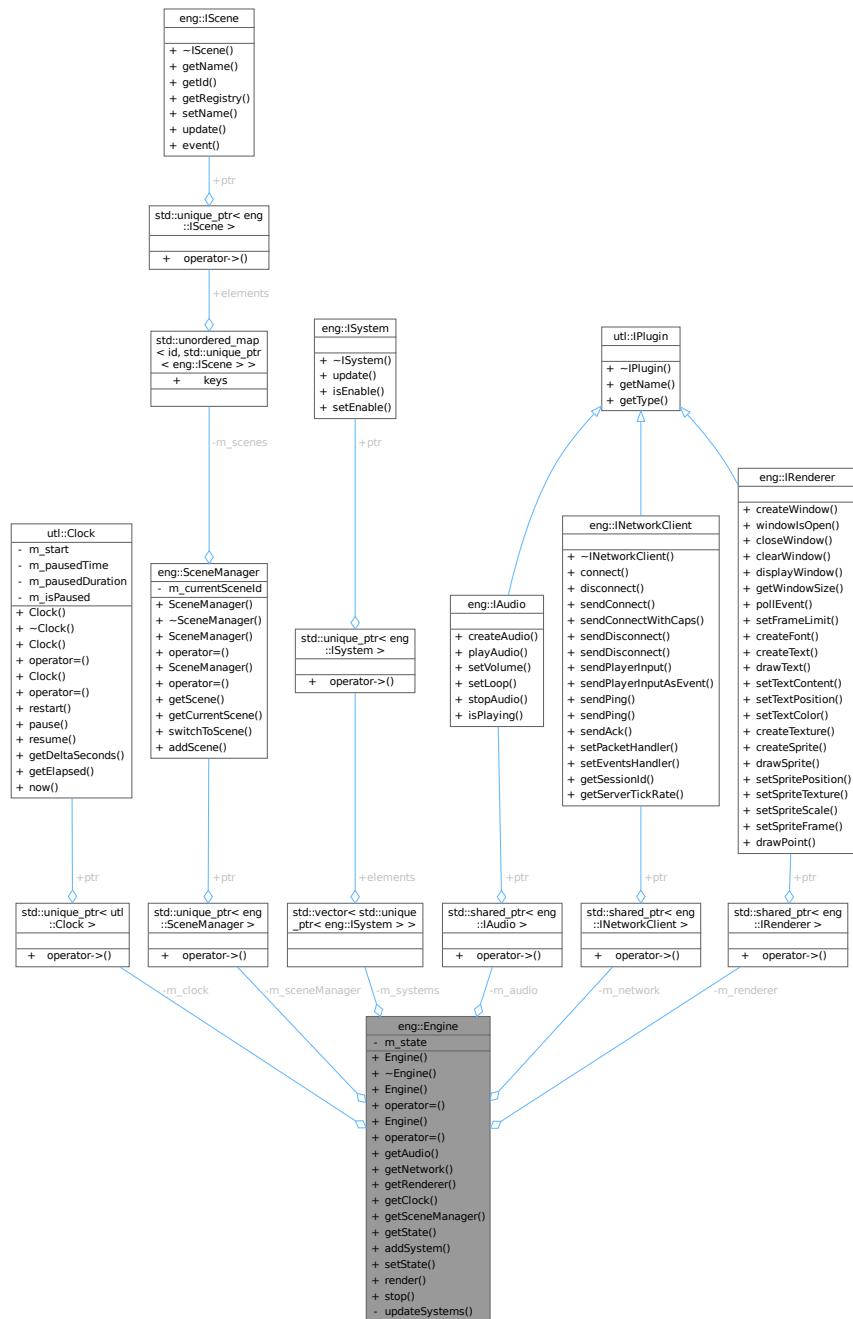
- /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/[Enemy.hpp](#)

7.27 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::shared_ptr< IAudio >()> &audioFactory, const std::function< std::shared_ptr< INetworkClient >()> &networkFactory, const std::function< std::shared_ptr< IRenderer >()> &rendererFactory)`
- `~Engine ()=default`
- `Engine (const Engine &)=delete`
- `Engine & operator=(const Engine &)=delete`
- `Engine (Engine &&)=delete`

- Engine & operator= (Engine &&) = delete
- std::shared_ptr< IAudio > & getAudio ()
- std::shared_ptr< INetworkClient > & getNetwork ()
- std::shared_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::shared_ptr< IAudio > m_audio
- std::shared_ptr< INetworkClient > m_network
- std::shared_ptr< IRenderer > m_renderer

7.27.1 Detailed Description

Class for the game engine.

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

7.27.2 Constructor & Destructor Documentation

7.27.2.1 Engine() [1/3]

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

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

7.27.2.2 ~Engine()

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

7.27.2.3 Engine() [2/3]

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

7.27.2.4 Engine() [3/3]

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

7.27.3 Member Function Documentation

7.27.3.1 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:



7.27.3.2 getAudio()

```
std::shared_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:



7.27.3.3 getClock()

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

Definition at line 52 of file [Engine.hpp](#).

References [m_clock](#).

7.27.3.4 getNetwork()

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

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

References [m_network](#).

7.27.3.5 getRenderer()

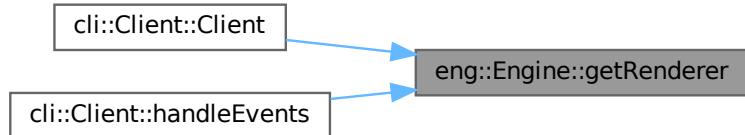
`std::shared_ptr< IRenderer > & eng::Engine::getRenderer () [inline]`

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:



7.27.3.6 getSceneManager()

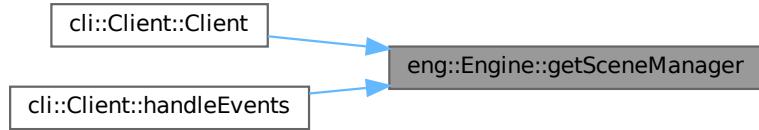
`std::unique_ptr< SceneManager > & eng::Engine::getSceneManager () [inline]`

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:



7.27.3.7 getState()

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

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

References [m_state](#).

7.27.3.8 operator=() [1/2]

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

7.27.3.9 operator=() [2/2]

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

7.27.3.10 render()

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

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

7.27.3.11 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:



7.27.3.12 stop()

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

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

References [m_renderer](#).

7.27.3.13 updateSystems()

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

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

7.27.4 Member Data Documentation

7.27.4.1 m_audio

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

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

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

7.27.4.2 m_clock

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

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

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

7.27.4.3 m_network

```
std::shared_ptr<INetworkClient> eng::Engine::m_network [private]
```

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

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

7.27.4.4 m_renderer

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

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

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

7.27.4.5 m_sceneManager

std::unique_ptr<[SceneManager](#)> eng::Engine::m_sceneManager [private]

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

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

7.27.4.6 m_state

[State](#) eng::Engine::m_state = [RUN](#) [private]

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

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

7.27.4.7 m_systems

std::vector<std::unique_ptr<[ISystem](#)> > eng::Engine::m_systems [private]

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

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

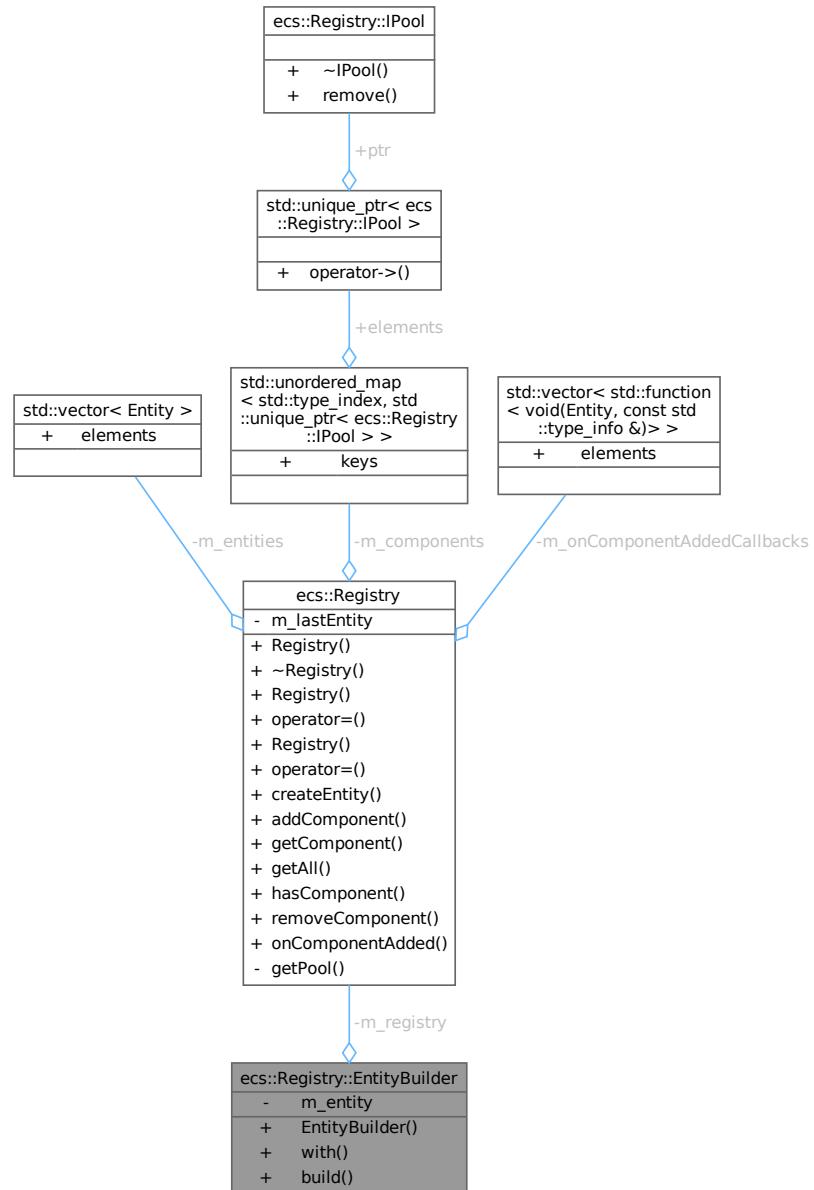
The documentation for this class was generated from the following files:

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

7.28 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`

7.28.1 Detailed Description

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

7.28.2 Constructor & Destructor Documentation

7.28.2.1 EntityBuilder()

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

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

7.28.3 Member Function Documentation

7.28.3.1 build()

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

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

References [m_entity](#).

7.28.3.2 with()

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

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

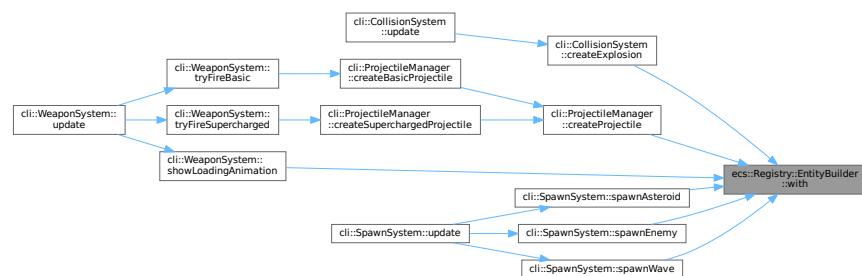
References [ecs::Registry::addComponent\(\)](#), [m_entity](#), and [m_registry](#).

Referenced by [cli::CollisionSystem::createExplosion\(\)](#), [cli::ProjectileManager::createProjectile\(\)](#), [cli::WeaponSystem::showLoadingAnimation\(\)](#), [cli::SpawnSystem::spawnAsteroid\(\)](#), [cli::SpawnSystem::spawnEnemy\(\)](#), and [cli::SpawnSystem::spawnWave\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.28.4 Member Data Documentation

7.28.4.1 m_entity

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

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

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

7.28.4.2 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:

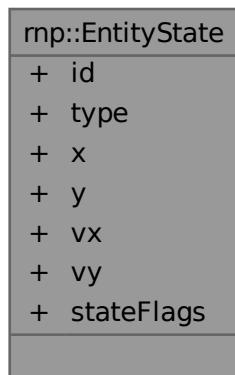
- /home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/[Registry.hpp](#)

7.29 rnp:: EntityState Struct Reference

Entity state for WORLD_STATE packet.

```
#include <Protocol.hpp>
```

Collaboration diagram for rnp:: EntityState:



Public Attributes

- std::uint32_t **id**
- std::uint16_t **type**
- float **x**
- float **y**
- float **vx**
- float **vy**
- std::uint8_t **stateFlags**

7.29.1 Detailed Description

Entity state for WORLD_STATE packet.

Definition at line 156 of file [Protocol.hpp](#).

7.29.2 Member Data Documentation

7.29.2.1 id

std::uint32_t rnp:: EntityState::id

Definition at line 158 of file [Protocol.hpp](#).

7.29.2.2 stateFlags

std::uint8_t rnp:: EntityState::stateFlags

Definition at line 162 of file [Protocol.hpp](#).

7.29.2.3 type

std::uint16_t rnp:: EntityState::type

Definition at line 159 of file [Protocol.hpp](#).

7.29.2.4 vx

float rnp:: EntityState::vx

Definition at line 161 of file [Protocol.hpp](#).

7.29.2.5 vy

float rnp:: EntityState::vy

Definition at line 161 of file [Protocol.hpp](#).

7.29.2.6 x

```
float rnp:: EntityState::x
```

Definition at line 160 of file [Protocol.hpp](#).

7.29.2.7 y

```
float rnp:: EntityState::y
```

Definition at line 160 of file [Protocol.hpp](#).

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

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

7.30 cli::EnvConfig Struct Reference

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

Collaboration diagram for cli::EnvConfig:



7.30.1 Detailed Description

Definition at line 33 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](#)

7.31 srv::EnvConfig Struct Reference

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

Collaboration diagram for srv::EnvConfig:



7.31.1 Detailed Description

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

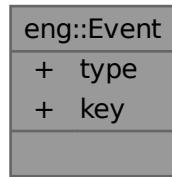
The documentation for this struct was generated from the following file:

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

7.32 eng::Event Struct Reference

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

Collaboration diagram for eng::Event:



Public Attributes

- `EventType type = EventType::None`
- `Key key = Key::Unknown`

7.32.1 Detailed Description

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

7.32.2 Member Data Documentation

7.32.2.1 key

`Key eng::Event::key = Key::Unknown`

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

Referenced by [cli::Game::event\(\)](#), [cli::Lobby::event\(\)](#), and [cli::Settings::event\(\)](#).

7.32.2.2 type

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

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

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

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

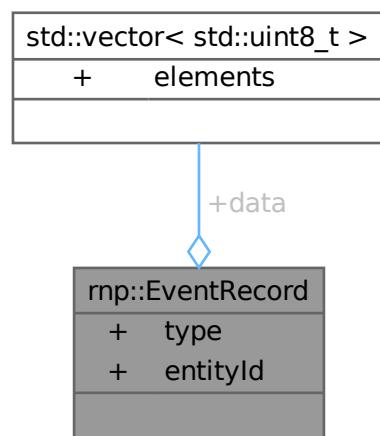
- /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/[IRenderer.hpp](#)

7.33 rnp::EventRecord Struct Reference

Event record for ENTITY_EVENT packets (TLV format)

#include <Protocol.hpp>

Collaboration diagram for rnp::EventRecord:



Public Attributes

- `EventType type`
- `std::uint32_t entityId`
- `std::vector< std::uint8_t > data`

7.33.1 Detailed Description

Event record for ENTITY_EVENT packets (TLV format)

Definition at line 103 of file [Protocol.hpp](#).

7.33.2 Member Data Documentation

7.33.2.1 data

`std::vector<std::uint8_t> rnp::EventRecord::data`

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

Referenced by [rnp::deserializeEvents\(\)](#).

7.33.2.2 entityId

`std::uint32_t rnp::EventRecord::entityId`

Definition at line 106 of file [Protocol.hpp](#).

7.33.2.3 type

`EventType rnp::EventRecord::type`

Definition at line 105 of file [Protocol.hpp](#).

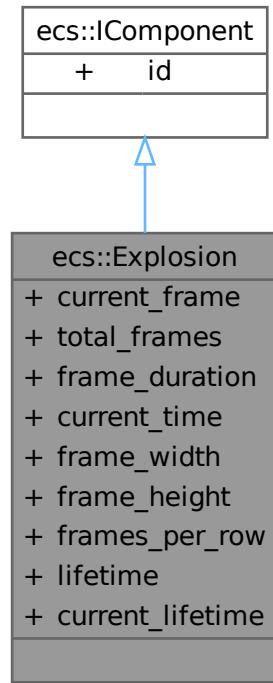
The documentation for this struct was generated from the following file:

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

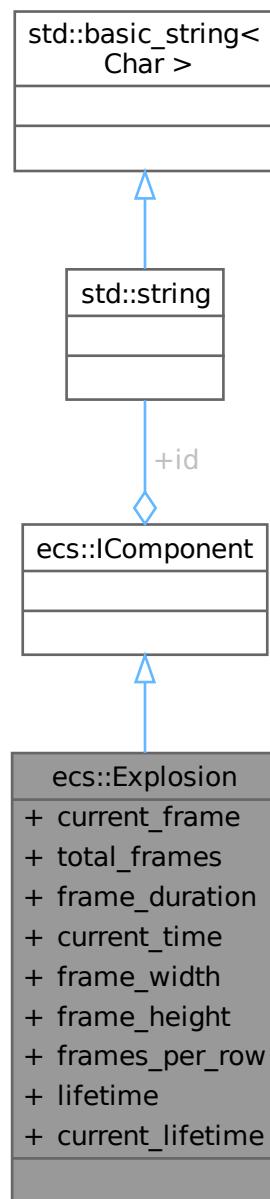
7.34 ecs::Explosion Struct Reference

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

Inheritance diagram for ecs::Explosion:



Collaboration diagram for ecs::Explosion:



Public Attributes

- int `current_frame`
- int `total_frames`
- float `frame_duration`
- float `current_time`
- float `frame_width`
- float `frame_height`
- int `frames_per_row`

- float `lifetime`
- float `current_lifetime`

Public Attributes inherited from `ecs::IComponent`

- std::string `id`

7.34.1 Detailed Description

Definition at line 139 of file `Component.hpp`.

7.34.2 Member Data Documentation

7.34.2.1 `current_frame`

int `ecs::Explosion::current_frame`

Definition at line 141 of file `Component.hpp`.

7.34.2.2 `current_lifetime`

float `ecs::Explosion::current_lifetime`

Definition at line 149 of file `Component.hpp`.

7.34.2.3 `current_time`

float `ecs::Explosion::current_time`

Definition at line 144 of file `Component.hpp`.

7.34.2.4 `frame_duration`

float `ecs::Explosion::frame_duration`

Definition at line 143 of file `Component.hpp`.

7.34.2.5 `frame_height`

float `ecs::Explosion::frame_height`

Definition at line 146 of file `Component.hpp`.

7.34.2.6 frame_width

```
float ecs::Explosion::frame_width
```

Definition at line 145 of file [Component.hpp](#).

7.34.2.7 frames_per_row

```
int ecs::Explosion::frames_per_row
```

Definition at line 147 of file [Component.hpp](#).

7.34.2.8 lifetime

```
float ecs::Explosion::lifetime
```

Definition at line 148 of file [Component.hpp](#).

7.34.2.9 total_frames

```
int ecs::Explosion::total_frames
```

Definition at line 142 of file [Component.hpp](#).

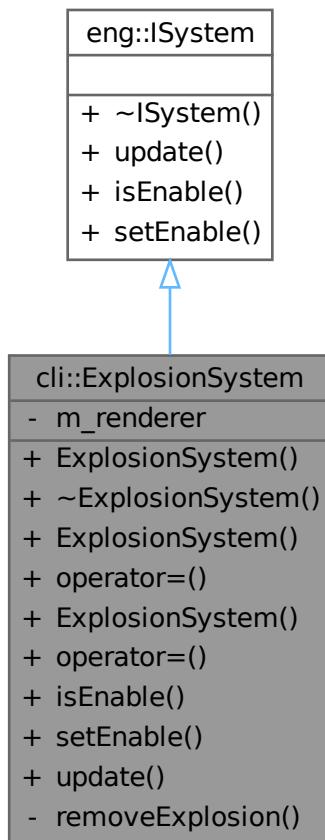
The documentation for this struct was generated from the following file:

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

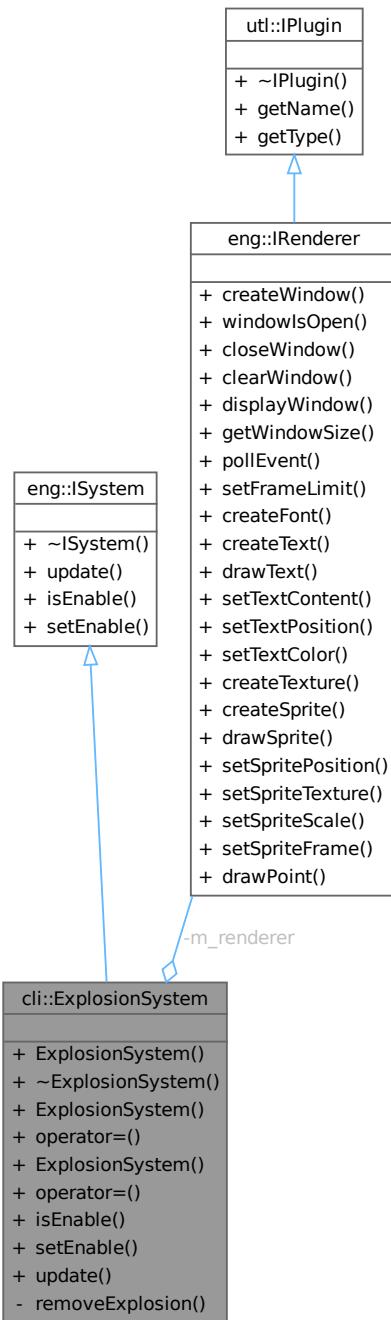
7.35 cli::ExplosionSystem Class Reference

```
#include <Explosion.hpp>
```

Inheritance diagram for cli::ExplosionSystem:



Collaboration diagram for cli::ExplosionSystem:



Public Member Functions

- `ExplosionSystem (eng::IRenderer &renderer)`
- `~ExplosionSystem () override=default`
- `ExplosionSystem (const ExplosionSystem &)=delete`
- `ExplosionSystem & operator= (const ExplosionSystem &)=delete`
- `ExplosionSystem (ExplosionSystem &&)=delete`

- `ExplosionSystem & operator= (ExplosionSystem &&) = delete`
- `bool isEnabled () override`
- `void setEnable (bool enable) override`
- `void update (ecs::Registry ®istry, float dt) override`

Public Member Functions inherited from `eng::ISystem`

- `virtual ~ISystem () = default`

Private Member Functions

- `void removeExplosion (ecs::Registry ®istry, ecs::Entity entity)`

Private Attributes

- `eng::IRenderer & m_renderer`

7.35.1 Detailed Description

Definition at line 18 of file `Explosion.hpp`.

7.35.2 Constructor & Destructor Documentation

7.35.2.1 `ExplosionSystem()` [1/3]

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

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

7.35.2.2 `~ExplosionSystem()`

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

7.35.2.3 `ExplosionSystem()` [2/3]

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

7.35.2.4 `ExplosionSystem()` [3/3]

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

7.35.3 Member Function Documentation

7.35.3.1 isEnabled()

```
bool cli::ExplosionSystem::isEnabled () [inline], [override], [virtual]
```

Implements [eng::ISystem](#).

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

7.35.3.2 operator=() [1/2]

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

7.35.3.3 operator=() [2/2]

```
ExplosionSystem & cli::ExplosionSystem::operator= (
    ExplosionSystem && ) [delete]
```

7.35.3.4 removeExplosion()

```
void cli::ExplosionSystem::removeExplosion (
    ecs::Registry & registry,
    ecs::Entity entity) [inline], [private]
```

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

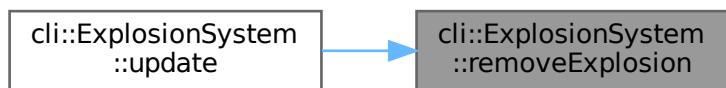
References [ecs::Registry::hasComponent\(\)](#), and [ecs::Registry::removeComponent\(\)](#).

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

Here is the call graph for this function:



Here is the caller graph for this function:



7.35.3.5 setEnable()

```
void cli::ExplosionSystem::setEnable (
    bool enable) [inline], [override], [virtual]
```

Implements [eng::ISystem](#).

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

7.35.3.6 update()

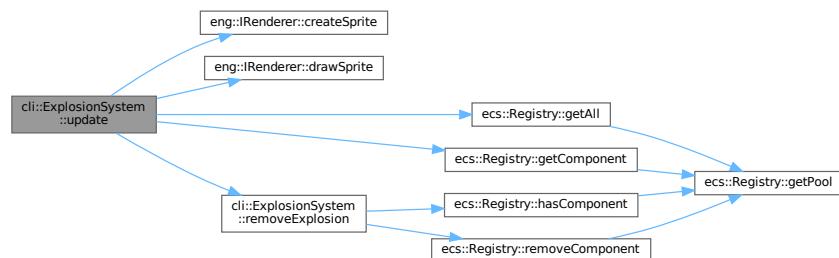
```
void cli::ExplosionSystem::update (
    ecs::Registry & registry,
    float dt) [inline], [override], [virtual]
```

Implements [eng::ISystem](#).

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

References [eng::IRenderer::createSprite\(\)](#), [eng::IRenderer::drawSprite\(\)](#), [ecs::Registry::getAll\(\)](#), [ecs::Registry::getComponent\(\)](#), [ecs::IComponent::id](#), [m_renderer](#), and [removeExplosion\(\)](#).

Here is the call graph for this function:



7.35.4 Member Data Documentation

7.35.4.1 m_renderer

```
eng::IRenderer& cli::ExplosionSystem::m_renderer [private]
```

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

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

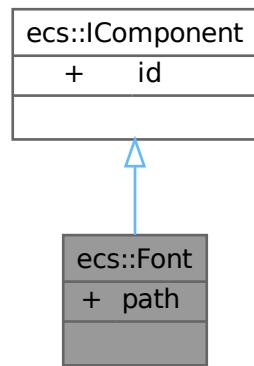
The documentation for this class was generated from the following file:

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

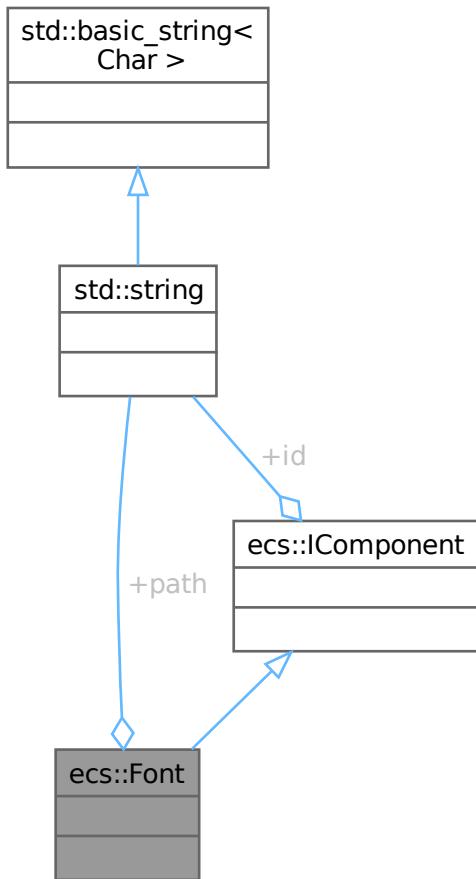
7.36 ecs::Font Struct Reference

```
#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`

7.36.1 Detailed Description

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

7.36.2 Member Data Documentation

7.36.2.1 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:

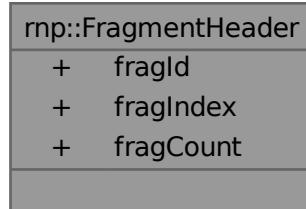
- /home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/[Component.hpp](#)

7.37 rnp::FragmentHeader Struct Reference

Fragmentation header (when FRAG flag is set)

```
#include <Protocol.hpp>
```

Collaboration diagram for rnp::FragmentHeader:



Public Attributes

- std::uint16_t **fragId**
- std::uint16_t **fragIndex**
- std::uint16_t **fragCount**

7.37.1 Detailed Description

Fragmentation header (when FRAG flag is set)

Definition at line 206 of file [Protocol.hpp](#).

7.37.2 Member Data Documentation

7.37.2.1 fragCount

```
std::uint16_t rnp::FragmentHeader::fragCount
```

Definition at line 210 of file [Protocol.hpp](#).

7.37.2.2 fragId

```
std::uint16_t rnp::FragmentHeader::fragId
```

Definition at line 208 of file [Protocol.hpp](#).

7.37.2.3 fragIndex

```
std::uint16_t rnp::FragmentHeader::fragIndex
```

Definition at line 209 of file [Protocol.hpp](#).

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

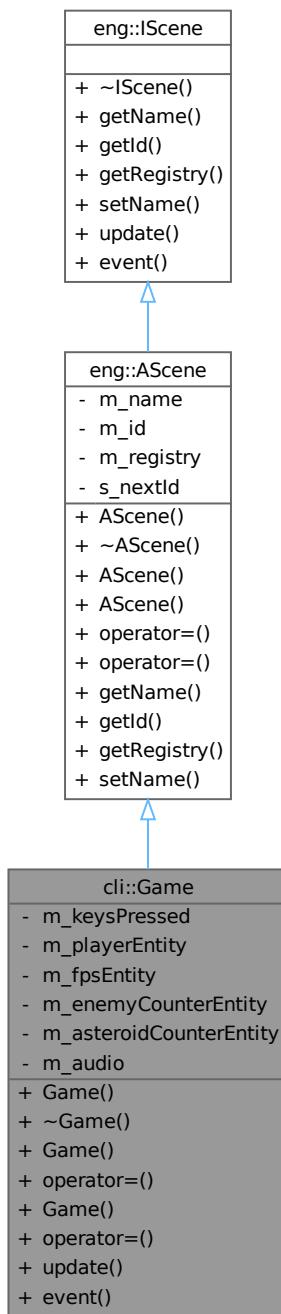
- /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/Protocol/[Protocol.hpp](#)

7.38 cli::Game Class Reference

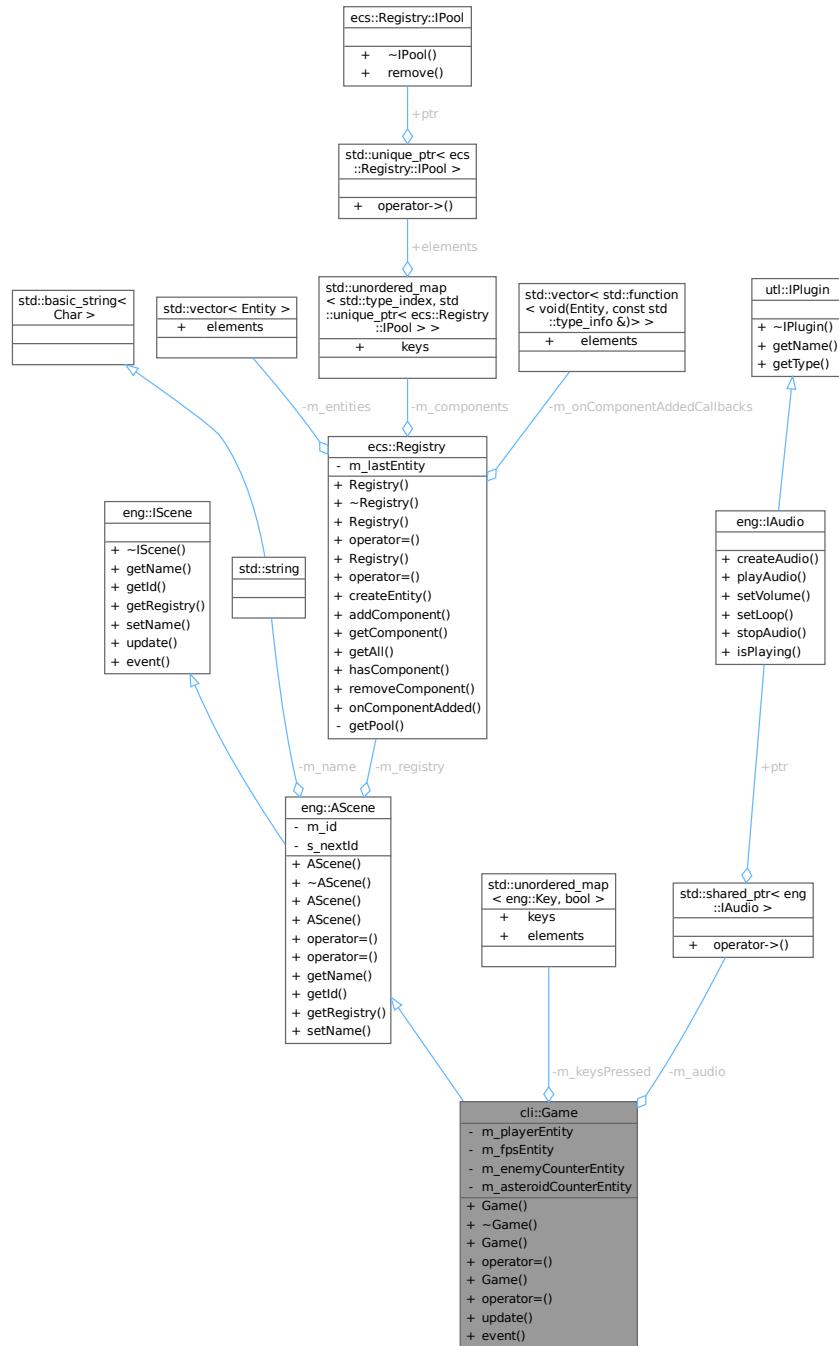
[Game](#) scene.

```
#include <Game.hpp>
```

Inheritance diagram for cli::Game:



Collaboration diagram for cli::Game:



Public Member Functions

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

- `Game & operator= (Game &&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`
- `ecs::Entity m_enemyCounterEntity`
- `ecs::Entity m_asteroidCounterEntity`
- `const std::shared_ptr< eng::IAudio > & m_audio`

7.38.1 Detailed Description

`Game` scene.

Definition at line 22 of file `Game.hpp`.

7.38.2 Constructor & Destructor Documentation

7.38.2.1 Game() [1/3]

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

Definition at line 15 of file `game.cpp`.

References `eng::Color::a`, `cli::Path::Audio::AUDIO_TITLE`, `eng::Color::b`, `BLUE`, `BLUE_SECOND`, `cli::Path::Font::FONTS_RTYPE`, `eng::Color::g`, `GREEN`, `ecs::IComponent::id`, `m_asteroidCounterEntity`, `m_enemyCounterEntity`, `m_fpsEntity`, `m_playerEntity`, `cli::GameConfig::Beam::MAX_CHARGE`, `cli::GameConfig::Hitbox::PLAYER_RADIUS`, `PURPLE`, `eng::Color::r`, `cli::GameConfig::Player::SCALE`, `cli::GameConfig::Player::SPRITE_HEIGHT`, `cli::GameConfig::Player::SPRITE_WIDTH`, `cli::Path::Texture::TEXTURE_WHITE`, `WHITE_TRANS`, `ecs::Scale::x`, and `YELLOW`.

7.38.2.2 ~Game()

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

7.38.2.3 Game() [2/3]

```
cli::Game::Game (
    const Game & other) [delete]
```

7.38.2.4 Game() [3/3]

```
cli::Game::Game (
    Game && other) [delete]
```

7.38.3 Member Function Documentation

7.38.3.1 event()

```
void cli::Game::event (
    const eng::Event & event) [override], [virtual]
```

Implements [eng::IScene](#).

Definition at line 330 of file [game.cpp](#).

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

7.38.3.2 operator=() [1/2]

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

7.38.3.3 operator=() [2/2]

```
Game & cli::Game::operator= (
    Game && other) [delete]
```

7.38.3.4 update()

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

Implements [eng::IScene](#).

Definition at line 212 of file [game.cpp](#).

References [cli::GameConfig::Player::DIAGONAL_SPEED_MULTIPLIER](#), [eng::Down](#), [eng::WindowSize::height](#), [eng::Left](#), [eng::Playing](#), [eng::Right](#), [cli::GameConfig::Player::SCALE](#), [cli::GameConfig::Player::SPEED](#), [cli::GameConfig::Player::SPRITE_HEIGHT](#), [cli::GameConfig::Player::SPRITE_WIDTH](#), [eng::Up](#), [eng::WindowSize::width](#), [ecs::Transform::x](#), and [ecs::Transform::y](#).

7.38.4 Member Data Documentation

7.38.4.1 m_asteroidCounterEntity

`ecs::Entity` `cli::Game::m_asteroidCounterEntity` [private]

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

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

7.38.4.2 m_audio

`const std::shared_ptr<eng::IAudio>&` `cli::Game::m_audio` [private]

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

7.38.4.3 m_enemyCounterEntity

`ecs::Entity` `cli::Game::m_enemyCounterEntity` [private]

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

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

7.38.4.4 m_fpsEntity

`ecs::Entity` `cli::Game::m_fpsEntity` [private]

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

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

7.38.4.5 m_keysPressed

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

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

7.38.4.6 m_playerEntity

`ecs::Entity` `cli::Game::m_playerEntity` [private]

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

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

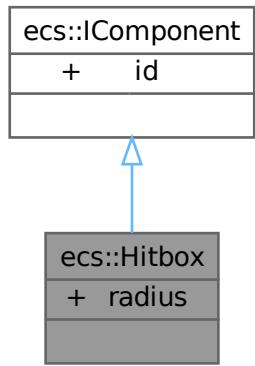
The documentation for this class was generated from the following files:

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

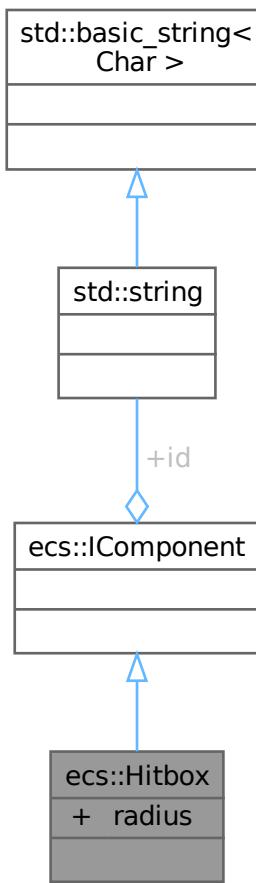
7.39 ecs::Hitbox Struct Reference

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

Inheritance diagram for ecs::Hitbox:



Collaboration diagram for ecs::Hitbox:



Public Attributes

- float `radius`

Public Attributes inherited from `ecs::IComponent`

- std::string `id`

7.39.1 Detailed Description

Definition at line 152 of file [Component.hpp](#).

7.39.2 Member Data Documentation

7.39.2.1 radius

```
float ecs::Hitbox::radius
```

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

Referenced by [cli::CollisionSystem::checkCircularCollision\(\)](#).

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

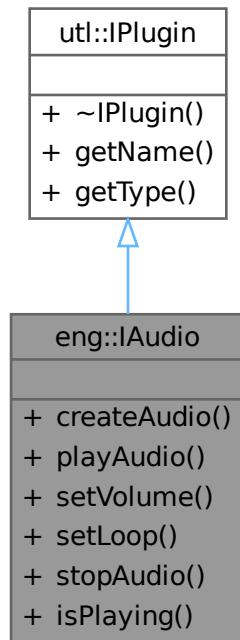
- /home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/[Component.hpp](#)

7.40 eng::IAudio Class Reference

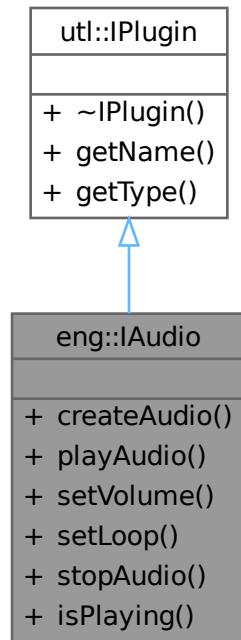
Interface for the audio.

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

Inheritance diagram for eng::IAudio:



Collaboration diagram for eng::IAudio:



Public Member Functions

- virtual 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

Public Member Functions inherited from `utl::IPlugin`

- virtual `~IPlugin` ()=default
- virtual const std::string `getName` () const =0
- virtual `PluginType getType` () const =0

7.40.1 Detailed Description

Interface for the audio.

Definition at line 28 of file `IAudio.hpp`.

7.40.2 Member Function Documentation

7.40.2.1 createAudio()

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

7.40.2.2 isPlaying()

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

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

Here is the caller graph for this function:



7.40.2.3 playAudio()

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

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

Here is the caller graph for this function:



7.40.2.4 setLoop()

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

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

Here is the caller graph for this function:



7.40.2.5 setVolume()

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

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

Here is the caller graph for this function:



7.40.2.6 stopAudio()

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

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](#)

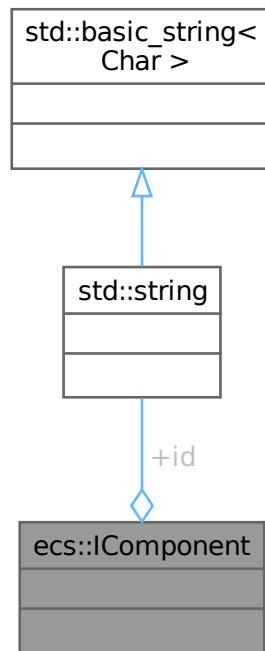
7.41 ecs::IComponent Struct Reference

#include <Component.hpp>

Inheritance diagram for ecs::IComponent:



Collaboration diagram for ecs::IComponent:



Public Attributes

- `std::string id`

7.41.1 Detailed Description

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

7.41.2 Member Data Documentation

7.41.2.1 id

std::string ecs::IComponent::id

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

Referenced by [cli::Game::Game\(\)](#), [cli::Lobby::Lobby\(\)](#), [cli::Settings::Settings\(\)](#), [cli::AsteroidSystem::update\(\)](#), [cli::EnemySystem::update\(\)](#), [cli::ExplosionSystem::update\(\)](#), and [cli::LoadingAnimationSystem::update\(\)](#).

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

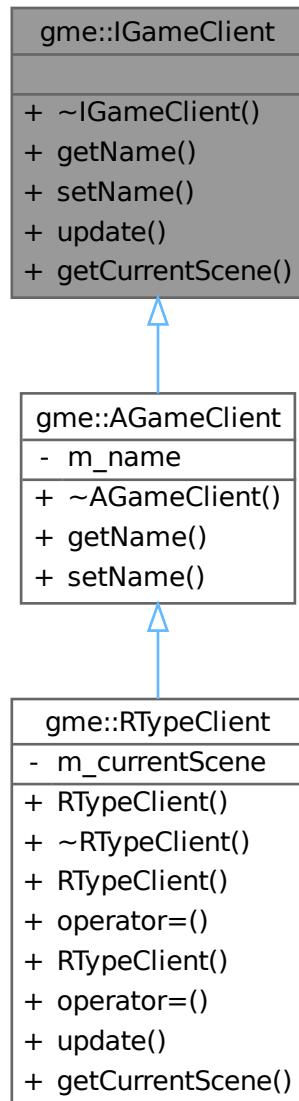
- /home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/[Component.hpp](#)

7.42 gme::IGameClient Class Reference

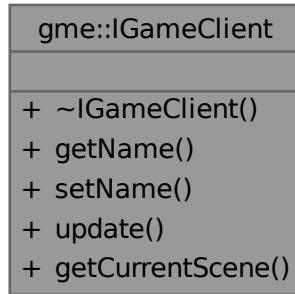
Interface for the games.

```
#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

7.42.1 Detailed Description

Interface for the games.

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

7.42.2 Constructor & Destructor Documentation

7.42.2.1 ~IGameClient()

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

7.42.3 Member Function Documentation

7.42.3.1 getCurrentScene()

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

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

7.42.3.2 getName()

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

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

7.42.3.3 setName()

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

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

7.42.3.4 update()

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

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

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

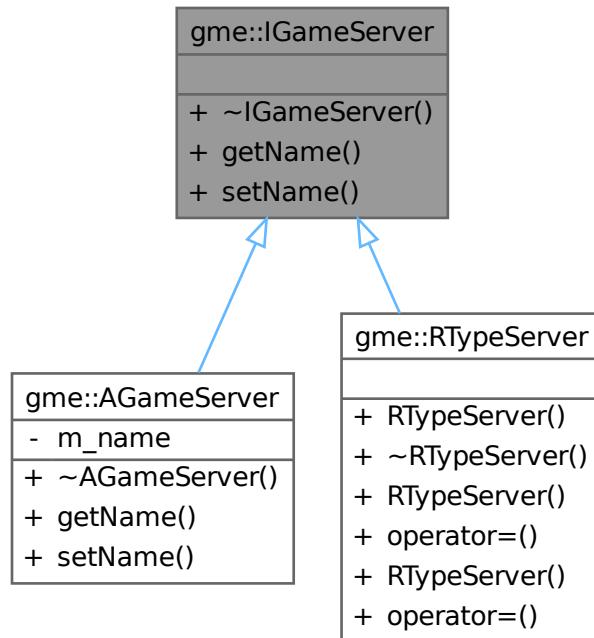
- /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/[IGameClient.hpp](#)

7.43 gme::IGameServer Class Reference

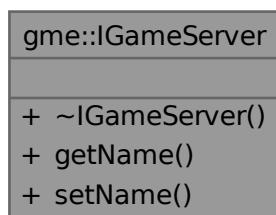
Interface for the games.

```
#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)`

7.43.1 Detailed Description

Interface for the games.

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

7.43.2 Constructor & Destructor Documentation

7.43.2.1 ~IGameServer()

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

7.43.3 Member Function Documentation

7.43.3.1 getName()

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

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

7.43.3.2 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:

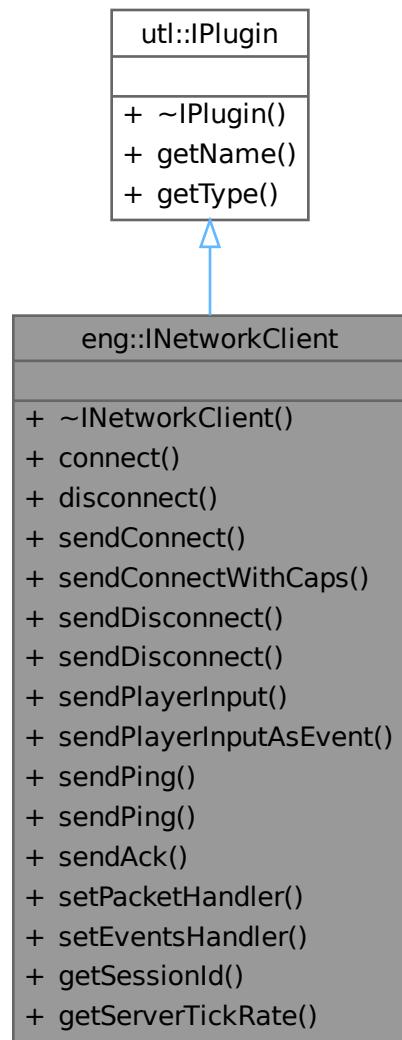
- /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/[IGameServer.hpp](#)

7.44 eng::INetworkClient Class Reference

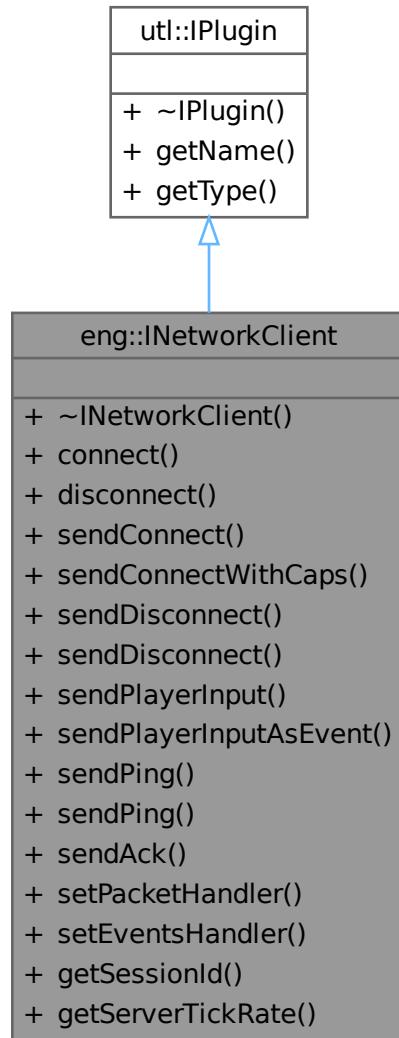
Interface for the client network.

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

Inheritance diagram for eng::INetworkClient:



Collaboration diagram for eng::INetworkClient:



Public Types

- using `PacketHandler` = std::function<void(const `rmp::PacketHeader` &, const std::vector<uint8_t> &)>

Public Member Functions

- virtual ~INetworkClient ()=default
- virtual void `connect` (const std::string &host, uint16_t port)=0
- virtual void `disconnect` ()=0
- virtual void `sendConnect` (const std::string &playerName)=0
- virtual void `sendConnectWithCaps` (const std::string &playerName, std::uint32_t clientCaps)=0
- virtual void `sendDisconnect` ()=0

- virtual void `sendDisconnect (rnp::DisconnectReason reason)=0`
- virtual void `sendPlayerInput (uint8_t direction, uint8_t shooting)=0`
- virtual void `sendPlayerInputAsEvent (std::uint16_t playerId, uint8_t direction, uint8_t shooting, std::uint32_t clientTimeMs)=0`
- virtual void `sendPing ()=0`
- virtual void `sendPing (std::uint32_t nonce, std::uint32_t sendTimeMs)=0`
- virtual void `sendAck (std::uint32_t cumulative, std::uint32_t ackBits)=0`
- virtual void `setPacketHandler (rnp::PacketType type, PacketHandler handler)=0`
- virtual void `setEventsHandler (std::function< void(const std::vector< rnp::EventRecord > &) > handler)=0`
- virtual std::uint32_t `getSessionId () const =0`
- virtual std::uint16_t `getServerTickRate () const =0`

Public Member Functions inherited from [utl::IPlugin](#)

- virtual `~IPlugin ()=default`
- virtual const std::string `getName () const =0`
- virtual `PluginType getType () const =0`

7.44.1 Detailed Description

Interface for the client network.

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

7.44.2 Member Typedef Documentation

7.44.2.1 PacketHandler

```
using eng::INetworkClient::PacketHandler = std::function<void(const rnp::PacketHeader &, const std::vector<uint8_t> &)>
```

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

7.44.3 Constructor & Destructor Documentation

7.44.3.1 ~INetworkClient()

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

7.44.4 Member Function Documentation

7.44.4.1 connect()

```
virtual void eng::INetworkClient::connect (
    const std::string & host,
    std::uint16_t port) [pure virtual]
```

7.44.4.2 disconnect()

```
virtual void eng::INetworkClient::disconnect () [pure virtual]
```

7.44.4.3 getServerTickRate()

```
virtual std::uint16_t eng::INetworkClient::getServerTickRate () const [pure virtual]
```

7.44.4.4 getSessionId()

```
virtual std::uint32_t eng::INetworkClient::getSessionId () const [pure virtual]
```

7.44.4.5 sendAck()

```
virtual void eng::INetworkClient::sendAck (
    std::uint32_t cumulative,
    std::uint32_t ackBits) [pure virtual]
```

7.44.4.6 sendConnect()

```
virtual void eng::INetworkClient::sendConnect (
    const std::string & playerName) [pure virtual]
```

7.44.4.7 sendConnectWithCaps()

```
virtual void eng::INetworkClient::sendConnectWithCaps (
    const std::string & playerName,
    std::uint32_t clientCaps) [pure virtual]
```

7.44.4.8 sendDisconnect() [1/2]

```
virtual void eng::INetworkClient::sendDisconnect () [pure virtual]
```

7.44.4.9 sendDisconnect() [2/2]

```
virtual void eng::INetworkClient::sendDisconnect (
    rnp::DisconnectReason reason) [pure virtual]
```

7.44.4.10 sendPing() [1/2]

```
virtual void eng::INetworkClient::sendPing () [pure virtual]
```

7.44.4.11 sendPing() [2/2]

```
virtual void eng::INetworkClient::sendPing (
    std::uint32_t nonce,
    std::uint32_t sendTimeMs) [pure virtual]
```

7.44.4.12 sendPlayerInput()

```
virtual void eng::INetworkClient::sendPlayerInput (
    uint8_t direction,
    uint8_t shooting) [pure virtual]
```

7.44.4.13 sendPlayerInputAsEvent()

```
virtual void eng::INetworkClient::sendPlayerInputAsEvent (
    std::uint16_t playerId,
    uint8_t direction,
    uint8_t shooting,
    std::uint32_t clientTimeMs) [pure virtual]
```

7.44.4.14 setEventsHandler()

```
virtual void eng::INetworkClient::setEventsHandler (
    std::function< void(const std::vector< rnp::EventRecord > &)> handler) [pure virtual]
```

7.44.4.15 setPacketHandler()

```
virtual void eng::INetworkClient::setPacketHandler (
    rnp::PacketType type,
    PacketHandler handler) [pure virtual]
```

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

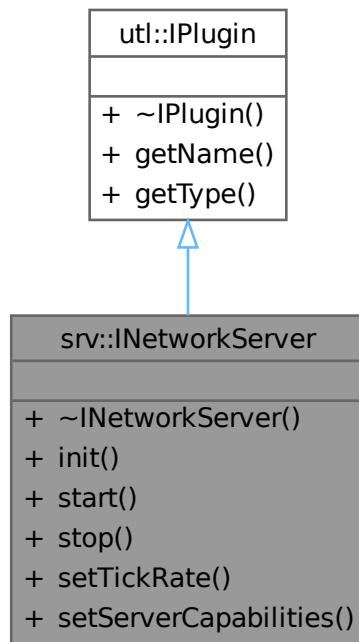
- /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/[INetworkClient.hpp](#)

7.45 srv::INetworkServer Class Reference

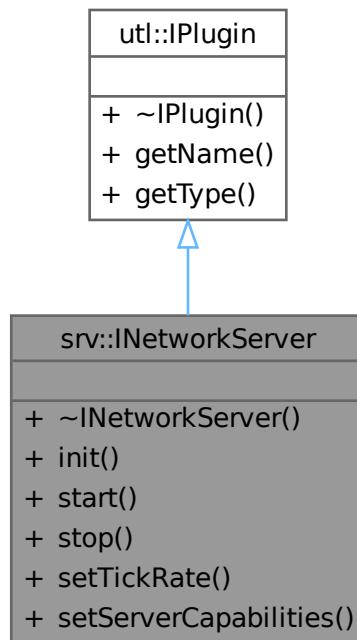
Interface for the server network.

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

Inheritance diagram for `srv::INetworkServer`:



Collaboration diagram for srv::INetworkServer:



Public Member Functions

- virtual `~INetworkServer ()=default`
- virtual void `init (const std::string &host, uint16_t port)=0`
- virtual void `start ()=0`
- virtual void `stop ()=0`
- virtual void `setTickRate (std::uint16_t tickRate)=0`
- virtual void `setServerCapabilities (std::uint32_t caps)=0`

Public Member Functions inherited from `utl::IPlugin`

- virtual `~IPlugin ()=default`
- virtual const std::string `getName () const =0`
- virtual `PluginType getType () const =0`

7.45.1 Detailed Description

Interface for the server network.

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

7.45.2 Constructor & Destructor Documentation

7.45.2.1 ~INetworkServer()

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

7.45.3 Member Function Documentation

7.45.3.1 init()

```
virtual void srv::INetworkServer::init (
    const std::string & host,
    uint16_t port) [pure virtual]
```

7.45.3.2 setServerCapabilities()

```
virtual void srv::INetworkServer::setServerCapabilities (
    std::uint32_t caps) [pure virtual]
```

7.45.3.3 setTickRate()

```
virtual void srv::INetworkServer::setTickRate (
    std::uint16_t tickRate) [pure virtual]
```

7.45.3.4 start()

```
virtual void srv::INetworkServer::start () [pure virtual]
```

7.45.3.5 stop()

```
virtual void srv::INetworkServer::stop () [pure virtual]
```

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

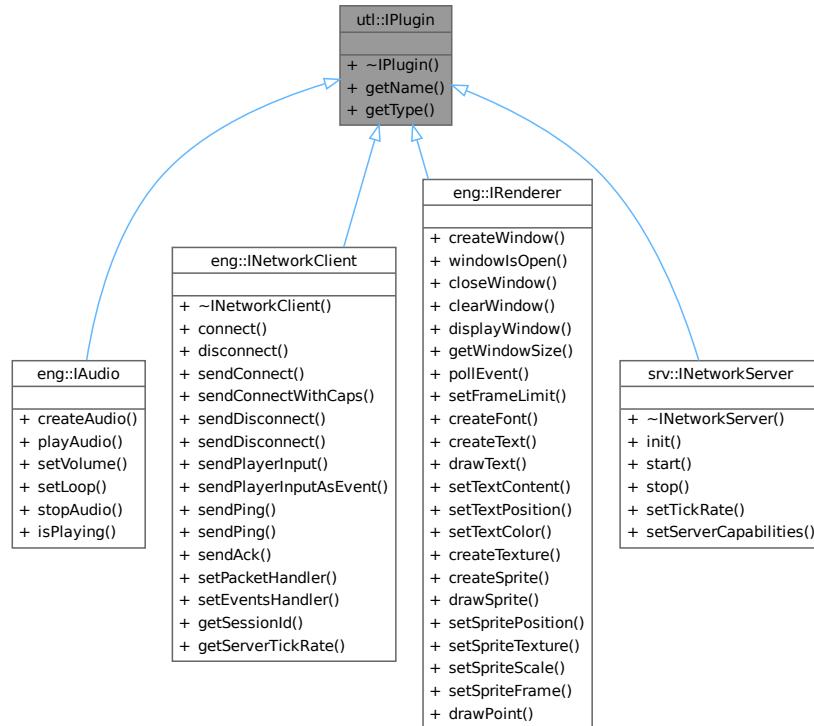
- /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/[INetworkServer.hpp](#)

7.46 utl::IPlugin Interface Reference

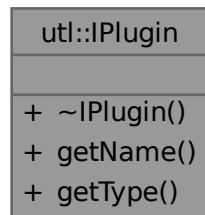
Interface for plugins.

```
#include <IPlugin.hpp>
```

Inheritance diagram for utl::IPlugin:



Collaboration diagram for utl::IPlugin:



Public Member Functions

- virtual `~IPlugin ()=default`
- virtual const std::string `getName () const =0`
- virtual `PluginType getType () const =0`

7.46.1 Detailed Description

Interface for plugins.

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

7.46.2 Constructor & Destructor Documentation

7.46.2.1 ~IPlugin()

```
virtual utl::IPlugin::~IPlugin () [virtual], [default]
```

7.46.3 Member Function Documentation

7.46.3.1 getName()

```
virtual const std::string utl::IPlugin::getName () const [nodiscard], [pure virtual]
```

7.46.3.2 getType()

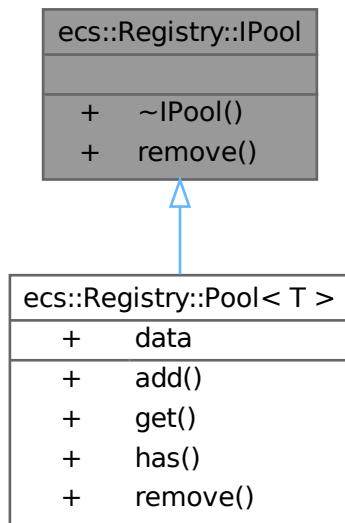
```
virtual PluginType utl::IPlugin::getType () const [nodiscard], [pure virtual]
```

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

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

7.47 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`

7.47.1 Detailed Description

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

7.47.2 Constructor & Destructor Documentation

7.47.2.1 ~IPool()

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

7.47.3 Member Function Documentation

7.47.3.1 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`

7.48 eng::IRenderer Class Reference

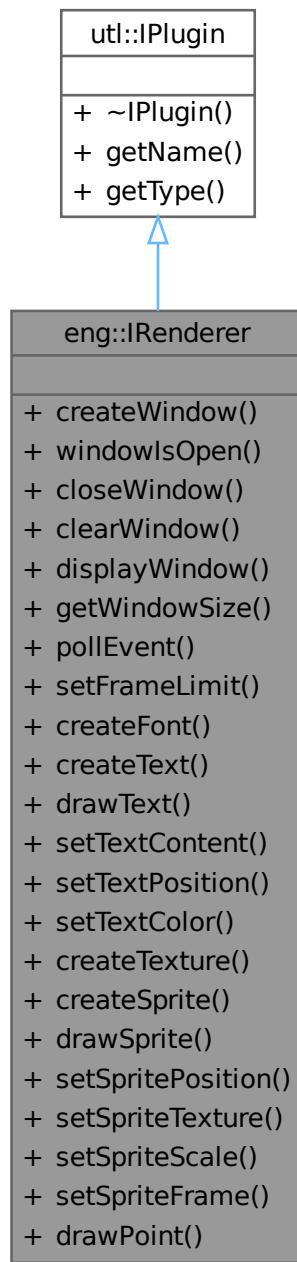
Interface for the renderer.

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

Inheritance diagram for eng::IRenderer:



Collaboration diagram for eng::IRenderer:



Public Member Functions

- `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`

Public Member Functions inherited from `utl::IPlugin`

- virtual `~IPlugin ()=default`
- virtual const `std::string getName () const =0`
- virtual `PluginType getType () const =0`

7.48.1 Detailed Description

Interface for the renderer.

Definition at line 105 of file `IRenderer.hpp`.

7.48.2 Member Function Documentation

7.48.2.1 `clearWindow()`

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

7.48.2.2 `closeWindow()`

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

7.48.2.3 `createFont()`

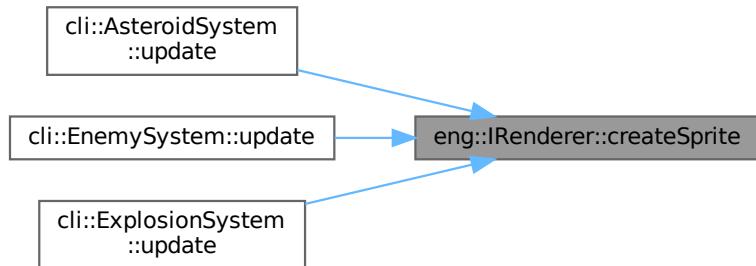
```
virtual void eng::IRenderer::createFont (
    const std::string & name,
    const std::string & path) [pure virtual]
```

7.48.2.4 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 fnx = -1) [pure virtual]
```

Referenced by [cli::AsteroidSystem::update\(\)](#), [cli::EnemySystem::update\(\)](#), and [cli::ExplosionSystem::update\(\)](#).

Here is the caller graph for this function:



7.48.2.5 createText()

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

7.48.2.6 createTexture()

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

7.48.2.7 createWindow()

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

7.48.2.8 displayWindow()

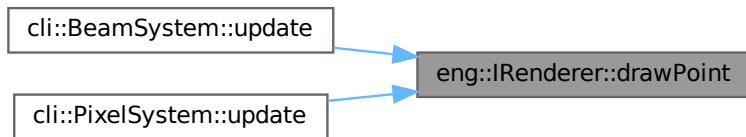
```
virtual void eng::IRenderer::displayWindow () [pure virtual]
```

7.48.2.9 drawPoint()

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

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

Here is the caller graph for this function:

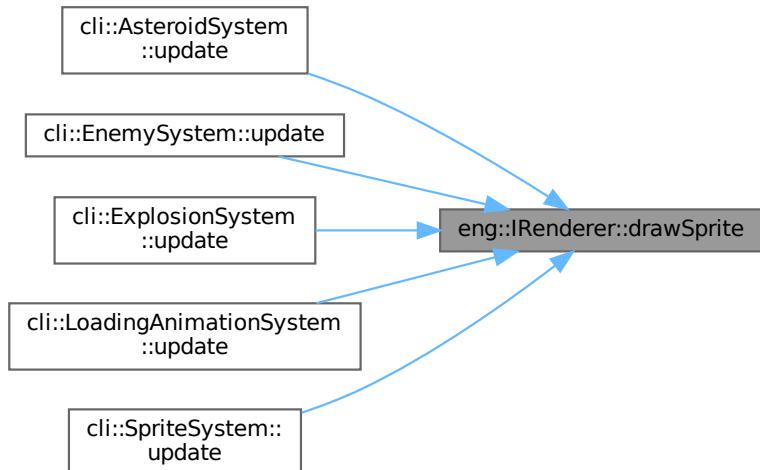


7.48.2.10 drawSprite()

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

Referenced by [cli::AsteroidSystem::update\(\)](#), [cli::EnemySystem::update\(\)](#), [cli::ExplosionSystem::update\(\)](#), [cli::LoadingAnimationSystem::update\(\)](#), and [cli::SpriteSystem::update\(\)](#).

Here is the caller graph for this function:



7.48.2.11 drawText()

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

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

Here is the caller graph for this function:



7.48.2.12 getWindowSize()

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

7.48.2.13 pollEvent()

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

7.48.2.14 setFrameLimit()

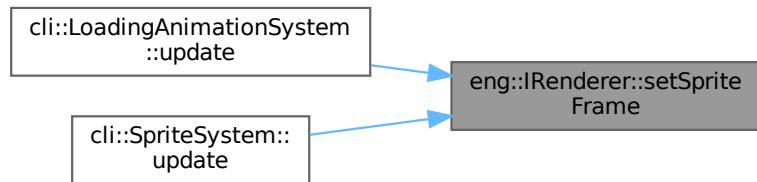
```
virtual void eng::IRenderer::setFrameLimit (
    unsigned int frameLimit) [pure virtual]
```

7.48.2.15 setSpriteFrame()

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

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

Here is the caller graph for this function:

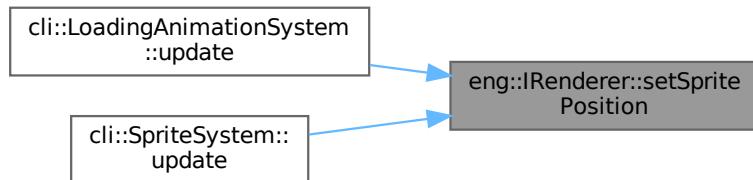


7.48.2.16 setSpritePosition()

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

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

Here is the caller graph for this function:



7.48.2.17 setSpriteScale()

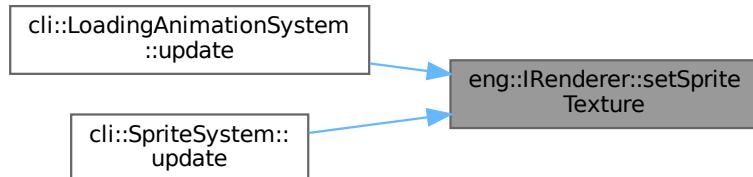
```
virtual void eng::IRenderer::setSpriteScale (
    const std::string & name,
    int x,
    int y) [pure virtual]
```

7.48.2.18 setSpriteTexture()

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

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

Here is the caller graph for this function:



7.48.2.19 setTextColor()

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

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

Here is the caller graph for this function:



7.48.2.20 setTextContent()

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

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

Here is the caller graph for this function:



7.48.2.21 setTextPosition()

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

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

Here is the caller graph for this function:



7.48.2.22 windowIsOpen()

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

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

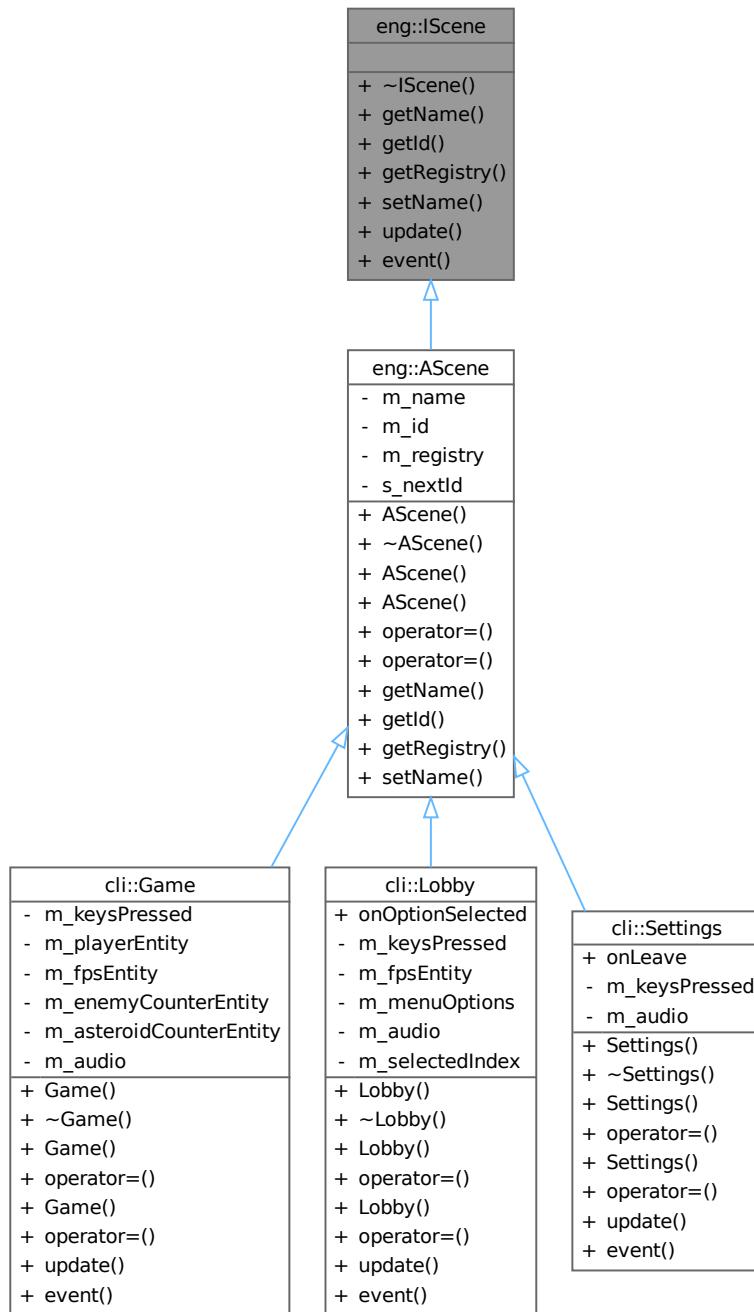
- /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IRenderer.hpp

7.49 eng::IScene Class Reference

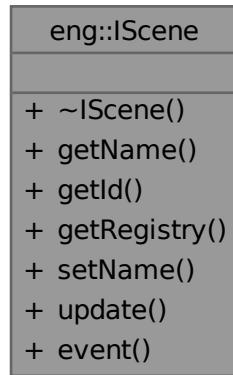
interface class for scene

```
#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`

7.49.1 Detailed Description

interface class for scene

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

7.49.2 Constructor & Destructor Documentation

7.49.2.1 ~IScene()

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

7.49.3 Member Function Documentation

7.49.3.1 event()

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

Implemented in [cli::Game](#), [cli::Lobby](#), and [cli::Settings](#).

7.49.3.2 getId()

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

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

7.49.3.3 getName()

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

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

7.49.3.4 getRegistry()

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

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

7.49.3.5 setName()

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

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

7.49.3.6 update()

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

Implemented in [cli::Game](#), [cli::Lobby](#), and [cli::Settings](#).

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

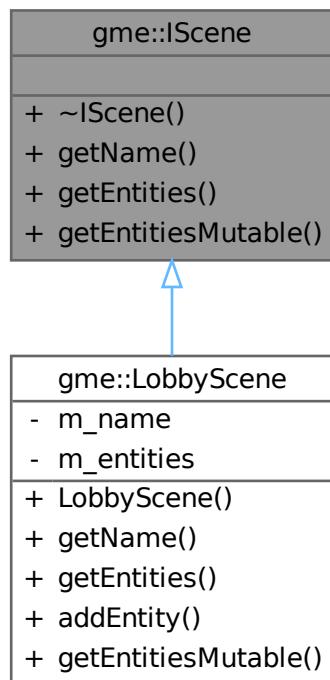
- /home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/Interfaces/[IScene.hpp](#)

7.50 gme::IScene Class Reference

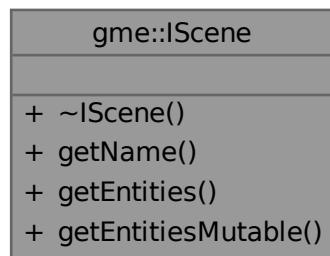
Interface for scenes.

```
#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

7.50.1 Detailed Description

Interface for scenes.

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

7.50.2 Constructor & Destructor Documentation

7.50.2.1 ~IScene()

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

7.50.3 Member Function Documentation

7.50.3.1 getEntities()

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

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

7.50.3.2 getEntitiesMutable()

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

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

7.50.3.3 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:

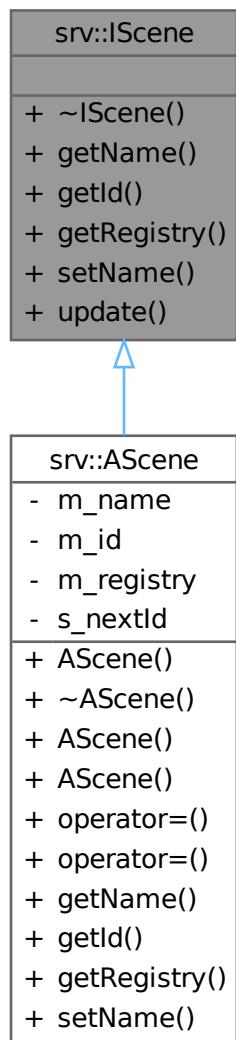
- /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/[IGameClient.hpp](#)

7.51 srv::IScene Class Reference

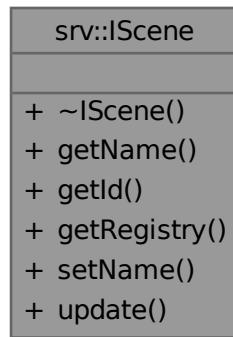
interface class for scene

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

Inheritance diagram for srv::IScene:



Collaboration diagram for srv::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)=0`

7.51.1 Detailed Description

interface class for scene

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

7.51.2 Constructor & Destructor Documentation

7.51.2.1 ~IScene()

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

7.51.3 Member Function Documentation

7.51.3.1 getId()

`virtual id srv::IScene::getId () const [nodiscard], [pure virtual]`

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

7.51.3.2 getName()

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

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

7.51.3.3 getRegistry()

```
virtual ecs::Registry & srv::IScene::getRegistry () [nodiscard], [pure virtual]
```

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

7.51.3.4 setName()

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

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

7.51.3.5 update()

```
virtual void srv::IScene::update (
    float dt) [pure virtual]
```

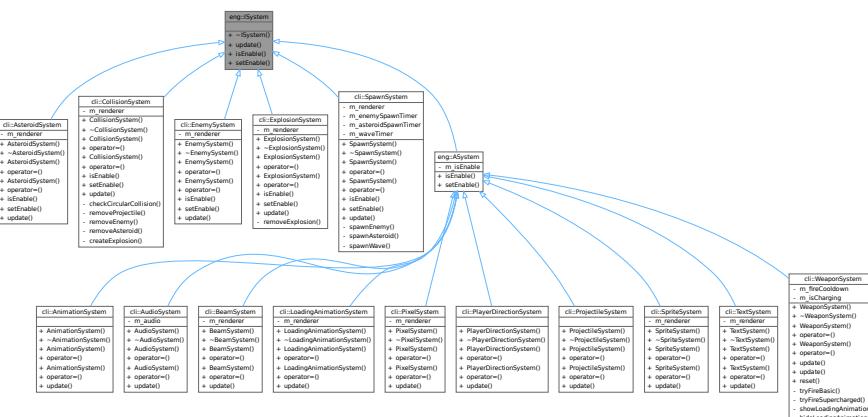
The documentation for this class was generated from the following file:

- /home/masina/Projects/Epitech/rtype/server/include/Server/Interfaces/IScene.hpp

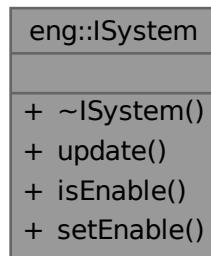
7.52 eng::ISystem Class Reference

```
#include <ISystems.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`

7.52.1 Detailed Description

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

7.52.2 Constructor & Destructor Documentation

7.52.2.1 ~ISystem()

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

7.52.3 Member Function Documentation

7.52.3.1 isEnabled()

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

Implemented in [cli::AsteroidSystem](#), [cli::CollisionSystem](#), [cli::EnemySystem](#), [cli::ExplosionSystem](#), [cli::SpawnSystem](#), and [eng::ASystem](#).

7.52.3.2 setEnable()

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

Implemented in [cli::AsteroidSystem](#), [cli::CollisionSystem](#), [cli::EnemySystem](#), [cli::ExplosionSystem](#), [cli::SpawnSystem](#), and [eng::ASystem](#).

7.52.3.3 update()

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

Implemented in [cli::AnimationSystem](#), [cli::AsteroidSystem](#), [cli::AudioSystem](#), [cli::BeamSystem](#), [cli::CollisionSystem](#), [cli::EnemySystem](#), [cli::ExplosionSystem](#), [cli::LoadingAnimationSystem](#), [cli::PixelSystem](#), [cli::PlayerDirectionSystem](#), [cli::ProjectileSystem](#), [cli::SpawnSystem](#), [cli::SpriteSystem](#), [cli::TextSystem](#), and [cli::WeaponSystem](#).

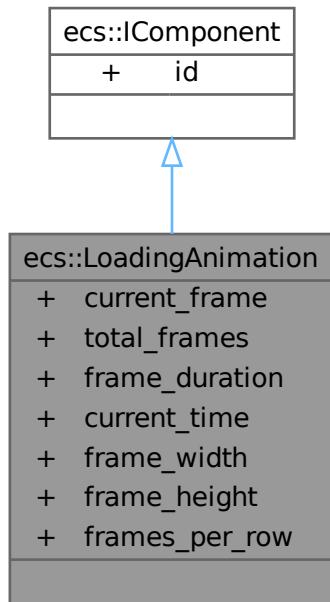
The documentation for this class was generated from the following file:

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

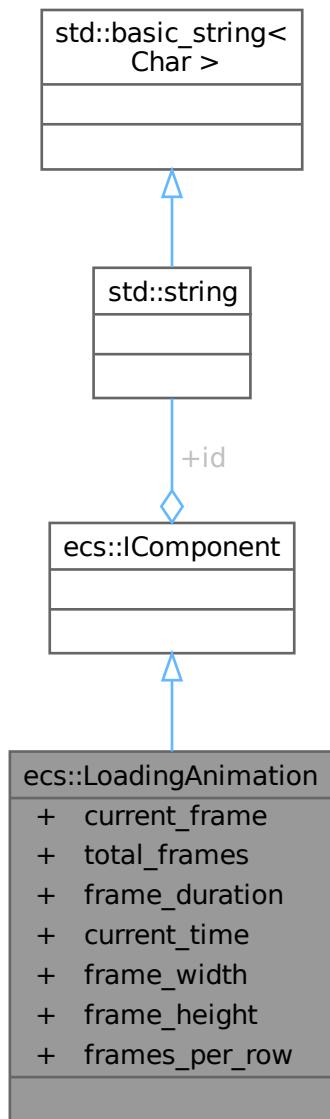
7.53 ecs::LoadingAnimation Struct Reference

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

Inheritance diagram for ecs::LoadingAnimation:



Collaboration diagram for ecs::LoadingAnimation:



Public Attributes

- int `current_frame`
- int `total_frames`
- float `frame_duration`
- float `current_time`
- float `frame_width`
- float `frame_height`
- int `frames_per_row`

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

- std::string [id](#)

7.53.1 Detailed Description

Definition at line 105 of file [Component.hpp](#).

7.53.2 Member Data Documentation

7.53.2.1 current_frame

int ecs::LoadingAnimation::current_frame

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

7.53.2.2 current_time

float ecs::LoadingAnimation::current_time

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

7.53.2.3 frame_duration

float ecs::LoadingAnimation::frame_duration

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

7.53.2.4 frame_height

float ecs::LoadingAnimation::frame_height

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

7.53.2.5 frame_width

float ecs::LoadingAnimation::frame_width

Definition at line 111 of file [Component.hpp](#).

7.53.2.6 frames_per_row

int ecs::LoadingAnimation::frames_per_row

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

7.53.2.7 total_frames

```
int ecs::LoadingAnimation::total_frames
```

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

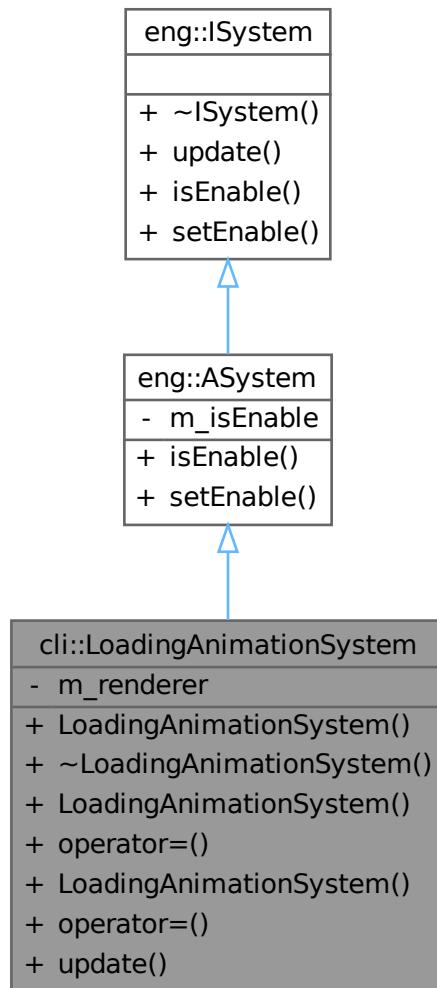
The documentation for this struct was generated from the following file:

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

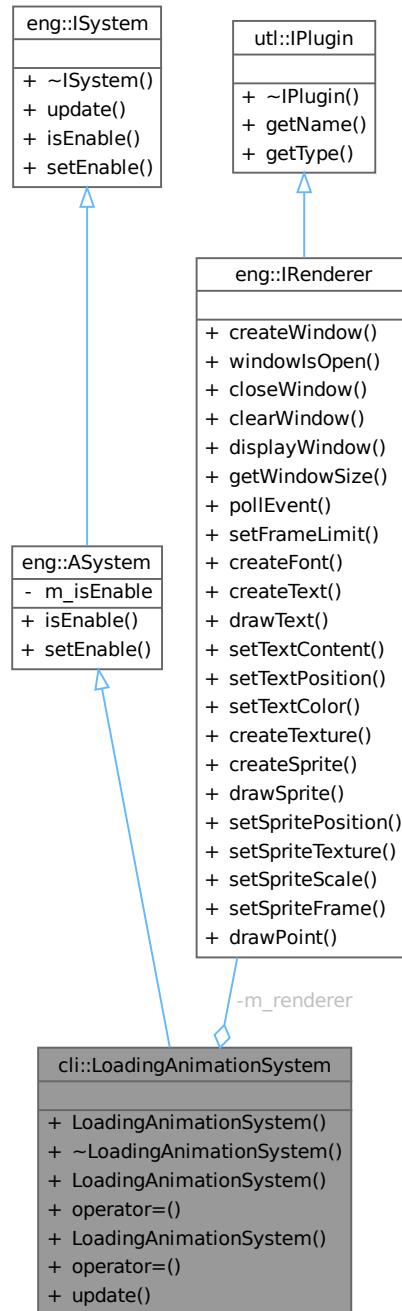
7.54 cli::LoadingAnimationSystem Class Reference

```
#include <LoadingAnimation.hpp>
```

Inheritance diagram for cli::LoadingAnimationSystem:



Collaboration diagram for cli::LoadingAnimationSystem:



Public Member Functions

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

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

- bool [isEnable \(\)](#) override
- void [setEnable \(const bool enable\)](#) override

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

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

Private Attributes

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

7.54.1 Detailed Description

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

7.54.2 Constructor & Destructor Documentation

7.54.2.1 LoadingAnimationSystem() [1/3]

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

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

7.54.2.2 ~LoadingAnimationSystem()

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

7.54.2.3 LoadingAnimationSystem() [2/3]

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

7.54.2.4 LoadingAnimationSystem() [3/3]

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

7.54.3 Member Function Documentation

7.54.3.1 operator=() [1/2]

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

7.54.3.2 operator=() [2/2]

```
>LoadingAnimationSystem & cli::LoadingAnimationSystem::operator= (
    LoadingAnimationSystem &&) [delete]
```

7.54.3.3 update()

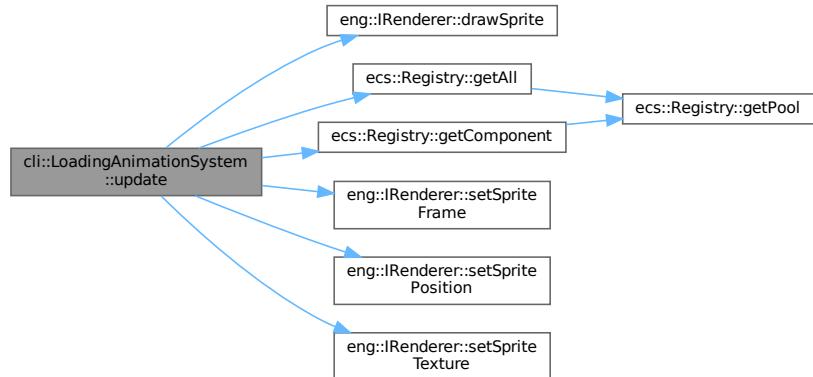
```
void cli::LoadingAnimationSystem::update (
    ecs::Registry & registry,
    float dt) [inline], [override], [virtual]
```

Implements [eng::ISystem](#).

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

References [eng::IRenderer::drawSprite\(\)](#), [ecs::Registry::getAll\(\)](#), [ecs::Registry::getComponent\(\)](#), [ecs::IComponent::id](#), [m_renderer](#), [eng::IRenderer::setSpriteFrame\(\)](#), [eng::IRenderer::setSpritePosition\(\)](#), and [eng::IRenderer::setSpriteTexture\(\)](#).

Here is the call graph for this function:



7.54.4 Member Data Documentation

7.54.4.1 m_renderer

```
eng::IRenderer& cli::LoadingAnimationSystem::m_renderer [private]
```

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

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

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

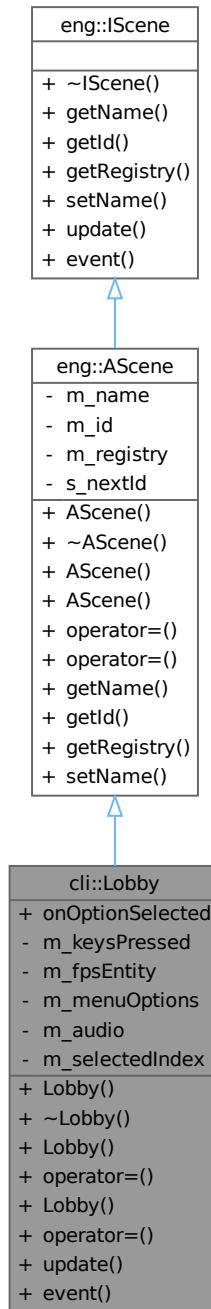
- /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/[LoadingAnimation.hpp](#)

7.55 cli::Lobby Class Reference

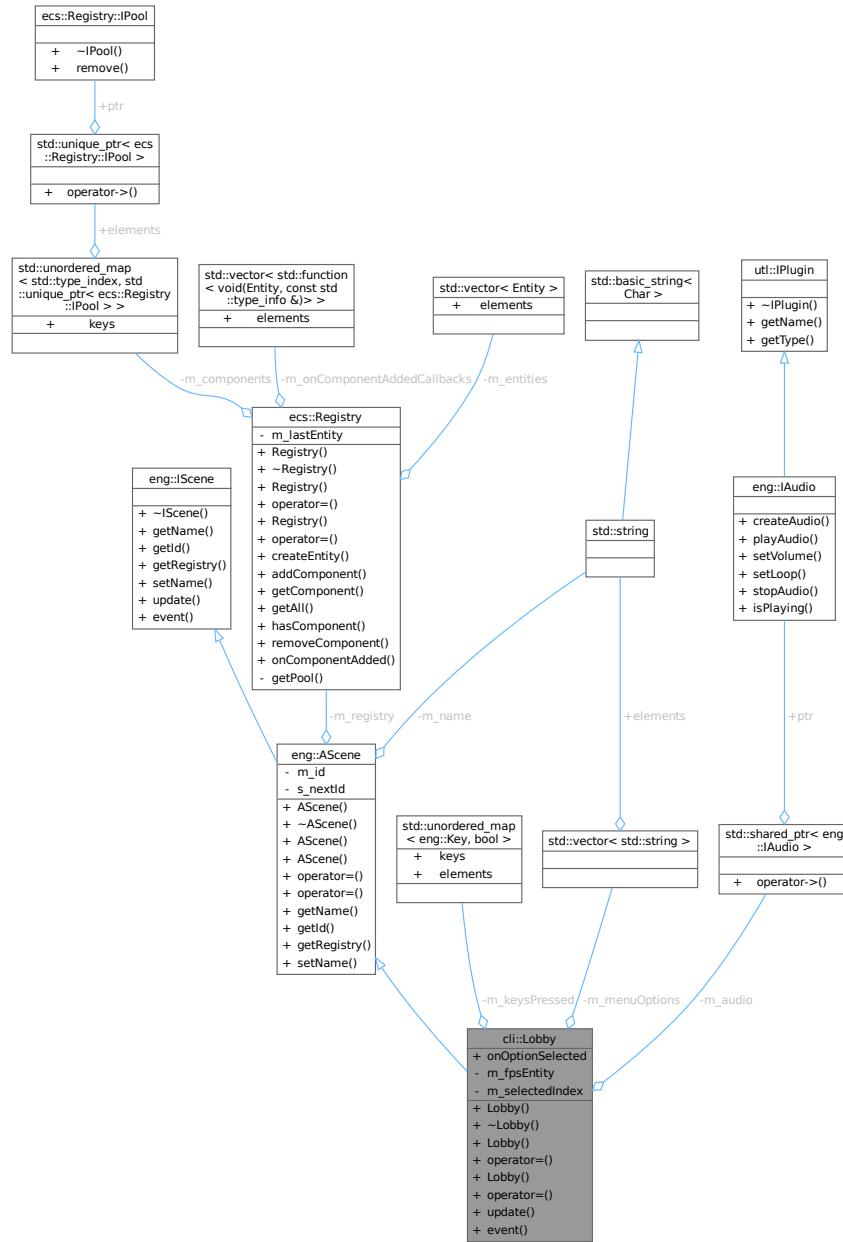
[Lobby](#) scene.

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

Inheritance diagram for cli::Lobby:



Collaboration diagram for cli::Lobby:



Public Member Functions

- `Lobby` (const std::shared_ptr< eng::IRenderer > &renderer, const std::shared_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`

Public Attributes

- `std::function< void(const std::string &option)> onOptionSelected`

Private Attributes

- `std::unordered_map< eng::Key, bool > m_keysPressed`
- `ecs::Entity m_fpsEntity`
- `const std::vector< std::string > m_menuOptions = {"Solo", "Multi", "Settings"}`
- `const std::shared_ptr< eng::IAudio > & m_audio`
- `int m_selectedIndex = 0`

7.55.1 Detailed Description

`Lobby` scene.

Definition at line 22 of file `Lobby.hpp`.

7.55.2 Constructor & Destructor Documentation

7.55.2.1 `Lobby()` [1/3]

```
cli::Lobby::Lobby (
    const std::shared_ptr< eng::IRenderer > & renderer,
    const std::shared_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_menuOptions`, `m_selectedIndex`, `eng::Color::r`, `WHITE`, and `ecs::Scale::x`.

7.55.2.2 ~Lobby()

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

7.55.2.3 Lobby() [2/3]

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

7.55.2.4 Lobby() [3/3]

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

7.55.3 Member Function Documentation

7.55.3.1 event()

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

Implements **eng::IScene**.

Definition at line 144 of file [lobby.cpp](#).

References **eng::Down**, **eng::Enter**, **eng::Event::key**, **eng::KeyPressed**, **eng::KeyReleased**, **eng::Left**, **eng::Right**, **eng::Space**, **eng::Event::type**, and **eng::Up**.

7.55.3.2 operator=() [1/2]

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

7.55.3.3 operator=() [2/2]

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

7.55.3.4 update()

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

Implements **eng::IScene**.

Definition at line 98 of file [lobby.cpp](#).

References **eng::Playing**.

7.55.4 Member Data Documentation

7.55.4.1 m_audio

```
const std::shared_ptr<eng::IAudio>& cli::Lobby::m_audio [private]
```

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

7.55.4.2 m_fpsEntity

```
ecs::Entity cli::Lobby::m_fpsEntity [private]
```

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

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

7.55.4.3 m_keysPressed

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

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

7.55.4.4 m_menuOptions

```
const std::vector<std::string> cli::Lobby::m_menuOptions = {"Solo", "Multi", "Settings"} [private]
```

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

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

7.55.4.5 m_selectedIndex

```
int cli::Lobby::m_selectedIndex = 0 [private]
```

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

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

7.55.4.6 onOptionSelected

```
std::function<void(const std::string &option)> cli::Lobby::onOptionSelected
```

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

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

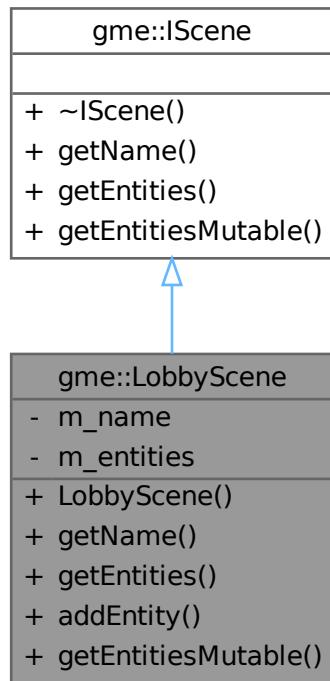
- /home/masina/Projects/Epitech/rtype/client/include/Client/Scenes/[Lobby.hpp](#)
- /home/masina/Projects/Epitech/rtype/client/src/scenes/[lobby.cpp](#)

7.56 gme::LobbyScene Class Reference

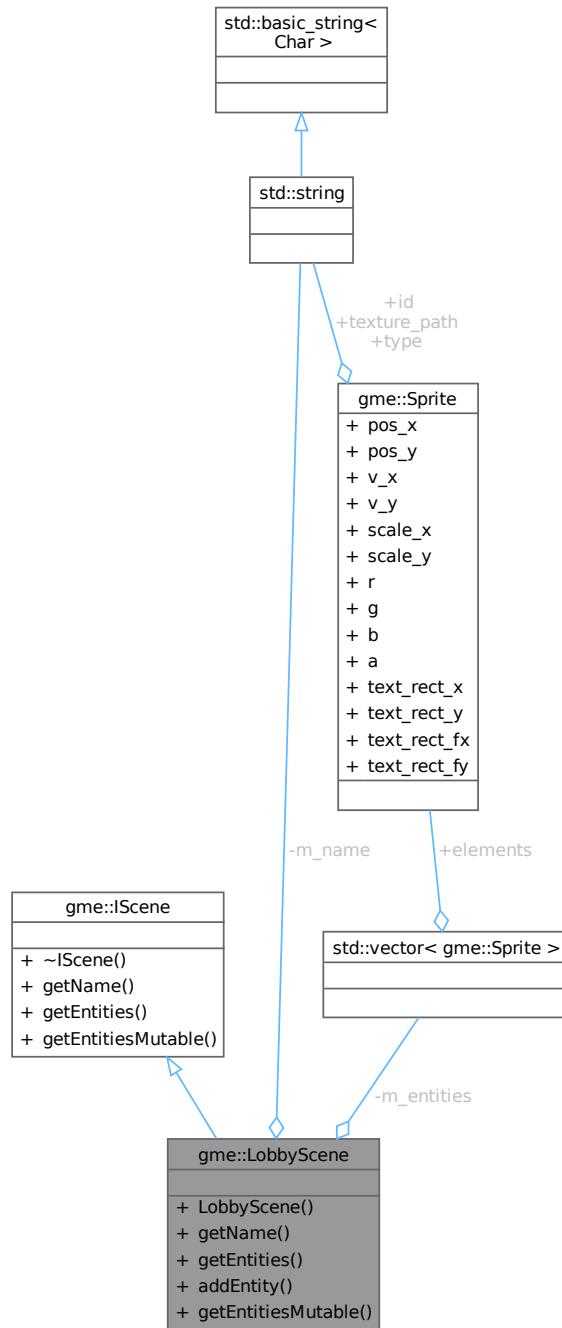
Class for the Lobby scene.

```
#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](#)

7.56.1 Detailed Description

Class for the Lobby scene.

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

7.56.2 Constructor & Destructor Documentation

7.56.2.1 LobbyScene()

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

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

7.56.3 Member Function Documentation

7.56.3.1 addEntity()

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

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

References [m_entities](#).

7.56.3.2 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](#).

7.56.3.3 getEntitiesMutable()

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

Implements [gme::IScene](#).

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

References [m_entities](#).

7.56.3.4 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](#).

7.56.4 Member Data Documentation

7.56.4.1 m_entities

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

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

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

7.56.4.2 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:

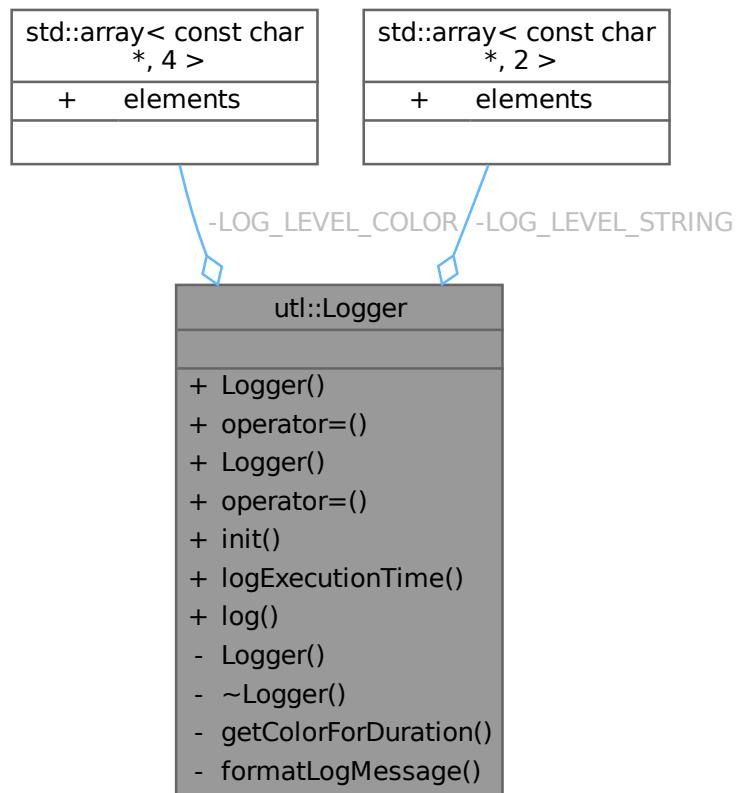
- /home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Client/include/R-TypeClient/[LobbyScene.hpp](#)

7.57 utl::Logger Class Reference

Class for logging.

```
#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"}

7.57.1 Detailed Description

Class for logging.

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

7.57.2 Member Enumeration Documentation

7.57.2.1 ColorIndex

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

Enumerator

<code>COLOR_ERROR</code>	
<code>COLOR_INFO</code>	
<code>COLOR_WARNING</code>	
<code>COLOR_RESET</code>	

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

7.57.3 Constructor & Destructor Documentation

7.57.3.1 `Logger()` [1/3]

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

7.57.3.2 Logger() [2/3]

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

7.57.3.3 Logger() [3/3]

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

7.57.3.4 ~Logger()

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

7.57.4 Member Function Documentation

7.57.4.1 formatLogMessage()

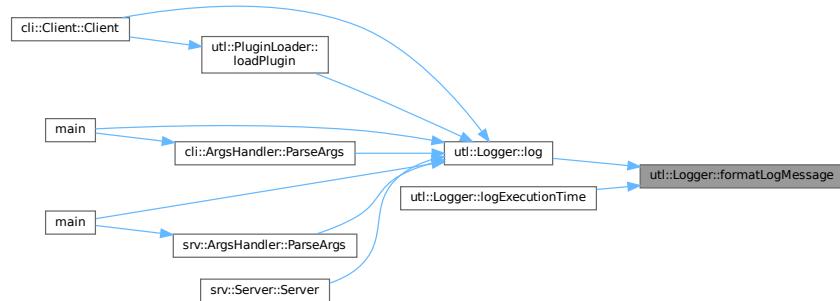
```
static std::string utl::Logger::formatLogMessage (
    LogLevel level,
    const std::string & message)  [inline], [static], [nodiscard], [private]
```

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

References [LOG_LEVEL_STRING](#).

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

Here is the caller graph for this function:



7.57.4.2 getColorForDuration()

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

Definition at line [78](#) 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:



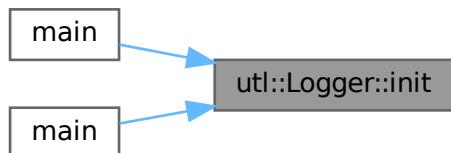
7.57.4.3 init()

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

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

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

Here is the caller graph for this function:



7.57.4.4 log()

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

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

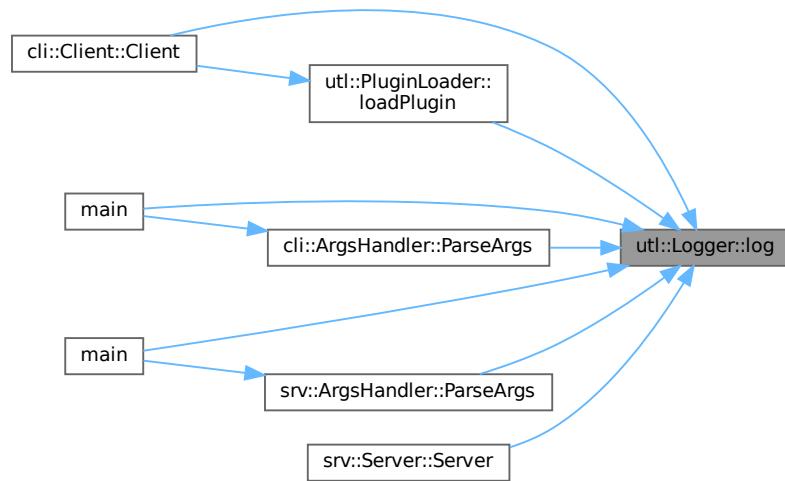
References [COLOR_INFO](#), [COLOR_RESET](#), [COLOR_WARNING](#), [formatLogMessage\(\)](#), [utl::INFO](#), and [LOG_LEVEL_COLOR](#).

Referenced by [cli::Client::Client\(\)](#), [utl::PluginLoader::loadPlugin\(\)](#), [main\(\)](#), [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:



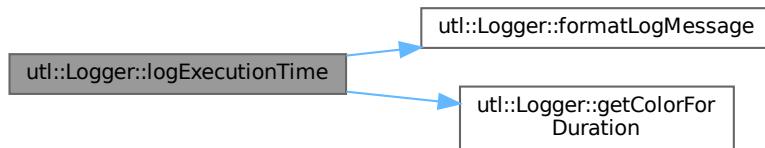
7.57.4.5 logExecutionTime()

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

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

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

Here is the call graph for this function:



7.57.4.6 operator=() [1/2]

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

7.57.4.7 operator=() [2/2]

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

7.57.5 Member Data Documentation

7.57.5.1 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 66 of file [Logger.hpp](#).

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

7.57.5.2 LOG_LEVEL_STRING

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

Definition at line 73 of file [Logger.hpp](#).

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

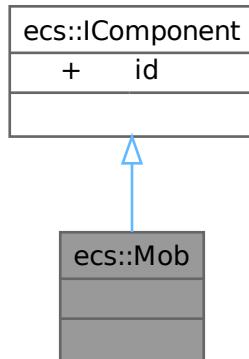
The documentation for this class was generated from the following files:

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

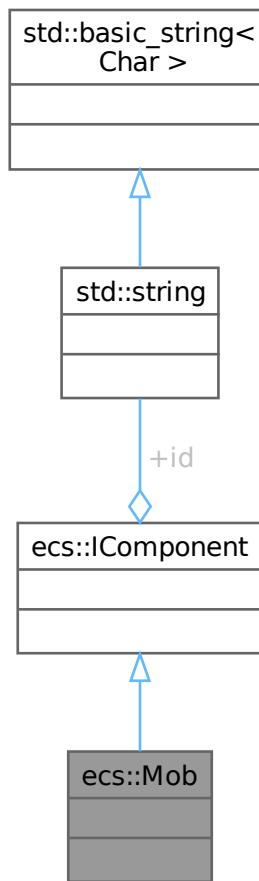
7.58 ecs::Mob Struct Reference

```
#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`

7.58.1 Detailed Description

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

The documentation for this struct was generated from the following file:

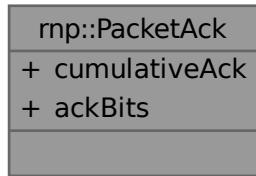
- `/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.hpp`

7.59 rnp::PacketAck Struct Reference

ACK packet payload.

```
#include <Protocol.hpp>
```

Collaboration diagram for rnp::PacketAck:



Public Attributes

- std::uint32_t [cumulativeAck](#)
- std::uint32_t [ackBits](#)

7.59.1 Detailed Description

ACK packet payload.

Definition at line [187](#) of file [Protocol.hpp](#).

7.59.2 Member Data Documentation

7.59.2.1 ackBits

```
std::uint32_t rnp::PacketAck::ackBits
```

Definition at line [190](#) of file [Protocol.hpp](#).

7.59.2.2 cumulativeAck

```
std::uint32_t rnp::PacketAck::cumulativeAck
```

Definition at line [189](#) of file [Protocol.hpp](#).

The documentation for this struct was generated from the following file:

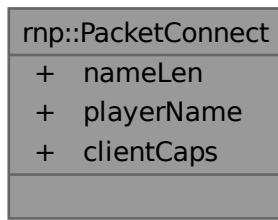
- /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/Protocol/[Protocol.hpp](#)

7.60 rnp::PacketConnect Struct Reference

CONNECT packet payload.

```
#include <Protocol.hpp>
```

Collaboration diagram for rnp::PacketConnect:



Public Attributes

- std::uint8_t `nameLen`
- char `playerName` [32]
- std::uint32_t `clientCaps`

7.60.1 Detailed Description

CONNECT packet payload.

Definition at line 127 of file [Protocol.hpp](#).

7.60.2 Member Data Documentation

7.60.2.1 clientCaps

```
std::uint32_t rnp::PacketConnect::clientCaps
```

Definition at line 131 of file [Protocol.hpp](#).

7.60.2.2 nameLen

```
std::uint8_t rnp::PacketConnect::nameLen
```

Definition at line 129 of file [Protocol.hpp](#).

7.60.2.3 playerName

char rnp::PacketConnect::playerName[32]

Definition at line 130 of file [Protocol.hpp](#).

The documentation for this struct was generated from the following file:

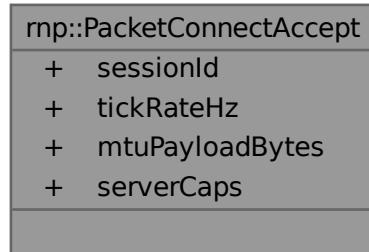
- /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/Protocol/[Protocol.hpp](#)

7.61 rnp::PacketConnectAccept Struct Reference

CONNECT_ACCEPT packet payload.

```
#include <Protocol.hpp>
```

Collaboration diagram for rnp::PacketConnectAccept:



Public Attributes

- std::uint32_t [sessionId](#)
- std::uint16_t [tickRateHz](#)
- std::uint16_t [mtuPayloadBytes](#)
- std::uint32_t [serverCaps](#)

7.61.1 Detailed Description

CONNECT_ACCEPT packet payload.

Definition at line 137 of file [Protocol.hpp](#).

7.61.2 Member Data Documentation

7.61.2.1 mtuPayloadBytes

std::uint16_t rnp::PacketConnectAccept::mtuPayloadBytes

Definition at line 141 of file [Protocol.hpp](#).

7.61.2.2 serverCaps

std::uint32_t rnp::PacketConnectAccept::serverCaps

Definition at line 142 of file [Protocol.hpp](#).

7.61.2.3 sessionId

std::uint32_t rnp::PacketConnectAccept::sessionId

Definition at line 139 of file [Protocol.hpp](#).

7.61.2.4 tickRateHz

std::uint16_t rnp::PacketConnectAccept::tickRateHz

Definition at line 140 of file [Protocol.hpp](#).

The documentation for this struct was generated from the following file:

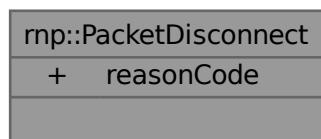
- /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/Protocol/[Protocol.hpp](#)

7.62 rnp::PacketDisconnect Struct Reference

DISCONNECT packet payload.

#include <Protocol.hpp>

Collaboration diagram for rnp::PacketDisconnect:



Public Attributes

- std::uint16_t [reasonCode](#)

7.62.1 Detailed Description

DISCONNECT packet payload.

Definition at line [148](#) of file [Protocol.hpp](#).

7.62.2 Member Data Documentation

7.62.2.1 reasonCode

std::uint16_t rnp::PacketDisconnect::reasonCode

Definition at line [150](#) of file [Protocol.hpp](#).

The documentation for this struct was generated from the following file:

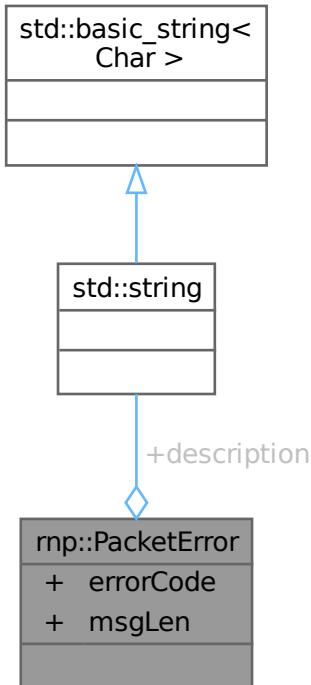
- [/home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/Protocol/Protocol.hpp](#)

7.63 rnp::PacketError Struct Reference

ERROR packet payload.

```
#include <Protocol.hpp>
```

Collaboration diagram for rnp::PacketError:



Public Attributes

- std::uint16_t [errorCode](#)
- std::uint16_t [msgLen](#)
- std::string [description](#)

7.63.1 Detailed Description

ERROR packet payload.

Definition at line [196](#) of file [Protocol.hpp](#).

7.63.2 Member Data Documentation

7.63.2.1 [description](#)

std::string rnp::PacketError::description

Definition at line [200](#) of file [Protocol.hpp](#).

7.63.2.2 [errorCode](#)

std::uint16_t rnp::PacketError::errorCode

Definition at line [198](#) of file [Protocol.hpp](#).

7.63.2.3 [msgLen](#)

std::uint16_t rnp::PacketError::msgLen

Definition at line [199](#) of file [Protocol.hpp](#).

The documentation for this struct was generated from the following file:

- /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/Protocol/[Protocol.hpp](#)

7.64 rnp::PacketHeader Struct Reference

Packet header according to RNP specification (Big Endian) Total size: 16 bytes.

```
#include <Protocol.hpp>
```

Collaboration diagram for rnp::PacketHeader:



Public Attributes

- std::uint8_t [type](#)
- std::uint16_t [length](#)
- std::uint16_t [flags](#)
- std::uint16_t [reserved](#)
- std::uint32_t [sequence](#)
- std::uint32_t [sessionId](#)

7.64.1 Detailed Description

Packet header according to RNP specification (Big Endian) Total size: 16 bytes.

Definition at line 114 of file [Protocol.hpp](#).

7.64.2 Member Data Documentation

7.64.2.1 flags

```
std::uint16_t rnp::PacketHeader::flags
```

Definition at line 118 of file [Protocol.hpp](#).

Referenced by [rnp::deserializeHeader\(\)](#), and [rnp::serializeHeader\(\)](#).

7.64.2.2 length

std::uint16_t rnp::PacketHeader::length

Definition at line 117 of file [Protocol.hpp](#).

Referenced by [rnp::deserializeHeader\(\)](#), [rnp::serialize\(\)](#), and [rnp::serializeHeader\(\)](#).

7.64.2.3 reserved

std::uint16_t rnp::PacketHeader::reserved

Definition at line 119 of file [Protocol.hpp](#).

Referenced by [rnp::deserializeHeader\(\)](#), and [rnp::serializeHeader\(\)](#).

7.64.2.4 sequence

std::uint32_t rnp::PacketHeader::sequence

Definition at line 120 of file [Protocol.hpp](#).

Referenced by [rnp::deserializeHeader\(\)](#), and [rnp::serializeHeader\(\)](#).

7.64.2.5 sessionId

std::uint32_t rnp::PacketHeader::sessionId

Definition at line 121 of file [Protocol.hpp](#).

Referenced by [rnp::deserializeHeader\(\)](#), and [rnp::serializeHeader\(\)](#).

7.64.2.6 type

std::uint8_t rnp::PacketHeader::type

Definition at line 116 of file [Protocol.hpp](#).

Referenced by [rnp::deserializeHeader\(\)](#), and [rnp::serializeHeader\(\)](#).

The documentation for this struct was generated from the following file:

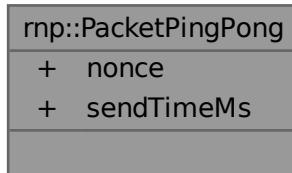
- /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/Protocol/[Protocol.hpp](#)

7.65 rnp::PacketPingPong Struct Reference

PING/PONG packet payload.

```
#include <Protocol.hpp>
```

Collaboration diagram for rnp::PacketPingPong:



Public Attributes

- std::uint32_t [nonce](#)
- std::uint32_t [sendTimeMs](#)

7.65.1 Detailed Description

PING/PONG packet payload.

Definition at line [178](#) of file [Protocol.hpp](#).

7.65.2 Member Data Documentation

7.65.2.1 nonce

```
std::uint32_t rnp::PacketPingPong::nonce
```

Definition at line [180](#) of file [Protocol.hpp](#).

7.65.2.2 sendTimeMs

```
std::uint32_t rnp::PacketPingPong::sendTimeMs
```

Definition at line [181](#) of file [Protocol.hpp](#).

The documentation for this struct was generated from the following file:

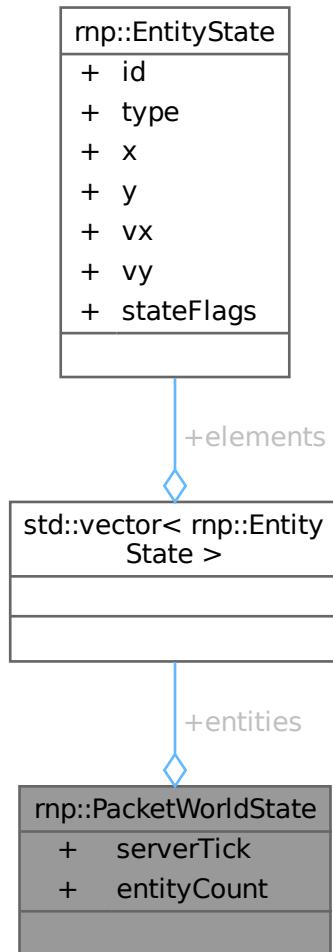
- /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/Protocol/[Protocol.hpp](#)

7.66 rnp::PacketWorldState Struct Reference

WORLD_STATE packet payload.

```
#include <Protocol.hpp>
```

Collaboration diagram for rnp::PacketWorldState:



Public Attributes

- std::uint32_t `serverTick`
- std::uint16_t `entityCount`
- std::vector<[EntityState](#)> `entities`

7.66.1 Detailed Description

WORLD_STATE packet payload.

Definition at line 168 of file [Protocol.hpp](#).

7.66.2 Member Data Documentation

7.66.2.1 entities

```
std::vector<EntityState> rnp::PacketWorldState::entities
```

Definition at line 172 of file [Protocol.hpp](#).

7.66.2.2 entityCount

```
std::uint16_t rnp::PacketWorldState::entityCount
```

Definition at line 171 of file [Protocol.hpp](#).

7.66.2.3 serverTick

```
std::uint32_t rnp::PacketWorldState::serverTick
```

Definition at line 170 of file [Protocol.hpp](#).

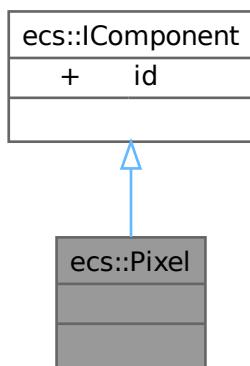
The documentation for this struct was generated from the following file:

- /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/Protocol/[Protocol.hpp](#)

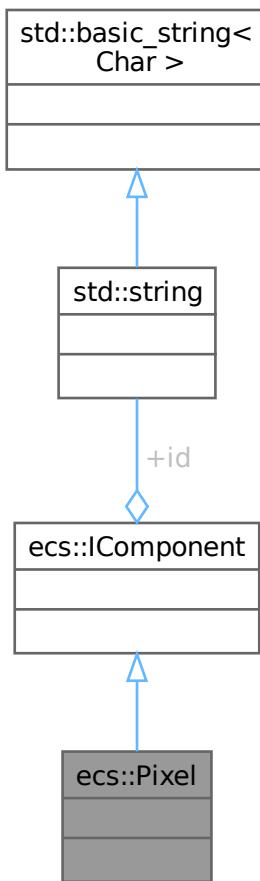
7.67 ecs::Pixel Struct Reference

```
#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`

7.67.1 Detailed Description

Definition at line 44 of file [Component.hpp](#).

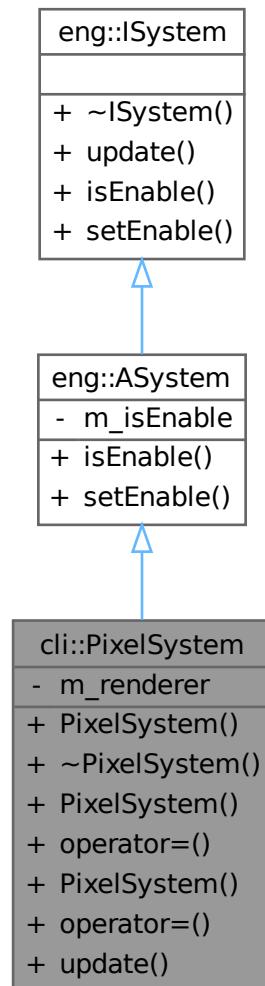
The documentation for this struct was generated from the following file:

- `/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.hpp`

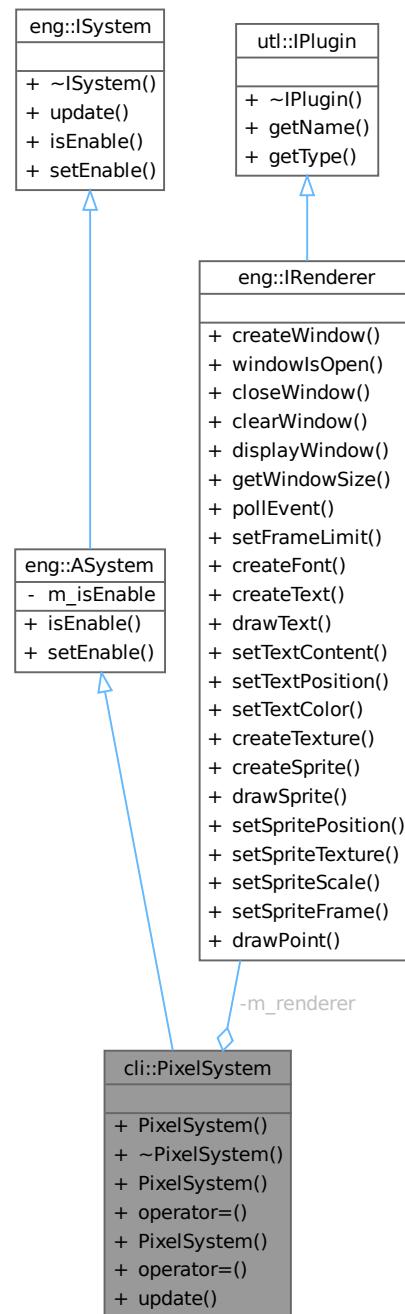
7.68 cli::PixelSystem Class Reference

```
#include <Pixel.hpp>
```

Inheritance diagram for cli::PixelSystem:



Collaboration diagram for cli::PixelSystem:



Public Member Functions

- `PixelSystem (eng::IRenderer &renderer)`
- `~PixelSystem () override=default`
- `PixelSystem (const PixelSystem &)=delete`
- `PixelSystem & operator= (const PixelSystem &)=delete`
- `PixelSystem (PixelSystem &&)=delete`
- `PixelSystem & operator= (PixelSystem &&)=delete`
- `void update (ecs::Registry ®istry, float) override`

Public Member Functions inherited from [eng::ASystem](#)

- bool [isEnable](#) () override
- void [setEnable](#) (const bool enable) override

Public Member Functions inherited from [eng::ISystem](#)

- virtual ~[ISystem](#) ()=default

Private Attributes

- [eng::IRenderer](#) & [m_renderer](#)

7.68.1 Detailed Description

Definition at line 16 of file [Pixel.hpp](#).

7.68.2 Constructor & Destructor Documentation

7.68.2.1 PixelSystem() [1/3]

```
cli::PixelSystem::PixelSystem (
    eng::IRenderer & renderer) [inline], [explicit]
```

Definition at line 19 of file [Pixel.hpp](#).

7.68.2.2 ~PixelSystem()

```
cli::PixelSystem::~PixelSystem () [override], [default]
```

7.68.2.3 PixelSystem() [2/3]

```
cli::PixelSystem::PixelSystem (
    const PixelSystem & ) [explicit], [delete]
```

7.68.2.4 PixelSystem() [3/3]

```
cli::PixelSystem::PixelSystem (
    PixelSystem && ) [explicit], [delete]
```

7.68.3 Member Function Documentation

7.68.3.1 operator=() [1/2]

```
PixelSystem & cli::PixelSystem::operator= (
    const PixelSystem & ) [delete]
```

7.68.3.2 operator=() [2/2]

```
PixelSystem & cli::PixelSystem::operator= (
    PixelSystem &&) [delete]
```

7.68.3.3 update()

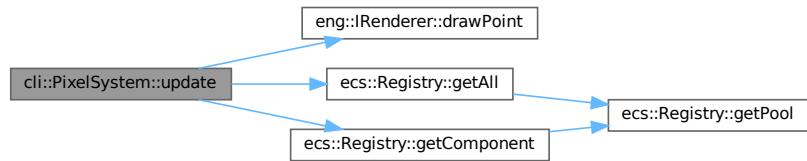
```
void cli::PixelSystem::update (
    ecs::Registry & registry,
    float ) [inline], [override], [virtual]
```

Implements [eng::ISystem](#).

Definition at line 27 of file [Pixel.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:



7.68.4 Member Data Documentation

7.68.4.1 m_renderer

```
eng::IRenderer& cli::PixelSystem::m_renderer [private]
```

Definition at line 39 of file [Pixel.hpp](#).

Referenced by [update\(\)](#).

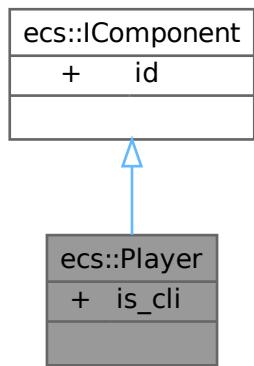
The documentation for this class was generated from the following file:

- /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/[Pixel.hpp](#)

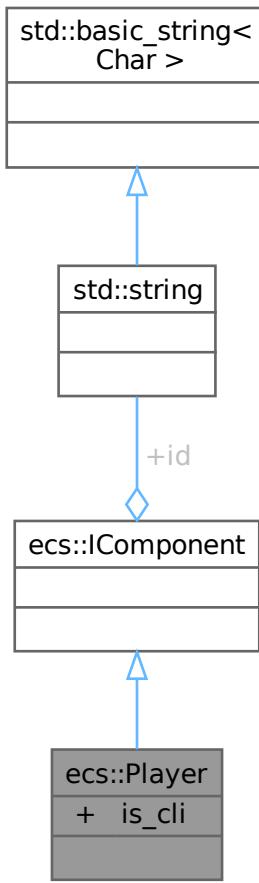
7.69 ecs::Player Struct Reference

```
#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`

7.69.1 Detailed Description

Definition at line 39 of file [Component.hpp](#).

7.69.2 Member Data Documentation

7.69.2.1 is_cli

```
bool ecs::Player::is_cli {}
```

Definition at line 41 of file [Component.hpp](#).

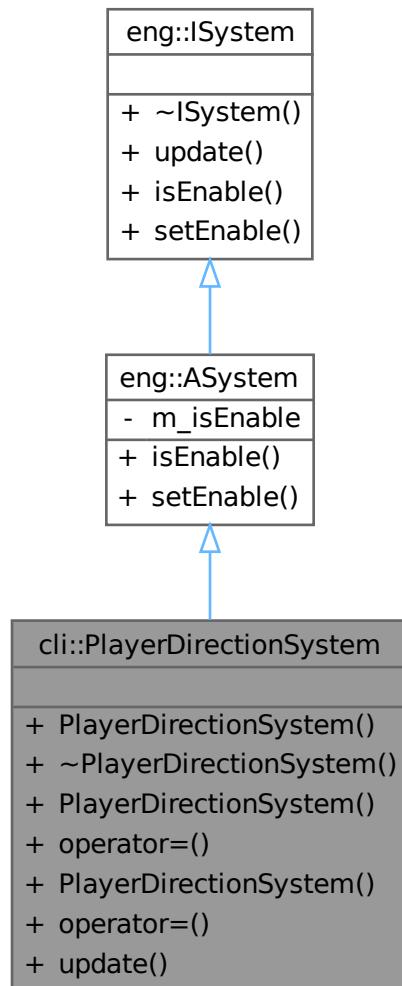
The documentation for this struct was generated from the following file:

- /home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/[Component.hpp](#)

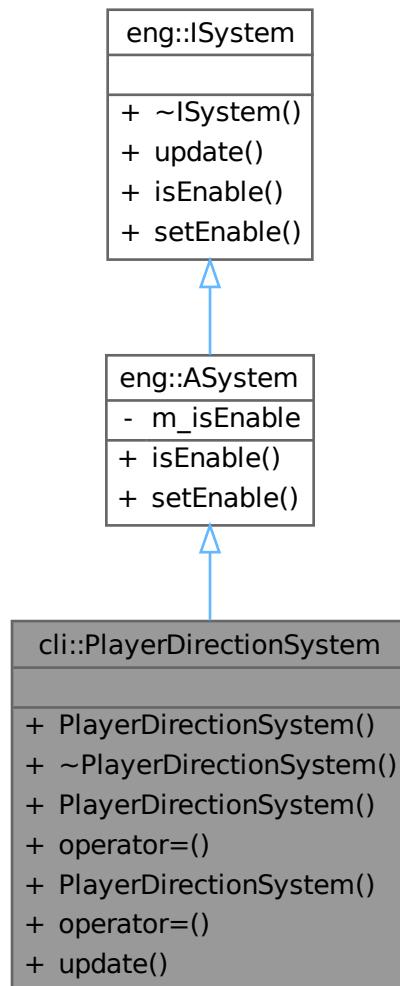
7.70 cli::PlayerDirectionSystem Class Reference

```
#include <PlayerDirection.hpp>
```

Inheritance diagram for cli::PlayerDirectionSystem:



Collaboration diagram for cli::PlayerDirectionSystem:



Public Member Functions

- `PlayerDirectionSystem ()=default`
- `~PlayerDirectionSystem () override=default`
- `PlayerDirectionSystem (const PlayerDirectionSystem &)=delete`
- `PlayerDirectionSystem & operator= (const PlayerDirectionSystem &)=delete`
- `PlayerDirectionSystem (PlayerDirectionSystem &&)=delete`
- `PlayerDirectionSystem & operator= (PlayerDirectionSystem &&)=delete`
- `void update (ecs::Registry ®istry, float) override`

Public Member Functions inherited from `eng::ASystem`

- `bool isEnabled () override`
- `void setEnable (const bool enable) override`

Public Member Functions inherited from `eng::ISystem`

- virtual `~ISystem ()=default`

7.70.1 Detailed Description

Definition at line 22 of file `PlayerDirection.hpp`.

7.70.2 Constructor & Destructor Documentation

7.70.2.1 `PlayerDirectionSystem()` [1/3]

`cli::PlayerDirectionSystem::PlayerDirectionSystem () [explicit], [default]`

7.70.2.2 `~PlayerDirectionSystem()`

`cli::PlayerDirectionSystem::~PlayerDirectionSystem () [override], [default]`

7.70.2.3 `PlayerDirectionSystem()` [2/3]

`cli::PlayerDirectionSystem::PlayerDirectionSystem (`
`const PlayerDirectionSystem &) [delete]`

7.70.2.4 `PlayerDirectionSystem()` [3/3]

`cli::PlayerDirectionSystem::PlayerDirectionSystem (`
`PlayerDirectionSystem &&) [delete]`

7.70.3 Member Function Documentation

7.70.3.1 `operator=()` [1/2]

`PlayerDirectionSystem & cli::PlayerDirectionSystem::operator= (`
`const PlayerDirectionSystem &) [delete]`

7.70.3.2 `operator=()` [2/2]

`PlayerDirectionSystem & cli::PlayerDirectionSystem::operator= (`
`PlayerDirectionSystem &&) [delete]`

7.70.3.3 update()

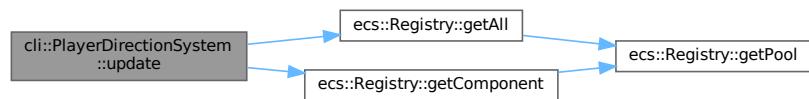
```
void cli::PlayerDirectionSystem::update (
    ecs::Registry & registry,
    float ) [inline], [override], [virtual]
```

Implements [eng::ISystem](#).

Definition at line 33 of file [PlayerDirection.hpp](#).

References [cli::GameConfig::Player::FRAMES_PER_ROW](#), [ecs::Registry::getAll\(\)](#), [ecs::Registry::getComponent\(\)](#), [M_PI](#), [cli::GameConfig::Player::SPRITE_HEIGHT](#), and [cli::GameConfig::Player::SPRITE_WIDTH](#).

Here is the call graph for this function:



The documentation for this class was generated from the following file:

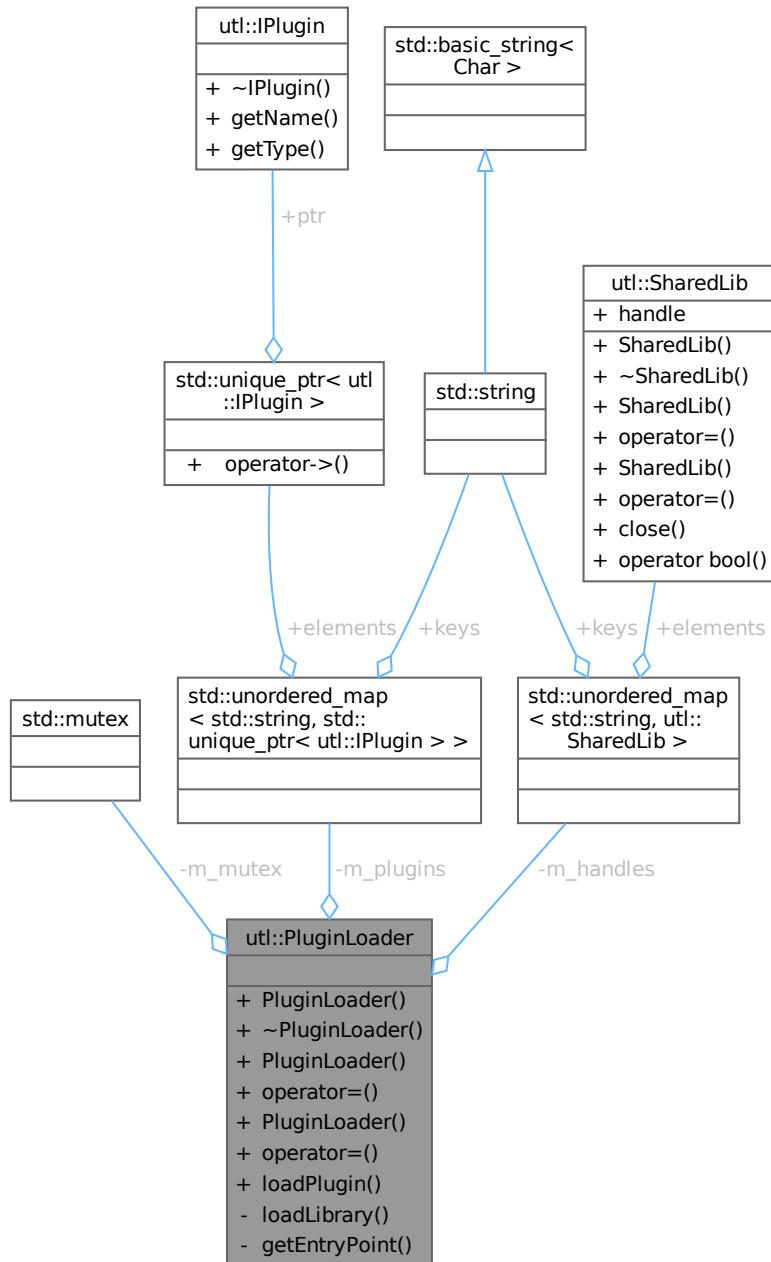
- /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/[PlayerDirection.hpp](#)

7.71 utl::PluginLoader Class Reference

Modern, type-safe plugin loader.

```
#include <PluginLoader.hpp>
```

Collaboration diagram for `utl::PluginLoader`:



Public Member Functions

- `PluginLoader ()=default`
- `~PluginLoader ()=default`
- `PluginLoader (const PluginLoader &)=delete`
- `PluginLoader & operator= (const PluginLoader &)=delete`
- `PluginLoader (PluginLoader &&)=delete`
- `PluginLoader & operator= (PluginLoader &&)=delete`

- template<std::derived_from< [IPlugin](#) > T>
std::shared_ptr< T > [loadPlugin](#) (const std::string &path)
Load a plugin of type T.

Private Member Functions

- [SharedLib loadLibrary](#) (const std::string &path)
- [EntryPointFn getEntryPoint](#) ([SharedLib](#) &lib, const std::string &path)

Private Attributes

- std::mutex [m_mutex](#)
- std::unordered_map< std::string, [SharedLib](#) > [m_handles](#)
- std::unordered_map< std::string, std::unique_ptr< [IPlugin](#) > > [m_plugins](#)

7.71.1 Detailed Description

Modern, type-safe plugin loader.

Definition at line 85 of file [PluginLoader.hpp](#).

7.71.2 Constructor & Destructor Documentation

7.71.2.1 PluginLoader() [1/3]

utl::PluginLoader::PluginLoader () [default]

7.71.2.2 ~PluginLoader()

utl::PluginLoader::~PluginLoader () [default]

7.71.2.3 PluginLoader() [2/3]

utl::PluginLoader::PluginLoader (const [PluginLoader](#) &) [delete]

7.71.2.4 PluginLoader() [3/3]

utl::PluginLoader::PluginLoader ([PluginLoader](#) &&) [delete]

7.71.3 Member Function Documentation

7.71.3.1 getEntryPoint()

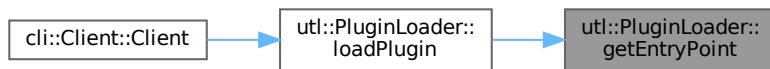
```
EntryPointFn utl::PluginLoader::getEntryPoint (
    SharedLib & lib,
    const std::string & path) [inline], [private]
```

Definition at line 163 of file [PluginLoader.hpp](#).

References [utl::SharedLib::handle](#).

Referenced by [loadPlugin\(\)](#).

Here is the caller graph for this function:



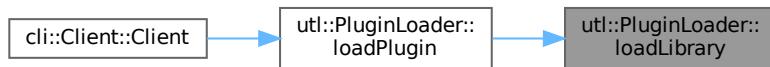
7.71.3.2 loadLibrary()

```
SharedLib utl::PluginLoader::loadLibrary (
    const std::string & path) [inline], [private]
```

Definition at line 137 of file [PluginLoader.hpp](#).

Referenced by [loadPlugin\(\)](#).

Here is the caller graph for this function:



7.71.3.3 loadPlugin()

```
template<std::derived_from< IPlugin > T>
std::shared_ptr< T > utl::PluginLoader::loadPlugin (
    const std::string & path) [inline]
```

Load a plugin of type T.

Template Parameters

T	Expected plugin interface (must derive from IPlugin)
---	---

Parameters

path	Path to the dynamic library
------	-----------------------------

Returns

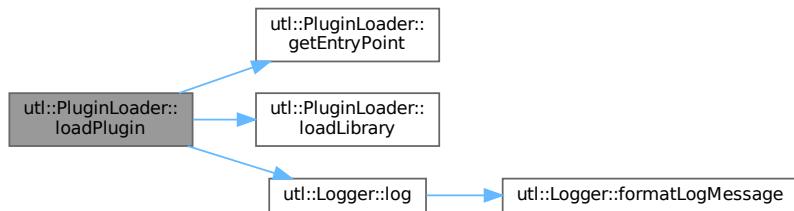
`shared_ptr<T>` instance

Definition at line 102 of file [PluginLoader.hpp](#).

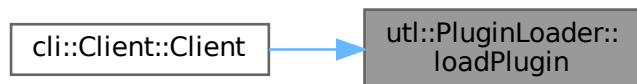
References [getEntryPoint\(\)](#), [utl::INFO](#), [loadLibrary\(\)](#), [utl::Logger::log\(\)](#), [m_handles](#), [m_mutex](#), and [m_plugins](#).

Referenced by [cli::Client::Client\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.71.3.4 operator=() [1/2]

```
PluginLoader & utl::PluginLoader::operator= (
    const PluginLoader & ) [delete]
```

7.71.3.5 operator=() [2/2]

```
PluginLoader & utl::PluginLoader::operator= (
    PluginLoader &&) [delete]
```

7.71.4 Member Data Documentation

7.71.4.1 m_handles

```
std::unordered_map<std::string, SharedLib> utl::PluginLoader::m_handles [private]
```

Definition at line 134 of file [PluginLoader.hpp](#).

Referenced by [loadPlugin\(\)](#).

7.71.4.2 m_mutex

```
std::mutex utl::PluginLoader::m_mutex [private]
```

Definition at line 133 of file [PluginLoader.hpp](#).

Referenced by [loadPlugin\(\)](#).

7.71.4.3 m_plugins

```
std::unordered_map<std::string, std::unique_ptr<IPlugin>> utl::PluginLoader::m_plugins [private]
```

Definition at line 135 of file [PluginLoader.hpp](#).

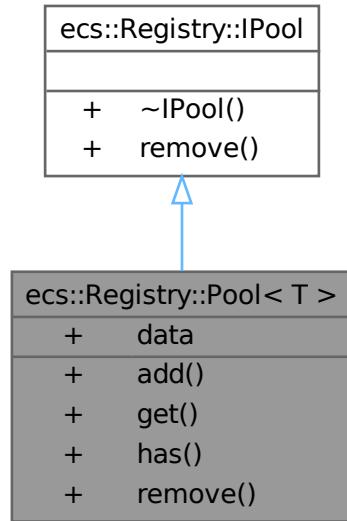
Referenced by [loadPlugin\(\)](#).

The documentation for this class was generated from the following file:

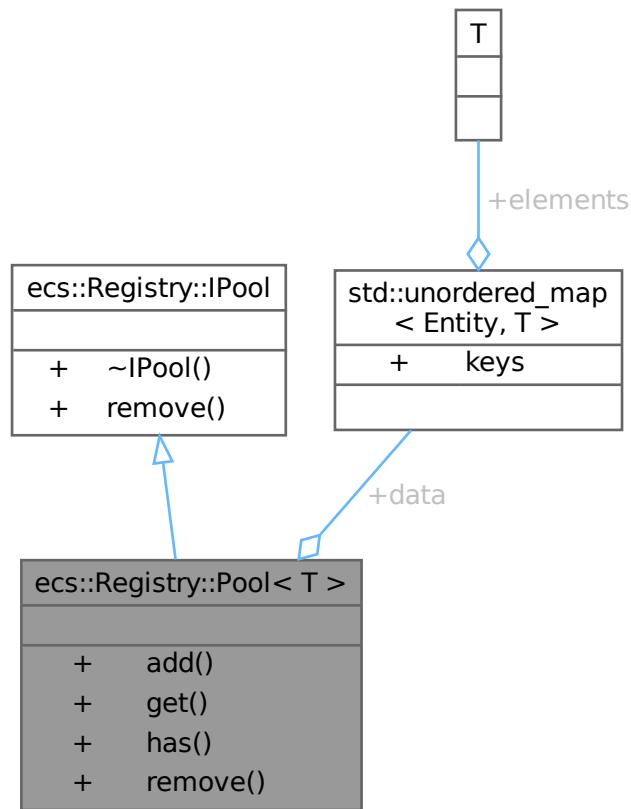
- /home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/[PluginLoader.hpp](#)

7.72 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`

7.72.1 Detailed Description

```
template<typename T>
class ecs::Registry::Pool< T >
```

Definition at line 104 of file [Registry.hpp](#).

7.72.2 Member Function Documentation

7.72.2.1 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](#).

7.72.2.2 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](#).

7.72.2.3 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](#).

7.72.2.4 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](#).

7.72.3 Member Data Documentation

7.72.3.1 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\(\)](#).

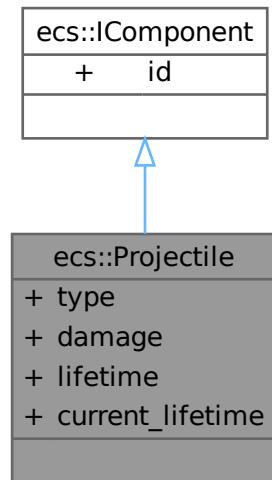
The documentation for this class was generated from the following file:

- [/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Registry.hpp](#)

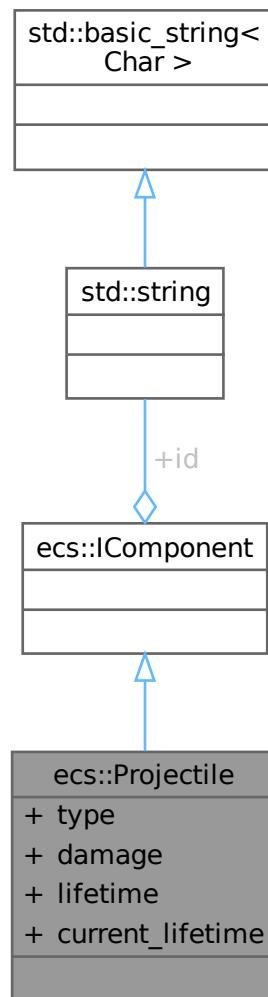
7.73 ecs::Projectile Struct Reference

```
#include <Component.hpp>
```

Inheritance diagram for ecs::Projectile:



Collaboration diagram for ecs::Projectile:



Public Types

- enum `Type` { `BASIC` , `SUPERCHARGED` }

Public Attributes

- `Type type`
- float `damage`
- float `lifetime`
- float `current_lifetime`

Public Attributes inherited from `ecs::IComponent`

- `std::string id`

7.73.1 Detailed Description

Definition at line 86 of file [Component.hpp](#).

7.73.2 Member Enumeration Documentation

7.73.2.1 Type

enum `ecs::Projectile::Type`

Enumerator

BASIC	
SUPERCHARGED	

Definition at line 88 of file [Component.hpp](#).

7.73.3 Member Data Documentation

7.73.3.1 current_lifetime

float `ecs::Projectile::current_lifetime`

Definition at line 96 of file [Component.hpp](#).

7.73.3.2 damage

float `ecs::Projectile::damage`

Definition at line 94 of file [Component.hpp](#).

7.73.3.3 lifetime

float `ecs::Projectile::lifetime`

Definition at line 95 of file [Component.hpp](#).

7.73.3.4 type

`Type` `ecs::Projectile::type`

Definition at line 93 of file [Component.hpp](#).

The documentation for this struct was generated from the following file:

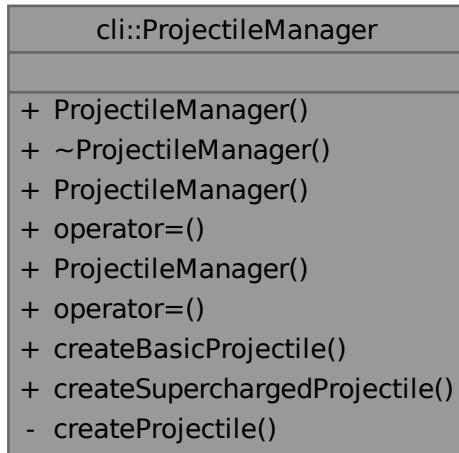
- /home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/[Component.hpp](#)

7.74 cli::ProjectileManager Class Reference

Handles projectile creation and management.

```
#include <ProjectileManager.hpp>
```

Collaboration diagram for cli::ProjectileManager:



Public Member Functions

- `ProjectileManager ()=default`
- `~ProjectileManager ()=default`
- `ProjectileManager (const ProjectileManager &)=delete`
- `ProjectileManager & operator= (const ProjectileManager &)=delete`
- `ProjectileManager (ProjectileManager &&)=delete`
- `ProjectileManager & operator= (ProjectileManager &&)=delete`

Static Public Member Functions

- static `ecs::Entity createBasicProjectile (ecs::Registry ®istry, float x, float y, float velocityX, float velocityY)`
Create a basic projectile.
- static `ecs::Entity createSuperchargedProjectile (ecs::Registry ®istry, float x, float y, float velocityX, float velocityY)`
Create a supercharged projectile.

Static Private Member Functions

- static `ecs::Entity createProjectile (ecs::Registry ®istry, ecs::Projectile::Type type, float x, float y, float velocityX, float velocityY)`
Create a projectile with given parameters.

7.74.1 Detailed Description

Handles projectile creation and management.

Definition at line 20 of file [ProjectileManager.hpp](#).

7.74.2 Constructor & Destructor Documentation

7.74.2.1 ProjectileManager() [1/3]

```
cli::ProjectileManager::ProjectileManager () [default]
```

7.74.2.2 ~ProjectileManager()

```
cli::ProjectileManager::~ProjectileManager () [default]
```

7.74.2.3 ProjectileManager() [2/3]

```
cli::ProjectileManager::ProjectileManager (
    const ProjectileManager & ) [delete]
```

7.74.2.4 ProjectileManager() [3/3]

```
cli::ProjectileManager::ProjectileManager (
    ProjectileManager && ) [delete]
```

7.74.3 Member Function Documentation

7.74.3.1 createBasicProjectile()

```
ecs::Entity cli::ProjectileManager::createBasicProjectile (
    ecs::Registry & registry,
    float x,
    float y,
    float velocityX,
    float velocityY) [static]
```

Create a basic projectile.

Parameters

registry	The ECS registry
x	X position
y	Y position
velocityX	X velocity
velocityY	Y velocity

Returns

The created entity

Definition at line 13 of file [ProjectileManager.cpp](#).

References [ecs::Projectile::BASIC](#), and [createProjectile\(\)](#).

Referenced by [cli::WeaponSystem::tryFireBasic\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.74.3.2 createProjectile()

```

ecs::Entity cli::ProjectileManager::createProjectile (
    ecs::Registry & registry,
    ecs::Projectile::Type type,
    float x,
    float y,
    float velocityX,
    float velocityY) [static], [private]
  
```

Create a projectile with given parameters.

Parameters

<code>registry</code>	The ECS registry
<code>type</code>	Projectile type
<code>x</code>	X position
<code>y</code>	Y position
<code>velocityX</code>	X velocity
<code>velocityY</code>	Y velocity

Returns

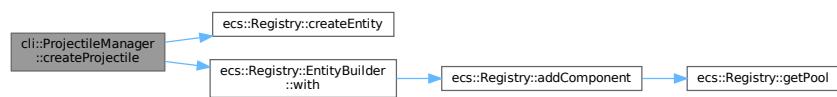
The created entity

Definition at line 25 of file [ProjectileManager.cpp](#).

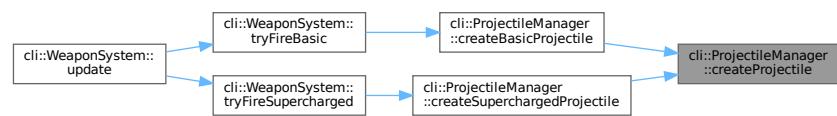
References `ecs::Projectile::BASIC`, `ecs::Registry::createEntity()`, `cli::GameConfig::Hitbox::PROJECTILE_BASIC_RADIUS`, `cli::GameConfig::Hitbox::PROJECTILE_SUPERCHARGED_RADIUS`, `cli::Path::Texture::TEXTURE_SHOOT`, `cli::Path::Texture::TEXTURE_SHOOT_CHARGED`, and `ecs::Registry::EntityBuilder::with()`.

Referenced by `createBasicProjectile()`, and `createSuperchargedProjectile()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.74.3.3 createSuperchargedProjectile()

```

ecs::Entity cli::ProjectileManager::createSuperchargedProjectile (
    ecs::Registry & registry,
    float x,
    float y,
    float velocityX,
    float velocityY) [static]
  
```

Create a supercharged projectile.

Parameters

<code>registry</code>	The ECS registry
<code>x</code>	X position
<code>y</code>	Y position
<code>velocityX</code>	X velocity
<code>velocityY</code>	Y velocity

Returns

The created entity

Definition at line 19 of file [ProjectileManager.cpp](#).

References [createProjectile\(\)](#), and [ecs::Projectile::SUPERCHARGED](#).

Referenced by [cli::WeaponSystem::tryFireSupercharged\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.74.3.4 operator=() [1/2]

```
ProjectileManager & cli::ProjectileManager::operator=
    const ProjectileManager & ) [delete]
```

7.74.3.5 operator=() [2/2]

```
ProjectileManager & cli::ProjectileManager::operator=
    ProjectileManager && ) [delete]
```

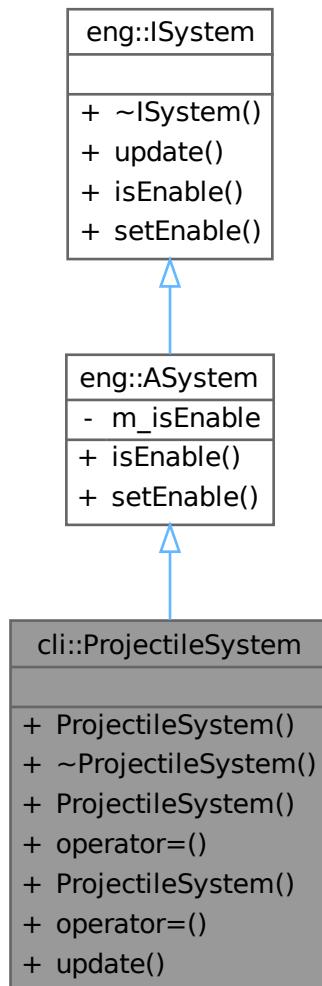
The documentation for this class was generated from the following files:

- /home/masina/Projects/Epitech/rtype/client/include/Client/[ProjectileManager.hpp](#)
- /home/masina/Projects/Epitech/rtype/client/src/[ProjectileManager.cpp](#)

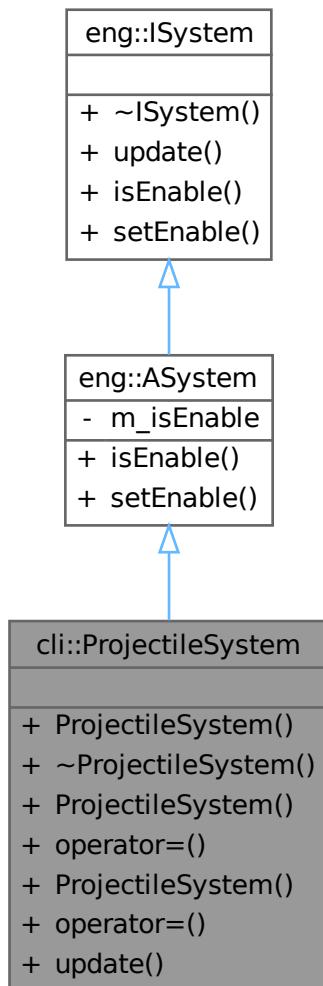
7.75 cli::ProjectileSystem Class Reference

```
#include <Projectile.hpp>
```

Inheritance diagram for cli::ProjectileSystem:



Collaboration diagram for cli::ProjectileSystem:



Public Member Functions

- `ProjectileSystem (eng::IRenderer &)`
- `~ProjectileSystem () override=default`
- `ProjectileSystem (const ProjectileSystem &)=delete`
- `ProjectileSystem & operator= (const ProjectileSystem &)=delete`
- `ProjectileSystem (ProjectileSystem &&)=delete`
- `ProjectileSystem & operator= (ProjectileSystem &&)=delete`
- `void update (ecs::Registry ®istry, float dt) override`

Public Member Functions inherited from `eng::ASystem`

- `bool isEnabled () override`
- `void setEnable (const bool enable) override`

Public Member Functions inherited from [eng::ISystem](#)

- virtual [~ISystem](#) ()=default

7.75.1 Detailed Description

Definition at line [18](#) of file [Projectile.hpp](#).

7.75.2 Constructor & Destructor Documentation

7.75.2.1 ProjectileSystem() [1/3]

```
cli::ProjectileSystem::ProjectileSystem (
    eng::IRenderer & ) [inline], [explicit]
```

Definition at line [21](#) of file [Projectile.hpp](#).

7.75.2.2 ~ProjectileSystem()

```
cli::ProjectileSystem::~ProjectileSystem () [override], [default]
```

7.75.2.3 ProjectileSystem() [2/3]

```
cli::ProjectileSystem::ProjectileSystem (
    const ProjectileSystem & ) [delete]
```

7.75.2.4 ProjectileSystem() [3/3]

```
cli::ProjectileSystem::ProjectileSystem (
    ProjectileSystem && ) [delete]
```

7.75.3 Member Function Documentation

7.75.3.1 operator=() [1/2]

```
ProjectileSystem & cli::ProjectileSystem::operator= (
    const ProjectileSystem & ) [delete]
```

7.75.3.2 operator=() [2/2]

```
ProjectileSystem & cli::ProjectileSystem::operator= (
    ProjectileSystem && ) [delete]
```

7.75.3.3 update()

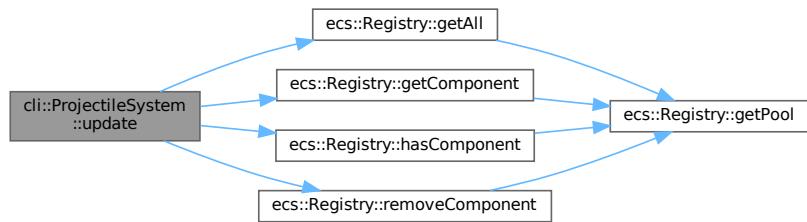
```
void cli::ProjectileSystem::update (
    ecs::Registry & registry,
    float dt) [inline], [override], [virtual]
```

Implements [eng::ISystem](#).

Definition at line 29 of file [Projectile.hpp](#).

References [ecs::Registry::getAll\(\)](#), [ecs::Registry::getComponent\(\)](#), [ecs::Registry::hasComponent\(\)](#), [ecs::Registry::removeComponent\(\)](#), [ecs::Transform::x](#), [ecs::Velocity::x](#), and [ecs::Transform::y](#).

Here is the call graph for this function:



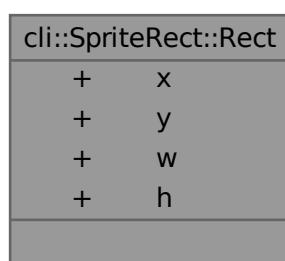
The documentation for this class was generated from the following file:

- /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/[Projectile.hpp](#)

7.76 cli::SpriteRect::Rect Struct Reference

```
#include <SpriteRect.hpp>
```

Collaboration diagram for cli::SpriteRect::Rect:



Public Attributes

- int `x`
- int `y`
- int `w`
- int `h`

7.76.1 Detailed Description

Definition at line 42 of file [SpriteRect.hpp](#).

7.76.2 Member Data Documentation

7.76.2.1 `h`

int cli::SpriteRect::Rect::`h`

Definition at line 44 of file [SpriteRect.hpp](#).

7.76.2.2 `w`

int cli::SpriteRect::Rect::`w`

Definition at line 44 of file [SpriteRect.hpp](#).

7.76.2.3 `x`

int cli::SpriteRect::Rect::`x`

Definition at line 44 of file [SpriteRect.hpp](#).

7.76.2.4 `y`

int cli::SpriteRect::Rect::`y`

Definition at line 44 of file [SpriteRect.hpp](#).

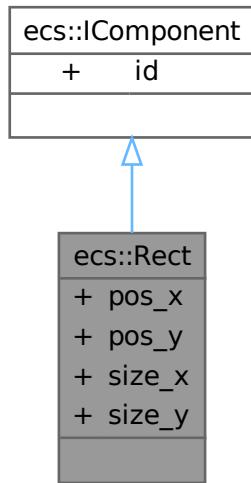
The documentation for this struct was generated from the following file:

- /home/masina/Projects/Epitech/rtype/client/include/Client/[SpriteRect.hpp](#)

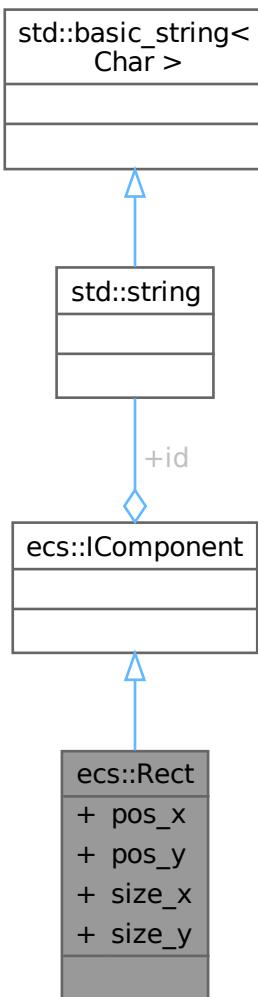
7.77 ecs::Rect Struct Reference

```
#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`

7.77.1 Detailed Description

Definition at line 47 of file [Component.hpp](#).

7.77.2 Member Data Documentation

7.77.2.1 pos_x

```
float ecs::Rect::pos_x {}
```

Definition at line 49 of file [Component.hpp](#).

7.77.2.2 pos_y

```
float ecs::Rect::pos_y {}
```

Definition at line 49 of file [Component.hpp](#).

7.77.2.3 size_x

```
int ecs::Rect::size_x {}
```

Definition at line 50 of file [Component.hpp](#).

7.77.2.4 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:

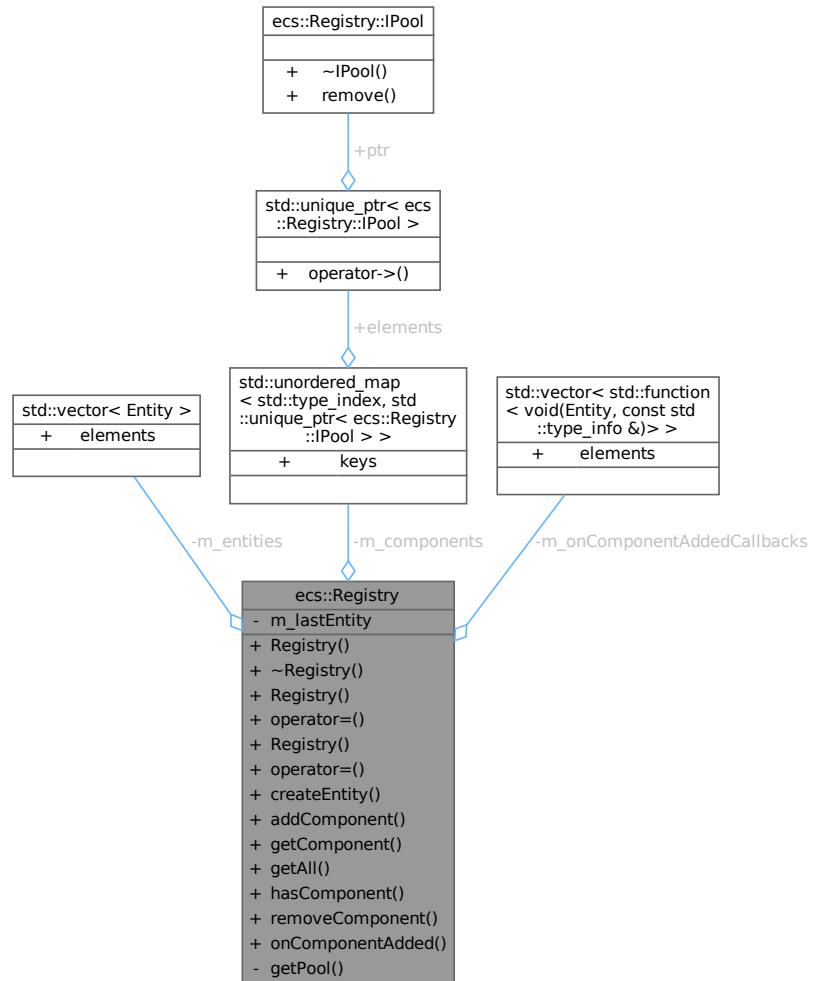
- /home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/[Component.hpp](#)

7.78 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 & operator=(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](#)

7.78.1 Detailed Description

Class for managing entities and their components.

Definition at line 24 of file [Registry.hpp](#).

7.78.2 Constructor & Destructor Documentation

7.78.2.1 Registry() [1/3]

`ecs::Registry::Registry () [default]`

7.78.2.2 ~Registry()

`ecs::Registry::~Registry () [default]`

7.78.2.3 Registry() [2/3]

`ecs::Registry::Registry (`
`const Registry &) [delete]`

7.78.2.4 Registry() [3/3]

`ecs::Registry::Registry (`
`Registry &&) [delete]`

7.78.3 Member Function Documentation

7.78.3.1 addComponent()

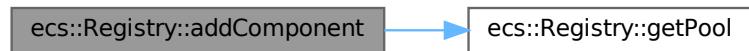
```
template<typename T , typename... Args>
T & ecs::Registry::addComponent (
    Entity e,
    Args &&... args) [inline]
```

Definition at line 60 of file [Registry.hpp](#).

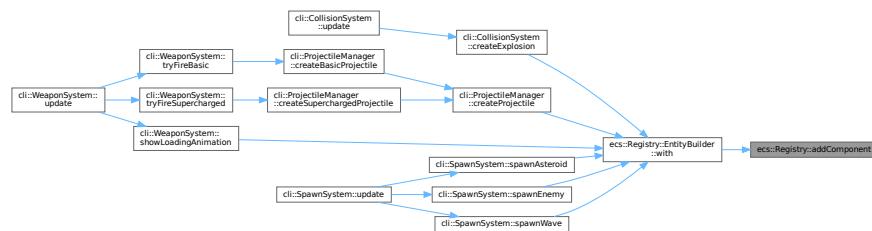
References [getPool\(\)](#), and [m_onComponentAddedCallbacks](#).

Referenced by [ecs::Registry::EntityBuilder::with\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.78.3.2 createEntity()

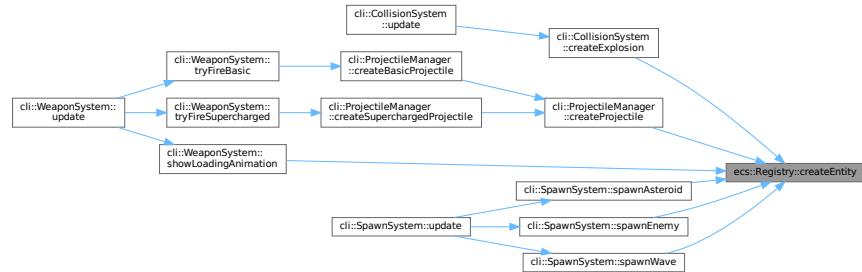
```
EntityBuilder ecs::Registry::createEntity () [inline]
```

Definition at line 53 of file [Registry.hpp](#).

References [m_entities](#), and [m_lastEntity](#).

Referenced by [cli::CollisionSystem::createExplosion\(\)](#), [cli::ProjectileManager::createProjectile\(\)](#), [cli::WeaponSystem::showLoadingAnimation\(\)](#), [cli::SpawnSystem::spawnAsteroid\(\)](#), [cli::SpawnSystem::spawnEnemy\(\)](#), and [cli::SpawnSystem::spawnWave\(\)](#).

Here is the caller graph for this function:



7.78.3.3 getAll()

```
template<typename T >
std::unordered_map< Entity, T > & ecs::Registry::getAll () [inline]
```

Definition at line 77 of file [Registry.hpp](#).

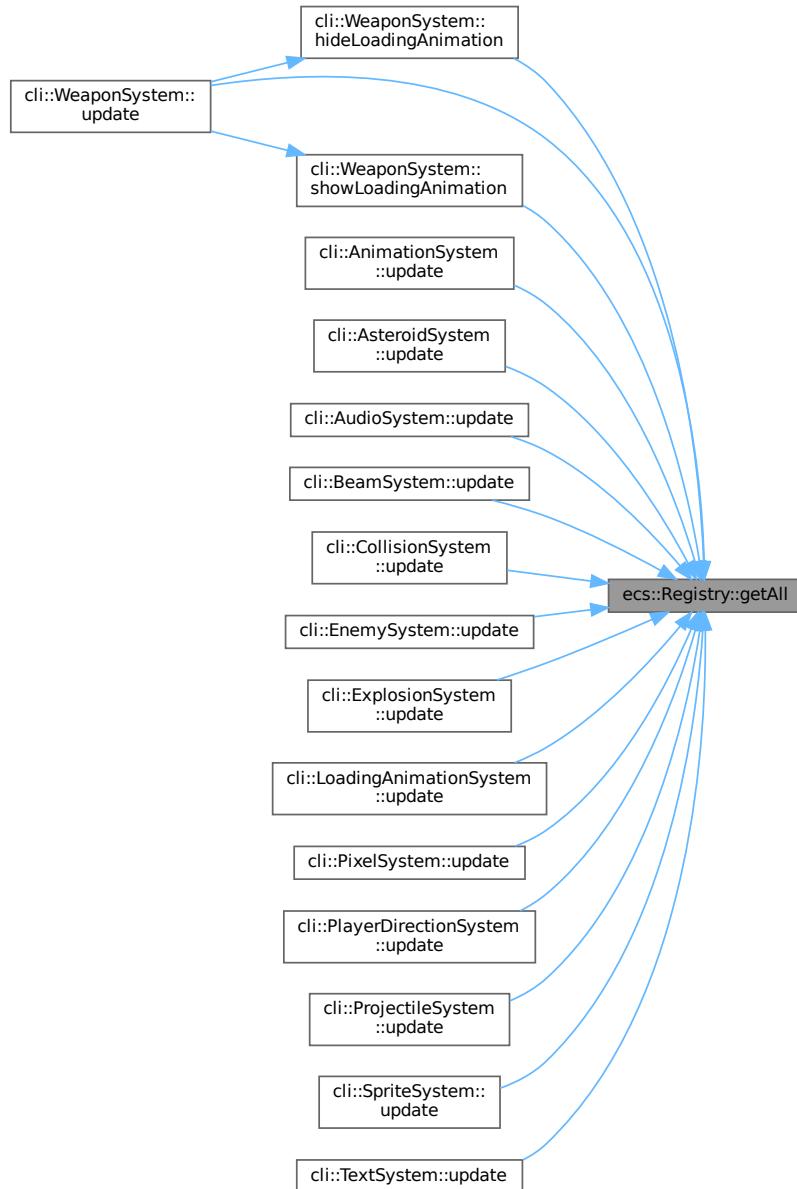
References [getPool\(\)](#).

Referenced by [cli::WeaponSystem::hideLoadingAnimation\(\)](#), [cli::WeaponSystem::showLoadingAnimation\(\)](#), [cli::AnimationSystem::update\(\)](#), [cli::AsteroidSystem::update\(\)](#), [cli::AudioSystem::update\(\)](#), [cli::BeamSystem::update\(\)](#), [cli::CollisionSystem::update\(\)](#), [cli::EnemySystem::update\(\)](#), [cli::ExplosionSystem::update\(\)](#), [cli::LoadingAnimationSystem::update\(\)](#), [cli::PixelSystem::update\(\)](#), [cli::PlayerDirectionSystem::update\(\)](#), [cli::ProjectileSystem::update\(\)](#), [cli::SpriteSystem::update\(\)](#), [cli::TextSystem::update\(\)](#), and [cli::WeaponSystem::update\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.78.3.4 getComponent()

```
template<typename T >
T * ecs::Registry::getComponent (
    Entity e) [inline]
```

Definition at line 71 of file [Registry.hpp](#).

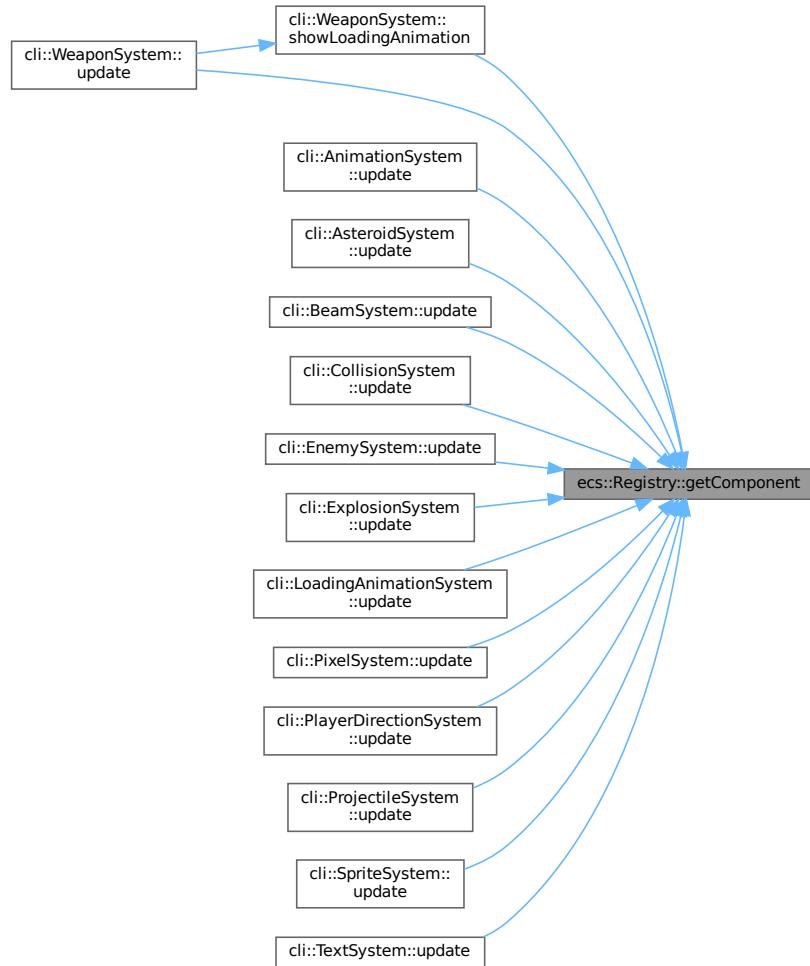
References [getPool\(\)](#).

Referenced by [cli::WeaponSystem::showLoadingAnimation\(\)](#), [cli::AnimationSystem::update\(\)](#), [cli::AsteroidSystem::update\(\)](#), [cli::BeamSystem::update\(\)](#), [cli::CollisionSystem::update\(\)](#), [cli::EnemySystem::update\(\)](#), [cli::ExplosionSystem::update\(\)](#), [cli::LoadingAnimationSystem::update\(\)](#), [cli::PixelSystem::update\(\)](#), [cli::PlayerDirectionSystem::update\(\)](#), [cli::ProjectileSystem::update\(\)](#), [cli::SpriteSystem::update\(\)](#), [cli::TextSystem::update\(\)](#), and [cli::WeaponSystem::update\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.78.3.5 getPool()

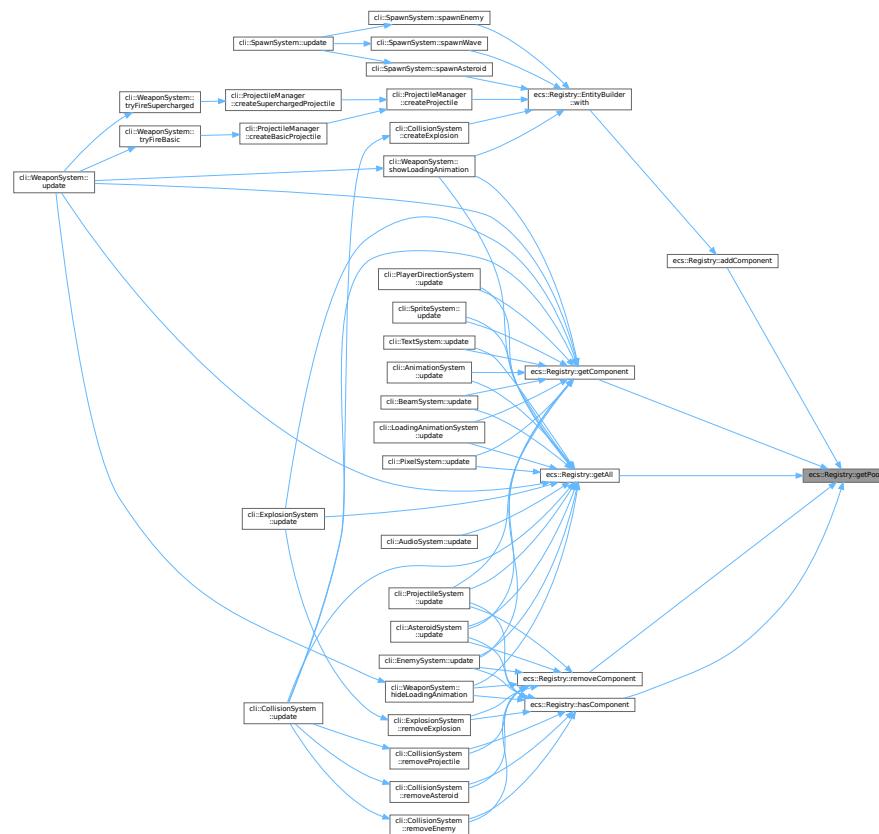
```
template<typename T >
Pool< T > & ecs::Registry::getPool () [inline], [private]
```

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:



7.78.3.6 hasComponent()

```
template<typename T >
bool ecs::Registry::hasComponent (
    Entity e) [inline]
```

Definition at line 79 of file [Registry.hpp](#).

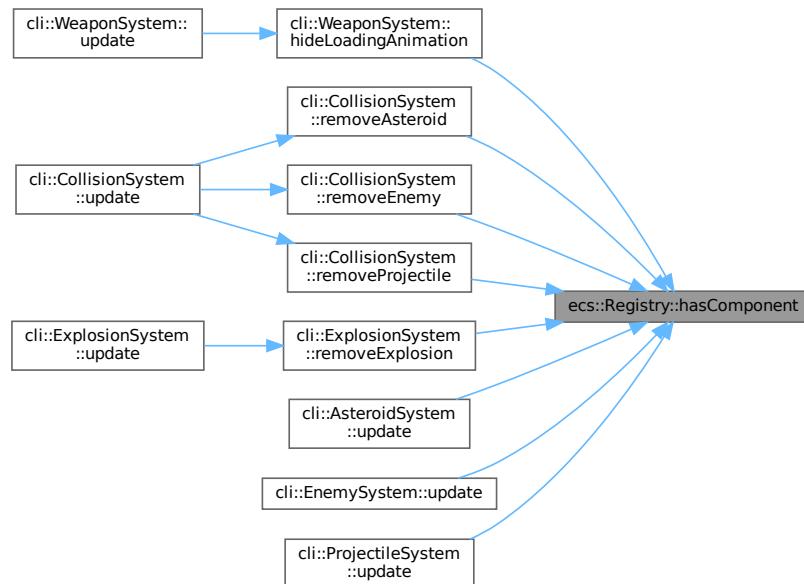
References [getPool\(\)](#).

Referenced by [cl::WeaponSystem::hideLoadingAnimation\(\)](#), [cl::CollisionSystem::removeAsteroid\(\)](#), [cl::CollisionSystem::removeEnemy\(\)](#), [cl::ExplosionSystem::removeExplosion\(\)](#), [cl::CollisionSystem::removeProjectile\(\)](#), [cl::AsteroidSystem::update\(\)](#), [cl::EnemySystem::update\(\)](#), and [cl::ProjectileSystem::update\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.78.3.7 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](#).

7.78.3.8 operator=() [1/2]

```
Registry & ecs::Registry::operator= (
    const Registry & ) [delete]
```

7.78.3.9 operator=() [2/2]

```
Registry & ecs::Registry::operator= (
    Registry && ) [delete]
```

7.78.3.10 removeComponent()

```
template<typename T >
void ecs::Registry::removeComponent (
    Entity e) [inline]
```

Definition at line 85 of file [Registry.hpp](#).

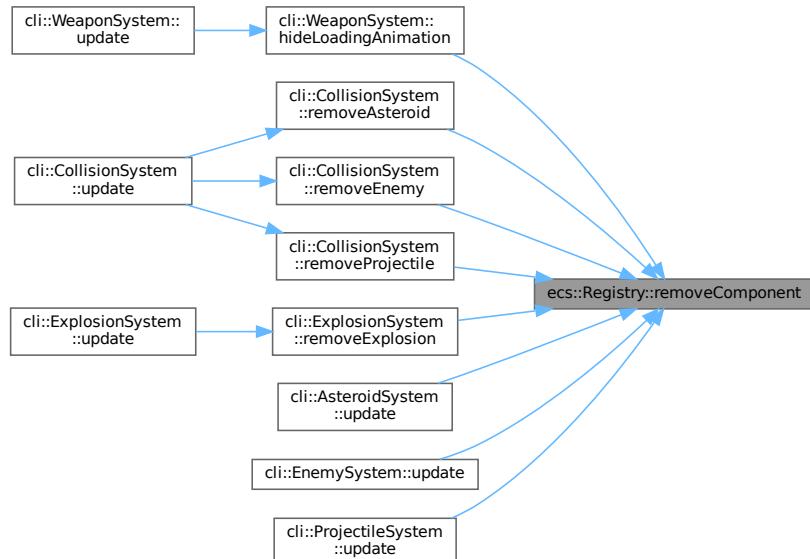
References [getPool\(\)](#).

Referenced by [cli::WeaponSystem::hideLoadingAnimation\(\)](#), [cli::CollisionSystem::removeAsteroid\(\)](#), [cli::CollisionSystem::removeEnemy\(\)](#), [cli::ExplosionSystem::removeExplosion\(\)](#), [cli::CollisionSystem::removeProjectile\(\)](#), [cli::AsteroidSystem::update\(\)](#), [cli::EnemySystem::update\(\)](#), and [cli::ProjectileSystem::update\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.78.4 Member Data Documentation

7.78.4.1 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\(\)](#).

7.78.4.2 m_entities

std::vector<[Entity](#)> ecs::Registry::m_entities [private]

Definition at line 139 of file [Registry.hpp](#).

Referenced by [createEntity\(\)](#).

7.78.4.3 m_lastEntity

[Entity](#) ecs::Registry::m_lastEntity = INVALID_ENTITY [private]

Definition at line 138 of file [Registry.hpp](#).

Referenced by [createEntity\(\)](#).

7.78.4.4 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:

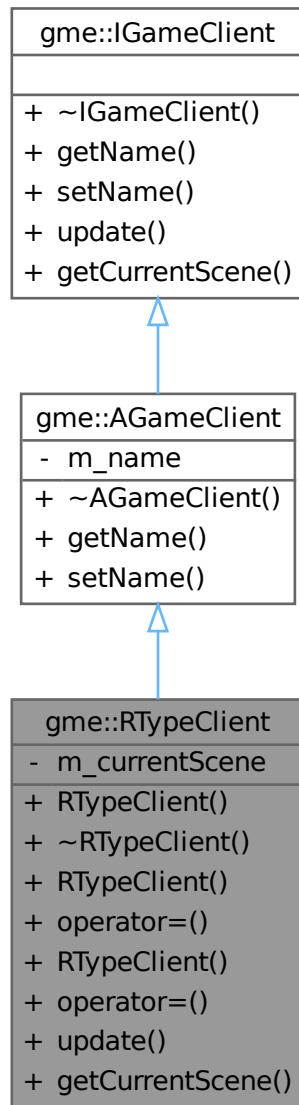
- /home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/[Registry.hpp](#)

7.79 gme::RTypeClient Class Reference

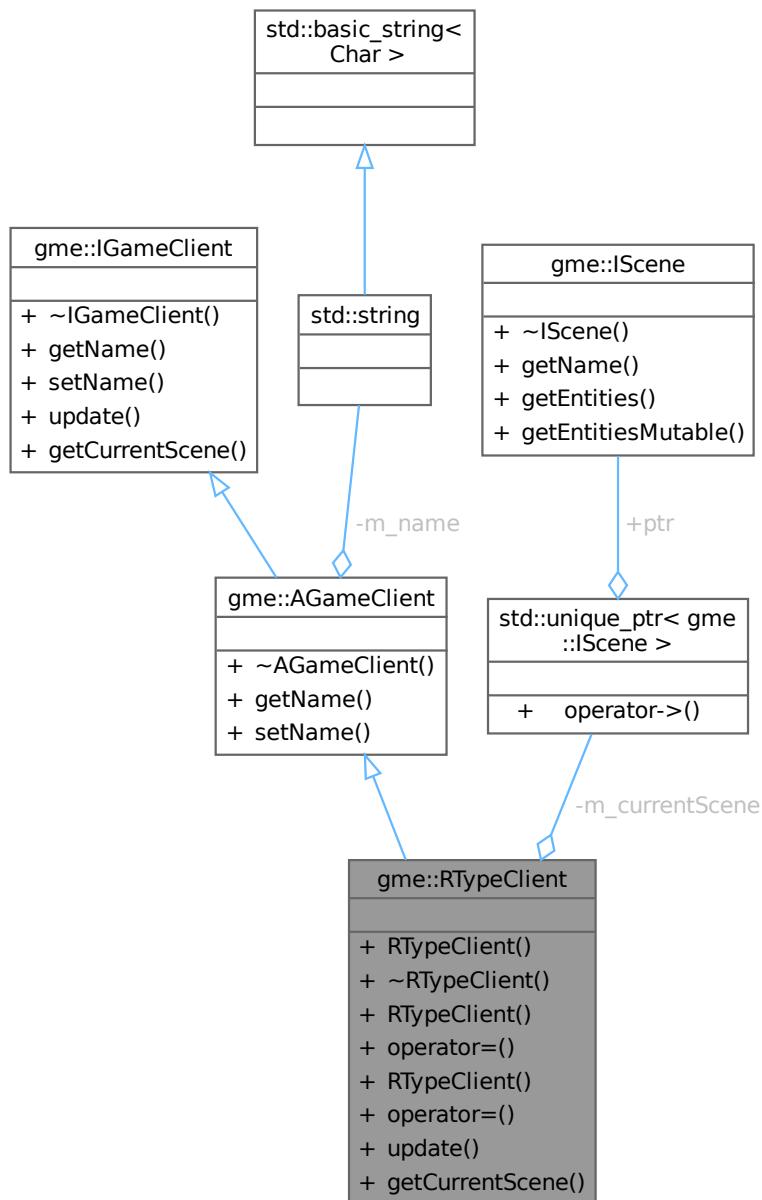
Class for the R-Type game.

```
#include <RTypeClient.hpp>
```

Inheritance diagram for gme::RTypeClient:



Collaboration diagram for gme::RTypClient:



Public Member Functions

- `RTypClient ()`
- `~RTypClient () override=default`
- `RTypClient (const RTypClient &)=delete`
- `RTypClient & operator= (const RTypClient &)=delete`
- `RTypClient (RTypClient &&)=delete`
- `RTypClient & operator= (RTypClient &&)=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`

7.79.1 Detailed Description

Class for the R-Type game.

Definition at line 21 of file [RTYPEClient.hpp](#).

7.79.2 Constructor & Destructor Documentation

7.79.2.1 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:



7.79.2.2 ~RTYPEClient()

`gme::RTYPEClient::~RTYPEClient () [override], [default]`

7.79.2.3 RTYPEClient() [2/3]

`gme::RTYPEClient::RTYPEClient (`
`const RTYPEClient &) [delete]`

7.79.2.4 RTypeClient() [3/3]

```
gme::RTypeClient::RTypeClient (
    RTypeClient && ) [delete]
```

7.79.3 Member Function Documentation

7.79.3.1 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](#).

7.79.3.2 operator=() [1/2]

```
RTypeClient & gme::RTypeClient::operator= (
    const RTypeClient & ) [delete]
```

7.79.3.3 operator=() [2/2]

```
RTypeClient & gme::RTypeClient::operator= (
    RTypeClient && ) [delete]
```

7.79.3.4 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](#).

7.79.4 Member Data Documentation

7.79.4.1 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:

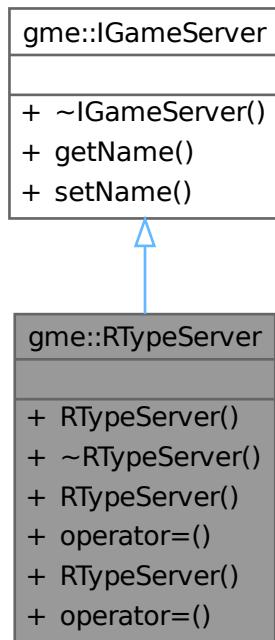
- /home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Client/include/R-TypeClient/[RTypeClient.hpp](#)
- /home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Client/src/[rtypeClient.cpp](#)

7.80 gme::RTypeServer Class Reference

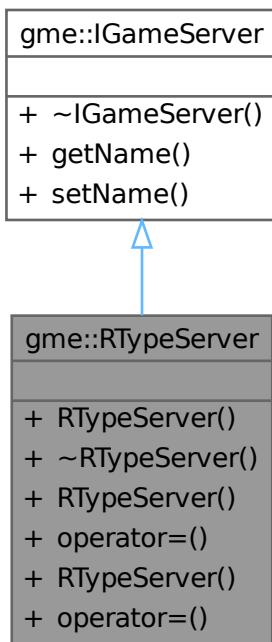
Class for the R-Type game.

```
#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)`

7.80.1 Detailed Description

Class for the R-Type game.

Definition at line 19 of file [RTypeServer.hpp](#).

7.80.2 Constructor & Destructor Documentation

7.80.2.1 RTypeServer() [1/3]

```
gme::RTypeServer::RTypeServer () [default]
```

7.80.2.2 ~RTypeServer()

```
gme::RTypeServer::~RTypeServer () [override], [default]
```

7.80.2.3 RTypeServer() [2/3]

```
gme::RTypeServer::RTypeServer (
    const RTypeServer & ) [delete]
```

7.80.2.4 RTypeServer() [3/3]

```
gme::RTypeServer::RTypeServer (
    RTypeServer && ) [delete]
```

7.80.3 Member Function Documentation

7.80.3.1 operator=() [1/2]

```
RTypeServer & gme::RTypeServer::operator= (
    const RTypeServer & ) [delete]
```

7.80.3.2 operator=() [2/2]

```
RTypeServer & gme::RTypeServer::operator= (
    RTypeServer && ) [delete]
```

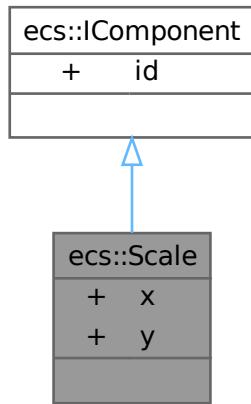
The documentation for this class was generated from the following file:

- /home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Server/include/R-TypeServer/[RTypeServer.hpp](#)

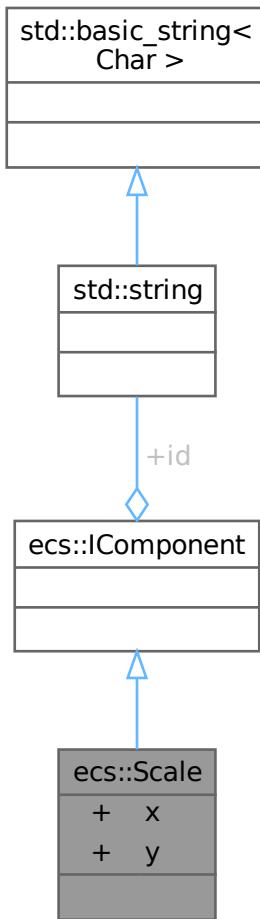
7.81 ecs::Scale Struct Reference

```
#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`

7.81.1 Detailed Description

Definition at line 52 of file [Component.hpp](#).

7.81.2 Member Data Documentation

7.81.2.1 x

```
float ecs::Scale::x {}
```

Definition at line 54 of file [Component.hpp](#).

Referenced by [cli::Game::Game\(\)](#), [cli::Lobby::Lobby\(\)](#), and [cli::Settings::Settings\(\)](#).

7.81.2.2 y

```
float ecs::Scale::y {}
```

Definition at line 54 of file [Component.hpp](#).

The documentation for this struct was generated from the following file:

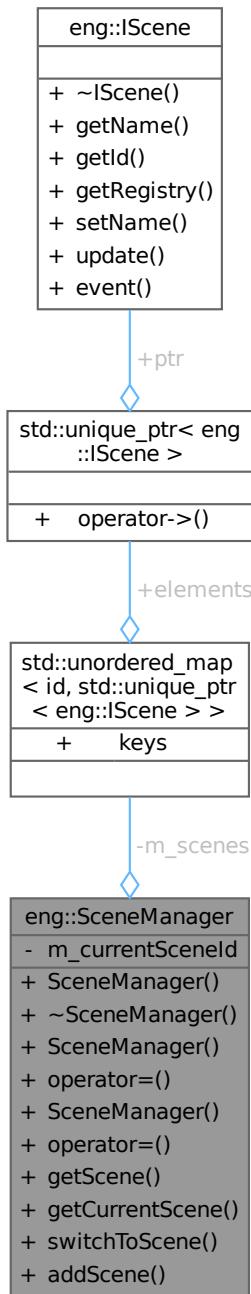
- /home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/[Component.hpp](#)

7.82 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`

7.82.1 Detailed Description

Class for managing scenes.

Definition at line 22 of file [SceneManager.hpp](#).

7.82.2 Constructor & Destructor Documentation

7.82.2.1 SceneManager() [1/3]

`eng::SceneManager::SceneManager () [default]`

7.82.2.2 ~SceneManager()

`eng::SceneManager::~SceneManager () [default]`

7.82.2.3 SceneManager() [2/3]

`eng::SceneManager::SceneManager (`
`const SceneManager &) [delete]`

7.82.2.4 SceneManager() [3/3]

`eng::SceneManager::SceneManager (`
`SceneManager &&) [delete]`

7.82.3 Member Function Documentation

7.82.3.1 addScene()

`void eng::SceneManager::addScene (`
`std::unique_ptr< IScene > scene) [inline]`

Definition at line 37 of file [SceneManager.hpp](#).

References [m_scenes](#).

7.82.3.2 getCurrentScene()

```
std::unique_ptr<IScene> & eng::SceneManager::getCurrentScene () [inline]
```

Definition at line 35 of file [SceneManager.hpp](#).

References [m_currentSceneId](#), and [m_scenes](#).

7.82.3.3 getScene()

```
std::unique_ptr<IScene> & eng::SceneManager::getScene (
    const id sceneId) [inline]
```

Definition at line 34 of file [SceneManager.hpp](#).

References [m_scenes](#).

7.82.3.4 operator=() [1/2]

```
SceneManager & eng::SceneManager::operator= (
    const SceneManager & ) [delete]
```

7.82.3.5 operator=() [2/2]

```
SceneManager & eng::SceneManager::operator= (
    SceneManager &&) [delete]
```

7.82.3.6 switchToScene()

```
void eng::SceneManager::switchToScene (
    const id sceneId) [inline]
```

Definition at line 36 of file [SceneManager.hpp](#).

References [m_currentSceneId](#).

7.82.4 Member Data Documentation

7.82.4.1 m_currentSceneId

```
id eng::SceneManager::m_currentSceneId = 1 [private]
```

Definition at line 57 of file [SceneManager.hpp](#).

Referenced by [getCurrentScene\(\)](#), and [switchToScene\(\)](#).

7.82.4.2 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:

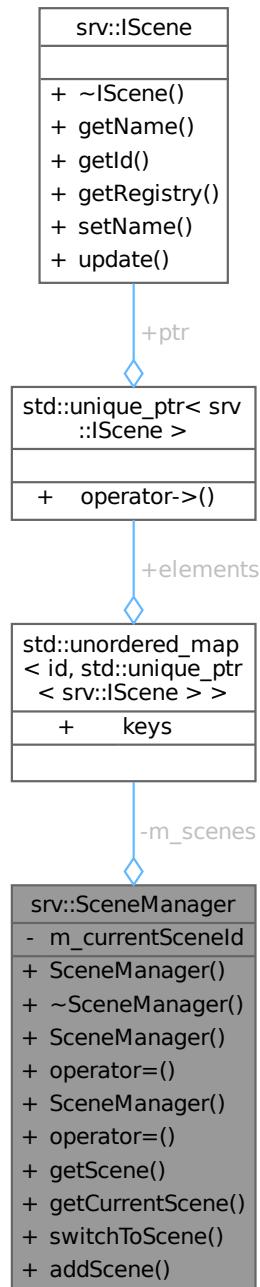
- /home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/[SceneManager.hpp](#)

7.83 srv::SceneManager Class Reference

Class for managing scenes.

```
#include <SceneManager.hpp>
```

Collaboration diagram for srv::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`

7.83.1 Detailed Description

Class for managing scenes.

Definition at line 22 of file [SceneManager.hpp](#).

7.83.2 Constructor & Destructor Documentation

7.83.2.1 SceneManager() [1/3]

`srv::SceneManager::SceneManager () [default]`

7.83.2.2 ~SceneManager()

`srv::SceneManager::~SceneManager () [default]`

7.83.2.3 SceneManager() [2/3]

`srv::SceneManager::SceneManager (`
 `const SceneManager &) [delete]`

7.83.2.4 SceneManager() [3/3]

`srv::SceneManager::SceneManager (`
 `SceneManager &&) [delete]`

7.83.3 Member Function Documentation

7.83.3.1 addScene()

`void srv::SceneManager::addScene (`
 `std::unique_ptr< IScene > scene) [inline]`

Definition at line 37 of file [SceneManager.hpp](#).

References [m_scenes](#).

7.83.3.2 getCurrentScene()

```
std::unique_ptr<IScene> & srv::SceneManager::getCurrentScene () [inline]
```

Definition at line 35 of file [SceneManager.hpp](#).

References [m_currentSceneId](#), and [m_scenes](#).

7.83.3.3 getScene()

```
std::unique_ptr<IScene> & srv::SceneManager::getScene (
    const id sceneId) [inline]
```

Definition at line 34 of file [SceneManager.hpp](#).

References [m_scenes](#).

7.83.3.4 operator=() [1/2]

```
SceneManager & srv::SceneManager::operator= (
    const SceneManager & ) [delete]
```

7.83.3.5 operator=() [2/2]

```
SceneManager & srv::SceneManager::operator= (
    SceneManager && ) [delete]
```

7.83.3.6 switchToScene()

```
void srv::SceneManager::switchToScene (
    const id sceneId) [inline]
```

Definition at line 36 of file [SceneManager.hpp](#).

References [m_currentSceneId](#).

7.83.4 Member Data Documentation

7.83.4.1 m_currentSceneId

```
id srv::SceneManager::m_currentSceneId = 1 [private]
```

Definition at line 57 of file [SceneManager.hpp](#).

Referenced by [getCurrentScene\(\)](#), and [switchToScene\(\)](#).

7.83.4.2 m_scenes

std::unordered_map<[id](#), std::unique_ptr<[IScene](#)> > srv::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:

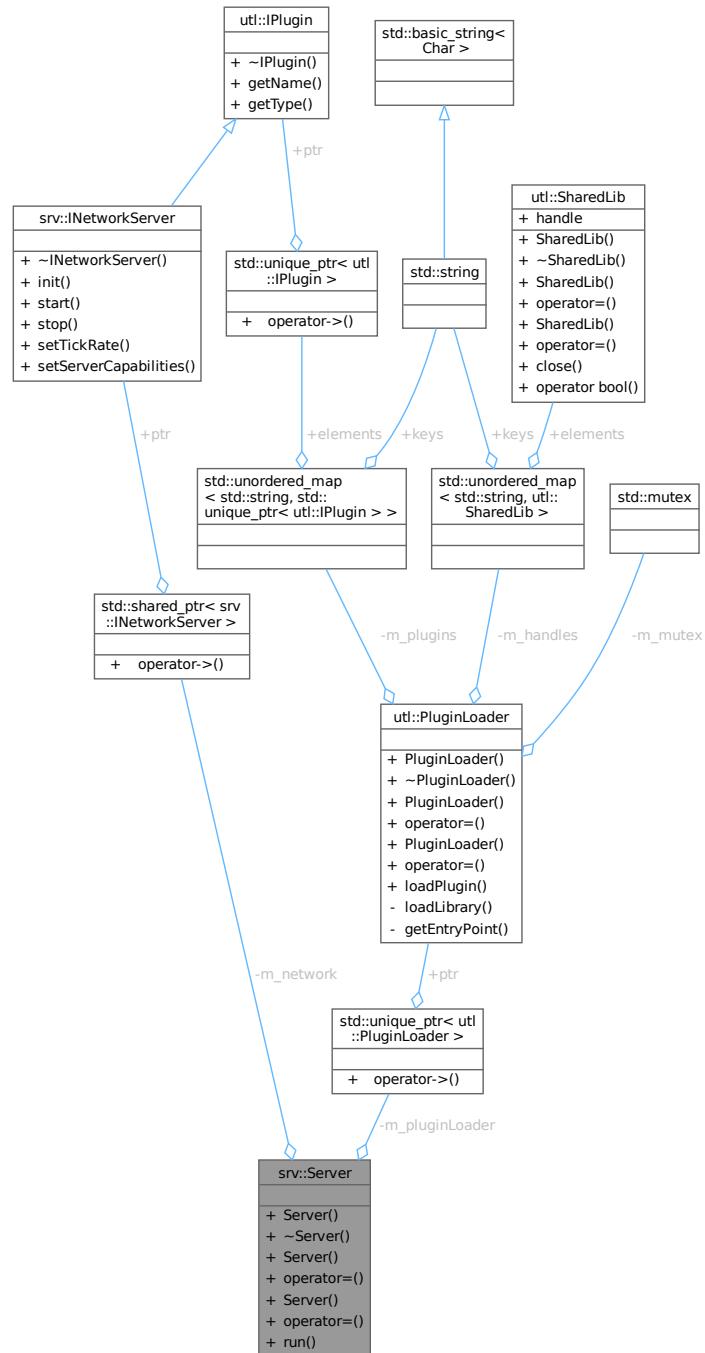
- /home/masina/Projects/Epitech/rtype/server/include/Server/[SceneManager.hpp](#)

7.84 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`
- `void run () const`

Private Attributes

- std::unique_ptr< [utl::PluginLoader](#) > [m_pluginLoader](#)
- std::shared_ptr< [INetworkServer](#) > [m_network](#)

7.84.1 Detailed Description

Class for the server.

Definition at line [23](#) of file [Server.hpp](#).

7.84.2 Constructor & Destructor Documentation

7.84.2.1 Server() [1/3]

```
srv::Server::Server (
    const ArgsConfig & config) [explicit]
```

Definition at line [9](#) of file [server.cpp](#).

References [BUILD_TYPE](#), [GIT_COMMIT_HASH](#), [GIT_TAG](#), [srv::ArgsConfig::host](#), [utl::INFO](#), [utl::Logger::log\(\)](#), [m_network](#), [srv::ArgsConfig::port](#), [PROJECT_NAME](#), and [PROJECT_VERSION](#).

Here is the call graph for this function:



7.84.2.2 ~Server()

```
srv::Server::~Server () [default]
```

7.84.2.3 Server() [2/3]

```
srv::Server::Server (
    const Server & ) [delete]
```

7.84.2.4 Server() [3/3]

```
srv::Server::Server (
    Server && ) [delete]
```

7.84.3 Member Function Documentation

7.84.3.1 operator=() [1/2]

```
Server & srv::Server::operator= (
    const Server & )  [delete]
```

7.84.3.2 operator=() [2/2]

```
Server & srv::Server::operator= (
    Server && )  [delete]
```

7.84.3.3 run()

```
void srv::Server::run () const
```

Definition at line 23 of file [server.cpp](#).

Referenced by [main\(\)](#).

Here is the caller graph for this function:



7.84.4 Member Data Documentation

7.84.4.1 m_network

```
std::shared_ptr<INetworkServer> srv::Server::m_network  [private]
```

Definition at line 39 of file [Server.hpp](#).

Referenced by [Server\(\)](#).

7.84.4.2 m_pluginLoader

```
std::unique_ptr<utl::PluginLoader> srv::Server::m_pluginLoader  [private]
```

Definition at line 38 of file [Server.hpp](#).

The documentation for this class was generated from the following files:

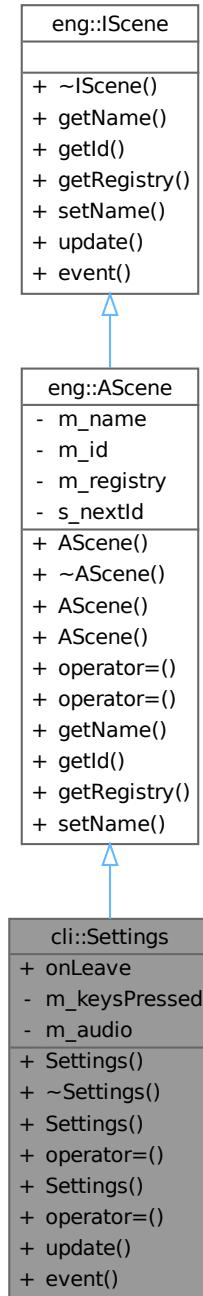
- /home/masina/Projects/Epitech/rtype/server/include/Server/[Server.hpp](#)
- /home/masina/Projects/Epitech/rtype/server/src/[server.cpp](#)

7.85 cli::Settings Class Reference

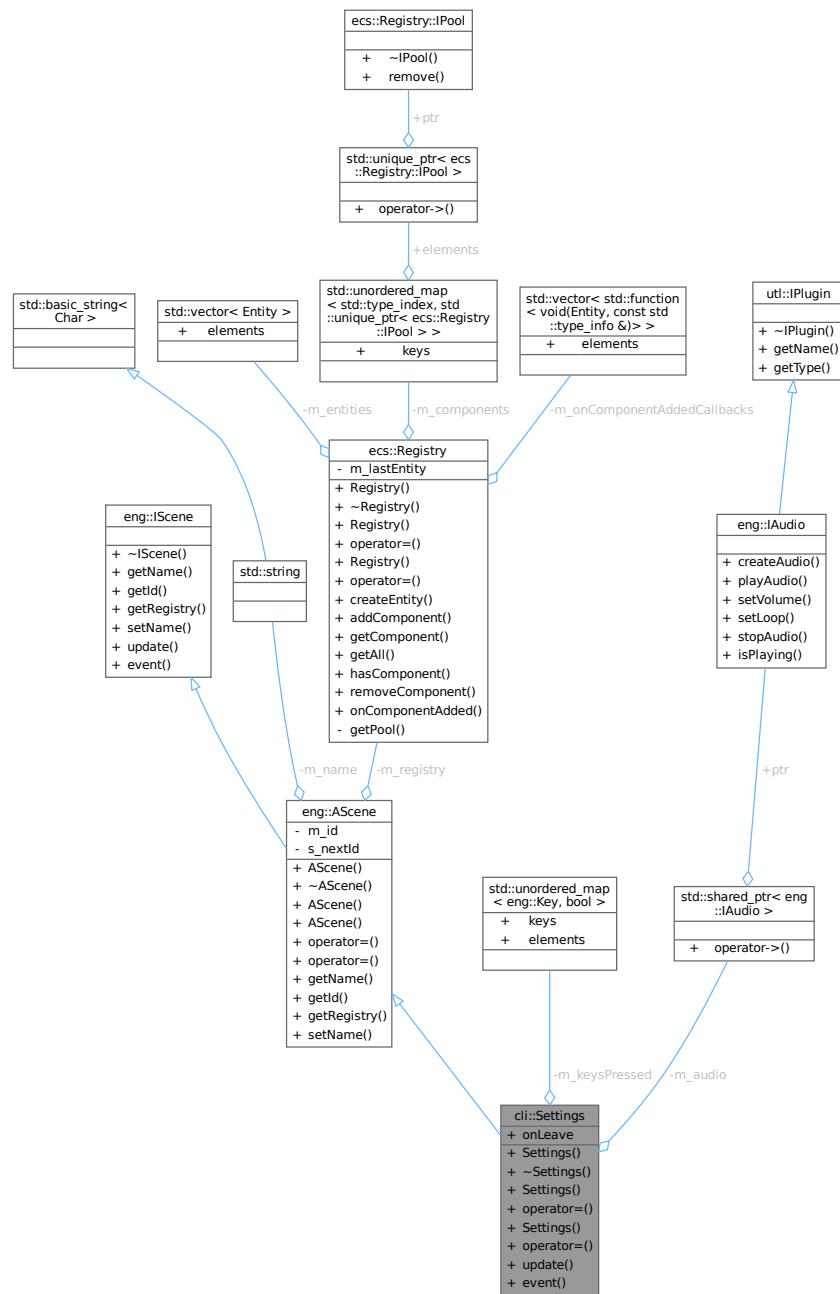
[Settings](#) scene.

```
#include <Settings.hpp>
```

Inheritance diagram for cli::Settings:



Collaboration diagram for cli::Settings:



Public Member Functions

- `Settings` (const std::shared_ptr<eng::IRenderer> &renderer, const std::shared_ptr<eng::IAudio> &audio)
- `~Settings()` override=default
- `Settings` (const `Settings` &other)=delete
- `Settings & operator=` (const `Settings` &other)=delete
- `Settings` (`Settings` &&other)=delete
- `Settings & operator=` (`Settings` &&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

Public Attributes

- [std::function< void\(\)> onLeave](#)

Private Attributes

- [std::unordered_map< eng::Key, bool > m_keysPressed](#)
- [const std::shared_ptr< eng::IAudio > & m_audio](#)

7.85.1 Detailed Description

[Settings](#) scene.

Definition at line 21 of file [Settings.hpp](#).

7.85.2 Constructor & Destructor Documentation

7.85.2.1 Settings() [1/3]

```
cli::Settings::Settings (
    const std::shared_ptr< eng::IRenderer > & renderer,
    const std::shared_ptr< eng::IAudio > & audio)
```

Definition at line 8 of file [settings.cpp](#).

References [eng::Color::a](#), [cli::Path::Audio::AUDIO_TITLE](#), [eng::Color::b](#), [cli::Path::Font::FONTS_RTYPE](#), [eng::Color::g](#), [ecs::IComponent::id](#), [eng::Color::r](#), [WHITE](#), and [ecs::Scale::x](#).

7.85.2.2 ~Settings()

```
cli::Settings::~Settings () [override], [default]
```

7.85.2.3 Settings() [2/3]

```
cli::Settings::Settings (
    const Settings & other) [delete]
```

7.85.2.4 Settings() [3/3]

```
cli::Settings::Settings (
    Settings && other) [delete]
```

7.85.3 Member Function Documentation

7.85.3.1 event()

```
void cli::Settings::event (
    const eng::Event & event) [override], [virtual]
```

Implements [eng::IScene](#).

Definition at line 99 of file [settings.cpp](#).

References [eng::Escape](#), [eng::Event::key](#), [eng::KeyPressed](#), and [eng::Event::type](#).

7.85.3.2 operator=() [1/2]

```
Settings & cli::Settings::operator= (
    const Settings & other) [delete]
```

7.85.3.3 operator=() [2/2]

```
Settings & cli::Settings::operator= (
    Settings && other) [delete]
```

7.85.3.4 update()

```
void cli::Settings::update (
    float dt,
    const eng::WindowSize & size) [override], [virtual]
```

Implements [eng::IScene](#).

Definition at line 81 of file [settings.cpp](#).

References [eng::Playing](#).

7.85.4 Member Data Documentation

7.85.4.1 m_audio

const std::shared_ptr<[eng::IAudio](#)>& cli::Settings::m_audio [private]

Definition at line [39](#) of file [Settings.hpp](#).

7.85.4.2 m_keysPressed

std::unordered_map<[eng::Key](#), bool> cli::Settings::m_keysPressed [private]

Definition at line [38](#) of file [Settings.hpp](#).

7.85.4.3 onLeave

std::function<void()> cli::Settings::onLeave

Definition at line [35](#) of file [Settings.hpp](#).

The documentation for this class was generated from the following files:

- /home/masina/Projects/Epitech/rtype/client/include/Client/Scenes/[Settings.hpp](#)
- /home/masina/Projects/Epitech/rtype/client/src/scenes/[settings.cpp](#)

7.86 utl::SharedLib Struct Reference

Handle to a dynamic library with RAII.

#include <PluginLoader.hpp>

Collaboration diagram for utl::SharedLib:

utl::SharedLib
+ handle
+ SharedLib()
+ ~SharedLib()
+ SharedLib()
+ operator=(())
+ SharedLib()
+ operator=(())
+ close()
+ operator bool()

Public Member Functions

- `SharedLib (const LibHandle h=nullptr)`
- `~SharedLib ()`
- `SharedLib (const SharedLib &)=delete`
- `SharedLib & operator= (const SharedLib &)=delete`
- `SharedLib (SharedLib &&other) noexcept`
- `SharedLib & operator= (SharedLib &&other) noexcept`
- `void close ()`
- `operator bool () const`

Public Attributes

- `LibHandle handle = nullptr`

7.86.1 Detailed Description

Handle to a dynamic library with RAII.

Definition at line 42 of file [PluginLoader.hpp](#).

7.86.2 Constructor & Destructor Documentation

7.86.2.1 SharedLib() [1/3]

```
utl::SharedLib::SharedLib (
    const LibHandle h = nullptr) [inline], [explicit]
```

Definition at line 46 of file [PluginLoader.hpp](#).

7.86.2.2 ~SharedLib()

```
utl::SharedLib::~SharedLib () [inline]
```

Definition at line 47 of file [PluginLoader.hpp](#).

References [close\(\)](#).

Here is the call graph for this function:



7.86.2.3 SharedLib() [2/3]

```
utl::SharedLib::SharedLib (
    const SharedLib & ) [delete]
```

7.86.2.4 SharedLib() [3/3]

```
utl::SharedLib::SharedLib (
    SharedLib && other) [inline], [noexcept]
```

Definition at line [51](#) of file [PluginLoader.hpp](#).

7.86.3 Member Function Documentation

7.86.3.1 close()

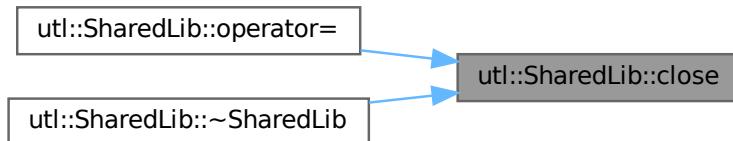
```
void utl::SharedLib::close () [inline]
```

Definition at line [63](#) of file [PluginLoader.hpp](#).

References [handle](#).

Referenced by [operator=\(\)](#), and [~SharedLib\(\)](#).

Here is the caller graph for this function:



7.86.3.2 operator bool()

```
utl::SharedLib::operator bool () const [inline], [explicit]
```

Definition at line [75](#) of file [PluginLoader.hpp](#).

References [handle](#).

7.86.3.3 operator=() [1/2]

```
SharedLib & utl::SharedLib::operator= (
    const SharedLib & ) [delete]
```

7.86.3.4 operator=() [2/2]

```
SharedLib & utl::SharedLib::operator= (
    SharedLib && other) [inline], [noexcept]
```

Definition at line 52 of file [PluginLoader.hpp](#).

References [close\(\)](#), and [handle](#).

Here is the call graph for this function:



7.86.4 Member Data Documentation

7.86.4.1 handle

```
LibHandle utl::SharedLib::handle = nullptr
```

Definition at line 44 of file [PluginLoader.hpp](#).

Referenced by [close\(\)](#), [utl::PluginLoader::getEntryPoint\(\)](#), [operator bool\(\)](#), and [operator=\(\)](#).

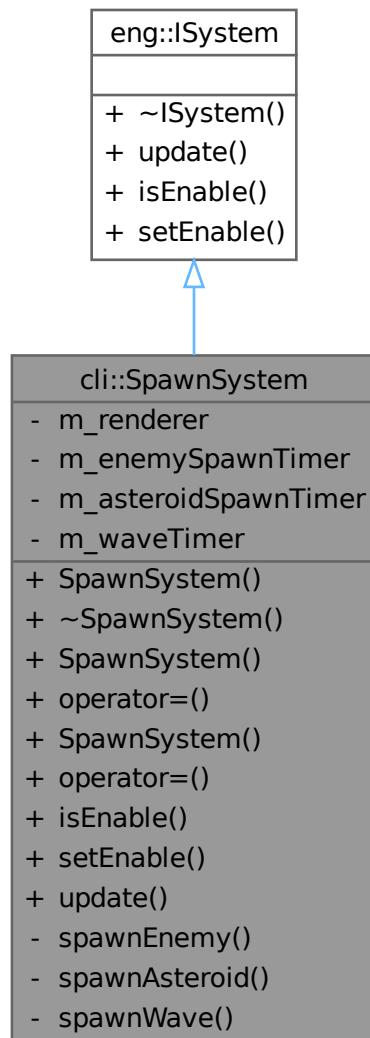
The documentation for this struct was generated from the following file:

- /home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/[PluginLoader.hpp](#)

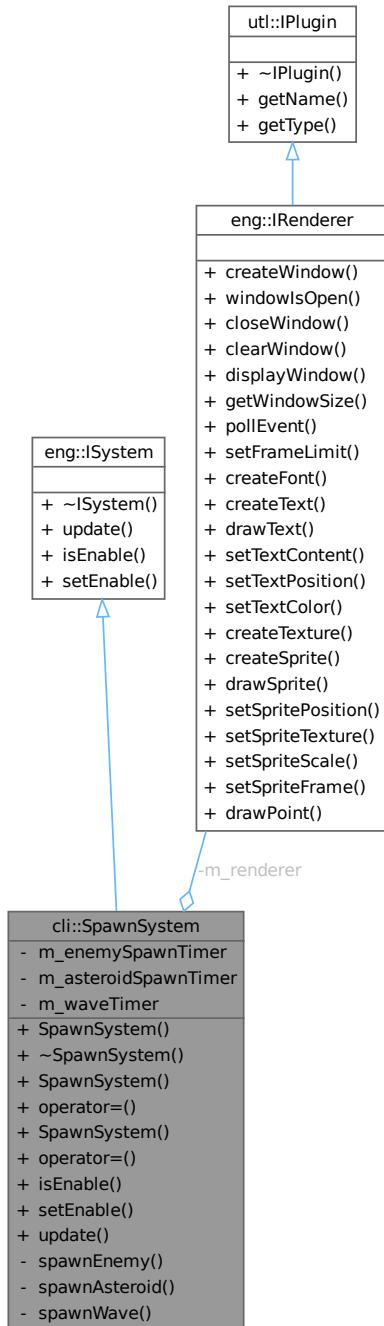
7.87 cli::SpawnSystem Class Reference

```
#include <Spawn.hpp>
```

Inheritance diagram for cli::SpawnSystem:



Collaboration diagram for cli::SpawnSystem:



Public Member Functions

- `SpawnSystem (eng::IRenderer &renderer)`
- `~SpawnSystem () override=default`
- `SpawnSystem (const SpawnSystem &)=delete`
- `SpawnSystem & operator= (const SpawnSystem &)=delete`
- `SpawnSystem (SpawnSystem &&)=delete`

- `SpawnSystem & operator= (SpawnSystem &&) =delete`
- `bool isEnabled () override`
- `void setEnable (bool enable) override`
- `void update (ecs::Registry ®istry, float dt) override`

Public Member Functions inherited from `eng::ISystem`

- `virtual ~ISystem () =default`

Private Member Functions

- `void spawnEnemy (ecs::Registry ®istry)`
- `void spawnAsteroid (ecs::Registry ®istry, ecs::Asteroid::Size size)`
- `void spawnWave (ecs::Registry ®istry)`

Private Attributes

- `eng::IRenderer & m_renderer`
- `float m_enemySpawnTimer = 0.0f`
- `float m_asteroidSpawnTimer = 0.0f`
- `float m_waveTimer = 0.0f`

7.87.1 Detailed Description

Definition at line 17 of file `Spawn.hpp`.

7.87.2 Constructor & Destructor Documentation

7.87.2.1 `SpawnSystem()` [1/3]

```
cli::SpawnSystem::SpawnSystem (
    eng::IRenderer & renderer) [inline], [explicit]
```

Definition at line 20 of file `Spawn.hpp`.

7.87.2.2 `~SpawnSystem()`

```
cli::SpawnSystem::~SpawnSystem () [override], [default]
```

7.87.2.3 `SpawnSystem()` [2/3]

```
cli::SpawnSystem::SpawnSystem (
    const SpawnSystem &) [delete]
```

7.87.2.4 SpawnSystem() [3/3]

```
cli::SpawnSystem::SpawnSystem (
    SpawnSystem && ) [delete]
```

7.87.3 Member Function Documentation

7.87.3.1 isEnabled()

```
bool cli::SpawnSystem::isEnabled () [inline], [override], [virtual]
```

Implements [eng::ISystem](#).

Definition at line 28 of file [Spawn.hpp](#).

7.87.3.2 operator=() [1/2]

```
SpawnSystem & cli::SpawnSystem::operator= (
    const SpawnSystem & ) [delete]
```

7.87.3.3 operator=() [2/2]

```
SpawnSystem & cli::SpawnSystem::operator= (
    SpawnSystem && ) [delete]
```

7.87.3.4 setEnable()

```
void cli::SpawnSystem::setEnable (
    bool enable) [inline], [override], [virtual]
```

Implements [eng::ISystem](#).

Definition at line 29 of file [Spawn.hpp](#).

7.87.3.5 spawnAsteroid()

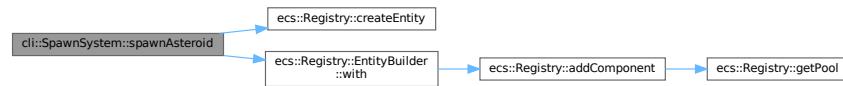
```
void cli::SpawnSystem::spawnAsteroid (
    ecs::Registry & registry,
    ecs::Asteroid::Size size) [inline], [private]
```

Definition at line 88 of file [Spawn.hpp](#).

References [cli::GameConfig::Asteroid::Small::ANIMATION_DURATION](#), [cli::GameConfig::Asteroid::Small::ANIMATION_DURATION](#), [cli::GameConfig::Hitbox::ASTEROID_SMALL_RADIUS](#), [ecs::Registry::createEntity\(\)](#), [cli::GameConfig::Asteroid::Small::HEALTH](#), [cli::GameConfig::Screen::MAX_Y](#), [cli::GameConfig::Screen::MIN_Y](#), [cli::GameConfig::Asteroid::Small::ROTATION_SPEED](#), [cli::GameConfig::Asteroid::Small::SCALE](#), [cli::GameConfig::Screen::SPAWN_X](#), [cli::GameConfig::Asteroid::Small::SPEED](#), [cli::GameConfig::Asteroid::Small::SPRITE_WIDTH](#), [cli::Path::Texture::TEXTURE_ASTEROID](#), and [ecs::Registry::EntityBuilder::with\(\)](#).

Referenced by [update\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.87.3.6 spawnEnemy()

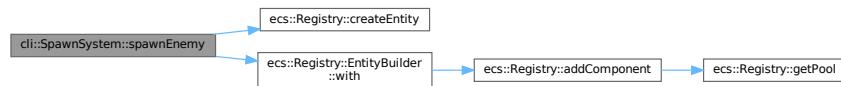
```
void cli::SpawnSystem::spawnEnemy (
    ecs::Registry & registry) [inline], [private]
```

Definition at line 62 of file [Spawn.hpp](#).

References [cli::GameConfig::Enemy::Easy::ANIMATION_DURATION](#), [cli::GameConfig::Enemy::Easy::ANIMATION_DURATION](#), [ecs::Registry::createEntity\(\)](#), [cli::GameConfig::Enemy::Easy::DAMAGE](#), [cli::GameConfig::Hitbox::ENEMY_RADIUS](#), [cli::GameConfig::Enemy::Easy::FRAMES_PER_ROW](#), [cli::GameConfig::Enemy::Easy::HEALTH](#), [cli::GameConfig::Screen::MAX_Y](#), [cli::GameConfig::Screen::MIN_Y](#), [cli::GameConfig::Enemy::Easy::SCALE](#), [cli::GameConfig::Enemy::Easy::SHOOT_COOLDOWN](#), [cli::GameConfig::Screen::SPAWN_X](#), [cli::GameConfig::Enemy::Easy::SPRITE_HEIGHT](#), [cli::GameConfig::Enemy::Easy::SPRITE_WIDTH](#), [cli::Path::Texture::TEXTURE_ENEMY_EASY](#), and [ecs::Registry::EntityBuilder::with\(\)](#).

Referenced by [update\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.87.3.7 spawnWave()

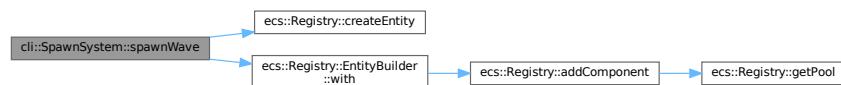
```
void cli::SpawnSystem::spawnWave (
    ecs::Registry & registry) [inline], [private]
```

Definition at line 115 of file [Spawn.hpp](#).

References [cli::GameConfig::Enemy::Easy::ANIMATION_DURATION](#), [cli::GameConfig::Enemy::Easy::ANIMATION_F](#)
[ecs::Registry::createEntity\(\)](#), [cli::GameConfig::Enemy::Easy::DAMAGE](#), [cli::GameConfig::Hitbox::ENEMY_RADIUS](#),
[cli::GameConfig::Enemy::Easy::FRAMES_PER_ROW](#), [cli::GameConfig::Enemy::Easy::HEALTH](#),
[cli::GameConfig::Screen::MAX_Y](#), [cli::GameConfig::Screen::MIN_Y](#), [cli::GameConfig::Enemy::Easy::SCALE](#),
[cli::GameConfig::Enemy::Easy::SHOOT_COOLDOWN](#), [cli::GameConfig::Screen::SPAWN_X](#), [cli::GameConfig::Enemy::E](#)
[cli::GameConfig::Enemy::Easy::SPRITE_HEIGHT](#), [cli::GameConfig::Enemy::Easy::SPRITE_WIDTH](#),
[cli::Path::Texture::TEXTURE_ENEMY_EASY](#), and [ecs::Registry::EntityBuilder::with\(\)](#).

Referenced by [update\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.87.3.8 update()

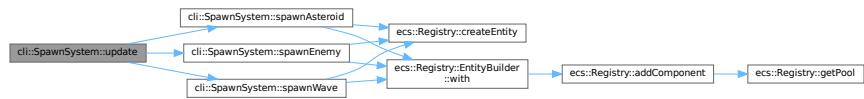
```
void cli::SpawnSystem::update (
    ecs::Registry & registry,
    float dt) [inline], [override], [virtual]
```

Implements [eng::ISystem](#).

Definition at line 31 of file [Spawn.hpp](#).

References [m_asteroidSpawnTimer](#), [m_enemySpawnTimer](#), [m_waveTimer](#), [ecs::Asteroid::SMALL](#), [cli::GameConfig::Asteroid::Small::SPAWN_RATE](#), [cli::GameConfig::Enemy::Easy::SPAWN_RATE](#), [spawnAsteroid\(\)](#), [spawnEnemy\(\)](#), and [spawnWave\(\)](#).

Here is the call graph for this function:



7.87.4 Member Data Documentation

7.87.4.1 m_asteroidSpawnTimer

```
float cli::SpawnSystem::m_asteroidSpawnTimer = 0.0f [private]
```

Definition at line 59 of file [Spawn.hpp](#).

Referenced by [update\(\)](#).

7.87.4.2 m_enemySpawnTimer

```
float cli::SpawnSystem::m_enemySpawnTimer = 0.0f [private]
```

Definition at line 58 of file [Spawn.hpp](#).

Referenced by [update\(\)](#).

7.87.4.3 m_renderer

```
eng::IRenderer& cli::SpawnSystem::m_renderer [private]
```

Definition at line 57 of file [Spawn.hpp](#).

7.87.4.4 m_waveTimer

float cli::SpawnSystem::m_waveTimer = 0.0f [private]

Definition at line 60 of file [Spawn.hpp](#).

Referenced by [update\(\)](#).

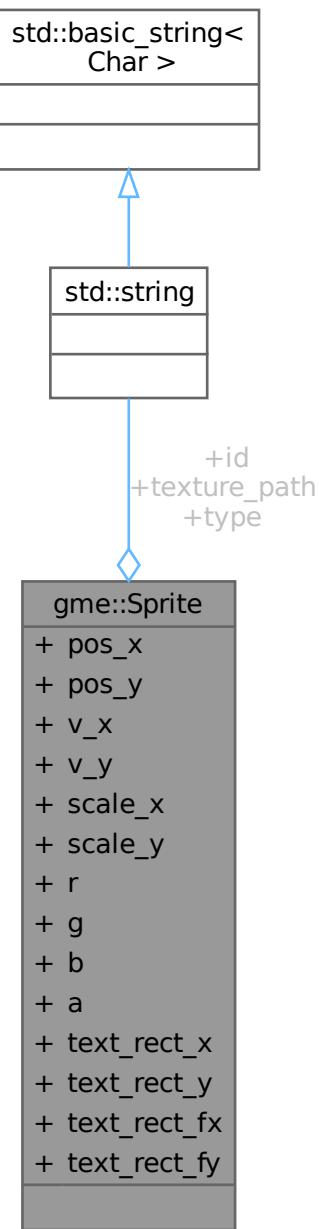
The documentation for this class was generated from the following file:

- /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/[Spawn.hpp](#)

7.88 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`

7.88.1 Detailed Description

Definition at line 15 of file [IGameClient.hpp](#).

7.88.2 Member Data Documentation

7.88.2.1 a

unsigned char gme::Sprite::a = 255u

Definition at line 21 of file [IGameClient.hpp](#).

7.88.2.2 b

unsigned char gme::Sprite::b = 255u

Definition at line 21 of file [IGameClient.hpp](#).

7.88.2.3 g

unsigned char gme::Sprite::g = 255u

Definition at line 21 of file [IGameClient.hpp](#).

7.88.2.4 id

std::string gme::Sprite::id

Definition at line 25 of file [IGameClient.hpp](#).

7.88.2.5 pos_x

float gme::Sprite::pos_x = 0.F

Definition at line 18 of file [IGameClient.hpp](#).

7.88.2.6 pos_y

```
float gme::Sprite::pos_y = 0.F
```

Definition at line 18 of file [IGameClient.hpp](#).

7.88.2.7 r

```
unsigned char gme::Sprite::r = 255u
```

Definition at line 21 of file [IGameClient.hpp](#).

7.88.2.8 scale_x

```
float gme::Sprite::scale_x = 1.F
```

Definition at line 20 of file [IGameClient.hpp](#).

7.88.2.9 scale_y

```
float gme::Sprite::scale_y = 1.F
```

Definition at line 20 of file [IGameClient.hpp](#).

7.88.2.10 text_rect_fx

```
int gme::Sprite::text_rect_fx = 0
```

Definition at line 24 of file [IGameClient.hpp](#).

7.88.2.11 text_rect_fy

```
int gme::Sprite::text_rect_fy = 0
```

Definition at line 24 of file [IGameClient.hpp](#).

7.88.2.12 text_rect_x

```
float gme::Sprite::text_rect_x = 0.F
```

Definition at line 23 of file [IGameClient.hpp](#).

7.88.2.13 text_rect_y

```
float gme::Sprite::text_rect_y = 0.F
```

Definition at line 23 of file [IGameClient.hpp](#).

7.88.2.14 texture_path

```
std::string gme::Sprite::texture_path = ""
```

Definition at line 22 of file [IGameClient.hpp](#).

7.88.2.15 type

```
std::string gme::Sprite::type
```

Definition at line 17 of file [IGameClient.hpp](#).

7.88.2.16 v_x

```
float gme::Sprite::v_x = 0.F
```

Definition at line 19 of file [IGameClient.hpp](#).

7.88.2.17 v_y

```
float gme::Sprite::v_y = 0.F
```

Definition at line 19 of file [IGameClient.hpp](#).

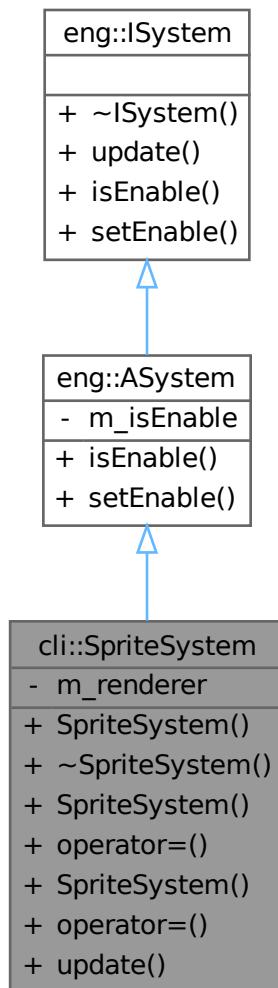
The documentation for this struct was generated from the following file:

- /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/[IGameClient.hpp](#)

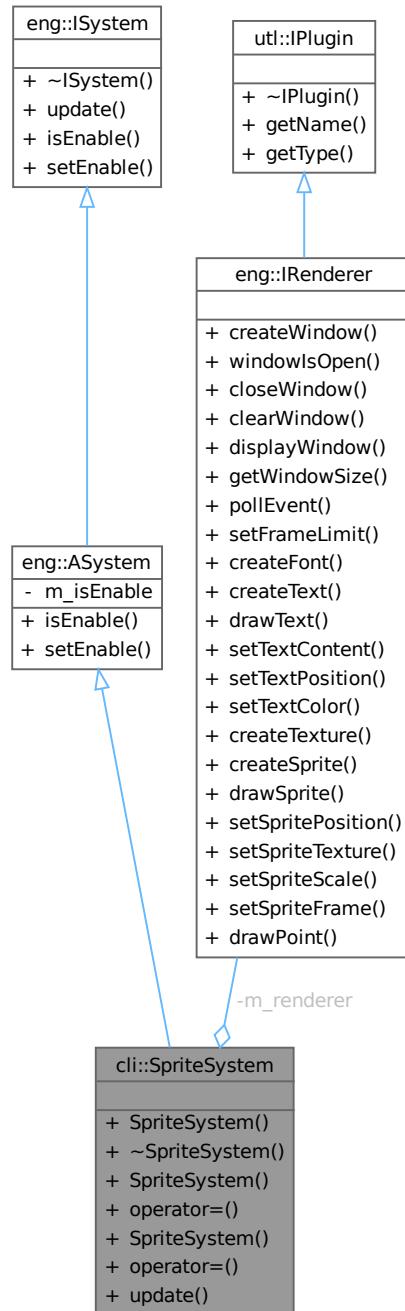
7.89 cli::SpriteSystem Class Reference

```
#include <Sprite.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) override`

Public Member Functions inherited from [eng::ASystem](#)

- bool [isEnable \(\)](#) override
- void [setEnable \(const bool enable\)](#) override

Public Member Functions inherited from [eng::ISystem](#)

- virtual [~ISystem \(\)=default](#)

Private Attributes

- [eng::IRenderer & m_renderer](#)

7.89.1 Detailed Description

Definition at line 16 of file [Sprite.hpp](#).

7.89.2 Constructor & Destructor Documentation

7.89.2.1 [SpriteSystem\(\)](#) [1/3]

```
cli::SpriteSystem::SpriteSystem (
    eng::IRenderer & renderer) [inline], [explicit]
```

Definition at line 19 of file [Sprite.hpp](#).

7.89.2.2 [~SpriteSystem\(\)](#)

```
cli::SpriteSystem::~SpriteSystem () [override], [default]
```

7.89.2.3 [SpriteSystem\(\)](#) [2/3]

```
cli::SpriteSystem::SpriteSystem (
    const SpriteSystem &) [delete]
```

7.89.2.4 [SpriteSystem\(\)](#) [3/3]

```
cli::SpriteSystem::SpriteSystem (
    SpriteSystem &&) [delete]
```

7.89.3 Member Function Documentation

7.89.3.1 [operator=\(\)](#) [1/2]

```
SpriteSystem & cli::SpriteSystem::operator= (
    const SpriteSystem &) [delete]
```

7.89.3.2 operator=() [2/2]

```
SpriteSystem & cli::SpriteSystem::operator=
    SpriteSystem && ) [delete]
```

7.89.3.3 update()

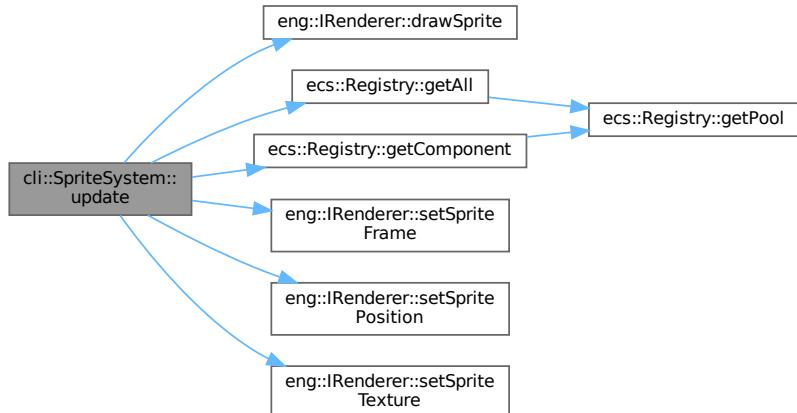
```
void cli::SpriteSystem::update (
    ecs::Registry & registry,
    float ) [inline], [override], [virtual]
```

Implements [eng::ISystem](#).

Definition at line 27 of file [Sprite.hpp](#).

References [eng::IRenderer::drawSprite\(\)](#), [ecs::Registry::getAll\(\)](#), [ecs::Registry::getComponent\(\)](#), [m_renderer](#), [eng::IRenderer::setSpriteFrame\(\)](#), [eng::IRenderer::setSpritePosition\(\)](#), and [eng::IRenderer::setSpriteTexture\(\)](#)

Here is the call graph for this function:



7.89.4 Member Data Documentation

7.89.4.1 m_renderer

```
eng::IRenderer& cli::SpriteSystem::m_renderer [private]
```

Definition at line 50 of file [Sprite.hpp](#).

Referenced by [update\(\)](#).

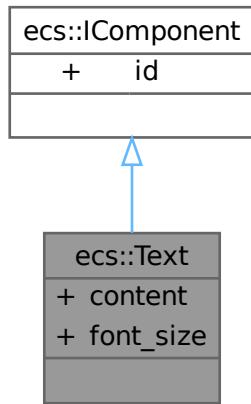
The documentation for this class was generated from the following file:

- /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/[Sprite.hpp](#)

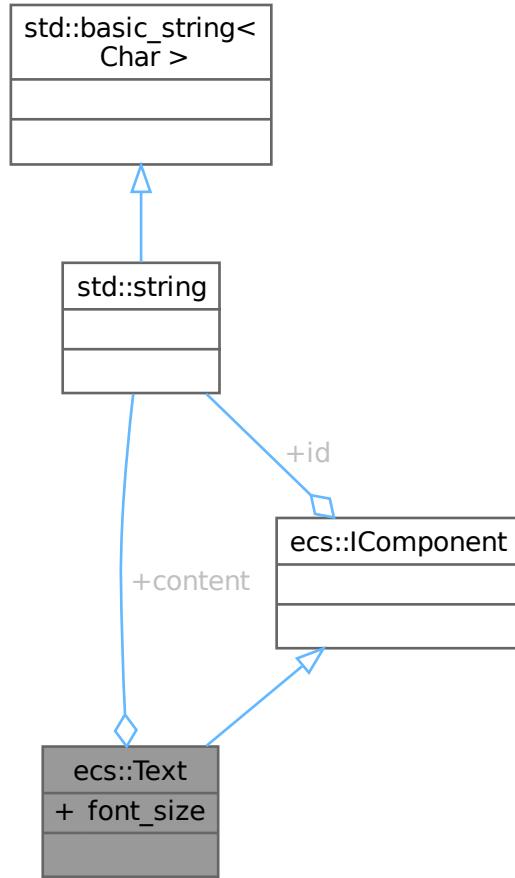
7.90 ecs::Text Struct Reference

```
#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`

7.90.1 Detailed Description

Definition at line 56 of file [Component.hpp](#).

7.90.2 Member Data Documentation

7.90.2.1 content

`std::string ecs::Text::content`

Definition at line 58 of file [Component.hpp](#).

7.90.2.2 font_size

`unsigned int ecs::Text::font_size`

Definition at line 59 of file [Component.hpp](#).

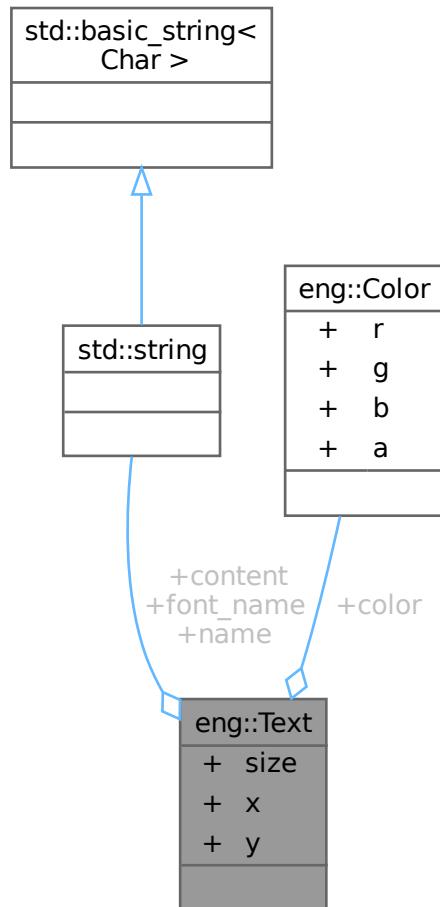
The documentation for this struct was generated from the following file:

- /home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/[Component.hpp](#)

7.91 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`

7.91.1 Detailed Description

Definition at line 22 of file [IRenderer.hpp](#).

7.91.2 Member Data Documentation

7.91.2.1 color

Color eng::Text::color

Definition at line 25 of file [IRenderer.hpp](#).

7.91.2.2 content

std::string eng::Text::content

Definition at line 26 of file [IRenderer.hpp](#).

7.91.2.3 font_name

std::string eng::Text::font_name

Definition at line 24 of file [IRenderer.hpp](#).

7.91.2.4 name

std::string eng::Text::name

Definition at line 30 of file [IRenderer.hpp](#).

7.91.2.5 size

unsigned int eng::Text::size

Definition at line 27 of file [IRenderer.hpp](#).

7.91.2.6 x

```
float eng::Text::x
```

Definition at line 28 of file [IRenderer.hpp](#).

7.91.2.7 y

```
float eng::Text::y
```

Definition at line 29 of file [IRenderer.hpp](#).

The documentation for this struct was generated from the following file:

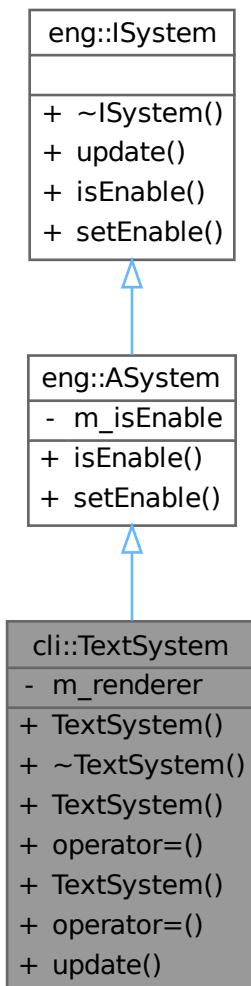
- /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/[IRenderer.hpp](#)

7.92 cli::TextSystem Class Reference

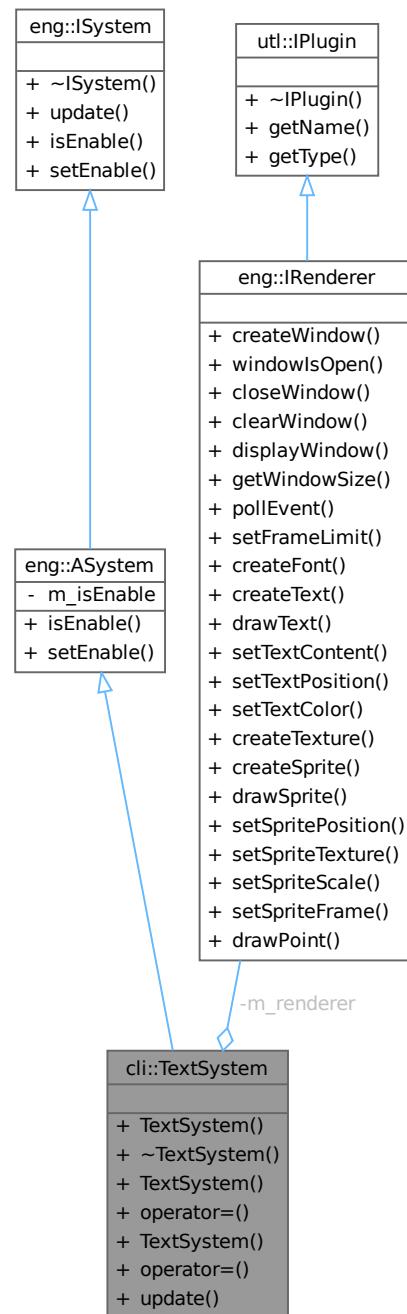
Class for managing entities and their components.

```
#include <Text.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) override`

Public Member Functions inherited from [eng::ASystem](#)

- bool [isEnable \(\)](#) override
- void [setEnable \(const bool enable\)](#) override

Public Member Functions inherited from [eng::ISystem](#)

- virtual ~[ISystem \(\)](#)=default

Private Attributes

- [eng::IRenderer & m_renderer](#)

7.92.1 Detailed Description

Class for managing entities and their components.

Definition at line [21](#) of file [Text.hpp](#).

7.92.2 Constructor & Destructor Documentation

7.92.2.1 [TextSystem\(\)](#) [1/3]

```
cli::TextSystem::TextSystem (
    eng::IRenderer & renderer) [inline], [explicit]
```

Definition at line [24](#) of file [Text.hpp](#).

7.92.2.2 [~TextSystem\(\)](#)

```
cli::TextSystem::~TextSystem () [override], [default]
```

7.92.2.3 [TextSystem\(\)](#) [2/3]

```
cli::TextSystem::TextSystem (
    const TextSystem &) [delete]
```

7.92.2.4 [TextSystem\(\)](#) [3/3]

```
cli::TextSystem::TextSystem (
    TextSystem &&) [delete]
```

7.92.3 Member Function Documentation

7.92.3.1 operator=() [1/2]

```
TextSystem & cli::TextSystem::operator= (
    const TextSystem & ) [delete]
```

7.92.3.2 operator=() [2/2]

```
TextSystem & cli::TextSystem::operator= (
    TextSystem && ) [delete]
```

7.92.3.3 update()

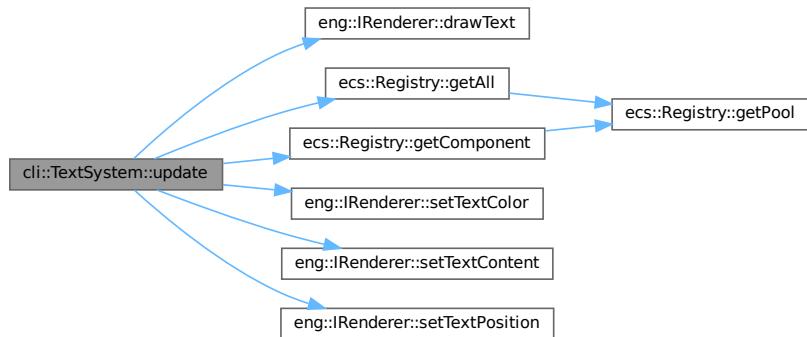
```
void cli::TextSystem::update (
    ecs::Registry & registry,
    float ) [inline], [override], [virtual]
```

Implements [eng::ISystem](#).

Definition at line 32 of file [Text.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:



7.92.4 Member Data Documentation

7.92.4.1 m_renderer

```
eng::IRenderer& cli::TextSystem::m_renderer [private]
```

Definition at line 56 of file [Text.hpp](#).

Referenced by [update\(\)](#).

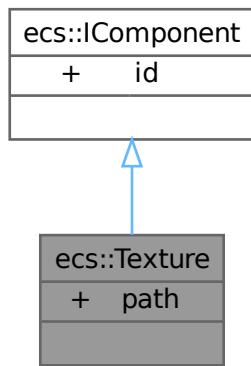
The documentation for this class was generated from the following file:

- /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/[Text.hpp](#)

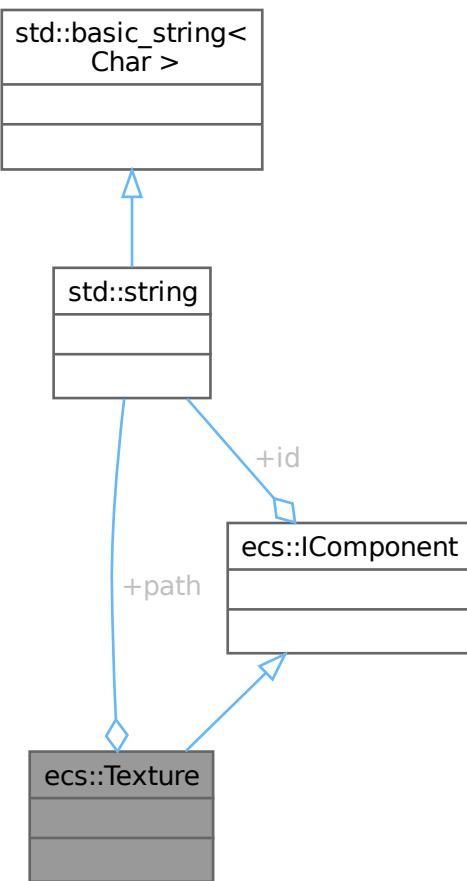
7.93 ecs::Texture Struct Reference

```
#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`

7.93.1 Detailed Description

Definition at line 61 of file [Component.hpp](#).

7.93.2 Member Data Documentation

7.93.2.1 path

```
std::string ecs::Texture::path
```

Definition at line 63 of file [Component.hpp](#).

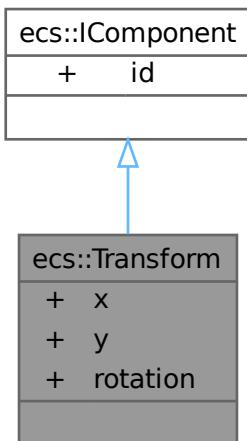
The documentation for this struct was generated from the following file:

- /home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/[Component.hpp](#)

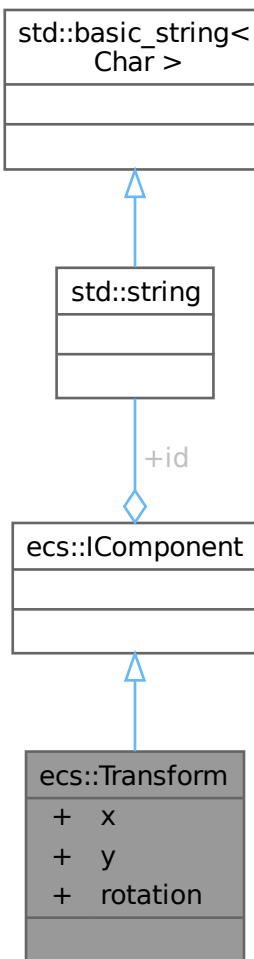
7.94 ecs::Transform Struct Reference

```
#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`

7.94.1 Detailed Description

Definition at line 67 of file [Component.hpp](#).

7.94.2 Member Data Documentation

7.94.2.1 rotation

```
float ecs::Transform::rotation {}
```

Definition at line 70 of file [Component.hpp](#).

7.94.2.2 x

```
float ecs::Transform::x {}
```

Definition at line 69 of file [Component.hpp](#).

Referenced by [cli::CollisionSystem::checkCircularCollision\(\)](#), [cli::WeaponSystem::showLoadingAnimation\(\)](#), [cli::AsteroidSystem::update\(\)](#), [cli::BeamSystem::update\(\)](#), [cli::EnemySystem::update\(\)](#), [cli::Game::update\(\)](#), and [cli::ProjectileSystem::update\(\)](#).

7.94.2.3 y

```
float ecs::Transform::y {}
```

Definition at line 69 of file [Component.hpp](#).

Referenced by [cli::CollisionSystem::checkCircularCollision\(\)](#), [cli::WeaponSystem::showLoadingAnimation\(\)](#), [cli::AsteroidSystem::update\(\)](#), [cli::EnemySystem::update\(\)](#), [cli::Game::update\(\)](#), and [cli::ProjectileSystem::update\(\)](#).

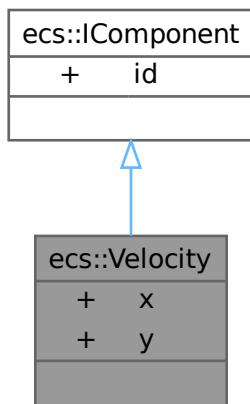
The documentation for this struct was generated from the following file:

- /home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/[Component.hpp](#)

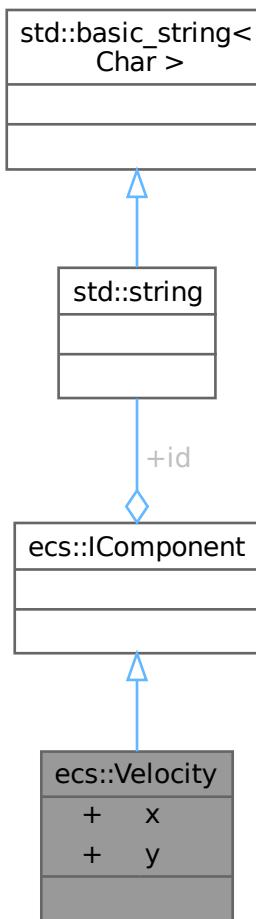
7.95 ecs::Velocity Struct Reference

```
#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`

7.95.1 Detailed Description

Definition at line 72 of file [Component.hpp](#).

7.95.2 Member Data Documentation

7.95.2.1 x

```
float ecs::Velocity::x {}
```

Definition at line 74 of file [Component.hpp](#).

Referenced by [cli::AsteroidSystem::update\(\)](#), and [cli::ProjectileSystem::update\(\)](#).

7.95.2.2 y

```
float ecs::Velocity::y {}
```

Definition at line 74 of file [Component.hpp](#).

The documentation for this struct was generated from the following file:

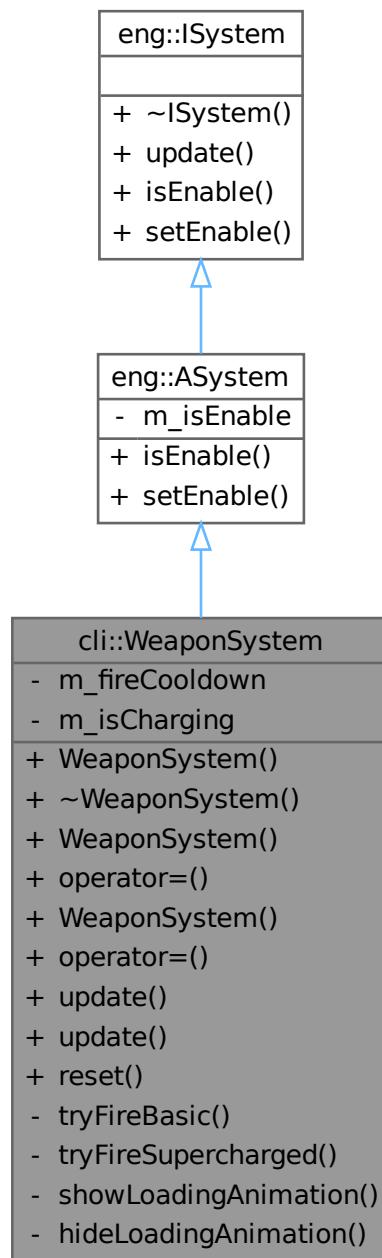
- /home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/[Component.hpp](#)

7.96 cli::WeaponSystem Class Reference

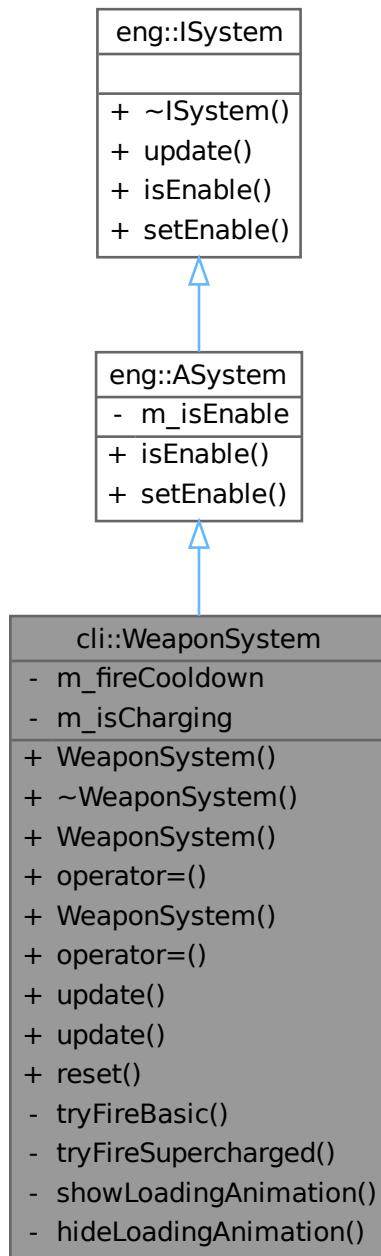
Manages weapon firing and charging.

```
#include <Weapon.hpp>
```

Inheritance diagram for cli::WeaponSystem:



Collaboration diagram for cli::WeaponSystem:



Public Member Functions

- [WeaponSystem \(\)](#)=default
- [~WeaponSystem \(\)](#) override=default
- [WeaponSystem \(const WeaponSystem &\)](#)=delete
- [WeaponSystem & operator= \(const WeaponSystem &\)](#)=delete
- [WeaponSystem \(WeaponSystem &&\)](#)=delete

- `WeaponSystem & operator= (WeaponSystem &&) = delete`
- `void update (ecs::Registry ®istry, float dt, bool spacePressed)`
 Update weapon system.
- `void update (ecs::Registry ®istry, float dt) override`
- `void reset ()`
 Reset weapon state.

Public Member Functions inherited from [eng::ASystem](#)

- `bool isEnabled () override`
- `void setEnable (const bool enable) override`

Public Member Functions inherited from [eng::ISystem](#)

- `virtual ~ISystem () = default`

Private Member Functions

- `bool tryFireBasic (ecs::Registry ®istry, float x, float y)`
 Try to fire basic projectile.
- `bool tryFireSupercharged (ecs::Registry ®istry, float x, float y)`
 Try to fire supercharged projectile.
- `void showLoadingAnimation (ecs::Registry ®istry, ecs::Entity playerEntity, const ecs::Transform *playerTransform)`
 Show loading animation in front of the player.
- `void hideLoadingAnimation (ecs::Registry ®istry, ecs::Entity playerEntity)`
 Hide loading animation.

Private Attributes

- `float m_fireCooldown = 0.0f`
- `bool m_isCharging = false`

7.96.1 Detailed Description

Manages weapon firing and charging.

Definition at line 21 of file [Weapon.hpp](#).

7.96.2 Constructor & Destructor Documentation

7.96.2.1 WeaponSystem() [1/3]

`cli::WeaponSystem::WeaponSystem () [default]`

7.96.2.2 ~WeaponSystem()

```
cli::WeaponSystem::~WeaponSystem () [override], [default]
```

7.96.2.3 WeaponSystem() [2/3]

```
cli::WeaponSystem::WeaponSystem (
    const WeaponSystem & ) [delete]
```

7.96.2.4 WeaponSystem() [3/3]

```
cli::WeaponSystem::WeaponSystem (
    WeaponSystem && ) [delete]
```

7.96.3 Member Function Documentation

7.96.3.1 hideLoadingAnimation()

```
void cli::WeaponSystem::hideLoadingAnimation (
    ecs::Registry & registry,
    ecs::Entity playerEntity) [private]
```

Hide loading animation.

Parameters

registry	The ECS registry
playerEntity	The player entity

Definition at line 154 of file [weapon.cpp](#).

References [ecs::Registry::getAll\(\)](#), [ecs::Registry::hasComponent\(\)](#), and [ecs::Registry::removeComponent\(\)](#).

Referenced by [update\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.96.3.2 operator=() [1/2]

```
WeaponSystem & cli::WeaponSystem::operator= (
    const WeaponSystem & ) [delete]
```

7.96.3.3 operator=() [2/2]

```
WeaponSystem & cli::WeaponSystem::operator= (
    WeaponSystem && ) [delete]
```

7.96.3.4 reset()

```
void cli::WeaponSystem::reset ()
```

Reset weapon state.

Definition at line 95 of file [weapon.cpp](#).

References [m_fireCooldown](#), and [m_isCharging](#).

7.96.3.5 showLoadingAnimation()

```
void cli::WeaponSystem::showLoadingAnimation (
    ecs::Registry & registry,
    ecs::Entity playerEntity,
    const ecs::Transform * playerTransform) [private]
```

Show loading animation in front of the player.

Parameters

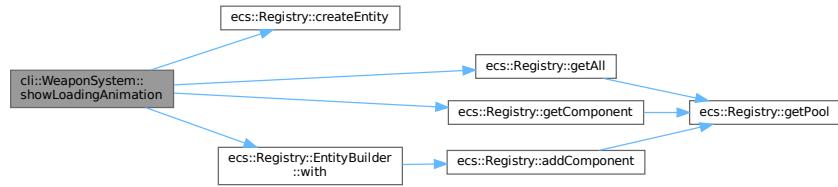
registry	The ECS registry
playerEntity	The player entity
playerTransform	The player transform

Definition at line 121 of file [weapon.cpp](#).

References `ecs::Registry::createEntity()`, `ecs::Registry::getAll()`, `ecs::Registry::getComponent()`, `cli::Path::Texture::TEXTURE`, `ecs::Registry::EntityBuilder::with()`, `ecs::Transform::x`, and `ecs::Transform::y`.

Referenced by [update\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.96.3.6 tryFireBasic()

```
bool cli::WeaponSystem::tryFireBasic (
    ecs::Registry & registry,
    float x,
    float y) [private]
```

Try to fire basic projectile.

Parameters

registry	The ECS registry
x	X position
y	Y position

Returns

True if fired successfully

Definition at line 101 of file [weapon.cpp](#).

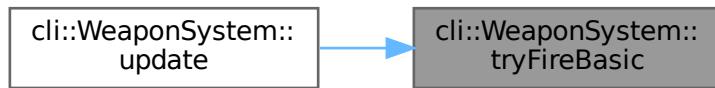
References [cli::ProjectileManager::createBasicProjectile\(\)](#), and [m_fireCooldown](#).

Referenced by [update\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.96.3.7 tryFireSupercharged()

```
bool cli::WeaponSystem::tryFireSupercharged (
    ecs::Registry & registry,
    float x,
    float y) [private]
```

Try to fire supercharged projectile.

Parameters

registry	The ECS registry
x	X position
y	Y position

Returns

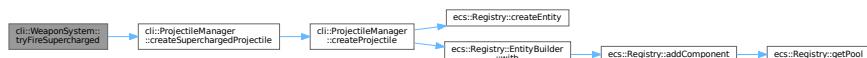
True if fired successfully

Definition at line 113 of file [weapon.cpp](#).

References [cli::ProjectileManager::createSuperchargedProjectile\(\)](#).

Referenced by [update\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.96.3.8 update() [1/2]

```
void cli::WeaponSystem::update (
    ecs::Registry & registry,
    float dt) [override], [virtual]
```

Implements [eng::ISystem](#).

7.96.3.9 update() [2/2]

```
void cli::WeaponSystem::update (
    ecs::Registry & registry,
    float dt,
    bool spacePressed)
```

Update weapon system.

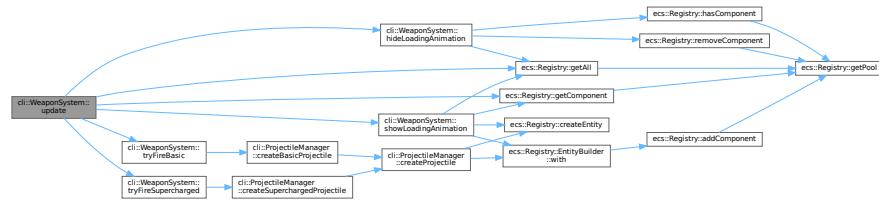
Parameters

registry	The ECS registry
dt	Delta time
spacePressed	Whether space is pressed

Definition at line 13 of file [weapon.cpp](#).

References `ecs::Registry::getAll()`, `ecs::Registry::getComponent()`, `hideLoadingAnimation()`, `m_fireCooldown`, `m_isCharging`, `showLoadingAnimation()`, `cli::GameConfig::Player::SPRITE_HEIGHT`, `cli::GameConfig::Player::SPRITE_WIDTH`, `tryFireBasic()`, and `tryFireSupercharged()`.

Here is the call graph for this function:



7.96.4 Member Data Documentation

7.96.4.1 m_fireCooldown

```
float cli::WeaponSystem::m_fireCooldown = 0.0f [private]
```

Definition at line 47 of file [Weapon.hpp](#).

Referenced by [reset\(\)](#), [tryFireBasic\(\)](#), and [update\(\)](#).

7.96.4.2 m_isCharging

```
bool cli::WeaponSystem::m_isCharging = false [private]
```

Definition at line 48 of file [Weapon.hpp](#).

Referenced by [reset\(\)](#), and [update\(\)](#).

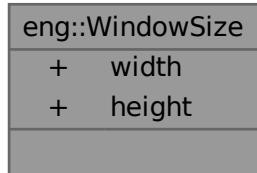
The documentation for this class was generated from the following files:

- /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/[Weapon.hpp](#)
- /home/masina/Projects/Epitech/rtype/client/src/systems/[weapon.cpp](#)

7.97 eng::WindowSize Struct Reference

```
#include <IRenderer.hpp>
```

Collaboration diagram for eng::WindowSize:



Public Attributes

- unsigned int [width](#)
- unsigned int [height](#)

7.97.1 Detailed Description

Definition at line [94](#) of file [IRenderer.hpp](#).

7.97.2 Member Data Documentation

7.97.2.1 height

unsigned int eng::WindowSize::height

Definition at line [97](#) of file [IRenderer.hpp](#).

Referenced by [cli::Game::update\(\)](#).

7.97.2.2 width

unsigned int eng::WindowSize::width

Definition at line [96](#) of file [IRenderer.hpp](#).

Referenced by [cli::Game::update\(\)](#).

The documentation for this struct was generated from the following file:

- /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/[IRenderer.hpp](#)

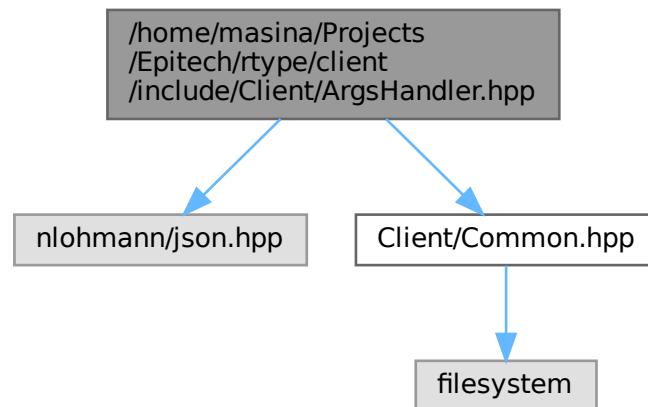
Chapter 8

File Documentation

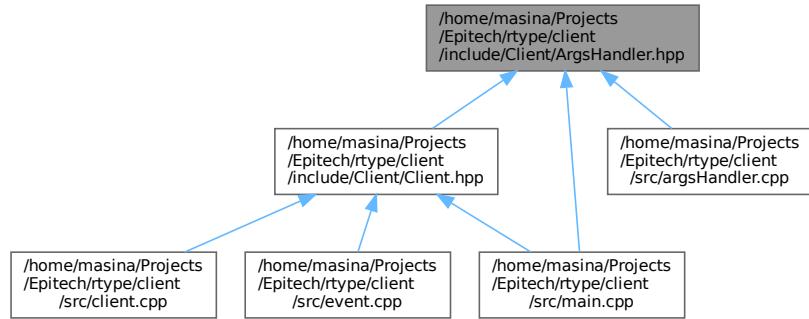
8.1 /home/masina/Projects/Epitech/rtype/client/include/Client/ArgsHandler.hpp File Reference

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 /**
00003 /**
00004 /**
00005 /**
00006
00007 #pragma once
00008
00009 #include <nlohmann/json.hpp>
00010
00011 #include "Client/Common.hpp"
00012
00013 namespace cli
  
```

```

00014 {
00015
00016     using json = nlohmann::json;
00017
00018     struct ArgsConfig
00019     {
00020         bool exit = false;
00021         unsigned int width = Config::Window::DEFAULT_WINDOW_WIDTH;
00022         unsigned int height = Config::Window::DEFAULT_WINDOW_HEIGHT;
00023         unsigned int frameLimit = Config::Window::DEFAULT_WINDOW_FRAME_LIMIT;
00024         bool fullscreen = Config::Window::DEFAULT_WINDOW_FULLSCREEN;
00025         std::string host = Config::Network::DEFAULT_NETWORK_HOST;
00026         unsigned int port = Config::Network::DEFAULT_NETWORK_PORT;
00027         std::string audio_lib_path = Path::Plugin::PLUGIN_AUDIO_SFML.string();
00028         std::string network_lib_path = Path::Plugin::PLUGIN_NETWORK_ASIO_CLIENT.string();
00029         std::string renderer_lib_path = Path::Plugin::PLUGIN_RENDERER_SFML.string();
00030
00031         static ArgsConfig fromFile(const std::string &path);
00032     }; // struct Config
00033     struct EnvConfig
00034     {
00035     };
00036
00037     /**
00038     * @class ArgsHandler
00039     * @brief Class to handle command line arguments
00040     * @namespace cli
00041     */
00042     class ArgsHandler
00043     {
00044
00045         public:
00046             ArgsHandler() = default;
00047             ~ArgsHandler() = default;
00048
00049             ArgsHandler(const ArgsHandler &) = delete;
00050             ArgsHandler &operator=(const ArgsHandler &) = delete;
00051             ArgsHandler(ArgsHandler &&) = delete;
00052             ArgsHandler &operator=(ArgsHandler &&) = delete;
00053
00054             static ArgsConfig ParseArgs(int argc, const char *const argv[]);
00055             static EnvConfig ParseEnv(const char *const env[]);
00056
00057         private:
00058     }; // class ArgsHandler
00059
00060 } // namespace cli

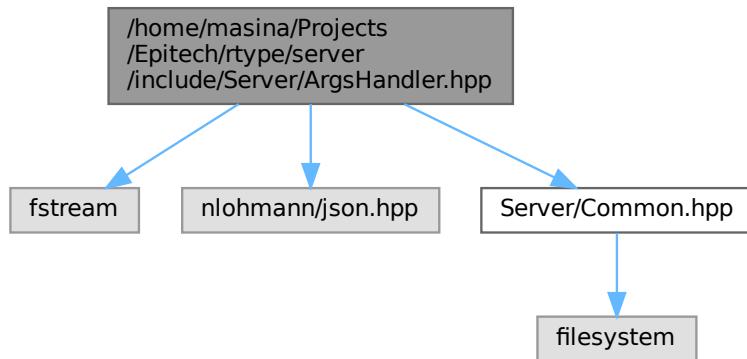
```

8.3 /home/masina/Projects/Epitech/rtype/server/include/Server/ArgsHandler.hpp File Reference

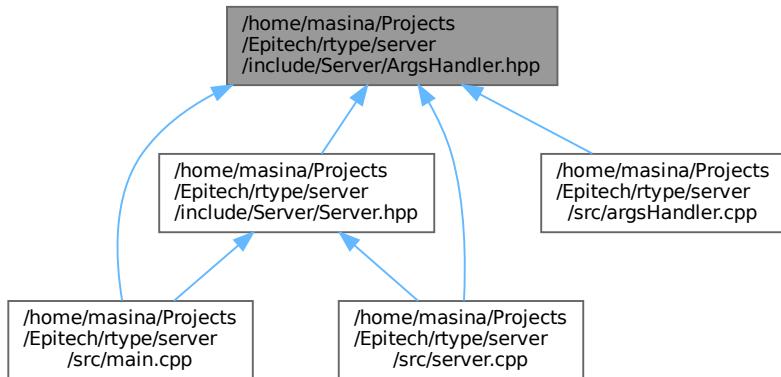
This file contains the ArgsHandler class declaration.

```
#include <fstream>
#include <nlohmann/json.hpp>
#include "Server/Common.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.](#)

```

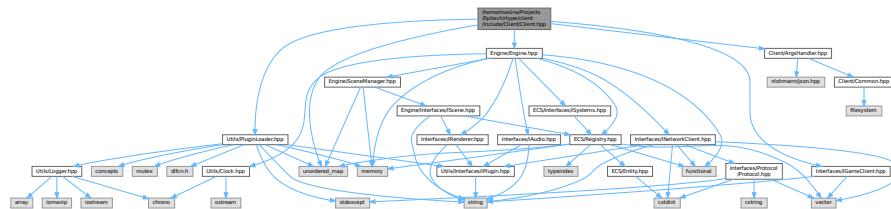
00001 /**
00002  * @file ArgsHandler.hpp
00003  * @brief This file contains the ArgsHandler class declaration
00004  * @namespace srv
00005 /**
00006
00007 #pragma once
00008
00009 #include <fstream>
00010
00011 #include <nlohmann/json.hpp>
00012
00013 #include "Server/Common.hpp"
00014
00015 namespace srv
00016 {
00017
00018     using json = nlohmann::json;
00019
00020     struct ArgsConfig
00021     {
00022         bool exit = false;
00023         std::string host = Config::Network::DEFAULT_NETWORK_HOST;
00024         uint16_t port = Config::Network::DEFAULT_NETWORK_PORT;
00025         std::string network_lib_path;
00026
00027         static ArgsConfig fromFile(const std::string &path);
00028     }; // struct Config
00029     struct EnvConfig
00030     {
00031     };
00032
00033 /**
00034  * @class ArgsHandler
00035  * @brief Class to handle command line arguments
00036  * @namespace srv
00037 /**
00038 class ArgsHandler
00039 {
00040
00041     public:
00042         ArgsHandler() = default;
00043         ~ArgsHandler() = default;
00044
00045         ArgsHandler(const ArgsHandler &) = delete;
00046         ArgsHandler &operator=(const ArgsHandler &) = delete;
00047         ArgsHandler(ArgsHandler &&) = delete;
00048         ArgsHandler &operator=(ArgsHandler &&) = delete;
00049
00050         static ArgsConfig ParseArgs(int argc, const char *const argv[]);
00051         static EnvConfig ParseEnv(const char *const env[]);
00052
00053     private:
00054 }; // class ArgsHandler
00055
00056 } // namespace srv

```

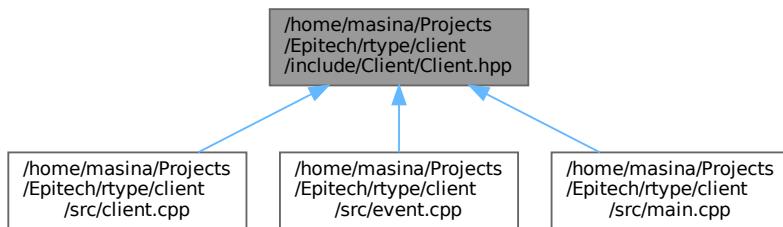
8.5 /home/masina/Projects/Epitech/rtype/client/include/Client/← Client.hpp File Reference

This file contains the Client class declaration.

```
#include <unordered_map>
#include "Client/ArgsHandler.hpp"
#include "Engine/Engine.hpp"
#include "Interfaces/IGameClient.hpp"
#include "Utils/PluginLoader.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 /// @file Client.hpp
00003 /// @brief This file contains the Client class declaration
00004 /// @namespace cli
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/PluginLoader.hpp"
00015
00016 namespace cli
00017 {
00018
00019 /**
00020 /// @class Client
00021 /// @brief Class for the client
00022 /// @namespace cli
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
00041         std::unique_ptr<util::PluginLoader> m_pluginLoader;
00042         std::unique_ptr<eng::Engine> m_engine;
00043         std::unique_ptr<gme::IGameClient> m_game;
00044         std::unordered_map<eng::Key, bool> m_keysPressed;
00045     }; // class Client
00046
00047 } // namespace cli

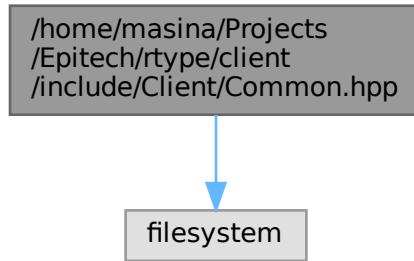
```

8.7 /home/masina/Projects/Epitech/rtype/client/include/Client/← Common.hpp File Reference

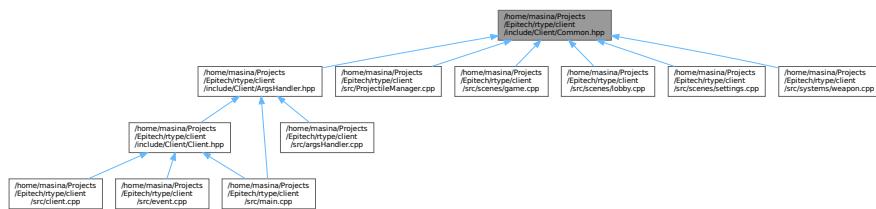
This file contains common definitions and constants.

```
#include <filesystem>
```

Include dependency graph for Common.hpp:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace `cli`
- namespace `cli::Config`
- namespace `cli::Config::Audio`
- namespace `cli::Config::Network`
- namespace `cli::Config::Window`
- namespace `cli::Path`
- namespace `cli::Path::Audio`
- namespace `cli::Path::Font`
- namespace `cli::Path::Plugin`
- namespace `cli::Path::Texture`

Variables

- constexpr auto `cli::Config::Audio::DEFAULT_AUDIO_VOLUME` = 50
- constexpr auto `cli::Config::Audio::DEFAULT_AUDIO_MUTED` = false
- constexpr auto `cli::Config::Network::DEFAULT_NETWORK_HOST` = "127.0.0.1"
- constexpr auto `cli::Config::Network::DEFAULT_NETWORK_PORT` = 2560
- 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::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"
- auto `cli::Path::Plugin::PLUGIN_AUDIO_SFML`
- auto `cli::Path::Plugin::PLUGIN_NETWORK_ASIO_CLIENT`
- auto `cli::Path::Plugin::PLUGIN_RENDERER_SFML`
- constexpr auto `cli::Path::Texture::TEXTURE_PLAYER` = "assets/sprites/r-typesheet42.gif"
- constexpr auto `cli::Path::Texture::TEXTURE_SHOOT` = "assets/sprites/shoot.gif"
- constexpr auto `cli::Path::Texture::TEXTURE_SHOOT_CHARGED` = "assets/sprites/shootcharged.gif"
- constexpr auto `cli::Path::Texture::TEXTURE_SHOOT_LOADING` = "assets/sprites/shootchargedloading.gif"
- constexpr auto `cli::Path::Texture::TEXTURE_ENEMY_EASY` = "assets/sprites/r-typesheet5.gif"
- constexpr auto `cli::Path::Texture::TEXTURE_ASTEROID` = "assets/sprites/r-typesheet3.gif"
- constexpr auto `cli::Path::Texture::TEXTURE_EXPLOSION` = "assets/sprites/r-typesheet44.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 /**
00003 /**
00004 /**
00005 /**
00006
00007 #pragma once
00008 #include <filesystem>
00009
00010 #ifdef _WIN32
00011 #define PLUGINS_EXTENSION ".dll"
00012 #elif __linux__
00013 #define PLUGINS_EXTENSION ".so"
00014 #elif __APPLE__
00015 #define PLUGINS_EXTENSION ".dylib"
00016 #endif
00017
00018 namespace cli
00019 {
00020     namespace Config
00021     {
00022         namespace Audio
00023         {
00024             inline constexpr auto DEFAULT_AUDIO_VOLUME = 50; // unused
00025             inline constexpr auto DEFAULT_AUDIO_MUTED = false; // unused
00026         } // namespace Audio
00027         namespace Network
00028         {
00029             inline constexpr auto DEFAULT_NETWORK_HOST = "127.0.0.1";
00030             inline constexpr auto DEFAULT_NETWORK_PORT = 2560;
00031         } // namespace Network
00032         namespace Window
00033         {
00034             inline constexpr auto DEFAULT_WINDOW_WIDTH = 960;
00035             inline constexpr auto DEFAULT_WINDOW_HEIGHT = 540;
00036             inline constexpr auto DEFAULT_WINDOW_FRAME_LIMIT = 240;
00037             inline constexpr auto DEFAULT_WINDOW_FULLSCREEN = false;
00038         } // namespace Window

```

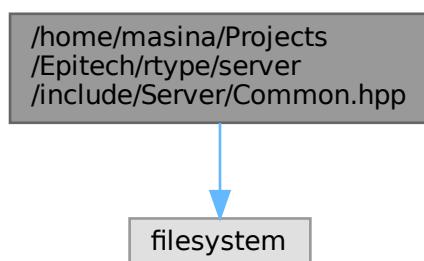
```

00039 } // namespace Config
00040 namespace Path
00041 {
00042     namespace Audio
00043     {
00044         inline constexpr auto AUDIO_TITLE = "assets/audio/title.mp3";
00045         inline constexpr auto AUDIO_COIN = "assets/audio/coin.mp3";
00046         inline constexpr auto AUDIO_BATTLE_THEME = "assets/audio/battle_theme.mp3";
00047     } // namespace Audio
00048     namespace Font
00049     {
00050         inline constexpr auto FONTS_RTYPE = "assets/fonts/r-type.otf";
00051     } // namespace Font
00052     namespace Plugin
00053     {
00054         inline auto PLUGIN_AUDIO_SFML =
00055             std::filesystem::path(PLUGINS_DIR) / ("audio_sfml" + std::string(PLUGINS_EXTENSION));
00056         inline auto PLUGIN_NETWORK_ASIOD_CLIENT =
00057             std::filesystem::path(PLUGINS_DIR) / ("network_asio_client" + std::string(PLUGINS_EXTENSION));
00058         inline auto PLUGIN_RENDERER_SFML =
00059             std::filesystem::path(PLUGINS_DIR) / ("renderer_sfml" + std::string(PLUGINS_EXTENSION));
00060     } // namespace Plugin
00061     namespace Texture
00062     {
00063         inline constexpr auto TEXTURE_PLAYER = "assets/sprites/r-typesheet42.gif";
00064         inline constexpr auto TEXTURE_SHOOT = "assets/sprites/shoot.gif";
00065         inline constexpr auto TEXTURE_SHOOT_CHARGED = "assets/sprites/shootcharged.gif";
00066         inline constexpr auto TEXTURE_SHOOT_LOADING = "assets/sprites/shootchargedloading.gif";
00067         inline constexpr auto TEXTURE_ENEMY_EASY = "assets/sprites/r-typesheet5.gif";
00068         inline constexpr auto TEXTURE_ASTEROID = "assets/sprites/r-typesheet3.gif";
00069         inline constexpr auto TEXTURE_EXPLOSION = "assets/sprites/r-typesheet44.gif";
00070     } // namespace Texture
00071 } // namespace Path
00072 } // namespace cli
00073 }
```

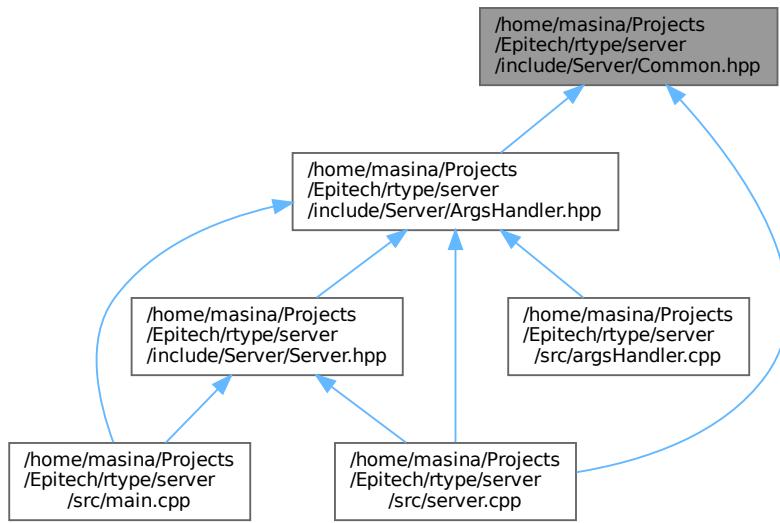
8.9 /home/masina/Projects/Epitech/rtype/server/include/Server/← Common.hpp File Reference

This file contains common definitions and constants.

```
#include <filesystem>
Include dependency graph for Common.hpp:
```



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace `srv`
- namespace `srv::Config`
- namespace `srv::Config::Network`
- namespace `srv::Game`
- namespace `srv::Path`
- namespace `srv::Path::Plugin`

Variables

- constexpr auto `srv::Config::Network::DEFAULT_NETWORK_HOST` = "0.0.0.0"
- constexpr auto `srv::Config::Network::DEFAULT_NETWORK_PORT` = 2560
- constexpr auto `srv::Config::Network::DEFAULT_MAX_CLIENT` = 4
- constexpr auto `srv::Game::DEFAULT_TICK_RATE` = 60
- constexpr auto `srv::Game::DEFAULT_UPDATE_INTERVAL` = 1 / 20.F
- auto `srv::Path::Plugin::PLUGINS_NETWORK_ASIO_SERVER`

8.9.1 Detailed Description

This file contains common definitions and constants.

Definition in file [Common.hpp](#).

8.10 Common.hpp

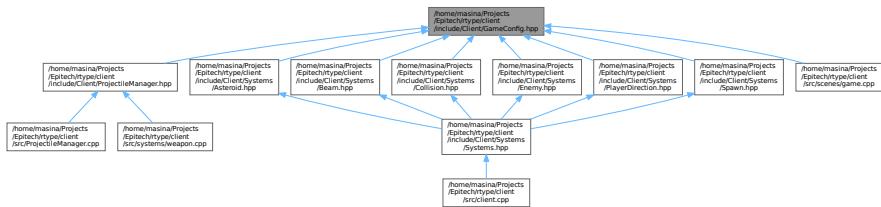
[Go to the documentation of this file.](#)

```
00001 /**
00002 /// @file Common.hpp
00003 /// @brief This file contains common definitions and constants
00004 /// @namespace srv
00005 /**
00006
00007 #pragma once
00008
00009 #include <filesystem>
00010
00011 #ifdef __WIN32
00012 #define PLUGINS_EXTENSION ".dll"
00013 #elif __linux
00014 #define PLUGINS_EXTENSION ".so"
00015 #elif __APPLE
00016 #define PLUGINS_EXTENSION ".dylib"
00017 #endif
00018
00019 namespace srv
00020 {
00021     namespace Config::Network
00022     {
00023         inline constexpr auto DEFAULT_NETWORK_HOST = "0.0.0.0";
00024         inline constexpr auto DEFAULT_NETWORK_PORT = 2560;
00025         inline constexpr auto DEFAULT_MAX_CLIENT = 4;
00026     } // namespace Config::Network
00027     namespace Game
00028     {
00029         inline constexpr auto DEFAULT_TICK_RATE = 60;
00030         inline constexpr auto DEFAULT_UPDATE_INTERVAL = 1 / 20.F;
00031     }
00032     namespace Path::Plugin
00033     {
00034         inline auto PLUGINS_NETWORK_ASIO_SERVER =
00035             std::filesystem::path(PLUGINS_DIR) / ("network_asio_server" + std::string(PLUGINS_EXTENSION));
00036     }
00037 } // namespace srv
```

8.11 /home/masina/Projects/Epitech/rtype/client/include/Client/←
GameConfig.hpp File Reference

Configuration constants for the game.

This graph shows which files directly or indirectly include this file:



Namespaces

- namespace `cli`
 - namespace `cli::GameConfig`
 - namespace `cli::GameConfig::Screen`
 - namespace `cli::GameConfig::Player`
 - namespace `cli::GameConfig::Projectile`

- namespace `cli::GameConfig::Projectile::Basic`
- namespace `cli::GameConfig::Projectile::Supercharged`
- namespace `cli::GameConfig::Animation`
- namespace `cli::GameConfig::Beam`
- namespace `cli::GameConfig::LoadingAnimation`
- namespace `cli::GameConfig::Enemy`
- namespace `cli::GameConfig::Enemy::Easy`
- namespace `cli::GameConfig::Explosion`
- namespace `cli::GameConfig::Asteroid`
- namespace `cli::GameConfig::Asteroid::Small`
- namespace `cli::GameConfig::Asteroid::Medium`
- namespace `cli::GameConfig::Asteroid::Large`
- namespace `cli::GameConfig::Hitbox`

Variables

- constexpr int `cli::GameConfig::Screen::WIDTH` = 1920
- constexpr int `cli::GameConfig::Screen::HEIGHT` = 1080
- constexpr float `cli::GameConfig::Screen::SPAWN_X` = 1950.0f
- constexpr float `cli::GameConfig::Screen::MIN_Y` = 50.0f
- constexpr float `cli::GameConfig::Screen::MAX_Y` = 1030.0f
- constexpr float `cli::GameConfig::Screen::REMOVE_X` = -100.0f
- constexpr float `cli::GameConfig::Screen::REMOVE_MIN_Y` = -50.0f
- constexpr float `cli::GameConfig::Screen::REMOVE_MAX_Y` = 1130.0f
- constexpr float `cli::GameConfig::Player::SPEED` = 500.0f
- constexpr float `cli::GameConfig::Player::DIAGONAL_SPEED_MULTIPLIER` = 0.707f
- constexpr float `cli::GameConfig::Player::SPRITE_WIDTH` = 33.0f
- constexpr float `cli::GameConfig::Player::SPRITE_HEIGHT` = 17.0f
- constexpr float `cli::GameConfig::Player::SCALE` = 2.0f
- constexpr int `cli::GameConfig::Player::FRAMES_PER_ROW` = 5
- constexpr int `cli::GameConfig::Player::TOTAL_FRAMES` = 5
- constexpr float `cli::GameConfig::Projectile::Basic::SPEED` = 800.0f
- constexpr float `cli::GameConfig::Projectile::Basic::DAMAGE` = 10.0f
- constexpr float `cli::GameConfig::Projectile::Basic::LIFETIME` = 3.0f
- constexpr float `cli::GameConfig::Projectile::Basic::SCALE` = 1.0f
- constexpr float `cli::GameConfig::Projectile::Basic::FIRE_COOLDOWN` = 0.3f
- constexpr float `cli::GameConfig::Projectile::Basic::SPRITE_WIDTH` = 16.0f
- constexpr float `cli::GameConfig::Projectile::Basic::SPRITE_HEIGHT` = 8.0f
- constexpr float `cli::GameConfig::Projectile::Supercharged::SPEED` = 1200.0f
- constexpr float `cli::GameConfig::Projectile::Supercharged::DAMAGE` = 25.0f
- constexpr float `cli::GameConfig::Projectile::Supercharged::LIFETIME` = 5.0f
- constexpr float `cli::GameConfig::Projectile::Supercharged::SCALE` = 1.5f
- constexpr float `cli::GameConfig::Projectile::Supercharged::FIRE_COOLDOWN` = 0.2f
- constexpr float `cli::GameConfig::Projectile::Supercharged::CHARGE_TIME` = 0.5f
- constexpr float `cli::GameConfig::Projectile::Supercharged::SPRITE_WIDTH` = 29.0f
- constexpr float `cli::GameConfig::Projectile::Supercharged::SPRITE_HEIGHT` = 24.0f
- constexpr int `cli::GameConfig::Projectile::Supercharged::ANIMATION_FRAMES` = 4
- constexpr float `cli::GameConfig::Projectile::Supercharged::ANIMATION_DURATION` = 0.15f
- constexpr float `cli::GameConfig::Animation::FRAME_DURATION` = 0.1f
- constexpr float `cli::GameConfig::Beam::MAX_CHARGE` = 1.0f
- constexpr float `cli::GameConfig::Beam::CHARGE_RATE` = 1.0f
- constexpr float `cli::GameConfig::Beam::BAR_WIDTH` = 120.0f
- constexpr float `cli::GameConfig::Beam::BAR_HEIGHT` = 12.0f
- constexpr float `cli::GameConfig::Beam::BAR_X` = 10.0f

- constexpr float `cli::GameConfig::Beam::BAR_Y` = 10.0f
- constexpr float `cli::GameConfig::LoadingAnimation::SPRITE_WIDTH` = 29.0f
- constexpr float `cli::GameConfig::LoadingAnimation::SPRITE_HEIGHT` = 24.0f
- constexpr int `cli::GameConfig::LoadingAnimation::ANIMATION_FRAMES` = 4
- constexpr float `cli::GameConfig::LoadingAnimation::ANIMATION_DURATION` = 0.15f
- constexpr float `cli::GameConfig::LoadingAnimation::OFFSET_X` = 40.0f
- constexpr float `cli::GameConfig::LoadingAnimation::OFFSET_Y` = 0.0f
- constexpr float `cli::GameConfig::Enemy::Easy::HEALTH` = 1.0f
- constexpr float `cli::GameConfig::Enemy::Easy::DAMAGE` = 5.0f
- constexpr float `cli::GameConfig::Enemy::Easy::SPEED` = 80.0f
- constexpr float `cli::GameConfig::Enemy::Easy::SPRITE_WIDTH` = 32.0f
- constexpr float `cli::GameConfig::Enemy::Easy::SPRITE_HEIGHT` = 32.0f
- constexpr float `cli::GameConfig::Enemy::Easy::SCALE` = 2.0f
- constexpr float `cli::GameConfig::Enemy::Easy::SHOOT_COOLDOWN` = 2.0f
- constexpr float `cli::GameConfig::Enemy::Easy::SPAWN_RATE` = 2.0f
- constexpr int `cli::GameConfig::Enemy::Easy::ANIMATION_FRAMES` = 4
- constexpr float `cli::GameConfig::Enemy::Easy::ANIMATION_DURATION` = 0.5f
- constexpr int `cli::GameConfig::Enemy::Easy::FRAMES_PER_ROW` = 4
- constexpr float `cli::GameConfig::Explosion::SPRITE_WIDTH` = 32.0f
- constexpr float `cli::GameConfig::Explosion::SPRITE_HEIGHT` = 32.0f
- constexpr int `cli::GameConfig::Explosion::ANIMATION_FRAMES` = 4
- constexpr float `cli::GameConfig::Explosion::ANIMATION_DURATION` = 0.1f
- constexpr int `cli::GameConfig::Explosion::FRAMES_PER_ROW` = 4
- constexpr float `cli::GameConfig::Explosion::LIFETIME` = 0.4f
- constexpr float `cli::GameConfig::Explosion::SCALE` = 2.0f
- constexpr float `cli::GameConfig::Asteroid::Small::HEALTH` = 20.0f
- constexpr float `cli::GameConfig::Asteroid::Small::SPEED` = 80.0f
- constexpr float `cli::GameConfig::Asteroid::Small::SPRITE_WIDTH` = 18.0f
- constexpr float `cli::GameConfig::Asteroid::Small::SPRITE_HEIGHT` = 18.0f
- constexpr float `cli::GameConfig::Asteroid::Small::SCALE` = 2.0f
- constexpr float `cli::GameConfig::Asteroid::Small::ROTATION_SPEED` = 90.0f
- constexpr float `cli::GameConfig::Asteroid::Small::SPAWN_RATE` = 1.0f
- constexpr int `cli::GameConfig::Asteroid::Small::ANIMATION_FRAMES` = 11
- constexpr float `cli::GameConfig::Asteroid::Small::ANIMATION_DURATION` = 0.5f
- constexpr int `cli::GameConfig::Asteroid::Small::FRAMES_PER_ROW` = 11
- constexpr float `cli::GameConfig::Asteroid::Medium::HEALTH` = 40.0f
- constexpr float `cli::GameConfig::Asteroid::Medium::SPEED` = 60.0f
- constexpr float `cli::GameConfig::Asteroid::Medium::SPRITE_WIDTH` = 32.0f
- constexpr float `cli::GameConfig::Asteroid::Medium::SPRITE_HEIGHT` = 32.0f
- constexpr float `cli::GameConfig::Asteroid::Medium::SCALE` = 1.0f
- constexpr float `cli::GameConfig::Asteroid::Medium::ROTATION_SPEED` = 60.0f
- constexpr float `cli::GameConfig::Asteroid::Medium::SPAWN_RATE` = 2.0f
- constexpr float `cli::GameConfig::Asteroid::Large::HEALTH` = 80.0f
- constexpr float `cli::GameConfig::Asteroid::Large::SPEED` = 40.0f
- constexpr float `cli::GameConfig::Asteroid::Large::SPRITE_WIDTH` = 64.0f
- constexpr float `cli::GameConfig::Asteroid::Large::SPRITE_HEIGHT` = 64.0f
- constexpr float `cli::GameConfig::Asteroid::Large::SCALE` = 1.5f
- constexpr float `cli::GameConfig::Asteroid::Large::ROTATION_SPEED` = 30.0f
- constexpr float `cli::GameConfig::Asteroid::Large::SPAWN_RATE` = 4.0f
- constexpr float `cli::GameConfig::Hitbox::PLAYER_RADIUS` = 20.0f
- constexpr float `cli::GameConfig::Hitbox::ENEMY_RADIUS` = 15.0f
- constexpr float `cli::GameConfig::Hitbox::PROJECTILE_BASIC_RADIUS` = 5.0f
- constexpr float `cli::GameConfig::Hitbox::PROJECTILE_SUPERCHARGED_RADIUS` = 8.0f
- constexpr float `cli::GameConfig::Hitbox::ASTEROID_SMALL_RADIUS` = 25.0f
- constexpr float `cli::GameConfig::Hitbox::ASTEROID_MEDIUM_RADIUS` = 40.0f
- constexpr float `cli::GameConfig::Hitbox::ASTEROID_LARGE_RADIUS` = 60.0f

8.11.1 Detailed Description

Configuration constants for the game.

Definition in file [GameConfig.hpp](#).

8.12 GameConfig.hpp

[Go to the documentation of this file.](#)

```

00001 /**
00002  * @file GameConfig.hpp
00003  * @brief Configuration constants for the game
00004  * @namespace cli
00005 /**
00006
00007 #pragma once
00008
00009 namespace cli
00010 {
00011     namespace GameConfig
00012     {
00013         namespace Screen
00014         {
00015             inline constexpr int WIDTH = 1920;
00016             inline constexpr int HEIGHT = 1080;
00017             inline constexpr float SPAWN_X = 1950.0f;
00018             inline constexpr float MIN_Y = 50.0f;
00019             inline constexpr float MAX_Y = 1030.0f;
00020             inline constexpr float REMOVE_X = -100.0f;
00021             inline constexpr float REMOVE_MIN_Y = -50.0f;
00022             inline constexpr float REMOVE_MAX_Y = 1130.0f;
00023         } // namespace Screen
00024         namespace Player
00025         {
00026             inline constexpr float SPEED = 500.0f;
00027             inline constexpr float DIAGONAL_SPEED_MULTIPLIER = 0.707f;
00028             inline constexpr float SPRITE_WIDTH = 33.0f;
00029             inline constexpr float SPRITE_HEIGHT = 17.0f;
00030             inline constexpr float SCALE = 2.0f;
00031             inline constexpr int FRAMES_PER_ROW = 5;
00032             inline constexpr int TOTAL_FRAMES = 5;
00033         } // namespace Player
00034         namespace Projectile
00035         {
00036             namespace Basic
00037             {
00038                 inline constexpr float SPEED = 800.0f;
00039                 inline constexpr float DAMAGE = 10.0f;
00040                 inline constexpr float LIFETIME = 3.0f;
00041                 inline constexpr float SCALE = 1.0f;
00042                 inline constexpr float FIRE_COOLDOWN = 0.3f;
00043                 inline constexpr float SPRITE_WIDTH = 16.0f;
00044                 inline constexpr float SPRITE_HEIGHT = 8.0f;
00045             } // namespace Basic
00046
00047             namespace Supercharged
00048             {
00049                 inline constexpr float SPEED = 1200.0f;
00050                 inline constexpr float DAMAGE = 25.0f;
00051                 inline constexpr float LIFETIME = 5.0f;
00052                 inline constexpr float SCALE = 1.5f;
00053                 inline constexpr float FIRE_COOLDOWN = 0.2f;
00054                 inline constexpr float CHARGE_TIME = 0.5f;
00055                 inline constexpr float SPRITE_WIDTH = 29.0f;
00056                 inline constexpr float SPRITE_HEIGHT = 24.0f;
00057                 inline constexpr int ANIMATION_FRAMES = 4;
00058                 inline constexpr float ANIMATION_DURATION = 0.15f;
00059             } // namespace Supercharged
00060         } // namespace Projectile
00061         namespace Animation
00062         {
00063             inline constexpr float FRAME_DURATION = 0.1f;
00064         }
00065         namespace Beam
00066         {
00067             inline constexpr float MAX_CHARGE = 1.0f;
00068             inline constexpr float CHARGE_RATE = 1.0f;
00069             inline constexpr float BAR_WIDTH = 120.0f;
00070             inline constexpr float BAR_HEIGHT = 12.0f;

```

```

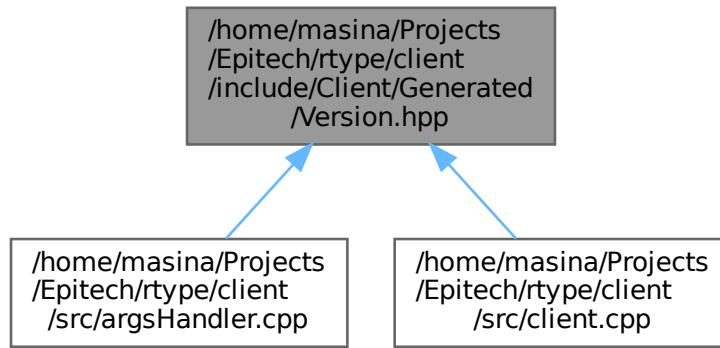
00071     inline constexpr float BAR_X = 10.0f;
00072     inline constexpr float BAR_Y = 10.0f;
00073 } // namespace Beam
00074 namespace LoadingAnimation
00075 {
00076     inline constexpr float SPRITE_WIDTH = 29.0f;
00077     inline constexpr float SPRITE_HEIGHT = 24.0f;
00078     inline constexpr int ANIMATION_FRAMES = 4;
00079     inline constexpr float ANIMATION_DURATION = 0.15f;
00080     inline constexpr float OFFSET_X = 40.0f;
00081     inline constexpr float OFFSET_Y = 0.0f;
00082 } // namespace LoadingAnimation
00083 namespace Enemy
00084 {
00085     namespace Easy
00086     {
00087         inline constexpr float HEALTH = 1.0f;
00088         inline constexpr float DAMAGE = 5.0f;
00089         inline constexpr float SPEED = 80.0f;
00090         inline constexpr float SPRITE_WIDTH = 32.0f;
00091         inline constexpr float SPRITE_HEIGHT = 32.0f;
00092         inline constexpr float SCALE = 2.0f;
00093         inline constexpr float SHOOT_COOLDOWN = 2.0f;
00094         inline constexpr float SPAWN_RATE = 2.0f;
00095         inline constexpr int ANIMATION_FRAMES = 4;
00096         inline constexpr float ANIMATION_DURATION = 0.5f;
00097         inline constexpr int FRAMES_PER_ROW = 4;
00098     } // namespace Easy
00099 } // namespace Enemy
00100 namespace Explosion
00101 {
00102     inline constexpr float SPRITE_WIDTH = 32.0f;
00103     inline constexpr float SPRITE_HEIGHT = 32.0f;
00104     inline constexpr int ANIMATION_FRAMES = 4;
00105     inline constexpr float ANIMATION_DURATION = 0.1f;
00106     inline constexpr int FRAMES_PER_ROW = 4;
00107     inline constexpr float LIFETIME = 0.4f;
00108     inline constexpr float SCALE = 2.0f;
00109 } // namespace Explosion
00110 namespace Asteroid
00111 {
00112     namespace Small
00113     {
00114         inline constexpr float HEALTH = 20.0f;
00115         inline constexpr float SPEED = 80.0f;
00116         inline constexpr float SPRITE_WIDTH = 18.0f;
00117         inline constexpr float SPRITE_HEIGHT = 18.0f;
00118         inline constexpr float SCALE = 2.0f;
00119         inline constexpr float ROTATION_SPEED = 90.0f;
00120         inline constexpr float SPAWN_RATE = 1.0f;
00121         inline constexpr int ANIMATION_FRAMES = 11;
00122         inline constexpr float ANIMATION_DURATION = 0.5f;
00123         inline constexpr int FRAMES_PER_ROW = 11;
00124     } // namespace Small
00125
00126     namespace Medium
00127     {
00128         inline constexpr float HEALTH = 40.0f;
00129         inline constexpr float SPEED = 60.0f;
00130         inline constexpr float SPRITE_WIDTH = 32.0f;
00131         inline constexpr float SPRITE_HEIGHT = 32.0f;
00132         inline constexpr float SCALE = 1.0f;
00133         inline constexpr float ROTATION_SPEED = 60.0f;
00134         inline constexpr float SPAWN_RATE = 2.0f;
00135     } // namespace Medium
00136
00137     namespace Large
00138     {
00139         inline constexpr float HEALTH = 80.0f;
00140         inline constexpr float SPEED = 40.0f;
00141         inline constexpr float SPRITE_WIDTH = 64.0f;
00142         inline constexpr float SPRITE_HEIGHT = 64.0f;
00143         inline constexpr float SCALE = 1.5f;
00144         inline constexpr float ROTATION_SPEED = 30.0f;
00145         inline constexpr float SPAWN_RATE = 4.0f;
00146     } // namespace Large
00147 } // namespace Asteroid
00148 namespace Hitbox
00149 {
00150     inline constexpr float PLAYER_RADIUS = 20.0f;
00151     inline constexpr float ENEMY_RADIUS = 15.0f;
00152     inline constexpr float PROJECTILE_BASIC_RADIUS = 5.0f;
00153     inline constexpr float PROJECTILE_SUPERCHARGED_RADIUS = 8.0f;
00154     inline constexpr float ASTEROID_SMALL_RADIUS = 25.0f;
00155     inline constexpr float ASTEROID_MEDIUM_RADIUS = 40.0f;
00156     inline constexpr float ASTEROID_LARGE_RADIUS = 60.0f;
00157 } // namespace Hitbox

```

```
00158 } // namespace GameConfig
00159 } // namespace cli
```

8.13 /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 "a8cb17a"
- #define GIT_TAG "a8cb17a"
- #define BUILD_TYPE "Release"

8.13.1 Macro Definition Documentation

8.13.1.1 BUILD_TYPE

```
#define BUILD_TYPE "Release"
```

Definition at line 15 of file [Version.hpp](#).

Referenced by [cli::Client::Client\(\)](#), and [srv::Server::Server\(\)](#).

8.13.1.2 GIT_COMMIT_HASH

```
#define GIT_COMMIT_HASH "a8cb17a"
```

Definition at line 13 of file [Version.hpp](#).

Referenced by [cli::Client::Client\(\)](#), and [srv::Server::Server\(\)](#).

8.13.1.3 GIT_TAG

```
#define GIT_TAG "a8cb17a"
```

Definition at line 14 of file [Version.hpp](#).

Referenced by [cli::Client::Client\(\)](#), and [srv::Server::Server\(\)](#).

8.13.1.4 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\(\)](#).

8.13.1.5 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\(\)](#).

8.13.1.6 PROJECT_VERSION_MAJOR

```
#define PROJECT_VERSION_MAJOR "0"
```

Definition at line 9 of file [Version.hpp](#).

8.13.1.7 PROJECT_VERSION_MINOR

```
#define PROJECT_VERSION_MINOR "0"
```

Definition at line 10 of file [Version.hpp](#).

8.13.1.8 PROJECT_VERSION_PATCH

```
#define PROJECT_VERSION_PATCH "0"
```

Definition at line 11 of file [Version.hpp](#).

8.14 Version.hpp

[Go to the documentation of this file.](#)

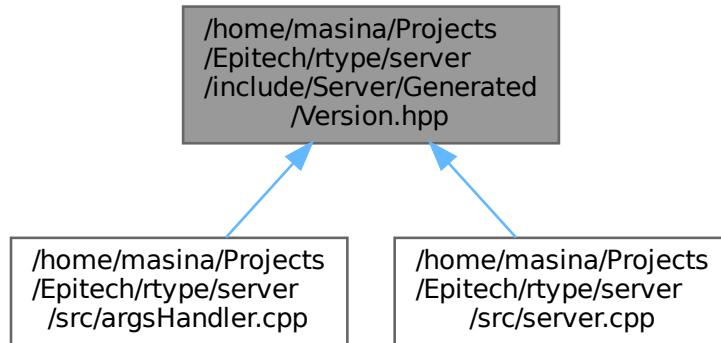
```

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 "a8cb17a"
00014 #define GIT_TAG "a8cb17a"
00015 #define BUILD_TYPE "Release"

```

8.15 /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 "a8cb17a"
- #define GIT_TAG "a8cb17a"
- #define BUILD_TYPE "Release"

8.15.1 Macro Definition Documentation

8.15.1.1 BUILD_TYPE

```
#define BUILD_TYPE "Release"
```

Definition at line 15 of file [Version.hpp](#).

8.15.1.2 GIT_COMMIT_HASH

```
#define GIT_COMMIT_HASH "a8cb17a"
```

Definition at line 13 of file [Version.hpp](#).

8.15.1.3 GIT_TAG

```
#define GIT_TAG "a8cb17a"
```

Definition at line 14 of file [Version.hpp](#).

8.15.1.4 PROJECT_NAME

```
#define PROJECT_NAME "r-type_server"
```

Definition at line 7 of file [Version.hpp](#).

8.15.1.5 PROJECT_VERSION

```
#define PROJECT_VERSION "0.0.0"
```

Definition at line 8 of file [Version.hpp](#).

8.15.1.6 PROJECT_VERSION_MAJOR

```
#define PROJECT_VERSION_MAJOR "0"
```

Definition at line 9 of file [Version.hpp](#).

8.15.1.7 PROJECT_VERSION_MINOR

```
#define PROJECT_VERSION_MINOR "0"
```

Definition at line 10 of file [Version.hpp](#).

8.15.1.8 PROJECT_VERSION_PATCH

```
#define PROJECT_VERSION_PATCH "0"
```

Definition at line 11 of file [Version.hpp](#).

8.16 Version.hpp

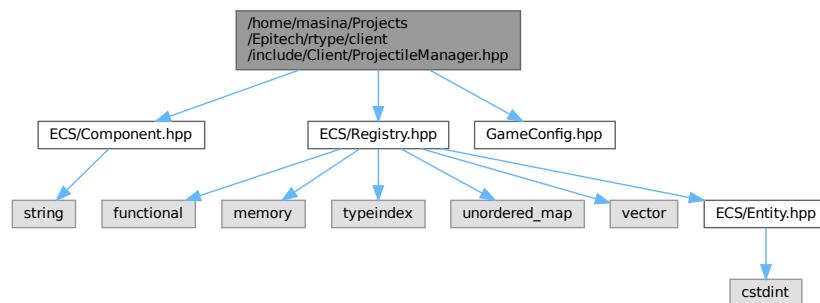
[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 //=====
00004 // DO NOT EDIT THIS FILE MANUALLY. IT IS GENERATED BY CMAKE DURING THE BUILD PROCESS.
00005 //=====
00006
00007 #define PROJECT_NAME "r-type_server"
00008 #define PROJECT_VERSION "0.0.0"
00009 #define PROJECT_VERSION_MAJOR "0"
00010 #define PROJECT_VERSION_MINOR "0"
00011 #define PROJECT_VERSION_PATCH "0"
00012
00013 #define GIT_COMMIT_HASH "a8cb17a"
00014 #define GIT_TAG "a8cb17a"
00015 #define BUILD_TYPE "Release"
```

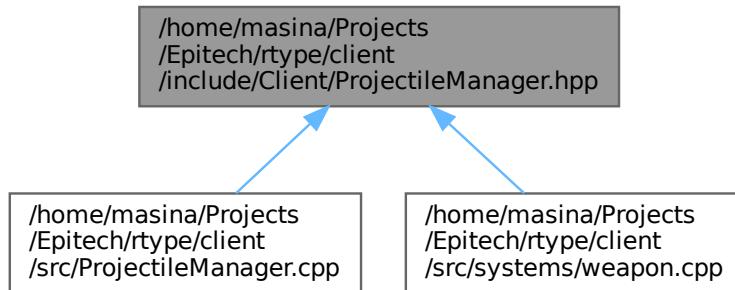
8.17 /home/masina/Projects/Epitech/rtype/client/include/Client/ProjectileManager.hpp File Reference

Manages projectile creation and configuration.

```
#include "ECS/Component.hpp"
#include "ECS/Registry.hpp"
#include "GameConfig.hpp"
Include dependency graph for ProjectileManager.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [cli::ProjectileManager](#)
Handles projectile creation and management.

Namespaces

- namespace [cli](#)

8.17.1 Detailed Description

Manages projectile creation and configuration.

Definition in file [ProjectileManager.hpp](#).

8.18 ProjectileManager.hpp

[Go to the documentation of this file.](#)

```

00001 /**
00002  * @file ProjectileManager.hpp
00003  * @brief Manages projectile creation and configuration
00004  * @namespace cli
00005 /**
00006
00007 #pragma once
00008
00009 #include "ECS/Component.hpp"
00010 #include "ECS/Registry.hpp"
00011 #include "GameConfig.hpp"
00012
00013 namespace cli
00014 {
00015 /**
00016  * @class ProjectileManager
00017  * @brief Handles projectile creation and management
00018  * @namespace cli
00019 /**
00020 class ProjectileManager
00021 {
00022     public:
  
```

```

00023     ProjectileManager() = default;
00024     ~ProjectileManager() = default;
00025
00026     ProjectileManager(const ProjectileManager &) = delete;
00027     ProjectileManager &operator=(const ProjectileManager &) = delete;
00028     ProjectileManager(ProjectileManager &&) = delete;
00029     ProjectileManager &operator=(ProjectileManager &&) = delete;
00030
00031     /**
00032      /// @brief Create a basic projectile
00033      /// @param registry The ECS registry
00034      /// @param x X position
00035      /// @param y Y position
00036      /// @param velocityX X velocity
00037      /// @param velocityY Y velocity
00038      /// @return The created entity
00039      /**
00040      static ecs::Entity createBasicProjectile(ecs::Registry &registry, float x, float y, float velocityX,
00041                                              float velocityY);
00042
00043     /**
00044      /// @brief Create a supercharged projectile
00045      /// @param registry The ECS registry
00046      /// @param x X position
00047      /// @param y Y position
00048      /// @param velocityX X velocity
00049      /// @param velocityY Y velocity
00050      /// @return The created entity
00051      /**
00052      static ecs::Entity createSuperchargedProjectile(ecs::Registry &registry, float x, float y, float velocityX,
00053                                              float velocityY);
00054
00055     private:
00056     /**
00057      /// @brief Create a projectile with given parameters
00058      /// @param registry The ECS registry
00059      /// @param type Projectile type
00060      /// @param x X position
00061      /// @param y Y position
00062      /// @param velocityX X velocity
00063      /// @param velocityY Y velocity
00064      /// @return The created entity
00065      /**
00066      static ecs::Entity createProjectile(ecs::Registry &registry, ecs::Projectile::Type type, float x, float y,
00067                                         float velocityX, float velocityY);
00068  }; // class ProjectileManager
00069 } // namespace cli

```

8.19 /home/masina/Projects/Epitech/rtype/client/include/Client/Scenes/Game.hpp File Reference

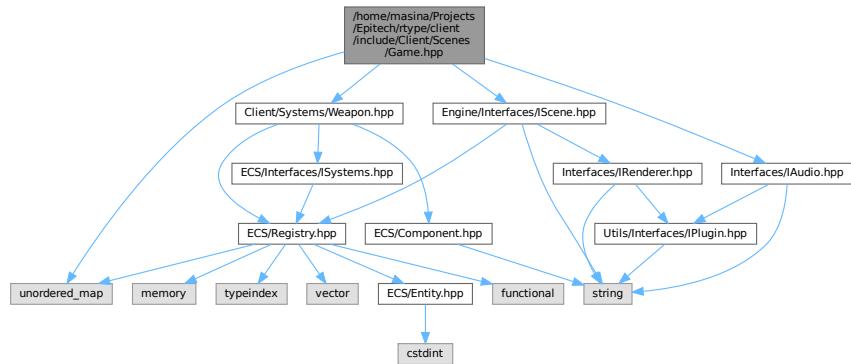
This file contains the Game scene.

```

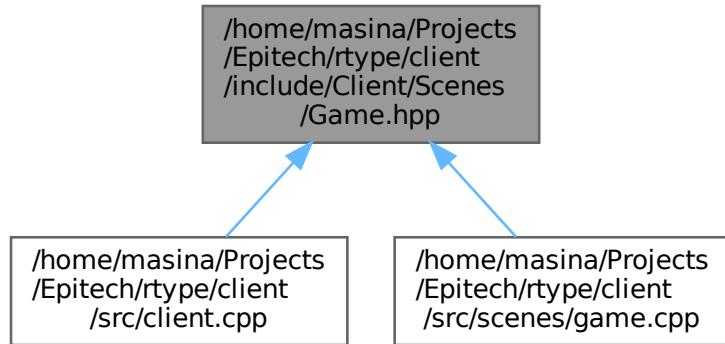
#include <unordered_map>
#include "Client/Systems/Weapon.hpp"
#include "Engine/Interfaces/IScene.hpp"
#include "Interfaces/IAudio.hpp"

```

Include dependency graph for Game.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [cli::Game](#)
Game scene.

Namespaces

- namespace [eng](#)
- namespace [cli](#)

8.19.1 Detailed Description

This file contains the Game scene.

Definition in file [Game.hpp](#).

8.20 Game.hpp

[Go to the documentation of this file.](#)

```

00001 /**
00002 /// @file Game.hpp
00003 /// @brief This file contains the Game scene
00004 /// @namespace eng
00005 /**
00006
00007 #pragma once
00008
00009 #include <unordered_map>
00010
00011 #include "Client/Systems/Weapon.hpp"
00012 #include "Engine/Interfaces/IScene.hpp"
00013 #include "Interfaces/IAudio.hpp"
00014
00015 namespace cli
00016 {
00017 /**
00018 /// @class Game
00019 /// @brief Game scene
00020 /// @namespace cli
00021 /**
00022 class Game final : public eng::AScene
00023 {
00024     public:
00025         Game(const std::shared_ptr<eng::IRenderer> &renderer, const std::shared_ptr<eng::IAudio> &audio);
00026         ~Game() override = default;
00027
00028         Game(const Game &other) = delete;
00029         Game &operator=(const Game &other) = delete;
00030         Game(Game &&other) = delete;
00031         Game &operator=(Game &&other) = delete;
00032
00033         void update(float dt, const eng::WindowSize &size) override;
00034         void event(const eng::Event &event) override;
00035
00036     private:
00037         std::unordered_map<eng::Key, bool> m_keysPressed;
00038
00039         ecs::Entity m_playerEntity;
00040         ecs::Entity m_fpsEntity;
00041         ecs::Entity m_enemyCounterEntity;
00042         ecs::Entity m_asteroidCounterEntity;
00043         const std::shared_ptr<eng::IAudio> &m_audio;
00044
00045         // WeaponSystem m_weaponSystem; TODO(bobis33): tofix
00046     }; // class Game
00047 } // namespace cli

```

8.21 /home/masina/Projects/Epitech/rtype/client/include/Client/Scenes/Lobby.hpp File Reference

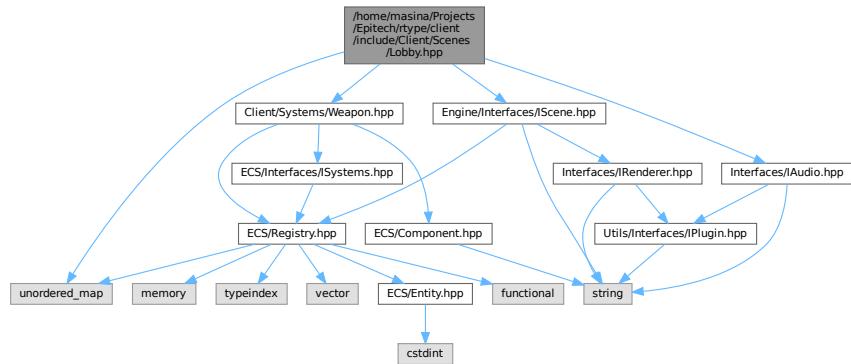
This file contains the lobby scene.

```

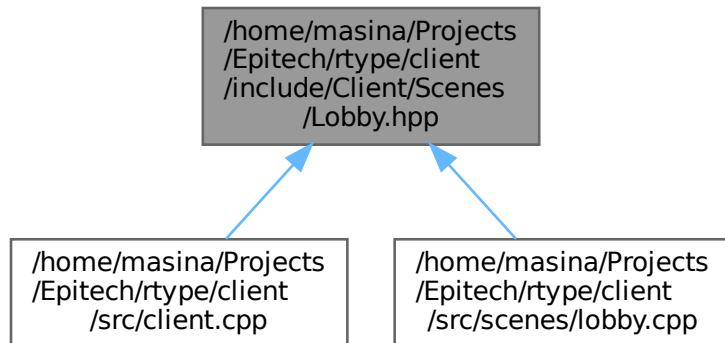
#include <unordered_map>
#include "Client/Systems/Weapon.hpp"
#include "Engine/Interfaces/IScene.hpp"
#include "Interfaces/IAudio.hpp"

```

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](#)

8.21.1 Detailed Description

This file contains the lobby scene.

Definition in file [Lobby.hpp](#).

8.22 Lobby.hpp

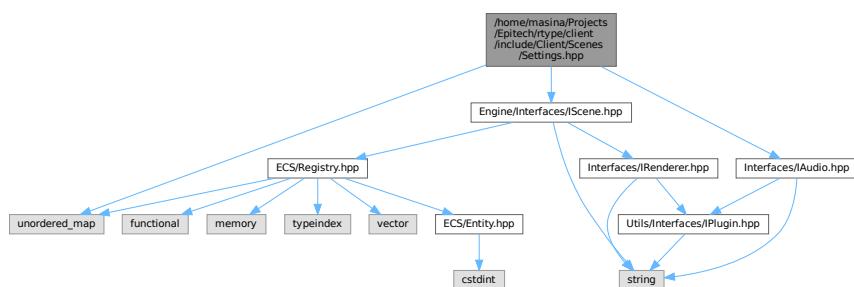
[Go to the documentation of this file.](#)

```
00001 /**
00002 /// @file Lobby.hpp
00003 /// @brief This file contains the lobby scene
00004 /// @namespace eng
00005 /**
00006
00007 #pragma once
00008
00009 #include <unordered_map>
00010
00011 #include "Client/Systems/Weapon.hpp"
00012 #include "Engine/Interfaces/IScene.hpp"
00013 #include "Interfaces/IAudio.hpp"
00014
00015 namespace cli
00016 {
00017 /**
00018 /// @class Lobby
00019 /// @brief Lobby scene
00020 /// @namespace cli
00021 /**
00022 class Lobby final : public eng::AScene
00023 {
00024     public:
00025         Lobby(const std::shared_ptr<eng::IRenderer> &renderer, const std::shared_ptr<eng::IAudio> &audio);
00026         ~Lobby() override = default;
00027
00028         Lobby(const Lobby &other) = delete;
00029         Lobby &operator=(const Lobby &other) = delete;
00030         Lobby(Lobby &&other) = delete;
00031         Lobby &operator=(Lobby &&other) = delete;
00032
00033         void update(float dt, const eng::WindowSize &size) override;
00034         void event(const eng::Event &event) override;
00035
00036         std::function<void(const std::string &option)> onOptionSelected;
00037
00038     private:
00039         std::unordered_map<eng::Key, bool> m_keysPressed;
00040         ecs::Entity m_fpsEntity;
00041         const std::vector<std::string> m_menuOptions = {"Solo", "Multi", "Settings"};
00042         const std::shared_ptr<eng::IAudio> &m_audio;
00043
00044         int m_selectedIndex = 0;
00045     }; // class Lobby
00046 } // namespace cli
```

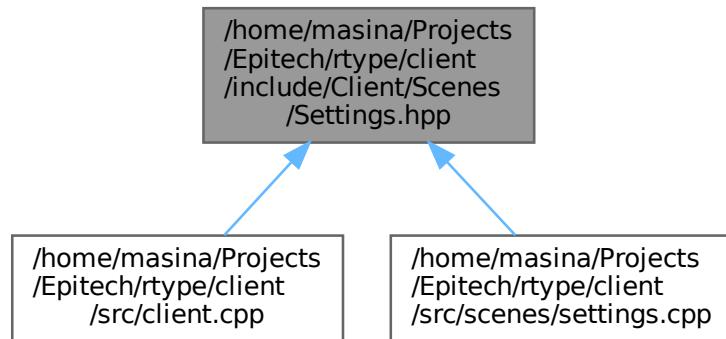
8.23 /home/masina/Projects/Epitech/rtype/client/include/Client/Scenes/Settings.hpp File Reference

This file contains the settings scene.

```
#include <unordered_map>
#include "Engine/Interfaces/IScene.hpp"
#include "Interfaces/IAudio.hpp"
Include dependency graph for Settings.hpp
```



This graph shows which files directly or indirectly include this file:



Classes

- class [cli::Settings](#)
Settings scene.

Namespaces

- namespace [eng](#)
- namespace [cli](#)

8.23.1 Detailed Description

This file contains the settings scene.

Definition in file [Settings.hpp](#).

8.24 Settings.hpp

[Go to the documentation of this file.](#)

```

00001 /**
00002 /// @file Settings.hpp
00003 /// @brief This file contains the settings scene
00004 /// @namespace eng
00005 /**
00006
00007 #pragma once
00008
00009 #include <unordered_map>
00010
00011 #include "Engine/Interfaces/IScene.hpp"
00012 #include "Interfaces/IAudio.hpp"
00013
00014 namespace cli
00015 {
00016 /**
00017 /// @class Settings
00018 /// @brief Settings scene
  
```

```

00019 // @namespace cli
00020 /**
00021 class Settings final : public eng::AScene
00022 {
00023     public:
00024         Settings(const std::shared_ptr<eng::IRenderer> &renderer, const std::shared_ptr<eng::IAudio> &audio);
00025         ~Settings() override = default;
00026
00027         Settings(const Settings &other) = delete;
00028         Settings &operator=(const Settings &other) = delete;
00029         Settings(Settings &&other) = delete;
00030         Settings &operator=(Settings &&other) = delete;
00031
00032         void update(float dt, const eng::WindowSize &size) override;
00033         void event(const eng::Event &event) override;
00034
00035         std::function<void()> onLeave;
00036
00037     private:
00038         std::unordered_map<eng::Key, bool> m_keysPressed;
00039         const std::shared_ptr<eng::IAudio> &m_audio;
00040     }; // class Settings
00041 } // namespace cli

```

8.25 /home/masina/Projects/Epitech/rtype/client/include/Client/SpriteRect.hpp File Reference

Helper functions for pixel-perfect sprite rectangles.

Classes

- struct `cli::SpriteRect::SpriteSheet::Asteroid`
- struct `cli::SpriteRect::SpriteSheet::Enemy`
- struct `cli::SpriteRect::Rect`

Namespaces

- namespace `cli`
- namespace `cli::SpriteRect`
- namespace `cli::SpriteRect::SpriteSheet`

Functions

- `Rect cli::SpriteRect::asteroidRect (int col, int row=0)`
- `Rect cli::SpriteRect::enemyRect (int col, int row=0)`

8.25.1 Detailed Description

Helper functions for pixel-perfect sprite rectangles.

Definition in file [SpriteRect.hpp](#).

8.26 SpriteRect.hpp

[Go to the documentation of this file.](#)

```

00001 /**
00002  * @file SpriteRect.hpp
00003  * @brief Helper functions for pixel-perfect sprite rectangles
00004  * @namespace cli
00005 /**
00006
00007 #pragma once
00008
00009 namespace cli
00010 {
00011     namespace SpriteRect
00012     {
00013         // SpriteSheet configuration for pixel-perfect rendering
00014         namespace SpriteSheet
00015         {
00016             // Asteroid spritesheet (r-typesheet3.gif)
00017             struct Asteroid
00018             {
00019                 static constexpr int frameW = 16;
00020                 static constexpr int frameH = 16;
00021                 static constexpr int marginX = 1;
00022                 static constexpr int marginY = 1;
00023                 static constexpr int spacingX = 1;
00024                 static constexpr int spacingY = 0;
00025                 static constexpr int totalFrames = 12;
00026             };
00027
00028             // Enemy spritesheet (r-typesheet5.gif)
00029             struct Enemy
00030             {
00031                 static constexpr int frameW = 16;
00032                 static constexpr int frameH = 16;
00033                 static constexpr int marginX = 1;
00034                 static constexpr int marginY = 1;
00035                 static constexpr int spacingX = 1;
00036                 static constexpr int spacingY = 0;
00037                 static constexpr int totalFrames = 8;
00038             };
00039         } // namespace SpriteSheet
00040
00041         // Simple rectangle structure
00042         struct Rect
00043         {
00044             int x, y, w, h;
00045         };
00046
00047         // Calculate exact rectangle for asteroid frame
00048         inline Rect asteroidRect(int col, int row = 0)
00049         {
00050             const int fx = SpriteSheet::Asteroid::marginX +
00051                         col * (SpriteSheet::Asteroid::frameW + SpriteSheet::Asteroid::spacingX);
00052             const int fy = SpriteSheet::Asteroid::marginY +
00053                         row * (SpriteSheet::Asteroid::frameH + SpriteSheet::Asteroid::spacingY);
00054             return {fx, fy, SpriteSheet::Asteroid::frameW, SpriteSheet::Asteroid::frameH};
00055         }
00056
00057         // Calculate exact rectangle for enemy frame
00058         inline Rect enemyRect(int col, int row = 0)
00059         {
00060             const int fx =
00061                 SpriteSheet::Enemy::marginX + col * (SpriteSheet::Enemy::frameW + SpriteSheet::Enemy::spacingX);
00062             const int fy =
00063                 SpriteSheet::Enemy::marginY + row * (SpriteSheet::Enemy::frameH + SpriteSheet::Enemy::spacingY);
00064             return {fx, fy, SpriteSheet::Enemy::frameW, SpriteSheet::Enemy::frameH};
00065         }
00066     } // namespace SpriteRect
00067 } // namespace cli

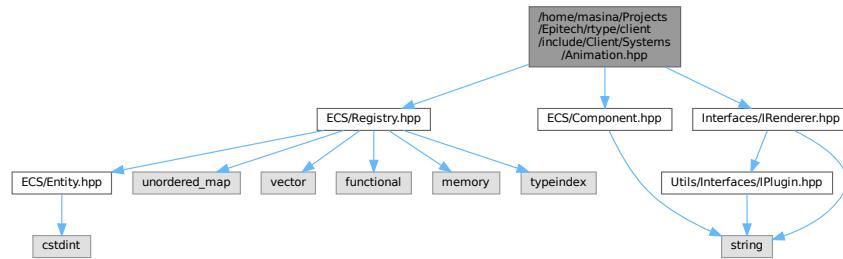
```

8.27 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Animation.hpp File Reference

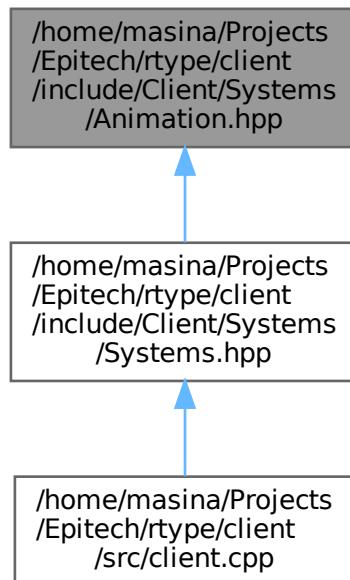
```
#include "ECS/Component.hpp"
#include "ECS/Registry.hpp"
```

#include "Interfaces/IRenderer.hpp"

Include dependency graph for Animation.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [cli::AnimationSystem](#)

Namespaces

- namespace [cli](#)

8.28 Animation.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/IRenderer.hpp"
00012
00013 namespace cli
00014 {
00015
00016     class AnimationSystem final : public eng::ASystem
00017     {
00018         public:
00019             explicit AnimationSystem(eng::IRenderer & /* renderer */) {}
00020             ~AnimationSystem() override = default;
00021
00022             AnimationSystem(const AnimationSystem &) = delete;
00023             AnimationSystem &operator=(const AnimationSystem &) = delete;
00024             AnimationSystem(AnimationSystem &&) = delete;
00025             AnimationSystem &operator=(AnimationSystem &&) = delete;
00026
00027             void update(ecs::Registry &registry, float dt) override
00028             {
00029                 for (auto &[entity, animation] : registry.getAll<ecs::Animation>())
00030                 {
00031                     animation.current_time += dt;
00032
00033                     if (animation.current_time >= animation.frame_duration)
00034                     {
00035                         animation.current_time = 0.0f;
00036                         animation.current_frame = (animation.current_frame + 1) % animation.total_frames;
00037                     }
00038
00039                     auto *rect = registry.getComponent<ecs::Rect>(entity);
00040
00041                     if (rect)
00042                     {
00043                         // Calculer la position du frame dans la spritesheet
00044                         int frame_x = (animation.current_frame % animation.frames_per_row) * animation.frame_width;
00045                         int frame_y = (animation.current_frame / animation.frames_per_row) * animation.frame_height;
00046                         if (rect->pos_x != static_cast<float>(frame_x) || rect->pos_y != static_cast<float>(frame_y))
00047                         {
00048                             rect->pos_x = static_cast<float>(frame_x);
00049                             rect->pos_y = static_cast<float>(frame_y);
00050                             rect->size_x = animation.frame_width;
00051                             rect->size_y = animation.frame_height;
00052                         }
00053                     }
00054                 }
00055             }
00056
00057     }; // class AnimationSystem
00058
00059 } // namespace cli

```

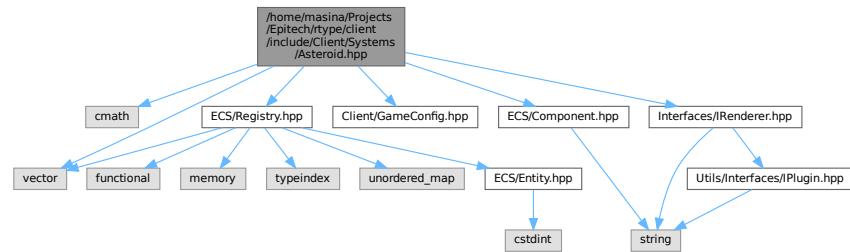
8.29 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Asteroid.hpp File Reference

```

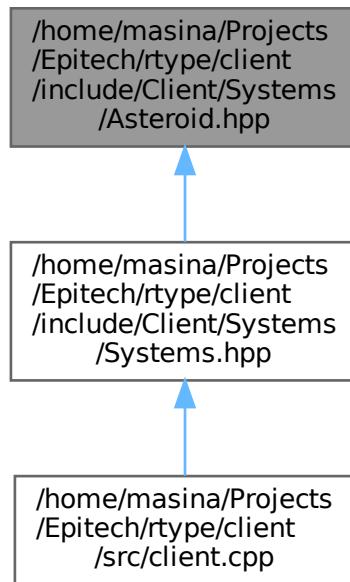
#include <cmath>
#include <vector>
#include "Client/GameConfig.hpp"
#include "ECS/Component.hpp"
#include "ECS/Registry.hpp"

```

```
#include "Interfaces/IRenderer.hpp"
Include dependency graph for Asteroid.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- class `cli::AsteroidSystem`

Namespaces

- namespace `cli`

8.30 Asteroid.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 <cmath>
00010 #include <vector>
00011
00012 #include "Client/GameConfig.hpp"
00013 #include "ECS/Component.hpp"
00014 #include "ECS/Registry.hpp"
00015 #include "Interfaces/IRenderer.hpp"
00016
00017 namespace cli
00018 {
00019
00020     class AsteroidSystem final : public eng::ISystem
00021     {
00022         public:
00023             explicit AsteroidSystem(eng::IRenderer &renderer) : m_renderer(renderer) {}
00024             ~AsteroidSystem() override = default;
00025
00026             AsteroidSystem(const AsteroidSystem &) = delete;
00027             AsteroidSystem &operator=(const AsteroidSystem &) = delete;
00028             AsteroidSystem(AsteroidSystem &&) = delete;
00029             AsteroidSystem &operator=(AsteroidSystem &&) = delete;
00030
00031             bool isEnabled() override { return true; }
00032             void setEnable(bool enable) override { (void)enable; }
00033
00034             void update(ecs::Registry &registry, float dt) override
00035             {
00036                 std::vector<ecs::Entity> asteroidsToRemove;
00037
00038                 for (auto &[entity, asteroid] : registry.getAll<ecs::Asteroid>())
00039                 {
00040                     auto *transform = registry.getComponent<ecs::Transform>(entity);
00041                     auto *velocity = registry.getComponent<ecs::Velocity>(entity);
00042                     auto *rect = registry.getComponent<ecs::Rect>(entity);
00043                     auto *texture = registry.getComponent<ecs::Texture>(entity);
00044                     auto *scale = registry.getComponent<ecs::Scale>(entity);
00045                     auto *animation = registry.getComponent<ecs::Animation>(entity);
00046
00047                     if (!transform || !velocity || !rect || !texture || !scale || !animation)
00048                         continue;
00049
00050                     animation->current_time += dt;
00051                     if (animation->current_time >= animation->frame_duration)
00052                     {
00053                         animation->current_time = 0.0f;
00054                         animation->current_frame = (animation->current_frame + 1) % animation->total_frames;
00055
00056                         const int frame_x =
00057                             animation->current_frame * static_cast<int>(GameConfig::Asteroid::Small::SPRITE_WIDTH);
00058                         const int frame_y = 0;
00059                         rect->pos_x = static_cast<float>(frame_x);
00060                         rect->pos_y = static_cast<float>(frame_y);
00061                     }
00062
00063                     transform->x += velocity->x * dt;
00064                     transform->y += velocity->y * dt;
00065                     transform->rotation += asteroid.rotation_speed * dt;
00066
00067                     m_renderer.createSprite(texture->id + std::to_string(entity), texture->path,
00068                                         std::round(transform->x), std::round(transform->y), scale->x, scale->y,
00069                                         static_cast<int>(rect->pos_x), static_cast<int>(rect->pos_y),
00070                                         static_cast<int>(rect->size_x), static_cast<int>(rect->size_y));
00071                     m_renderer.drawSprite(texture->id + std::to_string(entity));
00072
00073                     if (transform->x < GameConfig::Screen::REMOVE_X ||
00074                         transform->y < GameConfig::Screen::REMOVE_MIN_Y ||
00075                         transform->y > GameConfig::Screen::REMOVE_MAX_Y)
00076                     {
00077                         asteroidsToRemove.push_back(entity);
00078                     }
00079
00080                     for (ecs::Entity entity : asteroidsToRemove)
00081                     {

```

```

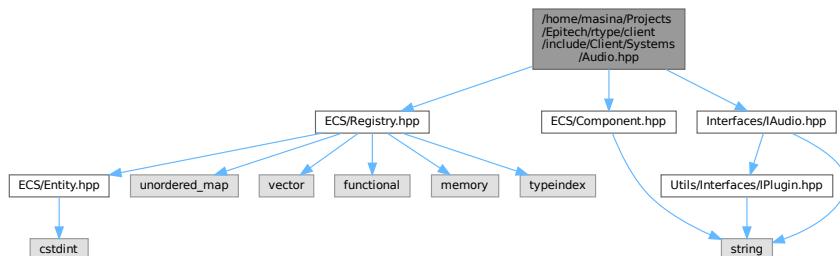
00083     if (registry.hasComponent<ecs::Asteroid>(entity))
00084         registry.removeComponent<ecs::Asteroid>(entity);
00085     if (registry.hasComponent<ecs::Transform>(entity))
00086         registry.removeComponent<ecs::Transform>(entity);
00087     if (registry.hasComponent<ecs::Velocity>(entity))
00088         registry.removeComponent<ecs::Velocity>(entity);
00089     if (registry.hasComponent<ecs::Rect>(entity))
00090         registry.removeComponent<ecs::Rect>(entity);
00091     if (registry.hasComponent<ecs::Texture>(entity))
00092         registry.removeComponent<ecs::Texture>(entity);
00093     if (registry.hasComponent<ecs::Scale>(entity))
00094         registry.removeComponent<ecs::Scale>(entity);
00095     if (registry.hasComponent<ecs::Hitbox>(entity))
00096         registry.removeComponent<ecs::Hitbox>(entity);
00097 }
00098 }
00099
00100 private:
00101     eng::IRenderer &m_renderer;
00102 }; // class AsteroidSystem
00103
00104 } // namespace cli

```

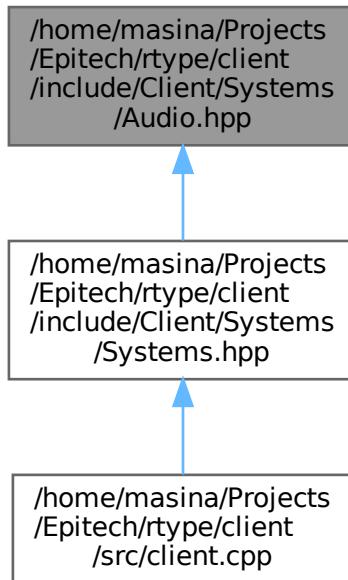
8.31 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Audio.hpp File Reference

```
#include "ECS/Component.hpp"
#include "ECS/Registry.hpp"
#include "Interfaces/IAudio.hpp"
```

Include dependency graph for Audio.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [cli::AudioSystem](#)
Class for managing entities and their components.

Namespaces

- namespace [cli](#)
- namespace [ecs](#)

8.32 Audio.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
00013 namespace cli
00014 {
00015
00016 /**
00017 /// @class AudioSystem
00018 /// @brief Class for managing entities and their components

```

```

00019 // @namespace ecs
00020 /**
00021 class AudioSystem final : public eng::ASystem
00022 {
00023     public:
00024         explicit AudioSystem(eng::IAudio &audio) : m_audio(audio) {}
00025         ~AudioSystem() override = default;
00026
00027         AudioSystem(const AudioSystem &) = delete;
00028         AudioSystem &operator=(const AudioSystem &) = delete;
00029         AudioSystem(AudioSystem &&) = delete;
00030         AudioSystem &operator=(AudioSystem &&) = delete;
00031
00032         void update(ecs::Registry &registry, float /* dt */) override
00033         {
00034             for (auto &[entity, audio] : registry.getAll<ecs::Audio>())
00035             {
00036                 m_audio.setVolume(audio.id + std::to_string(entity), audio.volume);
00037                 m_audio.setLoop(audio.id + std::to_string(entity), audio.loop);
00038                 if (audio.play && m_audio.isPlaying(audio.id + std::to_string(entity)) != eng::Status::Playing)
00039                 {
00040                     m_audio.playAudio(audio.id + std::to_string(entity));
00041                 }
00042                 else if (!audio.play &&
00043                         m_audio.isPlaying(audio.id + std::to_string(entity)) != eng::Status::Stopped)
00044                 {
00045                     m_audio.stopAudio(audio.id + std::to_string(entity));
00046                 }
00047             }
00048         }
00049
00050     private:
00051         eng::IAudio &m_audio;
00052     }; // class AudioSystem
00053
00054 } // namespace cli

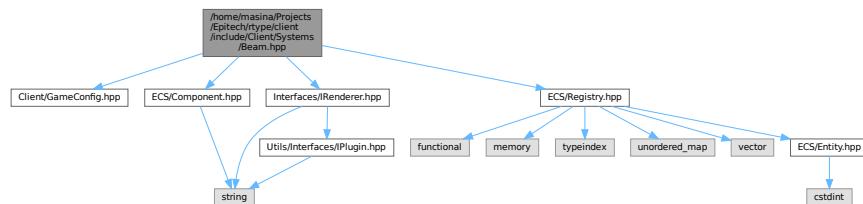
```

8.33 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Beam.hpp File Reference

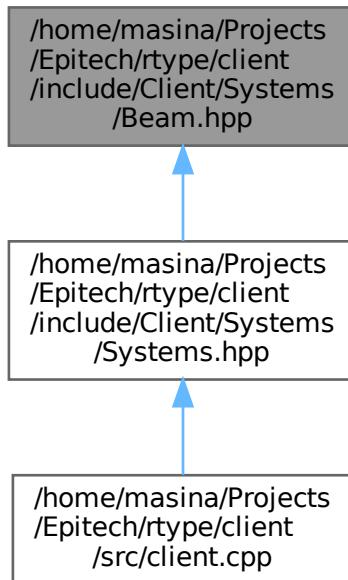
```

#include "Client/GameConfig.hpp"
#include "ECS/Component.hpp"
#include "ECS/Registry.hpp"
#include "Interfaces/IRenderer.hpp"
Include dependency graph for Beam.hpp:

```



This graph shows which files directly or indirectly include this file:



Classes

- class [cli::BeamSystem](#)

Namespaces

- namespace [cli](#)

8.34 Beam.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 "Client/GameConfig.hpp"
00010 #include "ECS/Component.hpp"
00011 #include "ECS/Registry.hpp"
00012 #include "Interfaces/IRenderer.hpp"
00013
00014 namespace cli
00015 {
00016
00017     class BeamSystem final : public eng::ASystem
00018     {
00019         public:
00020             explicit BeamSystem(eng::IRenderer &renderer) : m_renderer(renderer) {}
00021             ~BeamSystem() override = default;
  
```

```

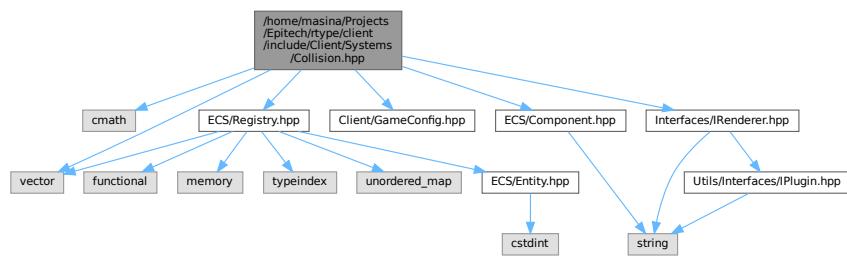
00022
00023     BeamSystem(const BeamSystem &) = delete;
00024     BeamSystem &operator=(const BeamSystem &) = delete;
00025     BeamSystem(BeamSystem &&) = delete;
00026     BeamSystem &operator=(BeamSystem &&) = delete;
00027
00028 void update(ecs::Registry &registry, float /* dt */) override
00029 {
00030     // Chercher seulement le joueur avec BeamCharge
00031     for (auto &[entity, beamCharge] : registry.getAll<ecs::BeamCharge>())
00032     {
00033         const auto *player = registry.getComponent<ecs::Player>(entity);
00034         if (!player)
00035             continue; // Seulement pour le joueur
00036
00037         const auto *transform = registry.getComponent<ecs::Transform>(entity);
00038         if (!transform)
00039             continue;
00040
00041         // Position de la barre au-dessus du joueur
00042         float barX =
00043             transform->x + GameConfig::Player::SPRITE_WIDTH / 2.0f - GameConfig::Beam::BAR_WIDTH / 2.0f;
00044         float barY = transform->y - GameConfig::Beam::BAR_HEIGHT - 10.0f; // 10 pixels au-dessus
00045
00046         // Dessiner la barre de fond avec des points (plus petite et plus fine)
00047         for (int x = 0; x < static_cast<int>(GameConfig::Beam::BAR_WIDTH); x += 3)
00048         {
00049             for (int y = 0; y < static_cast<int>(GameConfig::Beam::BAR_HEIGHT); y += 3)
00050             {
00051                 m_renderer.drawPoint(barX + x, barY + y, {r = 30, g = 30, b = 30, a = 200});
00052             }
00053         }
00054
00055         // Dessiner la barre de charge
00056         float chargeRatio = beamCharge.current_charge / beamCharge.max_charge;
00057         float chargeWidth = GameConfig::Beam::BAR_WIDTH * chargeRatio;
00058
00059         // Dessiner le seuil de 50% (ligne verticale)
00060         float thresholdX = barX + GameConfig::Beam::BAR_WIDTH * 0.5f;
00061         for (int y = 0; y < static_cast<int>(GameConfig::Beam::BAR_HEIGHT); y += 2)
00062         {
00063             m_renderer.drawPoint(
00064                 thresholdX, barY + y,
00065                 {r = 255, g = 255, b = 255, a = 150}); // Ligne blanche semi-transparente
00066         }
00067
00068         if (chargeWidth > 0)
00069         {
00070             for (int x = 0; x < static_cast<int>(chargeWidth); x += 3)
00071             {
00072                 for (int y = 0; y < static_cast<int>(GameConfig::Beam::BAR_HEIGHT); y += 3)
00073                 {
00074                     // Couleur qui change selon le niveau de charge
00075                     eng::Color chargeColor;
00076                     if (chargeRatio < 0.5f)
00077                     {
00078                         chargeColor = {
00079                             .r = 255, .g = 100, .b = 0, .a = 255}; // Orange (en dessous du seuil)
00080                     }
00081                     else if (chargeRatio < 0.8f)
00082                     {
00083                         chargeColor = {r = 255, g = 200, b = 0, a = 255}; // Jaune
00084                     }
00085                     else
00086                     {
00087                         chargeColor = {r = 0, g = 255, b = 0, a = 255}; // Vert
00088                     }
00089
00090                     m_renderer.drawPoint(barX + x, barY + y, chargeColor);
00091                 }
00092             }
00093         }
00094     }
00095 }
00096
00097 private:
00098     eng::IRenderer &m_renderer;
00099 }; // class BeamSystem
00100
00101 } // namespace cli

```

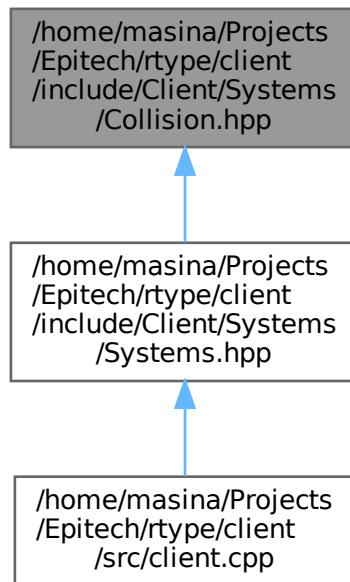
8.35 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Collision.hpp File Reference

```
#include <cmath>
#include <vector>
#include "Client/GameConfig.hpp"
#include "ECS/Component.hpp"
#include "ECS/Registry.hpp"
#include "Interfaces/IRenderer.hpp"
```

Include dependency graph for Collision.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [cli::CollisionSystem](#)

Namespaces

- namespace cli

8.36 Collision.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 <cmath>
00010 #include <vector>
00011
00012 #include "Client/GameConfig.hpp"
00013 #include "ECS/Component.hpp"
00014 #include "ECS/Registry.hpp"
00015 #include "Interfaces/IRenderer.hpp"
00016
00017 namespace cli
00018 {
00019
00020     class CollisionSystem final : public eng::ISystem
00021     {
00022         public:
00023             explicit CollisionSystem(eng::IRenderer &renderer) : m_renderer(renderer) {}
00024             ~CollisionSystem() override = default;
00025
00026             CollisionSystem(const CollisionSystem &) = delete;
00027             CollisionSystem &operator=(const CollisionSystem &) = delete;
00028             CollisionSystem(CollisionSystem &&) = delete;
00029             CollisionSystem &operator=(CollisionSystem &&) = delete;
00030
00031             bool isEnabled() override { return true; }
00032             void setEnable(bool enable) override { (void)enable; }
00033
00034             void update(ecs::Registry &registry, float dt) override
00035             {
00036                 std::vector<ecs::Entity> projectilesToRemove;
00037                 std::vector<ecs::Entity> enemiesToRemove;
00038                 std::vector<ecs::Entity> asteroidsToRemove;
00039
00040                 for (auto &[projectileEntity, projectile] : registry.getAll<ecs::Projectile>())
00041                 {
00042                     auto *projectileTransform = registry.getComponent<ecs::Transform>(projectileEntity);
00043                     auto *projectileHitbox = registry.getComponent<ecs::Hitbox>(projectileEntity);
00044                     if (!projectileTransform || !projectileHitbox)
00045                         continue;
00046
00047                     for (auto &[enemyEntity, enemy] : registry.getAll<ecs::Enemy>())
00048                     {
00049                         auto *enemyTransform = registry.getComponent<ecs::Transform>(enemyEntity);
00050                         auto *enemyHitbox = registry.getComponent<ecs::Hitbox>(enemyEntity);
00051                         if (!enemyTransform || !enemyHitbox)
00052                             continue;
00053
00054                         if (checkCircularCollision(*projectileTransform, *projectileHitbox, *enemyTransform,
00055                                         *enemyHitbox))
00056                         {
00057                             enemy.health -= projectile.damage;
00058
00059                             projectilesToRemove.push_back(projectileEntity);
00060
00061                             if (enemy.health <= 0.0f)
00062                             {
00063                                 createExplosion(registry, enemyTransform->x, enemyTransform->y);
00064                                 enemiesToRemove.push_back(enemyEntity);
00065                             }
00066                             break;
00067                         }
00068                     }
00069                 }
00070
00071                 for (auto &[projectileEntity, projectile] : registry.getAll<ecs::Projectile>())
00072                 {
00073                     auto *projectileTransform = registry.getComponent<ecs::Transform>(projectileEntity);

```

```

00074     auto *projectileHitbox = registry.getComponent<ecs::Hitbox>(projectileEntity);
00075     if (!projectileTransform || !projectileHitbox)
00076         continue;
00077
00078     for (auto &[asteroidEntity, asteroid] : registry.getAll<ecs::Asteroid>())
00079     {
00080         auto *asteroidTransform = registry.getComponent<ecs::Transform>(asteroidEntity);
00081         auto *asteroidHitbox = registry.getComponent<ecs::Hitbox>(asteroidEntity);
00082         if (!asteroidTransform || !asteroidHitbox)
00083             continue;
00084
00085         if (checkCircularCollision(*projectileTransform, *projectileHitbox, *asteroidTransform,
00086                                     *asteroidHitbox))
00087         {
00088             asteroid.health -= projectile.damage;
00089
00090             projectilesToRemove.push_back(projectileEntity);
00091
00092             if (asteroid.health <= 0.0f)
00093             {
00094                 createExplosion(registry, asteroidTransform->x, asteroidTransform->y);
00095                 asteroidsToRemove.push_back(asteroidEntity);
00096             }
00097             break;
00098         }
00099     }
00100 }
00101
00102 for (ecs::Entity entity : projectilesToRemove)
00103 {
00104     removeProjectile(registry, entity);
00105 }
00106 for (ecs::Entity entity : enemiesToRemove)
00107 {
00108     removeEnemy(registry, entity);
00109 }
00110 for (ecs::Entity entity : asteroidsToRemove)
00111 {
00112     removeAsteroid(registry, entity);
00113 }
00114 }
00115
00116 private:
00117     eng::IRenderer &m_renderer;
00118
00119     bool checkCircularCollision(const ecs::Transform &transform1, const ecs::Hitbox &hitbox1,
00120                                 const ecs::Transform &transform2, const ecs::Hitbox &hitbox2)
00121     {
00122         float dx = transform1.x - transform2.x;
00123         float dy = transform1.y - transform2.y;
00124         float distance = std::sqrt(dx * dx + dy * dy);
00125         float combinedRadius = hitbox1.radius + hitbox2.radius;
00126
00127         return distance < combinedRadius;
00128     }
00129
00130     void removeProjectile(ecs::Registry &registry, ecs::Entity entity)
00131     {
00132         if (registry.hasComponent<ecs::Projectile>(entity))
00133             registry.removeComponent<ecs::Projectile>(entity);
00134         if (registry.hasComponent<ecs::Transform>(entity))
00135             registry.removeComponent<ecs::Transform>(entity);
00136         if (registry.hasComponent<ecs::Velocity>(entity))
00137             registry.removeComponent<ecs::Velocity>(entity);
00138         if (registry.hasComponent<ecs::Rect>(entity))
00139             registry.removeComponent<ecs::Rect>(entity);
00140         if (registry.hasComponent<ecs::Texture>(entity))
00141             registry.removeComponent<ecs::Texture>(entity);
00142         if (registry.hasComponent<ecs::Scale>(entity))
00143             registry.removeComponent<ecs::Scale>(entity);
00144         if (registry.hasComponent<ecs::Animation>(entity))
00145             registry.removeComponent<ecs::Animation>(entity);
00146         if (registry.hasComponent<ecs::Hitbox>(entity))
00147             registry.removeComponent<ecs::Hitbox>(entity);
00148     }
00149
00150     void removeEnemy(ecs::Registry &registry, ecs::Entity entity)
00151     {
00152         if (registry.hasComponent<ecs::Enemy>(entity))
00153             registry.removeComponent<ecs::Enemy>(entity);
00154         if (registry.hasComponent<ecs::Transform>(entity))
00155             registry.removeComponent<ecs::Transform>(entity);
00156         if (registry.hasComponent<ecs::Velocity>(entity))
00157             registry.removeComponent<ecs::Velocity>(entity);
00158         if (registry.hasComponent<ecs::Rect>(entity))
00159             registry.removeComponent<ecs::Rect>(entity);
00160         if (registry.hasComponent<ecs::Texture>(entity))
00161     }

```

```

00161     registry.removeComponent<ecs::Texture>(entity);
00162     if (registry.hasComponent<ecs::Scale>(entity))
00163         registry.removeComponent<ecs::Scale>(entity);
00164     if (registry.hasComponent<ecs::Animation>(entity))
00165         registry.removeComponent<ecs::Animation>(entity);
00166     if (registry.hasComponent<ecs::Hitbox>(entity))
00167         registry.removeComponent<ecs::Hitbox>(entity);
00168 }
00169
00170 void removeAsteroid(ecs::Registry &registry, ecs::Entity entity)
00171 {
00172     if (registry.hasComponent<ecs::Asteroid>(entity))
00173         registry.removeComponent<ecs::Asteroid>(entity);
00174     if (registry.hasComponent<ecs::Transform>(entity))
00175         registry.removeComponent<ecs::Transform>(entity);
00176     if (registry.hasComponent<ecs::Velocity>(entity))
00177         registry.removeComponent<ecs::Velocity>(entity);
00178     if (registry.hasComponent<ecs::Rect>(entity))
00179         registry.removeComponent<ecs::Rect>(entity);
00180     if (registry.hasComponent<ecs::Texture>(entity))
00181         registry.removeComponent<ecs::Texture>(entity);
00182     if (registry.hasComponent<ecs::Scale>(entity))
00183         registry.removeComponent<ecs::Scale>(entity);
00184     if (registry.hasComponent<ecs::Hitbox>(entity))
00185         registry.removeComponent<ecs::Hitbox>(entity);
00186 }
00187
00188 void createExplosion(ecs::Registry &registry, float x, float y)
00189 {
00190     registry.createEntity()
00191         .with<ecs::Transform>("explosion_transform", x, y, 0.0f)
00192         .with<ecs::Rect>("explosion_rect", 0.0f, 0.0f,
00193             static_cast<int>(GameConfig::Explosion::SPRITE_WIDTH),
00194             static_cast<int>(GameConfig::Explosion::SPRITE_HEIGHT))
00195         .with<ecs::Scale>("explosion_scale", GameConfig::Explosion::SCALE, GameConfig::Explosion::SCALE)
00196         .with<ecs::Texture>("explosion_texture", Path::Texture::TEXTURE_EXPLOSION)
00197         .with<ecs::Explosion>("explosion", 0, GameConfig::Explosion::ANIMATION_FRAMES,
00198             GameConfig::Explosion::ANIMATION_DURATION, 0.0f,
00199             GameConfig::Explosion::SPRITE_WIDTH, GameConfig::Explosion::SPRITE_HEIGHT,
00200             GameConfig::Explosion::FRAMES_PER_ROW, GameConfig::Explosion::LIFETIME, 0.0f)
00201         .build();
00202 }
00203 }; // class CollisionSystem
00204
00205 } // namespace cli

```

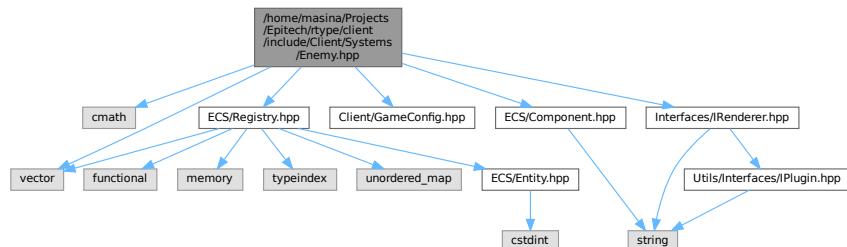
8.37 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Enemy.hpp File Reference

```

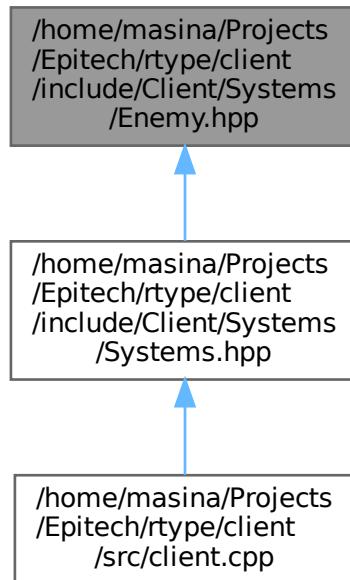
#include <cmath>
#include <vector>
#include "Client/GameConfig.hpp"
#include "ECS/Component.hpp"
#include "ECS/Registry.hpp"
#include "Interfaces/IRenderer.hpp"

```

Include dependency graph for Enemy.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [cli::EnemySystem](#)

Namespaces

- namespace [cli](#)

8.38 Enemy.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 <cmath>
00010 #include <vector>
00011
00012 #include "Client/GameConfig.hpp"
00013 #include "ECS/Component.hpp"
00014 #include "ECS/Registry.hpp"
00015 #include "Interfaces/IRenderer.hpp"
00016
00017 namespace cli
00018 {
00019
00020     class EnemySystem final : public eng::ISystem
00021     {
  
```

```

00022 public:
00023     explicit EnemySystem(eng::IRenderer &renderer) : m_renderer(renderer) {}
00024     ~EnemySystem() override = default;
00025
00026     EnemySystem(const EnemySystem &) = delete;
00027     EnemySystem &operator=(const EnemySystem &) = delete;
00028     EnemySystem(EnemySystem &&) = delete;
00029     EnemySystem &operator=(EnemySystem &&) = delete;
00030
00031     bool isEnabled() override { return true; }
00032     void setEnable(bool enable) override { (void)enable; }
00033
00034     void update(ecs::Registry &registry, float dt) override
00035     {
00036         std::vector<ecs::Entity> enemiesToRemove;
00037
00038         // Mettre à jour les ennemis existants
00039         for (auto &[entity, enemy] : registry.getAll<ecs::Enemy>())
00040         {
00041             auto *transform = registry.getComponent<ecs::Transform>(entity);
00042             auto *velocity = registry.getComponent<ecs::Velocity>(entity);
00043             auto *rect = registry.getComponent<ecs::Rect>(entity);
00044             auto *texture = registry.getComponent<ecs::Texture>(entity);
00045             auto *scale = registry.getComponent<ecs::Scale>(entity);
00046             auto *animation = registry.getComponent<ecs::Animation>(entity);
00047
00048             if (!transform || !velocity || !rect || !texture || !scale)
00049                 continue;
00050
00051             // Mettre à jour la position
00052             transform->x += velocity->x * dt;
00053             transform->y += velocity->y * dt;
00054
00055             // Animation simple
00056             if (animation)
00057             {
00058                 animation->current_time += dt;
00059                 if (animation->current_time >= animation->frame_duration)
00060                 {
00061                     animation->current_time = 0.0f;
00062                     animation->current_frame = (animation->current_frame + 1) % animation->total_frames;
00063
00064                     const int frame_x =
00065                         animation->current_frame * static_cast<int>(GameConfig::Enemy::Easy::SPRITE_WIDTH);
00066                     const int frame_y = 0;
00067                     rect->pos_x = static_cast<float>(frame_x);
00068                     rect->pos_y = static_cast<float>(frame_y);
00069                 }
00070             }
00071
00072             // Dessiner l'ennemi
00073             m_renderer.createSprite(texture->id + std::to_string(entity), texture->path,
00074                                     std::round(transform->x), std::round(transform->y), scale->x, scale->y,
00075                                     static_cast<int>(rect->pos_x), static_cast<int>(rect->pos_y),
00076                                     static_cast<int>(rect->size_x), static_cast<int>(rect->size_y));
00077             m_renderer.drawSprite(texture->id + std::to_string(entity));
00078
00079             if (transform->x < GameConfig::Screen::REMOVE_X ||
00080                 transform->y < GameConfig::Screen::REMOVE_MIN_Y ||
00081                 transform->y > GameConfig::Screen::REMOVE_MAX_Y)
00082             {
00083                 enemiesToRemove.push_back(entity);
00084             }
00085         }
00086
00087         for (ecs::Entity entity : enemiesToRemove)
00088         {
00089             if (registry.hasComponent<ecs::Enemy>(entity))
00090                 registry.removeComponent<ecs::Enemy>(entity);
00091             if (registry.hasComponent<ecs::Transform>(entity))
00092                 registry.removeComponent<ecs::Transform>(entity);
00093             if (registry.hasComponent<ecs::Velocity>(entity))
00094                 registry.removeComponent<ecs::Velocity>(entity);
00095             if (registry.hasComponent<ecs::Rect>(entity))
00096                 registry.removeComponent<ecs::Rect>(entity);
00097             if (registry.hasComponent<ecs::Texture>(entity))
00098                 registry.removeComponent<ecs::Texture>(entity);
00099             if (registry.hasComponent<ecs::Scale>(entity))
00100                 registry.removeComponent<ecs::Scale>(entity);
00101             if (registry.hasComponent<ecs::Animation>(entity))
00102                 registry.removeComponent<ecs::Animation>(entity);
00103             if (registry.hasComponent<ecs::Hitbox>(entity))
00104                 registry.removeComponent<ecs::Hitbox>(entity);
00105         }
00106     }
00107
00108 private:

```

```

00109     eng::IRenderer &m_renderer;
00110 } // class EnemySystem
00111
00112 } // namespace cli

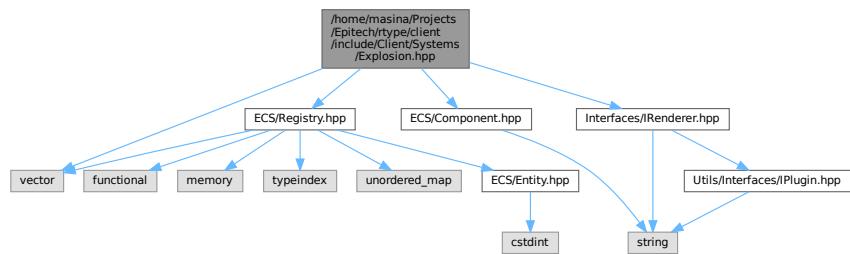
```

8.39 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Explosion.hpp File Reference

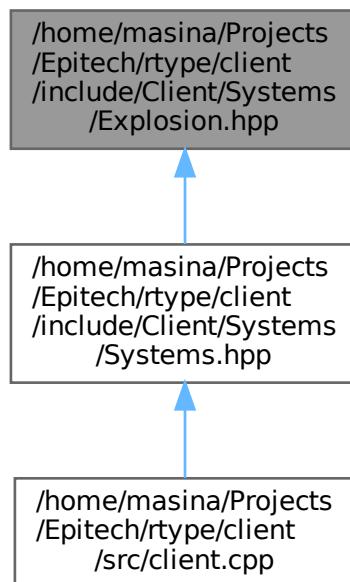
```

#include <vector>
#include "ECS/Component.hpp"
#include "ECS/Registry.hpp"
#include "Interfaces/IRenderer.hpp"
Include dependency graph for Explosion.hpp:

```



This graph shows which files directly or indirectly include this file:



Classes

- class [cli::ExplosionSystem](#)

Namespaces

- namespace [cli](#)

8.40 Explosion.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 <vector>
00010
00011 #include "ECS/Component.hpp"
00012 #include "ECS/Registry.hpp"
00013 #include "Interfaces/IRenderer.hpp"
00014
00015 namespace cli
00016 {
00017
00018     class ExplosionSystem final : public eng::ISystem
00019     {
00020         public:
00021             explicit ExplosionSystem(eng::IRenderer &renderer) : m_renderer(renderer) {}
00022             ~ExplosionSystem() override = default;
00023
00024             ExplosionSystem(const ExplosionSystem &) = delete;
00025             ExplosionSystem &operator=(const ExplosionSystem &) = delete;
00026             ExplosionSystem(ExplosionSystem &&) = delete;
00027             ExplosionSystem &operator=(ExplosionSystem &&) = delete;
00028
00029             bool isEnabled() override { return true; }
00030             void setEnable(bool enable) override { (void)enable; }
00031
00032             void update(ecs::Registry &registry, float dt) override
00033             {
00034                 std::vector<ecs::Entity> explosionsToRemove;
00035
00036                 for (auto &[entity, explosion] : registry.getAll<ecs::Explosion>())
00037                 {
00038                     auto *transform = registry.getComponent<ecs::Transform>(entity);
00039                     auto *rect = registry.getComponent<ecs::Rect>(entity);
00040                     auto *texture = registry.getComponent<ecs::Texture>(entity);
00041                     auto *scale = registry.getComponent<ecs::Scale>(entity);
00042
00043                     if (!transform || !rect || !texture || !scale)
00044                         continue;
00045
00046                     explosion.current_time += dt;
00047                     if (explosion.current_time >= explosion.frame_duration)
00048                     {
00049                         explosion.current_time = 0.0f;
00050                         explosion.current_frame = (explosion.current_frame + 1) % explosion.total_frames;
00051
00052                         int frame_x = (explosion.current_frame % explosion.frames_per_row) *
00053                             static_cast<int>(explosion.frame_width);
00054                         int frame_y = (explosion.current_frame / explosion.frames_per_row) *
00055                             static_cast<int>(explosion.frame_height);
00056
00057                         rect->pos_x = static_cast<float>(frame_x);
00058                         rect->pos_y = static_cast<float>(frame_y);
00059                 }
00060
00061                 m_renderer.createSprite(texture->id + std::to_string(entity), texture->path, transform->x,
00062                                         transform->y, scale->x, scale->y, static_cast<int>(rect->pos_x),
00063                                         static_cast<int>(rect->pos_y), static_cast<int>(rect->size_x),
00064                                         static_cast<int>(rect->size_y));
00065                 m_renderer.drawSprite(texture->id + std::to_string(entity));
00066

```

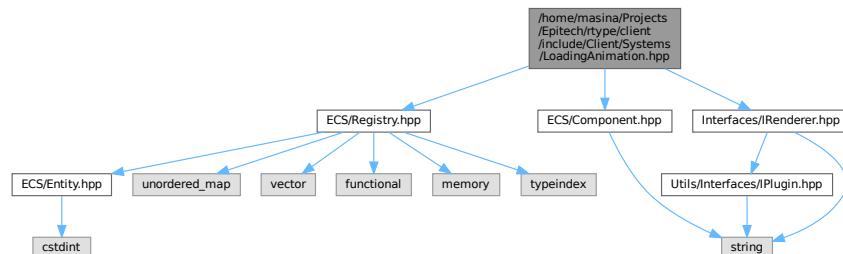
```

00067         explosion.current_lifetime += dt;
00068         if (explosion.current_lifetime >= explosion.lifetime)
00069         {
00070             explosionsToRemove.push_back(entity);
00071         }
00072     }
00073
00074     for (ecs::Entity entity : explosionsToRemove)
00075     {
00076         removeExplosion(registry, entity);
00077     }
00078 }
00079
00080 private:
00081     eng::IRenderer &m_renderer;
00082
00083     void removeExplosion(ecs::Registry &registry, ecs::Entity entity)
00084     {
00085         if (registry.hasComponent<ecs::Explosion>(entity))
00086             registry.removeComponent<ecs::Explosion>(entity);
00087         if (registry.hasComponent<ecs::Transform>(entity))
00088             registry.removeComponent<ecs::Transform>(entity);
00089         if (registry.hasComponent<ecs::Rect>(entity))
00090             registry.removeComponent<ecs::Rect>(entity);
00091         if (registry.hasComponent<ecs::Texture>(entity))
00092             registry.removeComponent<ecs::Texture>(entity);
00093         if (registry.hasComponent<ecs::Scale>(entity))
00094             registry.removeComponent<ecs::Scale>(entity);
00095     }
00096 };
00097
00098 } // namespace cli

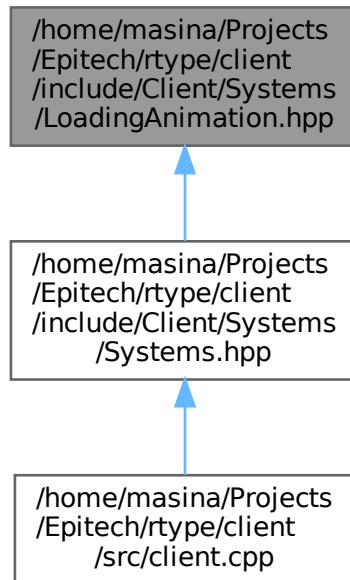
```

8.41 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/LoadingAnimation.hpp File Reference

```
#include "ECS/Component.hpp"
#include "ECS/Registry.hpp"
#include "Interfaces/IRenderer.hpp"
Include dependency graph for LoadingAnimation.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [cli::LoadingAnimationSystem](#)

Namespaces

- namespace [cli](#)

8.42 LoadingAnimation.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/IRenderer.hpp"
00012
00013 namespace cli
00014 {
00015
00016     class LoadingAnimationSystem final : public eng::ASystem
00017     {
00018         public:
00019             explicit LoadingAnimationSystem(eng::IRenderer &renderer) : m_renderer(renderer) {}
00020             ~LoadingAnimationSystem() override = default;
00021
  
```

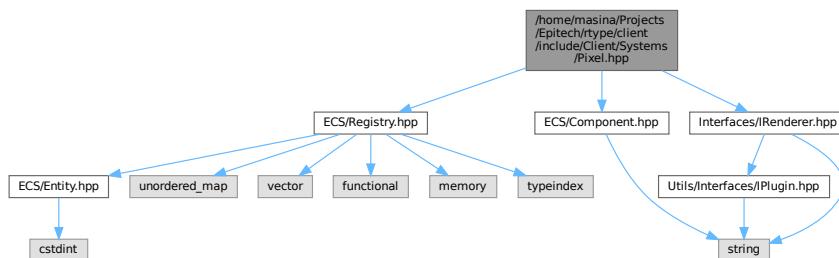
```

00022     LoadingAnimationSystem(const LoadingAnimationSystem &) = delete;
00023     LoadingAnimationSystem &operator=(const LoadingAnimationSystem &) = delete;
00024     LoadingAnimationSystem(LoadingAnimationSystem &&) = delete;
00025     LoadingAnimationSystem &operator=(LoadingAnimationSystem &&) = delete;
00026
00027     void update(ecs::Registry &registry, float dt) override
00028     {
00029         for (auto &[entity, animation] : registry.getAll<ecs::LoadingAnimation>())
00030         {
00031             const auto *transform = registry.getComponent<ecs::Transform>(entity);
00032             auto *rect = registry.getComponent<ecs::Rect>(entity);
00033             const auto *texture = registry.getComponent<ecs::Texture>(entity);
00034
00035             if (!transform || !rect || !texture)
00036                 continue;
00037
00038             // Mettre à jour l'animation
00039             animation.current_time += dt;
00040             if (animation.current_time >= animation.frame_duration)
00041             {
00042                 animation.current_time = 0.0f;
00043                 animation.current_frame = (animation.current_frame + 1) % animation.total_frames;
00044
00045                 // Mettre à jour le rectangle de texture
00046                 int frame_x = (animation.current_frame % animation.frames_per_row) *
00047                     static_cast<int>(animation.frame_width);
00048                 int frame_y = (animation.current_frame / animation.frames_per_row) *
00049                     static_cast<int>(animation.frame_height);
00050
00051                 rect->pos_x = static_cast<float>(frame_x);
00052                 rect->pos_y = static_cast<float>(frame_y);
00053             }
00054
00055             // Dessiner l'animation
00056             m_renderer.setSpriteTexture(texture->id + std::to_string(entity), texture->path);
00057             m_renderer.setSpritePosition(texture->id + std::to_string(entity), transform->x, transform->y);
00058             m_renderer.setSpriteFrame(texture->id + std::to_string(entity), static_cast<int>(rect->pos_x),
00059                                         static_cast<int>(rect->pos_y), rect->size_x, rect->size_y);
00060             m_renderer.drawSprite(texture->id + std::to_string(entity));
00061         }
00062     }
00063
00064     private:
00065         eng::IRenderer &m_renderer;
00066     }; // class LoadingAnimationSystem
00067
00068 } // namespace cli

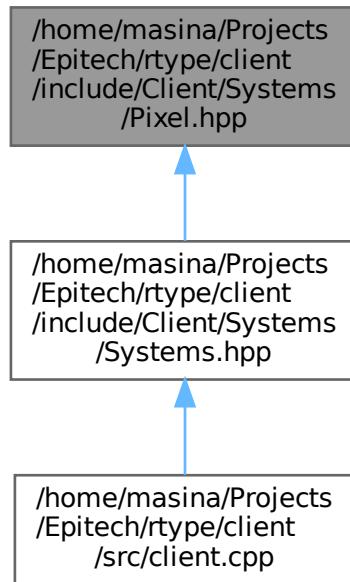
```

8.43 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Pixel.hpp File Reference

```
#include "ECS/Component.hpp"
#include "ECS/Registry.hpp"
#include "Interfaces/IRenderer.hpp"
Include dependency graph for Pixel.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [cli::PixelSystem](#)

Namespaces

- namespace [cli](#)

8.44 Pixel.hpp

[Go to the documentation of this file.](#)

```
00001 /**
00002 /**
00003 /**
00004 /**
00005 /**
00006
00007 #pragma once
00008
00009 #include "ECS/Component.hpp"
00010 #include "ECS/Registry.hpp"
00011 #include "Interfaces/IRenderer.hpp"
00012
00013 namespace cli
00014 {
00015
00016     class PixelSystem final : public eng::ASystem
00017     {
00018         public:
00019             explicit PixelSystem(eng::IRenderer &renderer) : m_renderer(renderer) {}
00020             ~PixelSystem() override = default;
00021 }
```

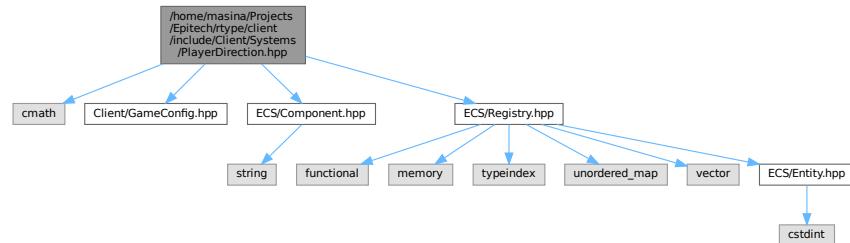
```

00022     explicit PixelSystem(const PixelSystem &) = delete;
00023     PixelSystem &operator=(const PixelSystem &) = delete;
00024     explicit PixelSystem(PixelSystem &&) = delete;
00025     PixelSystem &operator=(PixelSystem &&) = delete;
00026
00027     void update(ecs::Registry &registry, float /* dt */) override
00028     {
00029         for (auto &[entity, pixel] : registry.getAll<ecs::Pixel>())
00030         {
00031             const auto *color = registry.getComponent<ecs::Color>(entity);
00032             const auto *transform = registry.getComponent<ecs::Transform>(entity);
00033             m_renderer.drawPoint(transform->x, transform->y,
00034                                   {.r = color->r, .g = color->g, .b = color->b, .a = color->a});
00035         }
00036     }
00037
00038     private:
00039         eng::IRenderer &m_renderer;
00040     }; // class PixelSystem
00041
00042 } // namespace cli

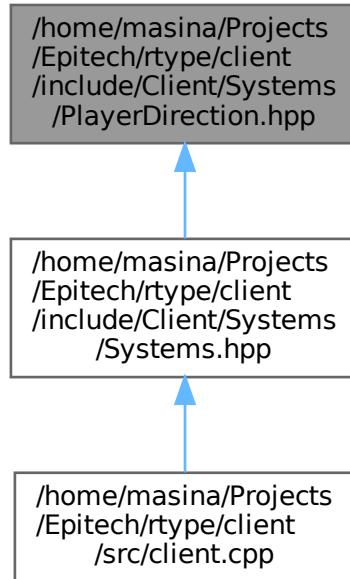
```

8.45 /home/masina/Projects/Epitech/rtype/client/include/Client/← Systems/PlayerDirection.hpp File Reference

```
#include <cmath>
#include "Client/GameConfig.hpp"
#include "ECS/Component.hpp"
#include "ECS/Registry.hpp"
Include dependency graph for PlayerDirection.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [cli::PlayerDirectionSystem](#)

Namespaces

- namespace [cli](#)

Macros

- `#define M_PI 3.14159265358979323846`

8.45.1 Macro Definition Documentation

8.45.1.1 M_PI

```
#define M_PI 3.14159265358979323846
```

Definition at line [16](#) of file [PlayerDirection.hpp](#).

Referenced by [cli::PlayerDirectionSystem::update\(\)](#).

8.46 PlayerDirection.hpp

[Go to the documentation of this file.](#)

```

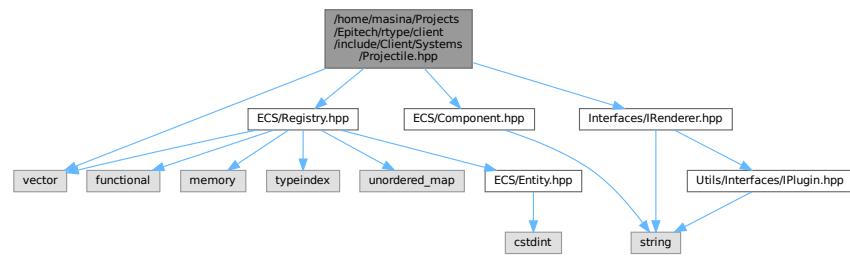
00001 /**
00002 /// @file Systems.hpp
00003 /// @brief This file contains the system definitions
00004 /// @namespace cli
00005 /**
00006
00007 #pragma once
00008
00009 #include <cmath>
00010
00011 #include "Client/GameConfig.hpp"
00012 #include "ECS/Component.hpp"
00013 #include "ECS/Registry.hpp"
00014
00015 #ifndef M_PI
00016 #define M_PI 3.14159265358979323846
00017 #endif
00018
00019 namespace cli
00020 {
00021
00022     class PlayerDirectionSystem final : public eng::ASystem
00023     {
00024         public:
00025             explicit PlayerDirectionSystem() = default;
00026             ~PlayerDirectionSystem() override = default;
00027
00028             PlayerDirectionSystem(const PlayerDirectionSystem &) = delete;
00029             PlayerDirectionSystem &operator=(const PlayerDirectionSystem &) = delete;
00030             PlayerDirectionSystem(PlayerDirectionSystem &&) = delete;
00031             PlayerDirectionSystem &operator=(PlayerDirectionSystem &&) = delete;
00032
00033             void update(ecs::Registry &registry, float /* dt */) override
00034             {
00035                 for (auto &[entity, player] : registry.getAll<ecs::Player>())
00036                 {
00037                     const auto *velocity = registry.getComponent<ecs::Velocity>(entity);
00038                     auto *rect = registry.getComponent<ecs::Rect>(entity);
00039
00040                     if (velocity && rect)
00041                     {
00042                         int frame = 0;
00043                         float angle = std::atan2(velocity->y, velocity->x);
00044                         if (std::abs(velocity->x) < 0.1f && std::abs(velocity->y) < 0.1f)
00045                         {
00046                             return;
00047                         }
00048                         if (angle < 0)
00049                             angle += 2.0f * static_cast<float>(M_PI);
00050                         if (angle >= 0 && angle < M_PI / 4)
00051                             frame = 0; // Droite
00052                         else if (angle >= M_PI / 4 && angle < 3 * M_PI / 4)
00053                             frame = 1; // Haut
00054                         else if (angle >= 3 * M_PI / 4 && angle < 5 * M_PI / 4)
00055                             frame = 2; // Gauche
00056                         else if (angle >= 5 * M_PI / 4 && angle < 7 * M_PI / 4)
00057                             frame = 3; // Bas
00058                         else
00059                             frame = 4; // Droite (retour)
00060                         int frame_width = static_cast<int>(GameConfig::Player::SPRITE_WIDTH);
00061                         int frame_height = static_cast<int>(GameConfig::Player::SPRITE_HEIGHT);
00062                         int frames_per_row = GameConfig::Player::FRAMES_PER_ROW;
00063                         int frame_x = (frame % frames_per_row) * frame_width;
00064                         int frame_y = (frame / frames_per_row) * frame_height;
00065
00066                         rect->pos_x = static_cast<float>(frame_x);
00067                         rect->pos_y = static_cast<float>(frame_y);
00068                         rect->size_x = frame_width;
00069                         rect->size_y = frame_height;
00070                     }
00071                 }
00072             }
00073     }; // class PlayerDirectionSystem
00074
00075 } // namespace cli

```

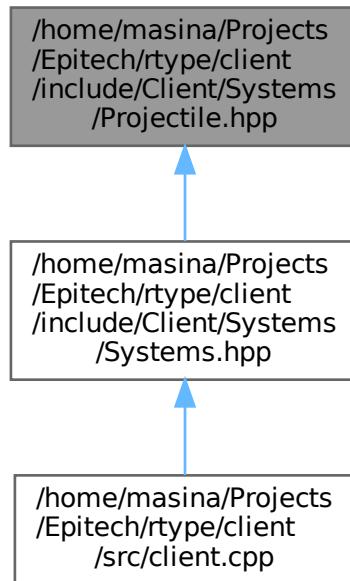
8.47 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Projectile.hpp File Reference

```
#include <vector>
#include "ECS/Component.hpp"
#include "ECS/Registry.hpp"
#include "Interfaces/IRenderer.hpp"
```

Include dependency graph for Projectile.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [cli::ProjectileSystem](#)

Namespaces

- namespace `cli`

8.48 Projectile.hpp

[Go to the documentation of this file.](#)

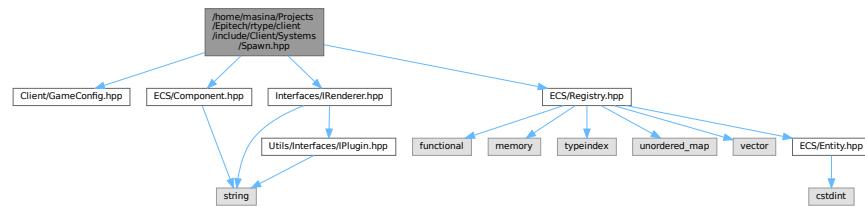
```

00001 /**
00002 /// @file Systems.hpp
00003 /// @brief This file contains the system definitions
00004 /// @namespace cli
00005 /**
00006
00007 #pragma once
00008
00009 #include <vector>
00010
00011 #include "ECS/Component.hpp"
00012 #include "ECS/Registry.hpp"
00013 #include "Interfaces/IRenderer.hpp"
00014
00015 namespace cli
00016 {
00017
00018     class ProjectileSystem final : public eng::ASystem
00019     {
00020         public:
00021             explicit ProjectileSystem(eng::IRenderer & /* renderer */) {}
00022             ~ProjectileSystem() override = default;
00023
00024             ProjectileSystem(const ProjectileSystem &) = delete;
00025             ProjectileSystem &operator=(const ProjectileSystem &) = delete;
00026             ProjectileSystem(ProjectileSystem &&) = delete;
00027             ProjectileSystem &operator=(ProjectileSystem &&) = delete;
00028
00029             void update(ecs::Registry &registry, float dt) override
00030             {
00031                 std::vector<ecs::Entity> entitiesToRemove;
00032
00033                 for (auto &[entity, projectile] : registry.getAll<ecs::Projectile>())
00034                 {
00035                     projectile.current_lifetime += dt;
00036                     if (projectile.current_lifetime >= projectile.lifetime)
00037                     {
00038                         entitiesToRemove.push_back(entity);
00039                         continue;
00040                     }
00041                     auto *transform = registry.getComponent<ecs::Transform>(entity);
00042                     auto *velocity = registry.getComponent<ecs::Velocity>(entity);
00043
00044                     if (transform && velocity)
00045                     {
00046                         transform->x += velocity->x * dt;
00047                         transform->y += velocity->y * dt;
00048                     }
00049                 }
00050
00051                 for (const auto &entity : entitiesToRemove)
00052                 {
00053                     if (registry.hasComponent<ecs::Projectile>(entity))
00054                         registry.removeComponent<ecs::Projectile>(entity);
00055                     if (registry.hasComponent<ecs::Transform>(entity))
00056                         registry.removeComponent<ecs::Transform>(entity);
00057                     if (registry.hasComponent<ecs::Velocity>(entity))
00058                         registry.removeComponent<ecs::Velocity>(entity);
00059                     if (registry.hasComponent<ecs::Rect>(entity))
00060                         registry.removeComponent<ecs::Rect>(entity);
00061                     if (registry.hasComponent<ecs::Scale>(entity))
00062                         registry.removeComponent<ecs::Scale>(entity);
00063                     if (registry.hasComponent<ecs::Texture>(entity))
00064                         registry.removeComponent<ecs::Texture>(entity);
00065                     if (registry.hasComponent<ecs::Animation>(entity))
00066                         registry.removeComponent<ecs::Animation>(entity);
00067                 }
00068             }
00069
00070     }; // class ProjectileSystem
00071 } // namespace cli

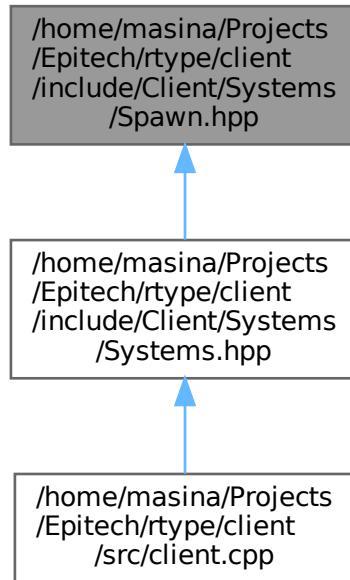
```

8.49 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Spawn.hpp File Reference

```
#include "Client/GameConfig.hpp"
#include "ECS/Component.hpp"
#include "ECS/Registry.hpp"
#include "Interfaces/IRenderer.hpp"
Include dependency graph for Spawn.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [cli::SpawnSystem](#)

Namespaces

- namespace [cli](#)

8.50 Spawn.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 "Client/GameConfig.hpp"
00010 #include "ECS/Component.hpp"
00011 #include "ECS/Registry.hpp"
00012 #include "Interfaces/IRenderer.hpp"
00013
00014 namespace cli
00015 {
00016
00017     class SpawnSystem final : public eng::ISystem
00018     {
00019         public:
00020             explicit SpawnSystem(eng::IRenderer &renderer) : m_renderer(renderer) {}
00021             ~SpawnSystem() override = default;
00022
00023             SpawnSystem(const SpawnSystem &) = delete;
00024             SpawnSystem &operator=(const SpawnSystem &) = delete;
00025             SpawnSystem(SpawnSystem &&) = delete;
00026             SpawnSystem &operator=(SpawnSystem &&) = delete;
00027
00028             bool isEnabled() override { return true; }
00029             void setEnable(bool enable) override { (void)enable; }
00030
00031             void update(ecs::Registry &registry, float dt) override
00032             {
00033                 m_enemySpawnTimer += dt;
00034                 m_asteroidSpawnTimer += dt;
00035                 m_waveTimer += dt;
00036
00037                 if (m_enemySpawnTimer >= GameConfig::Enemy::Easy::SPAWN_RATE)
00038                 {
00039                     spawnEnemy(registry);
00040                     m_enemySpawnTimer = 0.0f;
00041                 }
00042
00043                 if (m_waveTimer >= 10.0f)
00044                 {
00045                     spawnWave(registry);
00046                     m_waveTimer = 0.0f;
00047                 }
00048
00049                 if (m_asteroidSpawnTimer >= GameConfig::Asteroid::Small::SPAWN_RATE)
00050                 {
00051                     spawnAsteroid(registry, ecs::Asteroid::SMALL);
00052                     m_asteroidSpawnTimer = 0.0f;
00053                 }
00054             }
00055
00056         private:
00057             eng::IRenderer &m_renderer;
00058             float m_enemySpawnTimer = 0.0f;
00059             float m_asteroidSpawnTimer = 0.0f;
00060             float m_waveTimer = 0.0f;
00061
00062             void spawnEnemy(ecs::Registry &registry)
00063             {
00064                 float x = GameConfig::Screen::SPAWN_X;
00065                 float y = static_cast<float>(
00066                     GameConfig::Screen::MIN_Y +
00067                     (std::rand() % static_cast<int>(GameConfig::Screen::MAX_Y - GameConfig::Screen::MIN_Y)));
00068
00069                 registry.createEntity()
00070                     .with<ecs::Transform>"("enemy_transform", x, y, 0.0f)
00071                     .with<ecs::Velocity>"("enemy_velocity", -GameConfig::Enemy::Easy::SPEED, 0.0f)
00072                     .with<ecs::Rect>"("enemy_rect", 0.0f, 0.0f, static_cast<int>(GameConfig::Enemy::Easy::SPRITE_WIDTH),
00073                                     static_cast<int>(GameConfig::Enemy::Easy::SPRITE_HEIGHT))
00074                     .with<ecs::Scale>"("enemy_scale", GameConfig::Enemy::Easy::SCALE, GameConfig::Enemy::Easy::SCALE)
00075                     .with<ecs::Texture>"("enemy_texture", Path::Texture::TEXTURE_ENEMY_EASY)
00076                     .with<ecs::Animation>"("enemy_animation", 0, GameConfig::Enemy::Easy::ANIMATION_FRAMES,
00077                                     GameConfig::Enemy::Easy::ANIMATION_DURATION, 0.0f,
00078                                     static_cast<int>(GameConfig::Enemy::Easy::SPRITE_WIDTH),
00079                                     static_cast<int>(GameConfig::Enemy::Easy::SPRITE_HEIGHT),
00080                                     static_cast<int>(GameConfig::Enemy::Easy::FRAMES_PER_ROW))
00081                     .with<ecs::Enemy>"("enemy", GameConfig::Enemy::Easy::HEALTH, GameConfig::Enemy::Easy::HEALTH,
00082                                     GameConfig::Enemy::Easy::DAMAGE, GameConfig::Enemy::Easy::SPEED, 0.0f,

```

```

00083             GameConfig::Enemy::Easy::SHOOT_COOLDOWN)
00084         .with<ecs::Hitbox>("enemy_hitbox", GameConfig::Hitbox::ENEMY_RADIUS)
00085         .build();
00086     }
00087
00088     void spawnAsteroid(ecs::Registry &registry, ecs::Asteroid::Size size)
00089     {
00090         float x = GameConfig::Screen::SPAWN_X;
00091         float y = static_cast<float>(
00092             GameConfig::Screen::MIN_Y +
00093             (std::rand() % static_cast<int>(GameConfig::Screen::MAX_Y - GameConfig::Screen::MIN_Y)));
00094
00095         registry.createEntity()
00096             .with<ecs::Transform>("asteroid_transform", x, y, 0.0f)
00097             .with<ecs::Velocity>("asteroid_velocity", -GameConfig::Asteroid::Small::SPEED, 0.0f)
00098             .with<ecs::Rect>("asteroid_rect", 0.0f, 0.0f,
00099                 static_cast<int>(GameConfig::Asteroid::Small::SPRITE_WIDTH),
00100                 static_cast<int>(GameConfig::Asteroid::Small::SPRITE_HEIGHT))
00101             .with<ecs::Scale>("asteroid_scale", GameConfig::Asteroid::Small::SCALE,
00102                 GameConfig::Asteroid::Small::SCALE)
00103             .with<ecs::Texture>("asteroid_texture", Path::Texture::TEXTURE_ASTEROID)
00104             .with<ecs::Animation>("asteroid_animation", 0, GameConfig::Asteroid::Small::ANIMATION_FRAMES,
00105                 GameConfig::Asteroid::Small::ANIMATION_DURATION, 0.0f,
00106                 static_cast<int>(GameConfig::Asteroid::Small::SPRITE_WIDTH),
00107                 static_cast<int>(GameConfig::Asteroid::Small::SPRITE_HEIGHT),
00108                 static_cast<int>(GameConfig::Asteroid::Small::FRAMES_PER_ROW))
00109             .with<ecs::Asteroid>("asteroid", size, GameConfig::Asteroid::Small::ROTATION_SPEED,
00110                 GameConfig::Asteroid::Small::HEALTH)
00111             .with<ecs::Hitbox>("asteroid_hitbox", GameConfig::Hitbox::ASTEROID_SMALL_RADIUS)
00112             .build();
00113     }
00114
00115     void spawnWave(ecs::Registry &registry)
00116     {
00117         int waveSize = 5 + (std::rand() % 4);
00118
00119         for (int i = 0; i < waveSize; ++i)
00120         {
00121             float x = GameConfig::Screen::SPAWN_X + (i * 100.0f);
00122             float y = static_cast<float>(
00123                 GameConfig::Screen::MIN_Y +
00124                 (std::rand() % static_cast<int>(GameConfig::Screen::MAX_Y - GameConfig::Screen::MIN_Y)));
00125
00126             registry.createEntity()
00127                 .with<ecs::Transform>("enemy_wave_transform", x, y, 0.0f)
00128                 .with<ecs::Velocity>("enemy_wave_velocity", -GameConfig::Enemy::Easy::SPEED, 0.0f)
00129                 .with<ecs::Rect>("enemy_wave_rect", 0.0f, 0.0f,
00130                     static_cast<int>(GameConfig::Enemy::Easy::SPRITE_WIDTH),
00131                     static_cast<int>(GameConfig::Enemy::Easy::SPRITE_HEIGHT))
00132                 .with<ecs::Scale>("enemy_wave_scale", GameConfig::Enemy::Easy::SCALE,
00133                     GameConfig::Enemy::Easy::SCALE)
00134                 .with<ecs::Texture>("enemy_wave_texture", Path::Texture::TEXTURE_ENEMY_EASY)
00135                 .with<ecs::Animation>("enemy_wave_animation", 0,
00136                     static_cast<int>(GameConfig::Enemy::Easy::ANIMATION_FRAMES),
00137                     GameConfig::Enemy::Easy::ANIMATION_DURATION, 0.0f,
00138                     static_cast<int>(GameConfig::Enemy::Easy::SPRITE_WIDTH),
00139                     static_cast<int>(GameConfig::Enemy::Easy::SPRITE_HEIGHT),
00140                     static_cast<int>(GameConfig::Enemy::Easy::FRAMES_PER_ROW))
00141                 .with<ecs::Enemy>("enemy_wave", GameConfig::Enemy::Easy::HEALTH,
00142                     GameConfig::Enemy::Easy::HEALTH, GameConfig::Enemy::Easy::DAMAGE,
00143                     GameConfig::Enemy::Easy::SPEED, 0.0f, GameConfig::Enemy::Easy::SHOOT_COOLDOWN)
00144                 .with<ecs::Hitbox>("enemy_wave_hitbox", GameConfig::Hitbox::ENEMY_RADIUS)
00145                 .build();
00146         }
00147     }
00148 }; // class SpawnSystem
00149
00150 } // namespace cli

```

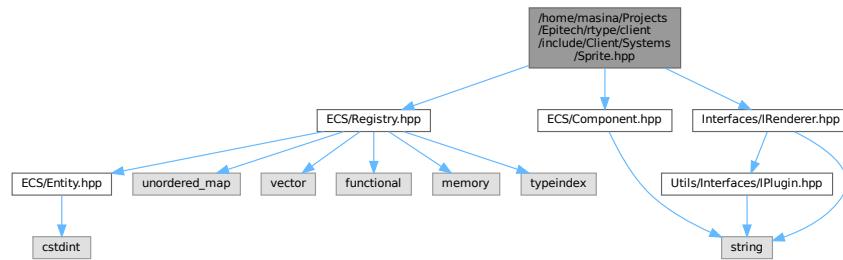
8.51 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Sprite.hpp File Reference

```

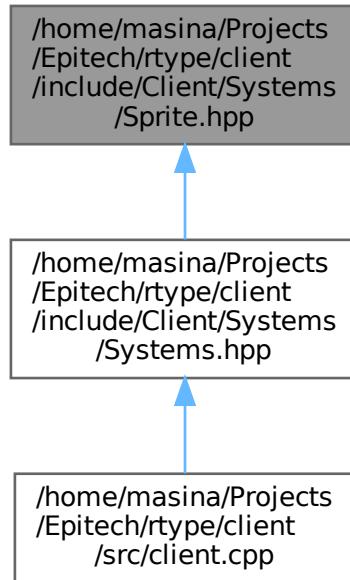
#include "ECS/Component.hpp"
#include "ECS/Registry.hpp"
#include "Interfaces/IRenderer.hpp"

```

Include dependency graph for Sprite.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [cli::SpriteSystem](#)

Namespaces

- namespace [cli](#)

8.52 Sprite.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/IRenderer.hpp"
00012
00013 namespace cli
00014 {
00015
00016     class SpriteSystem final : public eng::ASystem
00017     {
00018         public:
00019             explicit SpriteSystem(eng::IRenderer &renderer) : m_renderer(renderer) {}
00020             ~SpriteSystem() override = default;
00021
00022             SpriteSystem(const SpriteSystem &) = delete;
00023             SpriteSystem &operator=(const SpriteSystem &) = delete;
00024             SpriteSystem(SpriteSystem &&) = delete;
00025             SpriteSystem &operator=(SpriteSystem &&) = delete;
00026
00027             void update(ecs::Registry &registry, float /* dt */) override
00028             {
00029                 for (auto &[entity, sprite] : registry.getAll<ecs::Texture>())
00030                 {
00031                     const auto *transform = registry.getComponent<ecs::Transform>(entity);
00032                     const auto *rect = registry.getComponent<ecs::Rect>(entity);
00033
00034                     const float x = (transform != nullptr) ? transform->x : 0.F;
00035                     const float y = (transform != nullptr) ? transform->y : 0.F;
00036                     m_renderer.setSpriteTexture(sprite.id + std::to_string(entity), sprite.path);
00037                     m_renderer.setSpritePosition(sprite.id + std::to_string(entity), x, y);
00038
00039                     if (rect)
00040                     {
00041                         m_renderer.setSpriteFrame(sprite.id + std::to_string(entity), static_cast<int>(rect->pos_x),
00042                                         static_cast<int>(rect->pos_y), rect->size_x, rect->size_y);
00043                     }
00044
00045                     m_renderer.drawSprite(sprite.id + std::to_string(entity));
00046                 }
00047             }
00048
00049         private:
00050             eng::IRenderer &m_renderer;
00051     }; // class SpriteSystem
00052
00053 } // namespace cli

```

8.53 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Systems.hpp File Reference

This file contains the system definitions.

```

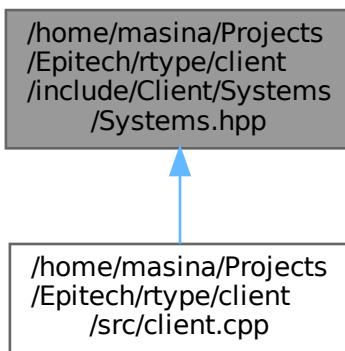
#include "Client/Systems/Animation.hpp"
#include "Client/Systems/Asteroid.hpp"
#include "Client/Systems/Audio.hpp"
#include "Client/Systems/Beam.hpp"
#include "Client/Systems/Collision.hpp"
#include "Client/Systems/Enemy.hpp"
#include "Client/Systems/Explosion.hpp"
#include "Client/Systems>LoadingAnimation.hpp"
#include "Client/Systems/Pixel.hpp"
#include "Client/Systems/PlayerDirection.hpp"

```

```
#include "Client/Systems/Projectile.hpp"
#include "Client/Systems/Spawn.hpp"
#include "Client/Systems/Sprite.hpp"
#include "Client/Systems/Text.hpp"
Include dependency graph for Systems.hpp:
```



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace `cli`

8.53.1 Detailed Description

This file contains the system definitions.

This file contains the TextSystem definitions.

Definition in file [Systems.hpp](#).

8.54 Systems.hpp

[Go to the documentation of this file.](#)

```
00001 /**
00002 /**
00003 /**
00004 /**
00005 /**
```

```

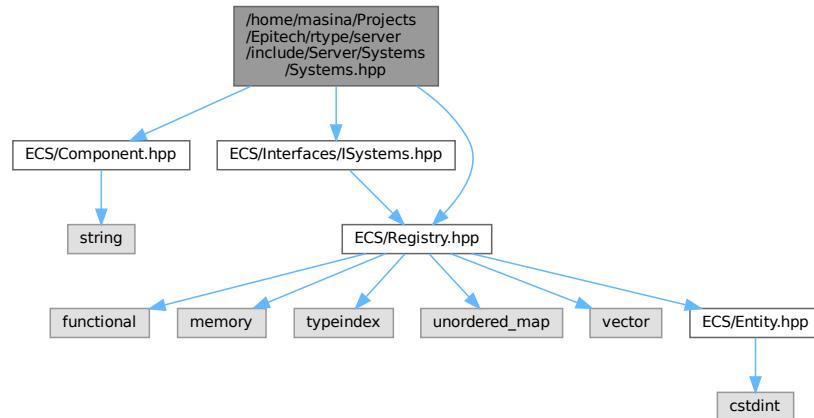
00006
00007 #pragma once
00008
00009 #include "Client/Systems/Animation.hpp"
00010 #include "Client/Systems/Asteroid.hpp"
00011 #include "Client/Systems/Audio.hpp"
00012 #include "Client/Systems/Beam.hpp"
00013 #include "Client/Systems/Collision.hpp"
00014 #include "Client/Systems/Enemy.hpp"
00015 #include "Client/Systems/Explosion.hpp"
00016 #include "Client/Systems>LoadingAnimation.hpp"
00017 #include "Client/Systems/Pixel.hpp"
00018 #include "Client/Systems/PlayerDirection.hpp"
00019 #include "Client/Systems/Projectile.hpp"
00020 #include "Client/Systems/Spawn.hpp"
00021 #include "Client/Systems/Sprite.hpp"
00022 #include "Client/Systems/Text.hpp"

```

8.55 /home/masina/Projects/Epitech/rtype/server/include/Server/Systems/Systems.hpp File Reference

This file contains the system definitions.

```
#include "ECS/Component.hpp"
#include "ECS/Interfaces/ISystems.hpp"
#include "ECS/Registry.hpp"
Include dependency graph for Systems.hpp:
```



Namespaces

- namespace **srv**

8.55.1 Detailed Description

This file contains the system definitions.

Definition in file [Systems.hpp](#).

8.56 Systems.hpp

[Go to the documentation of this file.](#)

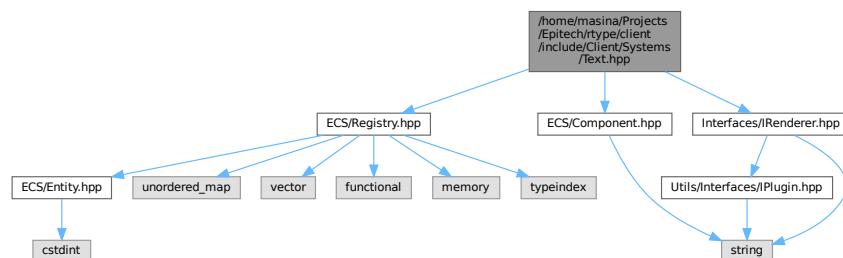
```

00001 /**
00002 /// @file Systems.hpp
00003 /// @brief This file contains the system definitions
00004 /// @namespace srv
00005 /**
00006
00007 #pragma once
00008
00009 #include "ECS/Component.hpp"
00010 #include "ECS/Interfaces/ISystems.hpp"
00011 #include "ECS/Registry.hpp"
00012
00013 namespace srv
00014 {
00015
00016 } // namespace srv

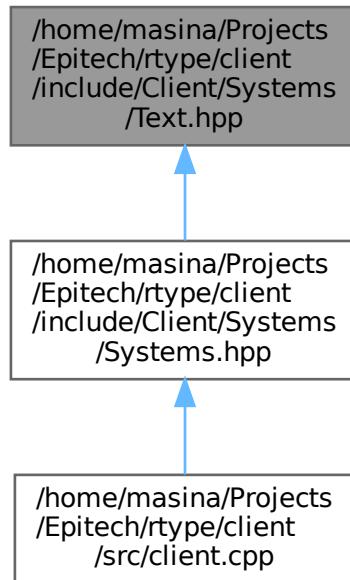
```

8.57 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Text.hpp File Reference

```
#include "ECS/Component.hpp"
#include "ECS/Registry.hpp"
#include "Interfaces/IRenderer.hpp"
Include dependency graph for Text.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- class `cli::TextSystem`
Class for managing entities and their components.

Namespaces

- namespace `cli`
- namespace `ecs`

8.58 Text.hpp

[Go to the documentation of this file.](#)

```
00001 /**
00002 /// @file Systems.hpp
00003 /// @brief This file contains the TextSystem definitions
00004 /// @namespace cli
00005 /**
00006
00007 #pragma once
00008
00009 #include "ECS/Component.hpp"
00010 #include "ECS/Registry.hpp"
00011 #include "Interfaces/IRenderer.hpp"
00012
00013 namespace cli
00014 {
00015
00016 /**
00017 /// @class TextSystem
00018 /// @brief Class for managing entities and their components
```

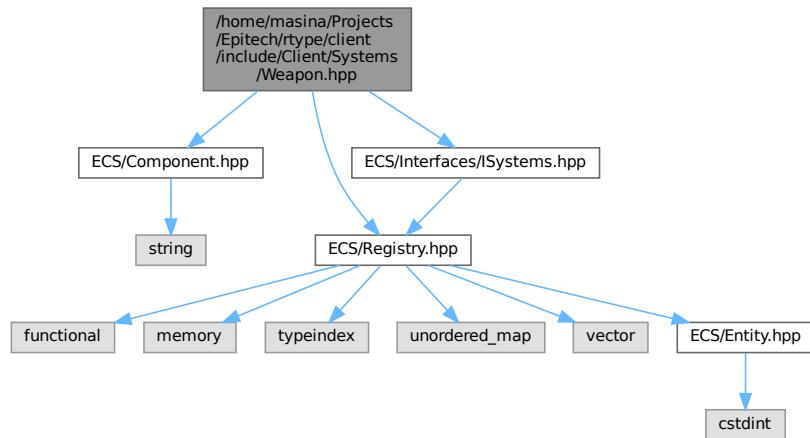
```

00019 // @namespace ecs
00020 /**
00021 class TextSystem final : public eng::ASystem
00022 {
00023     public:
00024         explicit TextSystem(eng::IRenderer &renderer) : m_renderer(renderer) {}
00025         ~TextSystem() override = default;
00026
00027         TextSystem(const TextSystem &) = delete;
00028         TextSystem &operator=(const TextSystem &) = delete;
00029         TextSystem(TextSystem &&) = delete;
00030         TextSystem &operator=(TextSystem &&) = delete;
00031
00032         void update(ecs::Registry &registry, float /* dt */) override
00033         {
00034
00035             for (auto &[entity, text] : registry.getAll<ecs::Text>())
00036             {
00037                 const auto *transform = registry.getComponent<ecs::Transform>(entity);
00038                 const auto *color = registry.getComponent<ecs::Color>(entity);
00039
00040                 const float x = (transform != nullptr) ? transform->x : 0.F;
00041                 const float y = (transform != nullptr) ? transform->y : 0.F;
00042
00043                 const std::uint8_t r = color ? color->r : 255u;
00044                 const std::uint8_t g = color ? color->g : 255u;
00045                 const std::uint8_t b = color ? color->b : 255u;
00046                 const std::uint8_t a = color ? color->a : 255u;
00047
00048                 m_renderer.setTextContent(text.id, text.content);
00049                 m_renderer.setTextPosition(text.id, x, y);
00050                 m_renderer.setTextColor(text.id, { .r = r, .g = g, .b = b, .a = a });
00051                 m_renderer.drawText(text.id);
00052             }
00053         }
00054
00055     private:
00056         eng::IRenderer &m_renderer;
00057     }; // class TextSystem
00058
00059 } // namespace cli

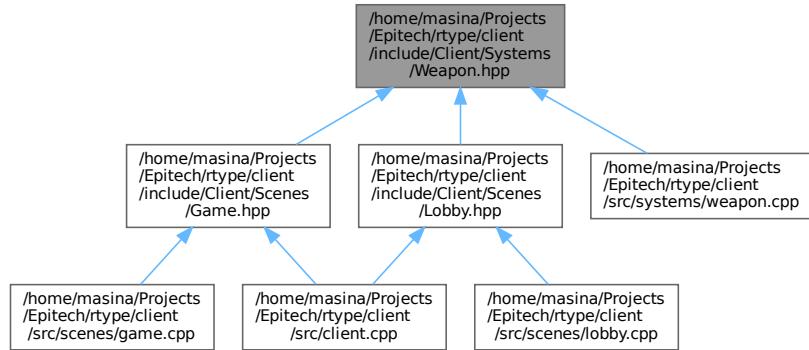
```

8.59 /home/masina/Projects/Epitech/rtype/client/include/Client/Systems/Weapon.hpp File Reference

```
#include "ECS/Component.hpp"
#include "ECS/Registry.hpp"
#include "ECS/Interfaces/ISystems.hpp"
Include dependency graph for Weapon.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- class [cli::WeaponSystem](#)
Manages weapon firing and charging.

Namespaces

- namespace [cli](#)

8.60 Weapon.hpp

[Go to the documentation of this file.](#)

```

00001 /**
00002 /**
00003 /**
00004 /**
00005 /**
00006
00007 #pragma once
00008
00009 #include "ECS/Component.hpp"
00010 #include "ECS/Registry.hpp"
00011
00012 #include "ECS/Interfaces/ISystems.hpp"
00013
00014 namespace cli
00015 {
00016 /**
00017 /**
00018 /**
00019 /**
00020 /**
00021 class WeaponSystem final : public eng::ASystem
00022 {
00023     public:
00024         WeaponSystem() = default;
00025         ~WeaponSystem() override = default;
00026
00027         WeaponSystem(const WeaponSystem &) = delete;
00028         WeaponSystem &operator=(const WeaponSystem &) = delete;
00029         WeaponSystem(WeaponSystem &&) = delete;
00030         WeaponSystem &operator=(WeaponSystem &&) = delete;
00031
00032 /**
00033 /**
00034 /**
        
```

```

00035     /// @param dt Delta time
00036     /// @param spacePressed Whether space is pressed
00037     ///
00038     void update(ecs::Registry &registry, float dt, bool spacePressed);
00039     void update(ecs::Registry &registry, float dt) override;
00040
00041     ///
00042     /// @brief Reset weapon state
00043     ///
00044     void reset();
00045
00046 private:
00047     float m_fireCooldown = 0.0f;
00048     bool m_isCharging = false;
00049
00050     ///
00051     /// @brief Try to fire basic projectile
00052     /// @param registry The ECS registry
00053     /// @param x X position
00054     /// @param y Y position
00055     /// @return True if fired successfully
00056     ///
00057     bool tryFireBasic(ecs::Registry &registry, float x, float y);
00058
00059     ///
00060     /// @brief Try to fire supercharged projectile
00061     /// @param registry The ECS registry
00062     /// @param x X position
00063     /// @param y Y position
00064     /// @return True if fired successfully
00065     ///
00066     bool tryFireSupercharged(ecs::Registry &registry, float x, float y);
00067
00068     ///
00069     /// @brief Show loading animation in front of the player
00070     /// @param registry The ECS registry
00071     /// @param playerEntity The player entity
00072     /// @param playerTransform The player transform
00073     ///
00074     void showLoadingAnimation(ecs::Registry &registry, ecs::Entity playerEntity,
00075                             const ecs::Transform *playerTransform);
00076
00077     ///
00078     /// @brief Hide loading animation
00079     /// @param registry The ECS registry
00080     /// @param playerEntity The player entity
00081     ///
00082     void hideLoadingAnimation(ecs::Registry &registry, ecs::Entity playerEntity);
00083 }; // class WeaponSystem
00084 } // namespace cli

```

8.61 /home/masina/Projects/Epitech/rtype/client/src/argsHandler.cpp

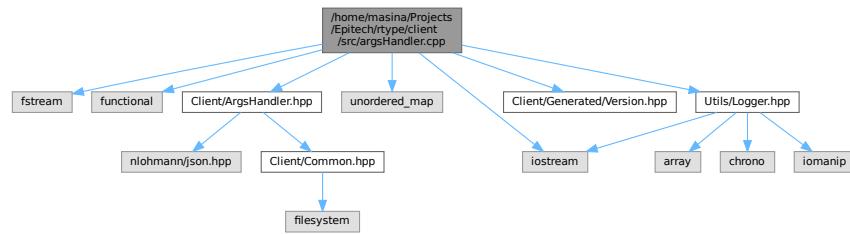
File Reference

```

#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`

8.61.1 Macro Definition Documentation

8.61.1.1 APP_EXTENSION

```
#define APP_EXTENSION ""
```

Definition at line 9 of file [argsHandler.cpp](#).

8.61.2 Variable Documentation

8.61.2.1 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\(\)](#).

8.61.2.2 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()`.

8.62 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                                     "  --help, -h      Show this help message\n"
00019                                     "  --version, -v   Show version information\n"
00020                                     "  --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         const auto &p = j["plugins"];
00058         if (p.contains("audio"))
00059         {
00060             cfg.audio_lib_path = p["audio"];
```

```

00061     }
00062     if (p.contains("network"))
00063     {
00064         cfg.network_lib_path = p["network"];
00065     }
00066     if (p.contains("renderer"))
00067     {
00068         cfg.renderer_lib_path = p["renderer"];
00069     }
00070     const auto &c = j["client"];
00071     if (c.contains("host"))
00072     {
00073         cfg.host = c["host"];
00074     }
00075     if (c.contains("port"))
00076     {
00077         cfg.port = c["port"];
00078     }
00079 }
00080 return cfg;
00081 }
00082
00083 cli::ArgsConfig cli::ArgsHandler::ParseArgs(const int argc, const char *const argv[])
00084 {
00085     if (argc <= 1)
00086     {
00087         return {};
00088     }
00089
00090     using ArgHandler = std::function<void(const char *arg)>;
00091     std::unordered_map<std::string_view, ArgHandler> handlers;
00092     ArgsConfig config{};
00093     for (const auto *const opt : {"-h", "--help"})
00094     {
00095         handlers[opt] = [&config](const char *)
00096         {
00097             std::cout « HELP_MESSAGE;
00098             config.exit = true;
00099         };
00100    }
00101    for (const auto *const opt : {"-v", "--version"})
00102    {
00103        handlers[opt] = [&config](const char *)
00104        {
00105            std::cout « VERSION_MESSAGE;
00106            config.exit = true;
00107        };
00108    }
00109
00110    for (const auto *const opt : {"-c", "--config"})
00111    {
00112        handlers[opt] = [&config](const char *arg)
00113        {
00114            if (!arg)
00115            {
00116                throw std::runtime_error("Missing config file argument");
00117            }
00118            config = ArgsConfig::fromFile(arg);
00119            utl::Logger::log("Loaded config from file: " + std::string(arg), utl::LogLevel::INFO);
00120            std::cout « "\tWidth: " « config.width « '\n'
00121            « "\tHeight: " « config.height « '\n'
00122            « "\tFrameLimit: " « config.frameLimit « '\n'
00123            « "\tFullscreen: " « (config.fullscreen ? "true" : "false") « '\n';
00124        };
00125    }
00126
00127    const std::string_view key = argv[1];
00128    const char *argValue = (argc > 2) ? argv[2] : nullptr;
00129
00130    if (const auto it = handlers.find(key); it != handlers.end())
00131    {
00132        it->second(argValue);
00133        return config;
00134    }
00135
00136    throw std::runtime_error("Unknown argument: " + std::string(key));
00137 }
00138
00139 cli::EnvConfig cli::ArgsHandler::ParseEnv(const char *const env[])
00140 {
00141     (void)env; // Currently unused
00142     return {};
00143 }

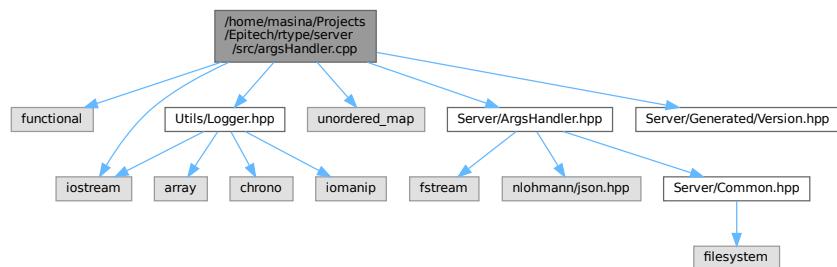
```

8.63 /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`

8.63.1 Macro Definition Documentation

8.63.1.1 APP_EXTENSION

```
#define APP_EXTENSION ""
```

Definition at line 9 of file [argsHandler.cpp](#).

8.63.2 Variable Documentation

8.63.2.1 HELP_MESSAGE

```
std::string_view HELP_MESSAGE [static], [constexpr]
```

Initial value:

```
= "Usage: " PROJECT_NAME APP_EXTENSION " [options]\n\n"
"Options:\n"
"\\"--help, -h      Show this help message\n"
"\\"--version, -v   Show version information\n"
"\\"--config, -c    Specify path to config file\n"
```

Definition at line 16 of file [argsHandler.cpp](#).

8.63.2.2 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`.

8.64 argsHandler.cpp

[Go to the documentation of this file.](#)

```
00001 #include <functional>
00002 #include <iostream>
00003 #include <unordered_map>
00004
00005 #ifdef _WIN32
00006 #include <windows.h>
00007 #define APP_EXTENSION ".exe"
00008 #else
00009 #define APP_EXTENSION ""
00010 #endif
00011
00012 #include "Server/ArgsHandler.hpp"
00013 #include "Server/Generated/Version.hpp"
00014 #include "Utils/Logger.hpp"
00015
00016 static constexpr std::string_view HELP_MESSAGE = "Usage: " PROJECT_NAME APP_EXTENSION " [options]\n\n"
00017             "Options:\n"
00018             "\t--help, -h      Show this help message\n"
00019             "\t--version, -v    Show version information\n"
00020             "\t--config, -c     Specify path to config file\n";
00021 static constexpr std::string_view VERSION_MESSAGE = PROJECT_NAME " version " PROJECT_VERSION "\n"
00022             "Build type: " BUILD_TYPE "\n"
00023             "Git tag: " GIT_TAG "\n"
00024             "Git commit hash: " GIT_COMMIT_HASH "\n";
00025
00026 srv::ArgsConfig srv::ArgsConfig::fromFile(const std::string &path)
00027 {
00028     ArgsConfig cfg;
00029     std::ifstream file(path);
00030     if (!file.is_open())
00031     {
00032         throw std::runtime_error("Cannot open config file: " + path);
00033     }
00034
00035     json j;
00036     file >> j;
00037
00038     if (j.contains("host"))
00039     {
00040         cfg.host = j["host"];
00041     }
00042     if (j.contains("port"))
00043     {
00044         cfg.port = j["port"];
00045     }
00046     if (const auto &p = j["plugins"]; p.contains("network"))
00047     {
00048         cfg.network_lib_path = p["network"];
00049     }
00050     return cfg;
00051 }
00052
00053 srv::ArgsConfig srv::ArgsHandler::ParseArgs(const int argc, const char *const argv[])
00054 {
00055     if (argc <= 1)
00056     {
00057         return {};
00058     }
00059
00060     using ArgHandler = std::function<void(const char *arg)>;
00061     std::unordered_map<std::string_view, ArgHandler> handlers;
00062     ArgsConfig config{};
00063     for (const auto *const opt : {"-h", "--help"})
```

```

00064  {
00065      handlers[opt] = [&config](const char *)
00066      {
00067          std::cout << HELP_MESSAGE;
00068          config.exit = true;
00069      };
00070  }
00071  for (const auto *const opt : {"-v", "--version"})
00072  {
00073      handlers[opt] = [&config](const char *)
00074      {
00075          std::cout << VERSION_MESSAGE;
00076          config.exit = true;
00077      };
00078  }
00079
00080  for (const auto *const opt : {"-c", "--config"})
00081  {
00082      handlers[opt] = [&config](const char *arg)
00083      {
00084          if (!arg)
00085          {
00086              throw std::runtime_error("Missing config file argument");
00087          }
00088          config = ArgsConfig::fromFile(arg);
00089          utl::Logger::log("Loaded config from file: " + std::string(arg), utl::LogLevel::INFO);
00090          std::cout << "\tHost: " << config.host << '\n' << "\tPort: " << config.port << '\n';
00091      };
00092  }
00093
00094  const std::string_view key = argv[1];
00095  const char *argValue = (argc > 2) ? argv[2] : nullptr;
00096
00097  if (const auto it = handlers.find(key); it != handlers.end())
00098  {
00099      it->second(argValue);
00100      return config;
00101  }
00102
00103  throw std::runtime_error("Unknown argument: " + std::string(key));
00104 }
00105
00106 srv::EnvConfig srv::ArgsHandler::ParseEnv(const char *const env[]) { return {}; }

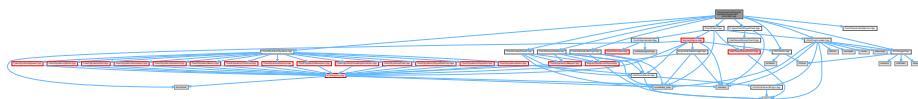
```

8.65 /home/masina/Projects/Epitech/rtype/client/src/client.cpp File Reference

```

#include "Client/Client.hpp"
#include "Client/Generated/Version.hpp"
#include "Client/Scenes/Game.hpp"
#include "Client/Scenes/Lobby.hpp"
#include "Client/Scenes/Settings.hpp"
#include "Client/Systems/Systems.hpp"
#include "R-TypeClient/RTypeClient.hpp"
#include "Utils/Clock.hpp"
#include "Utils/Logger.hpp"
#include "Utils/PluginLoader.hpp"
Include dependency graph for client.cpp:

```



Variables

- static constexpr eng::Color DARK = {.r = 0U, .g = 0U, .b = 0U, .a = 255U}

8.65.1 Variable Documentation

8.65.1.1 DARK

`eng::Color DARK = {.r = 0U, .g = 0U, .b = 0U, .a = 255U}` [static], [constexpr]

Definition at line 12 of file [client.cpp](#).

Referenced by [cli::Client::run\(\)](#).

8.66 client.cpp

[Go to the documentation of this file.](#)

```

00001 #include "Client/Client.hpp"
00002 #include "Client/Generated/Version.hpp"
00003 #include "Client/Scenes/Game.hpp"
00004 #include "Client/Scenes/Lobby.hpp"
00005 #include "Client/Scenes/Settings.hpp"
00006 #include "Client/Systems/Systems.hpp"
00007 #include "R-TypeClient/RTypeClient.hpp"
00008 #include "Utils/Clock.hpp"
00009 #include "Utils/Logger.hpp"
00010 #include "Utils/PluginLoader.hpp"
00011
00012 static constexpr eng::Color DARK = {.r = 0U, .g = 0U, .b = 0U, .a = 255U};
00013
00014 cli::Client::Client(const ArgsConfig &cfg)
00015 {
00016     utl::Logger::log("PROJECT INFO:", utl::LogLevel::INFO);
00017     std::cout << "\tName: " PROJECT_NAME "\n"
00018         << "\tVersion: " PROJECT_VERSION "\n"
00019         << "\tBuild type: " BUILD_TYPE "\n"
00020         << "\tGit tag: " GIT_TAG "\n"
00021         << "\tGit commit hash: " GIT_COMMIT_HASH "\n";
00022
00023     m_pluginLoader = std::make_unique<utl::PluginLoader>();
00024     m_engine = std::make_unique<eng::Engine>(
00025         [this, cfg]() { return m_pluginLoader->loadPlugin<eng::IAudio>(!cfg.audio_lib_path.empty() ? cfg.audio_lib_path
00026 : Path::Plugin::PLUGIN_AUDIO_SFML.string()); },
00027         [this, cfg]() { return m_pluginLoader->loadPlugin<eng::INetworkClient>(!cfg.network_lib_path.empty() ?
00028 cfg.network_lib_path : Path::Plugin::PLUGIN_NETWORK_ASIO_CLIENT.string()); },
00029         [this, cfg]() { return m_pluginLoader->loadPlugin<eng::IRenderer>(!cfg.renderer_lib_path.empty() ?
00030 cfg.renderer_lib_path : Path::Plugin::PLUGIN_RENDERER_SFML.string()); });
00031     // m_game = std::make_unique<gme::RTypeClient>();
00032     m_engine->getRenderer()->createWindow("R-Type Client", cfg.height, cfg.width, cfg.frameLimit, cfg.fullscreen);
00033
00034     m_engine->addSystem(std::make_unique<AnimationSystem>(*m_engine->getRenderer()));
00035     m_engine->addSystem(std::make_unique<AudioSystem>(*m_engine->getAudio()));
00036     // m_engine->addSystem(std::make_unique<SpawnSystem>(*m_engine->getRenderer())); TODO(bobis33): only in
00037     game
00038     m_engine->addSystem(std::make_unique<AsteroidSystem>(*m_engine->getRenderer()));
00039     m_engine->addSystem(std::make_unique<BeamSystem>(*m_engine->getRenderer()));
00040     m_engine->addSystem(std::make_unique<CollisionSystem>(*m_engine->getRenderer()));
00041     m_engine->addSystem(std::make_unique<EnemySystem>(*m_engine->getRenderer()));
00042     m_engine->addSystem(std::make_unique<ExplosionSystem>(*m_engine->getRenderer()));
00043     m_engine->addSystem(std::make_unique<LoadingAnimationSystem>(*m_engine->getRenderer()));
00044     m_engine->addSystem(std::make_unique<PixelSystem>(*m_engine->getRenderer()));
00045     m_engine->addSystem(std::make_unique<PlayerDirectionSystem>());
00046     m_engine->addSystem(std::make_unique<ProjectileSystem>(*m_engine->getRenderer()));
00047     m_engine->addSystem(std::make_unique<SpawnSystem>(*m_engine->getRenderer()));
00048     m_engine->addSystem(std::make_unique<SpriteSystem>(*m_engine->getRenderer()));
00049     m_engine->addSystem(std::make_unique<TextSystem>(*m_engine->getRenderer()));
00050
00051     auto lobby = std::make_unique<Lobby>(m_engine->getRenderer(), m_engine->getAudio());
00052     auto game = std::make_unique<Game>(m_engine->getRenderer(), m_engine->getAudio());
00053     auto settings = std::make_unique<Settings>(m_engine->getRenderer(), m_engine->getAudio());
00054     const auto lobbyId = lobby->getId();
00055     const auto gameId = game->getId();
00056     const auto settingsId = settings->getId();
00057     lobby->onOptionSelected = [this, gameId, settingsId](const std::string &option)
00058     {
00059         if (option == "Solo")
00060         {
00061             m_engine->getSceneManager()->switchToScene(gameId);
00062         }
00063     };

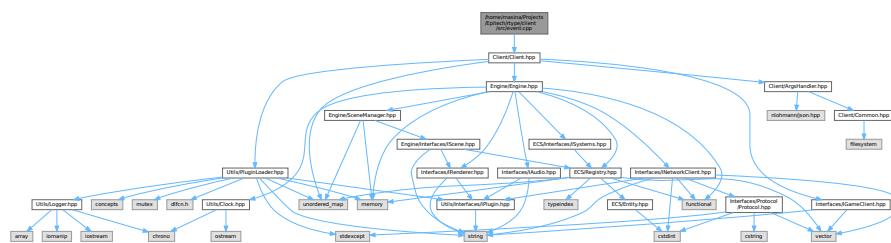
```

```
00060     else if (option == "Multi")
00061     {
00062         m_engine->getSceneManager()->switchToScene(gameId);
00063     }
00064     else if (option == "Settings")
00065     {
00066         m_engine->getSceneManager()->switchToScene(settingsId);
00067     }
00068 };
00069 settings->onLeave = [this, lobbyId]() { m_engine->getSceneManager()->switchToScene(lobbyId); };
00070 m_engine->getSceneManager()->addScene(std::move(lobby));
00071 m_engine->getSceneManager()->addScene(std::move(game));
00072 m_engine->getSceneManager()->addScene(std::move(settings));
00073 m_engine->getSceneManager()->switchToScene(lobbyId);
00074 }
00075
00076 void cli::Client::run()
00077 {
00078     eng::Event event;
00079
00080     while (m_engine->getState() == eng::State::RUN && m_engine->getRenderer()->windowIsOpen())
00081     {
00082         const float delta = m_engine->getClock()->getDeltaSeconds();
00083
00084         m_engine->getClock()->restart();
00085         m_engine->getSceneManager()->getCurrentScene()->update(delta, m_engine->getRenderer()->getWindowSize());
00086         handleEvents(event);
00087         m_engine->render(m_engine->getSceneManager()->getCurrentScene()->getRegistry(), DARK, delta);
00088     }
00089     m_engine->stop();
00090 }
```

8.67 /home/masina/Projects/Epitech/rtype/client/src/event.cpp File Reference

```
#include "Client/Client.hpp"
```

Include dependency graph for event.cpp:



8.68 event.cpp

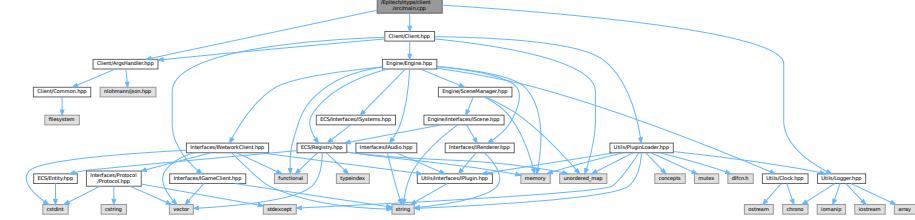
[Go to the documentation of this file.](#)

```
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->getRenderer()->closeWindow();
00021     //   m_engine->setState(eng::State::STOP);
00022     //}
00023     // else
00024     //{
00025     m_keysPressed[event.key] = true;
00026     //}
00027     break;
00028
00029 case eng::EventType::KeyReleased:
00030     m_keysPressed[event.key] = false;
00031     break;
00032
00033 default:
00034     break;
00035 }
00036 }
00037 }
```

8.69 /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)

8.69.1 Function Documentation

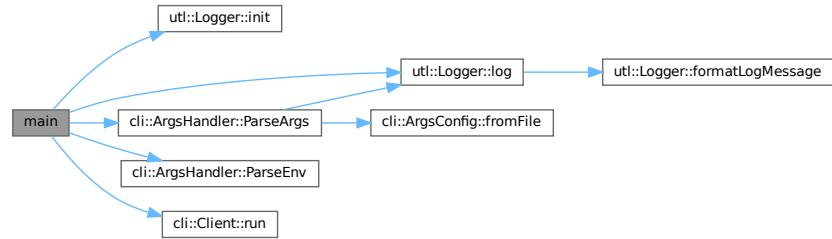
8.69.1.1 main()

```
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:



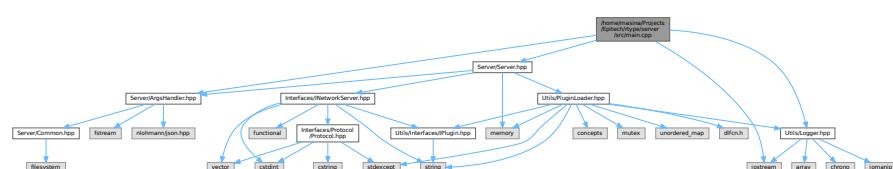
8.70 main.cpp

[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 }
```

8.71 /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[])

8.71.1 Function Documentation

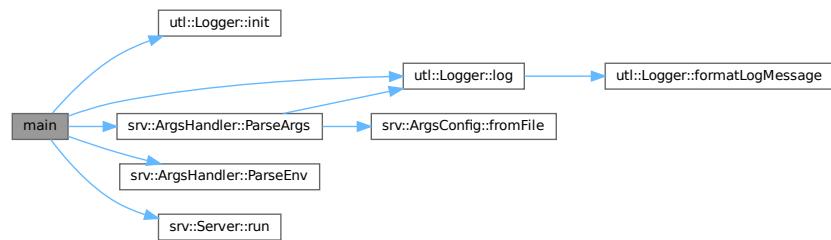
8.71.1.1 main()

```
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\(\)](#), [srv::Server::run\(\)](#), and [utl::WARNING](#).

Here is the call graph for this function:



8.72 main.cpp

[Go to the documentation of this file.](#)

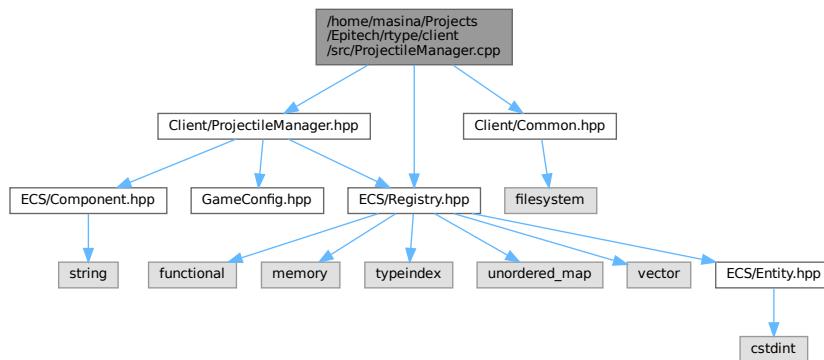
```
00001 #include <iostream>
00002
00003 #include "Server/ArgsHandler.hpp"
00004 #include "Server/Server.hpp"
00005 #include "Utils/Logger.hpp"
00006
00007 int main(const int argc, const char *const argv[], const char *const env[])
00008 {
00009     utl::Logger::init();
00010     try
00011     {
00012         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         const srv::Server server(argsConf);
00019         server.run();
00020     }
00021     catch (const std::exception &e)
00022     {
00023         utl::Logger::log(std::string("Exception: ") + e.what(), utl::LogLevel::WARNING);
00024         return EXIT_FAILURE;
00025     }
00026     catch (...)
00027     {
00028         utl::Logger::log("Unknown exception", utl::LogLevel::WARNING);
00029         return EXIT_FAILURE;
00030     }
00031     return EXIT_SUCCESS;
00032 }
```

8.73 /home/masina/Projects/Epitech/rtype/client/src/ProjectileManager.cpp File Reference

Implementation of ProjectileManager.

```
#include "Client/ProjectileManager.hpp"
#include "Client/Common.hpp"
#include "ECS/Registry.hpp"
```

Include dependency graph for ProjectileManager.cpp:



Namespaces

- namespace `cli`

8.73.1 Detailed Description

Implementation of ProjectileManager.

Definition in file [ProjectileManager.cpp](#).

8.74 ProjectileManager.cpp

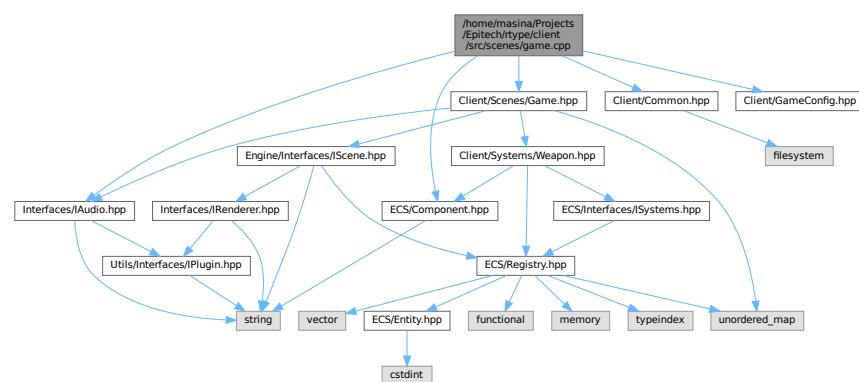
[Go to the documentation of this file.](#)

```
00001 /**
00002 /**
00003 /**
00004 /**
00005 /**
00006
00007 #include "Client/ProjectileManager.hpp"
00008 #include "Client/Common.hpp"
00009 #include "ECS/Registry.hpp"
00010
00011 namespace cli
00012 {
00013     ecs::Entity ProjectileManager::createBasicProjectile(ecs::Registry &registry, float x, float y, float velocityX,
00014                                                         float velocityY)
00015     {
00016         return createProjectile(registry, ecs::Projectile::BASIC, x, y, velocityX, velocityY);
00017     }
00018 }
```

```
00019 ecs::Entity ProjectileManager::createSuperchargedProjectile(ecs::Registry &registry, float x, float y,  
00020                                     float velocityX, float velocityY)  
00021 {  
00022     return createProjectile(registry, ecs::Projectile::SUPERCHARGED, x, y, velocityX, velocityY);  
00023 }  
00024  
00025 ecs::Entity ProjectileManager::createProjectile(ecs::Registry &registry, ecs::Projectile::Type type, float x,  
00026                                     float y, float velocityX, float velocityY)  
00027 {  
00028     using namespace GameConfig::Projectile;  
00029  
00030     if (type == ecs::Projectile::BASIC)  
00031     {  
00032         return registry.createEntity()  
00033             .with<ecs::Transform>("projectile_transform", x, y, 0.F)  
00034             .with<ecs::Velocity>("projectile_velocity", velocityX, velocityY)  
00035             .with<ecs::Rect>("projectile_rect", 0.F, 0.F, static_cast<int>(Basic::SPRITE_WIDTH),  
00036                             static_cast<int>(Basic::SPRITE_HEIGHT))  
00037             .with<ecs::Scale>("projectile_scale", Basic::SCALE, Basic::SCALE)  
00038             .with<ecs::Texture>("projectile_texture", Path::Texture::TEXTURE_SHOOT)  
00039             .with<ecs::Projectile>("projectile", type, Basic::DAMAGE, Basic::LIFETIME, 0.0f)  
00040             .with<ecs::Hitbox>("projectile_hitbox", GameConfig::Hitbox::PROJECTILE_BASIC_RADIUS)  
00041             .build();  
00042     }  
00043     else  
00044     {  
00045         return registry.createEntity()  
00046             .with<ecs::Transform>("projectile_transform", x, y, 0.F)  
00047             .with<ecs::Velocity>("projectile_velocity", velocityX, velocityY)  
00048             .with<ecs::Rect>("projectile_rect", 0.F, 0.F, static_cast<int>(Supercharged::SPRITE_WIDTH),  
00049                             static_cast<int>(Supercharged::SPRITE_HEIGHT))  
00050             .with<ecs::Scale>("projectile_scale", Supercharged::SCALE, Supercharged::SCALE)  
00051             .with<ecs::Texture>("projectile_texture", Path::Texture::TEXTURE_SHOOT_CHARGED)  
00052             .with<ecs::Projectile>("projectile", type, Supercharged::DAMAGE, Supercharged::LIFETIME, 0.0f)  
00053             .with<ecs::Animation>("projectile_animation", 0, Supercharged::ANIMATION_FRAMES,  
00054                             Supercharged::ANIMATION_DURATION, 0.0f,  
00055                             static_cast<int>(Supercharged::SPRITE_WIDTH),  
00056                             static_cast<int>(Supercharged::SPRITE_HEIGHT), Supercharged::ANIMATION_FRAMES)  
00057             .with<ecs::Hitbox>("projectile_hitbox", GameConfig::Hitbox::PROJECTILE_SUPERCHARGED_RADIUS)  
00058             .build();  
00059     }  
00060 } // namespace cli
```

8.75 /home/masina/Projects/Epitech/rtype/client/src/scenes/game.cpp
File Reference

```
#include "Client/Scenes/Game.hpp"
#include "Client/Common.hpp"
#include "Client/GameConfig.hpp"
#include "ECS/Component.hpp"
#include "Interfaces/IAudio.hpp"
Include dependency graph for game.cpp:
```



Variables

- static constexpr eng::Color **WHITE** = {.r = 255U, .g = 255U, .b = 255U, .a = 255U}
- static constexpr eng::Color **WHITE_TRANS** = {.r = 255U, .g = 255U, .b = 255U, .a = 100U}
- static constexpr eng::Color **BLUE** = {.r = 200U, .g = 200U, .b = 255U, .a = 150U}
- static constexpr eng::Color **BLUE_SECOND** = {.r = 50U, .g = 100U, .b = 200U, .a = 60U}
- static constexpr eng::Color **YELLOW** = {.r = 255U, .g = 255U, .b = 200U, .a = 200U}
- static constexpr eng::Color **PURPLE** = {.r = 100U, .g = 50U, .b = 150U, .a = 80U}
- static constexpr eng::Color **GREEN** = {.r = 200U, .g = 255U, .b = 200U, .a = 180U}

8.75.1 Variable Documentation

8.75.1.1 BLUE

`eng::Color` **BLUE** = {.r = 200U, .g = 200U, .b = 255U, .a = 150U} [static], [constexpr]

Definition at line 9 of file [game.cpp](#).

Referenced by [cli::Game::Game\(\)](#).

8.75.1.2 BLUE_SECOND

`eng::Color` **BLUE_SECOND** = {.r = 50U, .g = 100U, .b = 200U, .a = 60U} [static], [constexpr]

Definition at line 10 of file [game.cpp](#).

Referenced by [cli::Game::Game\(\)](#).

8.75.1.3 GREEN

`eng::Color` **GREEN** = {.r = 200U, .g = 255U, .b = 200U, .a = 180U} [static], [constexpr]

Definition at line 13 of file [game.cpp](#).

Referenced by [cli::Game::Game\(\)](#).

8.75.1.4 PURPLE

`eng::Color` **PURPLE** = {.r = 100U, .g = 50U, .b = 150U, .a = 80U} [static], [constexpr]

Definition at line 12 of file [game.cpp](#).

Referenced by [cli::Game::Game\(\)](#).

8.75.1.5 WHITE

`eng::Color` **WHITE** = {.r = 255U, .g = 255U, .b = 255U, .a = 255U} [static], [constexpr]

Definition at line 7 of file [game.cpp](#).

Referenced by [cli::Game::Game\(\)](#), [cli::Lobby::Lobby\(\)](#), and [cli::Settings::Settings\(\)](#).

8.75.1.6 WHITE_TRANS

```
eng::Color WHITE_TRANS = {.r = 255U, .g = 255U, .b = 255U, .a = 100U} [static], [constexpr]
```

Definition at line 8 of file [game.cpp](#).

Referenced by [cli::Game::Game\(\)](#).

8.75.1.7 YELLOW

```
eng::Color YELLOW = {.r = 255U, .g = 255U, .b = 200U, .a = 200U} [static], [constexpr]
```

Definition at line 11 of file [game.cpp](#).

Referenced by [cli::Game::Game\(\)](#).

8.76 game.cpp

[Go to the documentation of this file.](#)

```
00001 #include "Client/Scenes/Game.hpp"
00002 #include "Client/Common.hpp"
00003 #include "Client/GameConfig.hpp"
00004 #include "ECS/Component.hpp"
00005 #include "Interfaces/IAudio.hpp"
00006
00007 static constexpr eng::Color WHITE = {.r = 255U, .g = 255U, .b = 255U, .a = 255U};
00008 static constexpr eng::Color WHITE_TRANS = {.r = 255U, .g = 255U, .b = 255U, .a = 100U};
00009 static constexpr eng::Color BLUE = {.r = 200U, .g = 200U, .b = 255U, .a = 150U};
00010 static constexpr eng::Color BLUE_SECOND = {.r = 50U, .g = 100U, .b = 200U, .a = 60U};
00011 static constexpr eng::Color YELLOW = {.r = 255U, .g = 255U, .b = 200U, .a = 200U};
00012 static constexpr eng::Color PURPLE = {.r = 100U, .g = 50U, .b = 150U, .a = 80U};
00013 static constexpr eng::Color GREEN = {.r = 200U, .g = 255U, .b = 200U, .a = 180U};
00014
00015 cli::Game::Game(const std::shared_ptr<eng::IRenderer> &renderer, const std::shared_ptr<eng::IAudio> &audio)
00016 : m_audio(audio)
00017 {
00018     auto &registry = AScene::getRegistry();
00019
00020     registry.onComponentAdded(
00021         [&renderer, &audio, &registry](const ecs::Entity e, const std::type_info &type)
00022     {
00023         const auto *audioComp = registry.getComponent<ecs::Audio>(e);
00024         const auto *colorComp = registry.getComponent<ecs::Color>(e);
00025         const auto *fontComp = registry.getComponent<ecs::Font>(e);
00026         const auto *rectComp = registry.getComponent<ecs::Rect>(e);
00027         const auto *scaleComp = registry.getComponent<ecs::Scale>(e);
00028         const auto *textComp = registry.getComponent<ecs::Text>(e);
00029         const auto *textureComp = registry.getComponent<ecs::Texture>(e);
00030         const auto *transform = registry.getComponent<ecs::Transform>(e);
00031
00032         if (type == typeid(ecs::Text))
00033         {
00034             if (textComp && transform && fontComp)
00035             {
00036                 renderer->createFont(fontComp->id, fontComp->path);
00037                 renderer->createText(
00038                     {.font_name = fontComp->id,
00039                      .color = {.r = colorComp->r, .g = colorComp->g, .b = colorComp->b, .a = colorComp->a},
00040                      .content = textComp->content,
00041                      .size = textComp->font_size,
00042                      .x = transform->x,
00043                      .y = transform->y,
00044                      .name = textComp->id});
00045             }
00046         }
00047         else if (type == typeid(ecs::Texture))
00048         {
00049             const float scale_x = scaleComp ? scaleComp->x : 1.F;
00050             const float scale_y = scaleComp ? scaleComp->y : 1.F;
00051
00052             renderer->createTexture(textureComp->id, textureComp->path);
00053         }
00054     }
00055 }
```

```

00054     if (transform && textureComp)
00055     {
00056         if (rectComp)
00057         {
00058             renderer->createSprite(textureComp->id + std::to_string(e), textureComp->id, transform->x,
00059                                         transform->y, scale_x, scale_y, static_cast<int>(rectComp->pos_x),
00060                                         static_cast<int>(rectComp->pos_y), rectComp->size_x, rectComp->size_y);
00061         }
00062         else
00063         {
00064             renderer->createSprite(textureComp->id + std::to_string(e), textureComp->id, transform->x,
00065                                         transform->y);
00066         }
00067     }
00068 }
00069 else if (type == typeid(ecs::Audio))
00070 {
00071     if (audioComp)
00072     {
00073         audio->createAudio(audioComp->path, audioComp->volume, audioComp->loop,
00074                             audioComp->id + std::to_string(e));
00075     }
00076 }
00077 });
00078
00079 registry.createEntity().with<ecs::Audio>("id_audio", Path::Audio::AUDIO_TITLE, 5.F, true, true).build();
00080 registry.createEntity()
00081     .with<ecs::Font>("main_font", Path::Font::FONTS_RTYPE)
00082     .with<ecs::Transform>("transform_title", 10.F, 10.F, 0.F)
00083     .with<ecs::Color>("color_title", WHITE.r, WHITE.g, WHITE.b, WHITE.a)
00084     .with<ecs::Text>("id", std::string("RTType Client"), 50U)
00085     .build();
00086 m_fpsEntity = registry.createEntity()
00087     .with<ecs::Font>("main_font", Path::Font::FONTS_RTYPE)
00088     .with<ecs::Transform>("transform_fps", 10.F, 70.F, 0.F)
00089     .with<ecs::Color>("color_fps", WHITE.r, WHITE.g, WHITE.b, WHITE.a)
00090     .with<ecs::Text>("id_text", std::string("FPS: 0"), 20U)
00091     .build();
00092
00093 // Compteur d'ennemis
00094 m_enemyCounterEntity = registry.createEntity()
00095     .with<ecs::Font>("main_font", Path::Font::FONTS_RTYPE)
00096     .with<ecs::Transform>("transform_enemy_counter", 10.F, 100.F, 0.F)
00097     .with<ecs::Color>("color_enemy_counter", WHITE.r, WHITE.g, WHITE.b, WHITE.a)
00098     .with<ecs::Text>("id_enemy_counter", std::string("Enemies: 0"), 20U)
00099     .build();
00100
00101 // Compteur d'astéroïdes
00102 m_asteroidCounterEntity = registry.createEntity()
00103     .with<ecs::Font>("main_font", Path::Font::FONTS_RTYPE)
00104     .with<ecs::Transform>("transform_asteroid_counter", 10.F, 130.F, 0.F)
00105     .with<ecs::Color>("color_asteroid_counter", WHITE.r, WHITE.g, WHITE.b, WHITE.a)
00106     .with<ecs::Text>("id_asteroid_counter", std::string("Asteroids: 0"), 20U)
00107     .build();
00108
00109 m_playerEntity = registry.createEntity()
00110     .with<ecs::Transform>("player_transform", 200.F, 100.F, 0.F)
00111     .with<ecs::Velocity>("player_velocity", 0.F, 0.F)
00112     .with<ecs::Rect>("player_rect", 0.F, 0.F, static_cast<int>(GameConfig::Player::SPRITE_WIDTH),
00113                     static_cast<int>(GameConfig::Player::SPRITE_HEIGHT))
00114     .with<ecs::Scale>("player_scale", GameConfig::Player::SCALE, GameConfig::Player::SCALE)
00115     .with<ecs::Texture>("player_texture", Path::Texture::TEXTURE_PLAYER)
00116     .with<ecs::Player>("player", true)
00117     .with<ecs::BeamCharge>("beam_charge", 0.0f, GameConfig::Beam::MAX_CHARGE)
00118     .with<ecs::Hitbox>("player_hitbox", GameConfig::Hitbox::PLAYER_RADIUS)
00119     .build();
00120
00121 // La barre de Beam sera affichée directement au-dessus du joueur
00122 // Pas besoin d'une entité séparée
00123 // Crée des étoiles pour l'effet de parallax simple
00124 const int screenWidth = 1920;
00125 const int screenHeight = 1080;
00126
00127 // Étoiles lointaines (lentes)
00128 for (int i = 0; i < 50; ++i)
00129 {
00130     registry.createEntity()
00131         .with<ecs::Pixel>("star_far")
00132         .with<ecs::Transform>("star_far_transform", static_cast<float>(std::rand() % screenWidth),
00133                               static_cast<float>(std::rand() % screenHeight), 0.0f)
00134         .with<ecs::Color>("star_far_color", WHITE_TRANS.r, WHITE_TRANS.g, WHITE_TRANS.b,
00135                           WHITE_TRANS.a)
00136         .with<ecs::Velocity>("star_far_vel", -20.0f, 0.0f)
00137         .build();
00138
00139 // Étoiles moyennes

```

```

00140   for (int i = 0; i < 30; ++i)
00141   {
00142     registry.createEntity()
00143       .with<ecs::Pixel>("star_mid")
00144       .with<ecs::Transform>("star_mid_transform", static_cast<float>(std::rand() % screenWidth),
00145                               static_cast<float>(std::rand() % screenHeight), 0.0f)
00146       .with<ecs::Color>("star_mid_color", BLUE.r, BLUE.g, BLUE.b, BLUE.a)
00147       .with<ecs::Velocity>("star_mid_vel", -40.0f, 0.0f)
00148       .build();
00149   }
00150
00151 // Étoiles proches (rapides)
00152 for (int i = 0; i < 20; ++i)
00153 {
00154   registry.createEntity()
00155     .with<ecs::Pixel>("star_near")
00156     .with<ecs::Transform>("star_near_transform", static_cast<float>(std::rand() % screenWidth),
00157                               static_cast<float>(std::rand() % screenHeight), 0.0f)
00158     .with<ecs::Color>("star_near_color", YELLOW.r, YELLOW.g, YELLOW.b, YELLOW.a)
00159     .with<ecs::Velocity>("star_near_vel", -80.0f, 0.0f)
00160     .build();
00161 }
00162
00163 // Étoiles filantes
00164 for (int i = 0; i < 10; ++i)
00165 {
00166   registry.createEntity()
00167     .with<ecs::Pixel>("star_shooting")
00168     .with<ecs::Transform>("star_shooting_transform", static_cast<float>(std::rand() % screenWidth),
00169                               static_cast<float>(std::rand() % screenHeight), 0.0f)
00170     .with<ecs::Color>("star_shooting_color", GREEN.r, GREEN.g, GREEN.b, GREEN.a)
00171     .with<ecs::Velocity>("star_shooting_vel", -120.0f, static_cast<float>((std::rand() % 20) - 10))
00172     .build();
00173 }
00174
00175 // Planètes lointaines (très lentes)
00176 for (int i = 0; i < 5; ++i)
00177 {
00178   registry.createEntity()
00179     .with<ecs::Pixel>("planet_far")
00180     .with<ecs::Transform>("planet_far_transform", static_cast<float>(std::rand() % screenWidth),
00181                               static_cast<float>(std::rand() % screenHeight), 0.0f)
00182     .with<ecs::Color>("planet_far_color", PURPLE.r, PURPLE.g, PURPLE.b, PURPLE.a)
00183     .with<ecs::Velocity>("planet_far_vel", -5.0f, 0.0f)
00184     .build();
00185 }
00186
00187 // Nébuleuses (très lentes, grandes)
00188 for (int i = 0; i < 3; ++i)
00189 {
00190   registry.createEntity()
00191     .with<ecs::Pixel>("nebula")
00192     .with<ecs::Transform>("nebula_transform", static_cast<float>(std::rand() % screenWidth),
00193                               static_cast<float>(std::rand() % screenHeight), 0.0f)
00194     .with<ecs::Color>("nebula_color", BLUE_SECOND.r, BLUE_SECOND.g, BLUE_SECOND.b,
00195                         BLUE_SECOND.a)
00196     .with<ecs::Velocity>("nebula_vel", -8.0f, 0.0f)
00197     .build();
00198 }
00199
00200 // Comètes (mouvement diagonal)
00201 for (int i = 0; i < 8; ++i)
00202 {
00203   registry.createEntity()
00204     .with<ecs::Pixel>("comet")
00205     .with<ecs::Transform>("comet_transform", static_cast<float>(std::rand() % screenWidth),
00206                               static_cast<float>(std::rand() % screenHeight), 0.0f)
00207     .with<ecs::Color>("comet_color", GREEN.r, GREEN.g, GREEN.b, GREEN.a)
00208     .with<ecs::Velocity>("comet_vel", -60.0f, static_cast<float>((std::rand() % 40) - 20))
00209     .build();
00210 }
00211
00212 void cli::Game::update(const float dt, const eng::WindowSize &size)
00213 {
00214   auto &reg = getRegistry();
00215   auto *playerTransform = reg.getComponent<ecs::Transform>(m_playerEntity);
00216   auto *playerVelocity = reg.getComponent<ecs::Velocity>(m_playerEntity);
00217   auto &audios = reg.getAll<ecs::Audio>();
00218
00219   for (auto &audio : audios)
00220   {
00221     if (!audio.second.play && (m_audio->isPlaying(audio.second.id) == eng::Status::Playing))
00222     {
00223       m_audio->stopAudio(audio.second.id);
00224     }
00225   }

```

```

00226 // if (m_keysPressed[eng::Key::Space])
00227 //     m_weaponSystem.update(reg, dt);
00228 // m_weaponSystem.update(reg, dt, m_keysPressed[eng::Key::Space]); TODO(bobis33): tofix
00229 // Mise à jour des étoiles simples
00230 for (auto &[entity, pixel] : reg.getAll<ecs::Pixel>())
00231 {
00232     if (auto *transform = reg.getComponent<ecs::Transform>(entity))
00233     {
00234         if (auto *velocity = reg.getComponent<ecs::Velocity>(entity))
00235         {
00236             // Mise à jour de la position
00237             transform->x += velocity->x * dt;
00238             transform->y += velocity->y * dt;
00239
00240             // Réinitialiser si l'étoile sort de l'écran
00241             if (transform->x < -10.0f || transform->x > size.width + 10.0f || transform->y < -10.0f ||
00242                 transform->y > size.height + 10.0f)
00243             {
00244                 transform->x = static_cast<float>(size.width + std::rand() % 200);
00245                 transform->y = static_cast<float>(std::rand() % size.height);
00246             }
00247         }
00248     }
00249     if (auto *fpsText = reg.getComponent<ecs::Text>(m_fpsEntity))
00250     {
00251         fpsText->content = "FPS: " + std::to_string(static_cast<int>(1 / dt));
00252     }
00253
00254 // Mettre à jour le compteur d'ennemis
00255 if (auto *enemyCounterText = reg.getComponent<ecs::Text>(m_enemyCounterEntity))
00256 {
00257     int enemyCount = 0;
00258     for (auto &[entity, enemy] : reg.getAll<ecs::Enemy>())
00259     {
00260         enemyCount++;
00261     }
00262     enemyCounterText->content = "Enemies: " + std::to_string(enemyCount);
00263 }
00264
00265 // Mettre à jour le compteur d'astéroïdes
00266 if (auto *asteroidCounterText = reg.getComponent<ecs::Text>(m_asteroidCounterEntity))
00267 {
00268     int asteroidCount = 0;
00269     for (auto &[entity, asteroid] : reg.getAll<ecs::Asteroid>())
00270     {
00271         asteroidCount++;
00272     }
00273     asteroidCounterText->content = "Asteroids: " + std::to_string(asteroidCount);
00274 }
00275 float speed = GameConfig::Player::SPEED;
00276 float diagonal_speed = speed * GameConfig::Player::DIAGONAL_SPEED_MULTIPLIER;
00277
00278 playerVelocity->x = 0.0f;
00279 playerVelocity->y = 0.0f;
00280
00281
00282 bool up = m_keysPressed[eng::Key::Up];
00283 bool down = m_keysPressed[eng::Key::Down];
00284 bool left = m_keysPressed[eng::Key::Left];
00285 bool right = m_keysPressed[eng::Key::Right];
00286
00287 if (up && right)
00288 {
00289     playerVelocity->x = diagonal_speed;
00290     playerVelocity->y = -diagonal_speed;
00291 }
00292 else if (up && left)
00293 {
00294     playerVelocity->x = -diagonal_speed;
00295     playerVelocity->y = -diagonal_speed;
00296 }
00297 else if (down && right)
00298 {
00299     playerVelocity->x = diagonal_speed;
00300     playerVelocity->y = diagonal_speed;
00301 }
00302 else if (down && left)
00303 {
00304     playerVelocity->x = -diagonal_speed;
00305     playerVelocity->y = diagonal_speed;
00306 }
00307 else
00308 {
00309     if (up)
00310         playerVelocity->y = -speed;
00311     if (down)
00312         playerVelocity->y = speed;

```

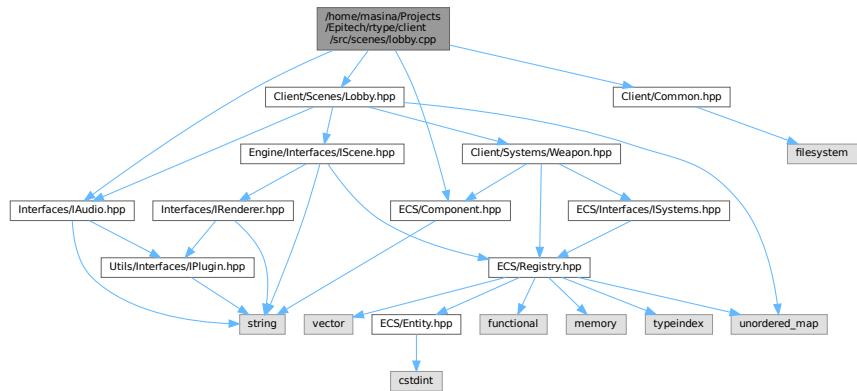
```

00313     if (left)
00314         playerVelocity->x = -speed;
00315     if (right)
00316         playerVelocity->x = speed;
00317 }
00318
00319 playerTransform->x += playerVelocity->x * dt;
00320 playerTransform->y += playerVelocity->y * dt;
00321 playerTransform->x = std::max(playerTransform->x, 0.F);
00322 playerTransform->y = std::max(playerTransform->y, 0.F);
00323 playerTransform->x = std::min(playerTransform->x, static_cast<float>(size.width) -
00324                                         GameConfig::Player::SPRITE_WIDTH * GameConfig::Player::SCALE);
00325 playerTransform->y =
00326     std::min(playerTransform->y,
00327             static_cast<float>(size.height) - GameConfig::Player::SPRITE_HEIGHT * GameConfig::Player::SCALE);
00328 }
00329
00330 void cli::Game::event(const eng::Event &event)
00331 {
00332     switch (event.type)
00333     {
00334         case eng::EventType::KeyPressed:
00335             if (event.key == eng::Key::Up)
00336                 m_keysPressed[eng::Key::Up] = true;
00337             if (event.key == eng::Key::Down)
00338                 m_keysPressed[eng::Key::Down] = true;
00339             if (event.key == eng::Key::Left)
00340                 m_keysPressed[eng::Key::Left] = true;
00341             if (event.key == eng::Key::Right)
00342                 m_keysPressed[eng::Key::Right] = true;
00343             if (event.key == eng::Key::Space)
00344                 m_keysPressed[eng::Key::Space] = true;
00345             break;
00346
00347         case eng::EventType::KeyReleased:
00348             if (event.key == eng::Key::Up)
00349                 m_keysPressed[eng::Key::Up] = false;
00350             if (event.key == eng::Key::Down)
00351                 m_keysPressed[eng::Key::Down] = false;
00352             if (event.key == eng::Key::Left)
00353                 m_keysPressed[eng::Key::Left] = false;
00354             if (event.key == eng::Key::Right)
00355                 m_keysPressed[eng::Key::Right] = false;
00356             if (event.key == eng::Key::Space)
00357                 m_keysPressed[eng::Key::Space] = false;
00358             break;
00359
00360         default:
00361             break;
00362     }
00363 }
```

8.77 /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}

8.77.1 Variable Documentation

8.77.1.1 WHITE

`eng::Color WHITE = {.r = 255U, .g = 255U, .b = 255U, .a = 255U}` [static], [constexpr]

Definition at line 6 of file [lobby.cpp](#).

8.78 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::shared_ptr<eng::IRenderer> &renderer, const std::shared_ptr<eng::IAudio> &audio)
00009 : m_audio(audio)
00010 {
00011     auto &registry = AScene::getRegistry();
00012
00013     registry.onComponentAdded(
00014         [&renderer, &audio, &registry](const ecs::Entity e, const std::type_info &type)
00015     {
00016         const auto *audioComp = registry.getComponent<ecs::Audio>(e);
00017         const auto *colorComp = registry.getComponent<ecs::Color>(e);
00018         const auto *fontComp = registry.getComponent<ecs::Font>(e);
00019         const auto *rectComp = registry.getComponent<ecs::Rect>(e);
00020         const auto *scaleComp = registry.getComponent<ecs::Scale>(e);
00021         const auto *textComp = registry.getComponent<ecs::Text>(e);
00022         const auto *textureComp = registry.getComponent<ecs::Texture>(e);
00023         const auto *transform = registry.getComponent<ecs::Transform>(e);
00024
00025         if (type == typeid(ecs::Text))
00026     {
00027             if (textComp && transform && fontComp)
00028         {
```

```

00029         renderer->createFont(fontComp->id, fontComp->path);
00030         renderer->createText(
00031             {.font_name = fontComp->id,
00032             .color = {.r = colorComp->r, .g = colorComp->g, .b = colorComp->b, .a = colorComp->a},
00033             .content = textComp->content,
00034             .size = textComp->font_size,
00035             .x = transform->x,
00036             .y = transform->y,
00037             .name = textComp->id});
00038     }
00039 }
00040 else if (type == typeid(ecs::Texture))
00041 {
00042     const float scale_x = scaleComp ? scaleComp->x : 1.F;
00043     const float scale_y = scaleComp ? scaleComp->y : 1.F;
00044
00045     renderer->createTexture(textureComp->id, textureComp->path);
00046
00047     if (transform && textureComp)
00048     {
00049         if (rectComp)
00050         {
00051             renderer->createSprite(textureComp->id + std::to_string(e), textureComp->id, transform->x,
00052                                         transform->y, scale_x, scale_y, static_cast<int>(rectComp->pos_x),
00053                                         static_cast<int>(rectComp->pos_y), rectComp->size_x, rectComp->size_y);
00054         }
00055         else
00056         {
00057             renderer->createSprite(textureComp->id + std::to_string(e), textureComp->id, transform->x,
00058                                         transform->y);
00059         }
00060     }
00061 }
00062 else if (type == typeid(ecs::Audio))
00063 {
00064     if (audioComp)
00065     {
00066         audio->createAudio(audioComp->path, audioComp->volume, audioComp->loop,
00067                               audioComp->id + std::to_string(e));
00068     }
00069 }
00070 );
00071
00072 registry.createEntity().with<ecs::Audio>("id_audio", Path::Audio::AUDIO_TITLE, 5.F, true, true).build();
00073 registry.createEntity()
00074     .with<ecs::Font>("main_font", Path::Font::FONTS_RTYPE)
00075     .with<ecs::Transform>("transform_title", 10.F, 10.F, 0.F)
00076     .with<ecs::Color>("color_title", WHITE.r, WHITE.g, WHITE.b, WHITE.a)
00077     .with<ecs::Text>("id", std::string("RTtype Client"), 50U)
00078     .build();
00079 m_fpsEntity = registry.createEntity()
00080     .with<ecs::Font>("main_font", Path::Font::FONTS_RTYPE)
00081     .with<ecs::Transform>("transform_fps", 10.F, 70.F, 0.F)
00082     .with<ecs::Color>("color_fps", WHITE.r, WHITE.g, WHITE.b, WHITE.a)
00083     .with<ecs::Text>("id_text", std::string("FPS: 0"), 20U)
00084     .build();
00085
00086 for (size_t i = 0; i < m_menuOptions.size(); ++i)
00087 {
00088     registry.createEntity()
00089         .with<ecs::Font>("main_font", Path::Font::FONTS_RTYPE)
00090         .with<ecs::Transform>("transform_menu", 100.F, 200.F + i * 60.F, 0.F)
00091         .with<ecs::Color>("color_menu", WHITE.r, WHITE.g, WHITE.b, WHITE.a)
00092         .with<ecs::Text>("menu_" + m_menuOptions[i], m_menuOptions[i], 40U)
00093         .build();
00094 }
00095 m_selectedIndex = 2;
00096 }
00097
00098 void cli::Lobby::update(const float dt, const eng::WindowSize &size)
00099 {
00100     auto &reg = getRegistry();
00101
00102     auto &transforms = reg.getAll<ecs::Transform>();
00103     auto &colors = reg.getAll<ecs::Color>();
00104     auto &texts = reg.getAll<ecs::Text>();
00105     auto &audios = reg.getAll<ecs::Audio>();
00106
00107     for (auto &audio : audios)
00108     {
00109         if (!audio.second.play && (m_audio->isPlaying(audio.second.id) == eng::Status::Playing))
00110         {
00111             m_audio->stopAudio(audio.second.id);
00112         }
00113     }
00114     size_t i = 0;
00115     for (auto &[entity, text] : texts)

```

```

00116    {
00117        if (text.content == "Solo" || text.content == "Multi" || text.content == "Settings")
00118    {
00119        auto &color = colors.at(entity);
00120
00121        if (i == m_selectedIndex)
00122        {
00123            color.r = 255;
00124            color.g = 200;
00125            color.b = 0;
00126        }
00127        else
00128        {
00129            color.r = 255;
00130            color.g = 255;
00131            color.b = 255;
00132        }
00133
00134        i++;
00135    }
00136
00137
00138    if (auto *fpsText = reg.getComponent<ecs::Text>(m_fpsEntity))
00139    {
00140        fpsText->content = "FPS: " + std::to_string(static_cast<int>(1 / dt));
00141    }
00142 }
00143
00144 void cli::Lobby::event(const eng::Event &event)
00145 {
00146     switch (event.type)
00147     {
00148         case eng::EventType::KeyPressed:
00149             if (event.key == eng::Key::Up)
00150             {
00151                 if (m_selectedIndex == 2)
00152                 {
00153                     m_selectedIndex = 0;
00154                 }
00155                 else
00156                 {
00157                     m_selectedIndex++;
00158                 }
00159             }
00160             else if (event.key == eng::Key::Down)
00161             {
00162                 if (m_selectedIndex == 0)
00163                 {
00164                     m_selectedIndex = 2;
00165                 }
00166                 else
00167                 {
00168                     m_selectedIndex--;
00169                 }
00170             }
00171             else if (event.key == eng::Key::Enter)
00172             {
00173                 const std::string &selectedOption =
00174                     m_menuOptions[static_cast<int>(m_menuOptions.size() - 1 - m_selectedIndex)];
00175                 if (onOptionSelected)
00176                 {
00177                     onOptionSelected(selectedOption);
00178                 }
00179             }
00180             break;
00181
00182         case eng::EventType::KeyReleased:
00183             if (event.key == eng::Key::Up)
00184             {
00185                 m_keysPressed[eng::Key::Up] = false;
00186             }
00187             if (event.key == eng::Key::Down)
00188             {
00189                 m_keysPressed[eng::Key::Down] = false;
00190             }
00191             if (event.key == eng::Key::Left)
00192             {
00193                 m_keysPressed[eng::Key::Left] = false;
00194             }
00195             if (event.key == eng::Key::Right)
00196             {
00197                 m_keysPressed[eng::Key::Right] = false;
00198             }
00199             if (event.key == eng::Key::Space)
00200             {
00201                 m_keysPressed[eng::Key::Space] = false;
00202             }

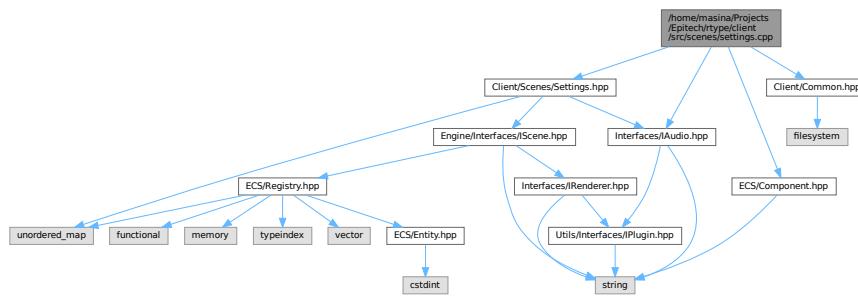
```

```

00203     break;
00204
00205     default:
00206         break;
00207     }
00208 }
```

8.79 /home/masina/Projects/← Epitech/rtype/client/src/scenes/settings.cpp File Reference

```
#include "Client/Scenes/Settings.hpp"
#include "Client/Common.hpp"
#include "ECS/Component.hpp"
#include "Interfaces/IAudio.hpp"
Include dependency graph for settings.cpp:
```



Variables

- static constexpr eng::Color WHITE = {.r = 255U, .g = 255U, .b = 255U, .a = 255U}

8.79.1 Variable Documentation

8.79.1.1 WHITE

`eng::Color` WHITE = {.r = 255U, .g = 255U, .b = 255U, .a = 255U} [static], [constexpr]

Definition at line 6 of file [settings.cpp](#).

8.80 settings.cpp

[Go to the documentation of this file.](#)

```

00001 #include "Client/Scenes/Settings.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::Settings::Settings(const std::shared_ptr<eng::IRenderer> &renderer, const std::shared_ptr<eng::IAudio> &audio)
00009   : m_audio(audio)
00010 {
00011     auto &registry = AScene::getRegistry();
```

```

00012
00013     registry.onComponentAdded(
00014         [&renderer, &audio, &registry](const ecs::Entity e, const std::type_info &type)
00015     {
00016         const auto *audioComp = registry.getComponent<ecs::Audio>(e);
00017         const auto *colorComp = registry.getComponent<ecs::Color>(e);
00018         const auto *fontComp = registry.getComponent<ecs::Font>(e);
00019         const auto *rectComp = registry.getComponent<ecs::Rect>(e);
00020         const auto *scaleComp = registry.getComponent<ecs::Scale>(e);
00021         const auto *textComp = registry.getComponent<ecs::Text>(e);
00022         const auto *textureComp = registry.getComponent<ecs::Texture>(e);
00023         const auto *transform = registry.getComponent<ecs::Transform>(e);
00024
00025         if (type == typeid(ecs::Text))
00026         {
00027             if (textComp && transform && fontComp)
00028             {
00029                 renderer->createFont(fontComp->id, fontComp->path);
00030                 renderer->createText(
00031                     {.font_name = fontComp->id,
00032                      .color = {.r = colorComp->r, .g = colorComp->g, .b = colorComp->b, .a = colorComp->a},
00033                      .content = textComp->content,
00034                      .size = textComp->font_size,
00035                      .x = transform->x,
00036                      .y = transform->y,
00037                      .name = textComp->id});
00038             }
00039         }
00040         else if (type == typeid(ecs::Texture))
00041         {
00042             const float scale_x = scaleComp ? scaleComp->x : 1.F;
00043             const float scale_y = scaleComp ? scaleComp->y : 1.F;
00044
00045             renderer->createTexture(textureComp->id, textureComp->path);
00046
00047             if (transform && textureComp)
00048             {
00049                 if (rectComp)
00050                 {
00051                     renderer->createSprite(textureComp->id + std::to_string(e), textureComp->id, transform->x,
00052                                         transform->y, scale_x, scale_y, static_cast<int>(rectComp->pos_x),
00053                                         static_cast<int>(rectComp->pos_y), rectComp->size_x, rectComp->size_y);
00054                 }
00055                 else
00056                 {
00057                     renderer->createSprite(textureComp->id + std::to_string(e), textureComp->id, transform->x,
00058                                         transform->y);
00059                 }
00060             }
00061         }
00062         else if (type == typeid(ecs::Audio))
00063         {
00064             if (audioComp)
00065             {
00066                 audio->createAudio(audioComp->path, audioComp->volume, audioComp->loop,
00067                                     audioComp->id + std::to_string(e));
00068             }
00069         }
00070     });
00071
00072     registry.createEntity().with<ecs::Audio>("id_audio", Path::Audio::AUDIO_TITLE, 5.F, true, true).build();
00073     registry.createEntity()
00074         .with<ecs::Font>("main_font", Path::Font::FONTS_RTYPE)
00075         .with<ecs::Transform>("transform_title", 10.F, 10.F, 0.F)
00076         .with<ecs::Color>("color_title", WHITE.r, WHITE.g, WHITE.b, WHITE.a)
00077         .with<ecs::Text>("id", std::string("RTtype Client"), 50U)
00078         .build();
00079 }
00080
00081 void cli::Settings::update(const float dt, const eng::WindowSize &size)
00082 {
00083     auto &reg = getRegistry();
00084
00085     auto &transforms = reg.getAll<ecs::Transform>();
00086     auto &colors = reg.getAll<ecs::Color>();
00087     auto &texts = reg.getAll<ecs::Text>();
00088     auto &audios = reg.getAll<ecs::Audio>();
00089
00090     for (auto &audio : audios)
00091     {
00092         if (!audio.second.play && (m_audio->isPlaying(audio.second.id) == eng::Status::Playing))
00093         {
00094             m_audio->stopAudio(audio.second.id);
00095         }
00096     }
00097 }
00098

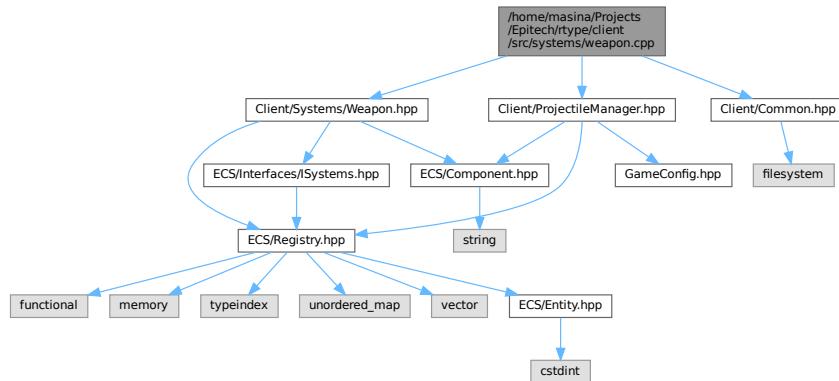
```

```

00099 void cli::Settings::event(const eng::Event &event)
00100 {
00101     switch (event.type)
00102     {
00103         case eng::EventType::KeyPressed:
00104             if (event.key == eng::Key::Escape)
00105                 onLeave();
00106             break;
00107         default:
00108             break;
00109     }
00110 }
00111 }
```

8.81 /home/masina/Projects/ ↵ Epitech/rtype/client/src/systems/weapon.cpp File Reference

```
#include "Client/Systems/Weapon.hpp"
#include "Client/Common.hpp"
#include "Client/ProjectileManager.hpp"
Include dependency graph for weapon.cpp:
```



Namespaces

- namespace `cli`

8.82 weapon.cpp

[Go to the documentation of this file.](#)

```

00001 /**
00002 /// @file WeaponSystem.cpp
00003 /// @brief Implementation of WeaponSystem
00004 /// @namespace cli
00005 /**
00006
00007 #include "Client/Systems/Weapon.hpp"
00008 #include "Client/Common.hpp"
00009 #include "Client/ProjectileManager.hpp"
00010
00011 namespace cli
00012 {
00013     void WeaponSystem::update(ecs::Registry &registry, float dt, bool spacePressed)
00014     {
00015         using namespace GameConfig::Projectile;
```

```

00016     using namespace GameConfig::Beam;
00017
00018     // Update cooldowns
00019     if (m_fireCooldown > 0.0f)
00020         m_fireCooldown -= dt;
00021
00022     // Get player entity and position
00023     auto playerEntities = registry.getAll<ecs::Player>();
00024     if (playerEntities.empty())
00025         return;
00026
00027     auto &[playerEntity, player] = *playerEntities.begin();
00028     auto *transform = registry.getComponent<ecs::Transform>(playerEntity);
00029     auto *beamCharge = registry.getComponent<ecs::BeamCharge>(playerEntity);
00030     if (!transform || !beamCharge)
00031         return;
00032
00033     float projectileX = transform->x + GameConfig::Player::SPRITE_WIDTH;
00034     float projectileY = transform->y + GameConfig::Player::SPRITE_HEIGHT / 2.0f;
00035
00036     if (spacePressed)
00037     {
00038         // Commencer le chargement si ce n'était pas déjà le cas
00039         if (!m_isCharging)
00040         {
00041             m_isCharging = true;
00042         }
00043
00044         // Charger la barre de Beam
00045         beamCharge->current_charge += CHARGE_RATE * dt;
00046         if (beamCharge->current_charge > beamCharge->max_charge)
00047             beamCharge->current_charge = beamCharge->max_charge;
00048
00049         // Afficher l'animation de chargement si on charge
00050         if (beamCharge->current_charge < beamCharge->max_charge)
00051         {
00052             showLoadingAnimation(registry, playerEntity, transform);
00053         }
00054         else
00055         {
00056             hideLoadingAnimation(registry, playerEntity);
00057         }
00058
00059         // PENDANT LE CHARGEMENT : NE RIEN TIRER DU TOUT
00060         return; // Sortir de la fonction sans rien tirer
00061     }
00062     else
00063     {
00064         // Si on était en train de charger et qu'on relâche espace
00065         if (m_isCharging)
00066         {
00067             m_isCharging = false;
00068
00069             // Cacher l'animation de chargement
00070             hideLoadingAnimation(registry, playerEntity);
00071
00072             // Quand on relâche espace, vérifier si on peut tirer
00073             if (m_fireCooldown <= 0.0f)
00074             {
00075                 // Si on a au moins 50% de charge, tirer un supercharged
00076                 float chargeThreshold = beamCharge->max_charge * 0.5f; // 50% minimum
00077                 if (beamCharge->current_charge >= chargeThreshold)
00078                 {
00079                     if (tryFireSupercharged(registry, projectileX, projectileY))
00080                     {
00081                         beamCharge->current_charge = 0.0f; // Consommer toute la charge
00082                         m_fireCooldown = Supercharged::FIRE_COOLDOWN;
00083                     }
00084                 }
00085                 else
00086                 {
00087                     // Si pas assez de charge, tirer un basic
00088                     tryFireBasic(registry, projectileX, projectileY);
00089                 }
00090             }
00091         }
00092     }
00093 }
00094
00095 void WeaponSystem::reset()
00096 {
00097     m_fireCooldown = 0.0f;
00098     m_isCharging = false;
00099 }
00100
00101 bool WeaponSystem::tryFireBasic(ecs::Registry &registry, float x, float y)
00102 {

```

```

00103     using namespace GameConfig::Projectile;
00104
00105     if (m_fireCooldown > 0.0f)
00106         return false;
00107
00108     ProjectileManager::createBasicProjectile(registry, x, y, Basic::SPEED, 0.0f);
00109     m_fireCooldown = Basic::FIRE_COOLDOWN;
00110     return true;
00111 }
00112
00113 bool WeaponSystem::tryFireSupercharged(ecs::Registry &registry, float x, float y)
00114 {
00115     using namespace GameConfig::Projectile;
00116
00117     ProjectileManager::createSuperchargedProjectile(registry, x, y, Supercharged::SPEED, 0.0f);
00118     return true;
00119 }
00120
00121 void WeaponSystem::showLoadingAnimation(ecs::Registry &registry, ecs::Entity playerEntity,
00122                                         const ecs::Transform *playerTransform)
00123 {
00124     using namespace GameConfig::LoadingAnimation;
00125
00126     // Chercher s'il y a déjà une animation de chargement
00127     auto loadingEntities = registry.getAll<ecs::LoadingAnimation>();
00128     for (auto &[entity, animation] : loadingEntities)
00129     {
00130         auto *loadingTransform = registry.getComponent<ecs::Transform>(entity);
00131         if (loadingTransform)
00132         {
00133             // Mettre à jour la position
00134             loadingTransform->x = playerTransform->x + OFFSET_X;
00135             loadingTransform->y = playerTransform->y + OFFSET_Y;
00136             return; // Animation déjà présente
00137         }
00138     }
00139
00140     // Créer une nouvelle animation de chargement
00141     auto loadingEntity =
00142         registry.createEntity()
00143             .with<ecs::Transform>("loading_transform", playerTransform->x + OFFSET_X, playerTransform->y +
00144             OFFSET_Y,
00145             0.0f)
00146             .with<ecs::Rect>("loading_rect", 0.0f, 0.0f, static_cast<int>(SPRITE_WIDTH),
00147                             static_cast<int>(SPRITE_HEIGHT))
00148             .with<ecs::Scale>("loading_scale", 1.0f, 1.0f)
00149             .with<ecs::Texture>("loading_texture", Path::Texture::TEXTURE_SHOOT_LOADING)
00150             .with<ecs::LoadingAnimation>("loading_animation", 0, ANIMATION_FRAMES,
ANIMATION_DURATION, 0.0f,
00151                                         SPRITE_WIDTH, SPRITE_HEIGHT, ANIMATION_FRAMES)
00152             .build();
00153 }
00154
00155 void WeaponSystem::hideLoadingAnimation(ecs::Registry &registry, ecs::Entity playerEntity)
00156 {
00157     // Supprimer toutes les animations de chargement
00158     auto loadingEntities = registry.getAll<ecs::LoadingAnimation>();
00159     std::vector<ecs::Entity> toRemove;
00160
00161     for (auto &[entity, animation] : loadingEntities)
00162     {
00163         toRemove.push_back(entity);
00164     }
00165
00166     for (auto entity : toRemove)
00167     {
00168         if (registry.hasComponent<ecs::Transform>(entity))
00169             registry.removeComponent<ecs::Transform>(entity);
00170         if (registry.hasComponent<ecs::Rect>(entity))
00171             registry.removeComponent<ecs::Rect>(entity);
00172         if (registry.hasComponent<ecs::Scale>(entity))
00173             registry.removeComponent<ecs::Scale>(entity);
00174         if (registry.hasComponent<ecs::Texture>(entity))
00175             registry.removeComponent<ecs::Texture>(entity);
00176         if (registry.hasComponent<ecs::LoadingAnimation>(entity))
00177             registry.removeComponent<ecs::LoadingAnimation>(entity);
00178     }
00179 } // namespace cli

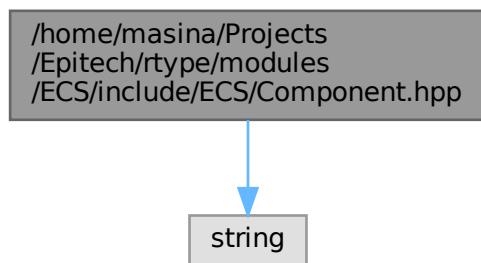
```

8.83 /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`
- struct `ecs::Animation`
- struct `ecs::Projectile`
- struct `ecs::BeamCharge`
- struct `ecs::LoadingAnimation`
- struct `ecs::Enemy`
- struct `ecs::Asteroid`
- struct `ecs::Explosion`
- struct `ecs::Hitbox`

Namespaces

- namespace `ecs`

8.83.1 Detailed Description

This file contains the component definitions.

Definition in file [Component.hpp](#).

8.84 Component.hpp

[Go to the documentation of this file.](#)

```
00001 /**
00002  * @file Component.hpp
00003  * @brief This file contains the component definitions
00004  * @namespace ecs
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 struct Animation final : IComponent
00077 {
00078     int current_frame{};
00079     int total_frames{};
00080     float frame_duration{};
00081     float current_time{};
00082     int frame_width{};
00083     int frame_height{};
00084     int frames_per_row{};
00085 };
00086 struct Projectile final : IComponent
00087 {
00088     enum Type
00089     {
00090         BASIC,
00091         SUPERCHARGED
00092     };
00093     Type type;
00094     float damage;
00095     float lifetime;
00096     float current_lifetime;
00097 };
00098
00099 struct BeamCharge final : IComponent
00100 {
00101     float current_charge;
00102     float max_charge;
00103 };
00104
00105 struct LoadingAnimation final : IComponent
00106 {
00107     int current_frame;
00108     int total_frames;
00109     float frame_duration;
00110     float current_time;
00111     float frame_width;
00112     float frame_height;
00113     int frames_per_row;
00114 };
00115
00116 struct Enemy final : IComponent
00117 {
00118     float health;
00119     float max_health;
00120     float damage;
00121     float speed;
00122     float last_shot_time;
00123     float shoot_cooldown;
00124 };
00125
00126 struct Asteroid final : IComponent
00127 {
00128     enum Size
00129     {
00130         SMALL,
00131         MEDIUM,
00132         LARGE
00133     };
00134     Size size;
00135     float rotation_speed;
00136     float health;
00137 };
00138
00139 struct Explosion final : IComponent
00140 {
00141     int current_frame;
00142     int total_frames;
00143     float frame_duration;
00144     float current_time;
00145     float frame_width;
00146     float frame_height;
00147     int frames_per_row;
00148     float lifetime;
00149     float current_lifetime;
```

```

00150     };
00151
00152     struct Hitbox final : IComponent
00153     {
00154         float radius;
00155     };
00156
00157 } // namespace ecs

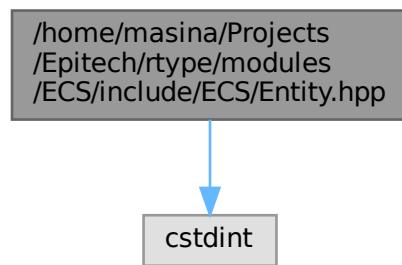
```

8.85 /home/masina/Projects/Epitech/rtype/modules/ECS/include/ecs/ECS/Entity.hpp File Reference

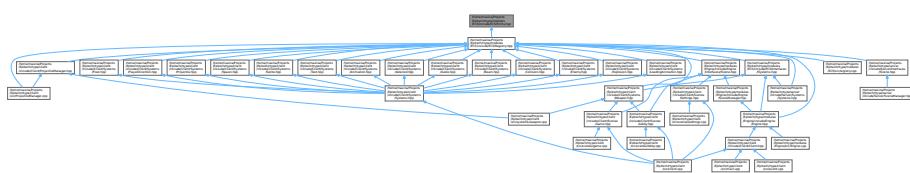
This file contains the entity definitions.

#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`

8.85.1 Detailed Description

This file contains the entity definitions.

Definition in file [Entity.hpp](#).

8.86 Entity.hpp

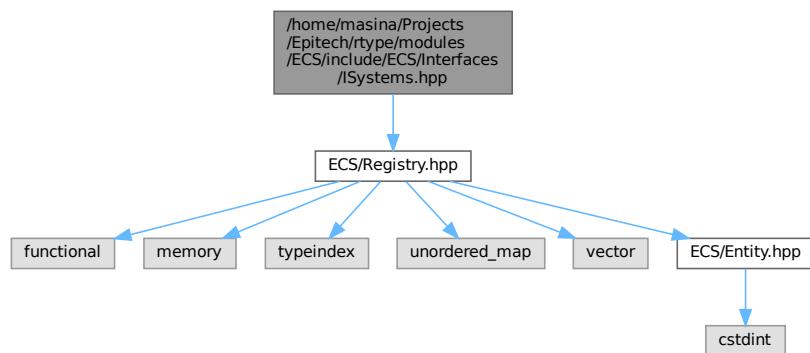
[Go to the documentation of this file.](#)

```
00001 /**
00002 /// @file Entity.hpp
00003 /// @brief This file contains the entity definitions
00004 /// @namespace ecs
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
```

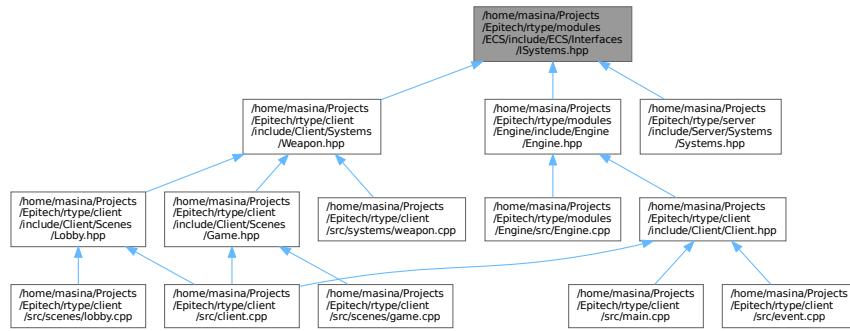
8.87 /home/masina/Projects/Epitech/rtype/modules/ECS/include/` ECS/Interfaces/ISystems.hpp File Reference

#include "ECS/Registry.hpp"

Include dependency graph for ISystems.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class eng::ISystem
- class eng::ASystem

Namespaces

- namespace eng

8.88 ISystems.hpp

[Go to the documentation of this file.](#)

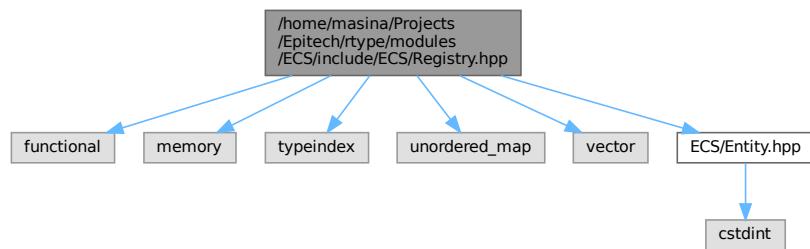
```

00001 /**
00002 /**
00003 /**
00004 /**
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_isEnabled; }
00027             void setEnable(const bool enable) override { m_isEnabled = enable; }
00028
00029         private:
00030             bool m_isEnabled = true;
00031     };
00032
00033 } // namespace eng
  
```

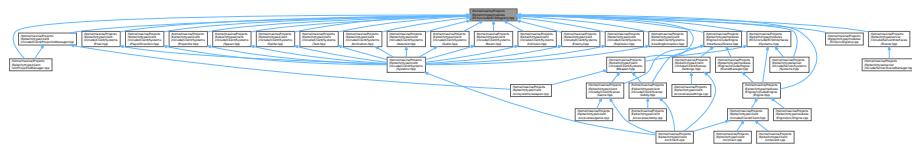
8.89 /home/masina/Projects/Epitech/rtype/modules/ECS/include/… ECS/Registry.hpp File Reference

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](#)

8.89.1 Detailed Description

This file contains the Registry class declaration.

Definition in file [Registry.hpp](#).

8.90 Registry.hpp

[Go to the documentation of this file.](#)

```

00001 /**
00002  * @file Registry.hpp
00003  * @brief This file contains the Registry class declaration
00004  * @namespace ecs
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  * @class Registry
00021  * @brief Class for managing entities and their components
00022  * @namespace ecs
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     }

```

```

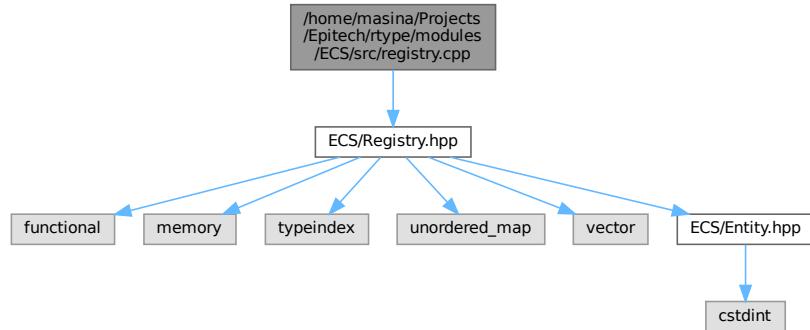
00083     }
00084
00085     template <typename T> void removeComponent(Entity e)
00086     {
00087         auto &pool = getPool<T>();
00088         pool.remove(e);
00089     }
00090
00091     void onComponentAdded(std::function<void(Entity, const std::type_info &)> cb)
00092     {
00093         m_onComponentAddedCallbacks.push_back(std::move(cb));
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

```

8.91 /home/masina/Projects/Epitech/rtype/modules/` ECS/src/registry.cpp File Reference

```
#include "ECS/Registry.hpp"
```

Include dependency graph for registry.cpp:



8.92 registry.cpp

[Go to the documentation of this file.](#)

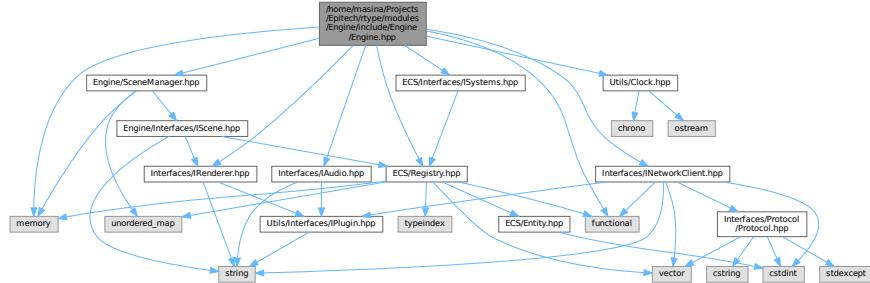
```
00001 #include "ECS/Registry.hpp"
```

8.93 /home/masina/Projects/Epitech/rtype/modules/Engine/include/` Engine/Engine.hpp File Reference

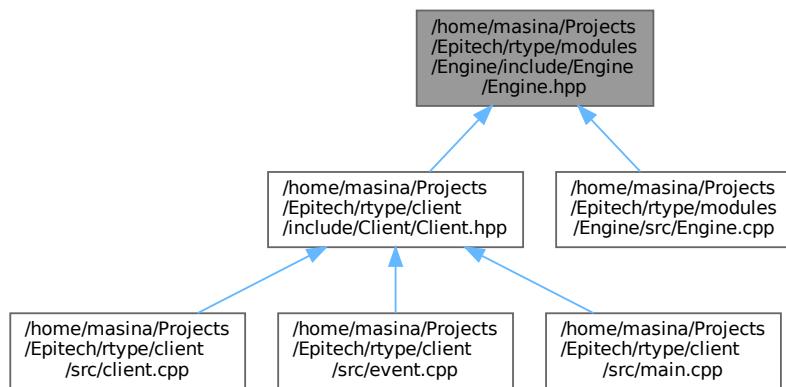
This file contains the Engine class declaration.

```
#include <functional>
#include <memory>
#include "ECS/Registry.hpp"
#include "Engine/SceneManager.hpp"
#include "ECS/Interfaces/ISystems.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 }

8.93.1 Detailed Description

This file contains the Engine class declaration.

Definition in file [Engine.hpp](#).

8.94 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"

```

```

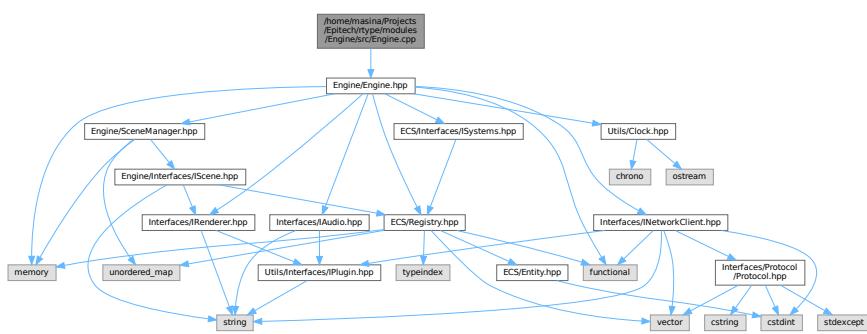
00013 #include "Engine/SceneManager.hpp"
00014 #include "ECS/Interfaces/ISystems.hpp"
00015 #include "Interfaces/IAudio.hpp"
00016 #include "Interfaces/INetworkClient.hpp"
00017 #include "Interfaces/IRenderer.hpp"
00018 #include "Utils/Clock.hpp"
00019
00020 namespace eng
00021 {
00022
00023     enum 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::shared_ptr<IAudio>()> &audioFactory,
00040                 const std::function<std::shared_ptr<INetworkClient>()> &networkFactory,
00041                 const std::function<std::shared_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::shared_ptr<IAudio> &getAudio() { return m_audio; }
00050         std::shared_ptr<INetworkClient> &getNetwork() { return m_network; }
00051         std::shared_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::shared_ptr<IAudio> m_audio;
00070         std::shared_ptr<INetworkClient> m_network;
00071         std::shared_ptr<IRenderer> m_renderer;
00072     }; // class Engine
00073 } // namespace eng

```

8.95 /home/masina/Projects/Epitech/rtype/modules/Engine/src/← Engine.cpp File Reference

```
#include "Engine/Engine.hpp"
```

Include dependency graph for Engine.cpp:



8.96 Engine.cpp

[Go to the documentation of this file.](#)

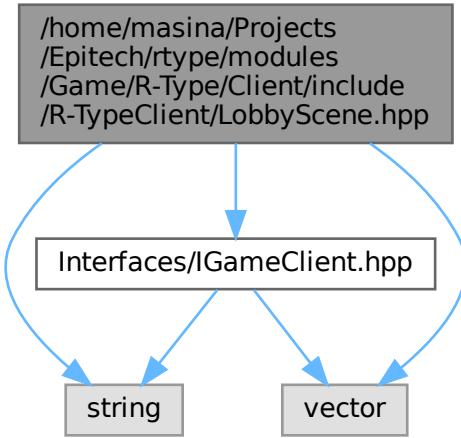
```
00001 #include "Engine/Engine.hpp"
00002
00003 eng::Engine::Engine(const std::function<std::shared_ptr<IAudio>()> &audioFactory,
00004             const std::function<std::shared_ptr<INetworkClient>()> &networkFactory,
00005             const std::function<std::shared_ptr<IRenderer>()> &rendererFactory)
00006 : m_clock(std::make_unique<util::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 }
```

8.97 /home/masina/Projects/Epitech/rtype/modules/Game/R-Type/← Client/include/R-TypeClient/LobbyScene.hpp File Reference

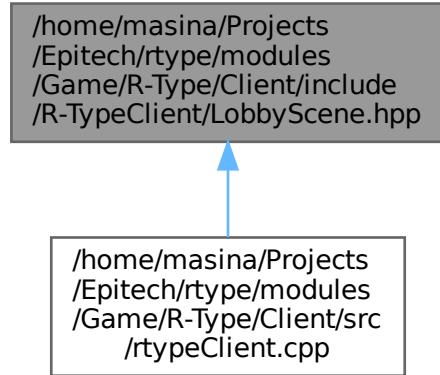
This file contains the lobby scene.

```
#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`

8.97.1 Detailed Description

This file contains the lobby scene.

Definition in file [LobbyScene.hpp](#).

8.98 LobbyScene.hpp

[Go to the documentation of this file.](#)

```
00001 /**
00002  * @file LobbyScene.hpp
00003  * @brief This file contains the lobby scene
00004  * @namespace gme
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     [[nodiscard]] const std::string &getName() const override { return m_name; }
00027     [[nodiscard]] const std::vector<Sprite> &getEntities() const override { return m_entities; }

00028     void addEntity(const Sprite &e) { m_entities.push_back(e); }
00029     std::vector<Sprite> &getEntitiesMutable() override { return m_entities; }

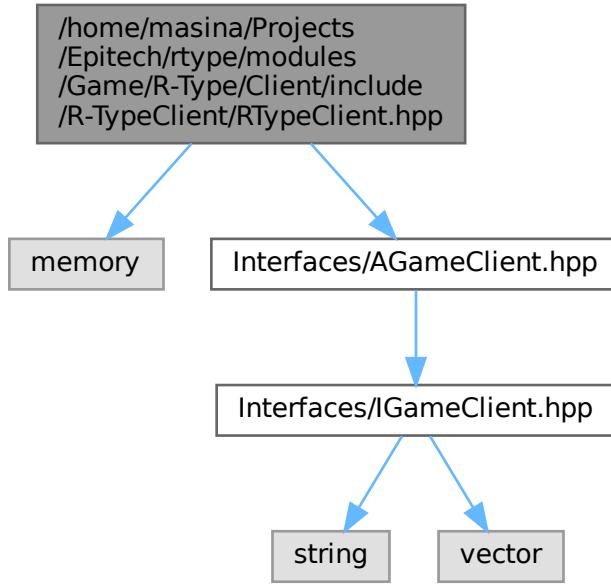
00030     private:
00031         std::string m_name;
00032         std::vector<Sprite> m_entities;
00033     }; // class LobbyScene
00034 } // namespace gme
```

8.99 /home/masina/Projects/Epitech/rtype/modules/Game/R-Type/← Client/include/R-TypeClient/RTypeClient.hpp File Reference

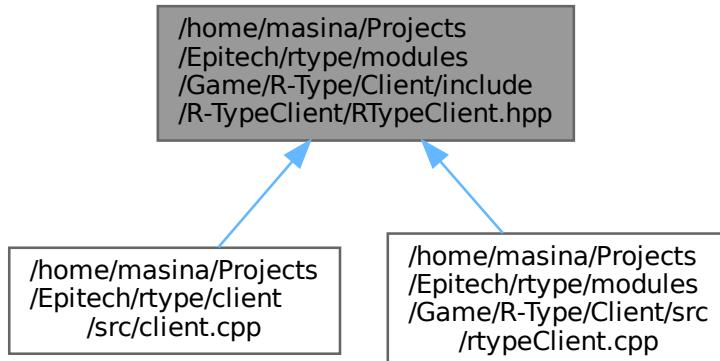
RType client class declaration.

```
#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](#)

8.99.1 Detailed Description

RType client class declaration.

Definition in file [RTypeClient.hpp](#).

8.100 RTypeClient.hpp

[Go to the documentation of this file.](#)

```

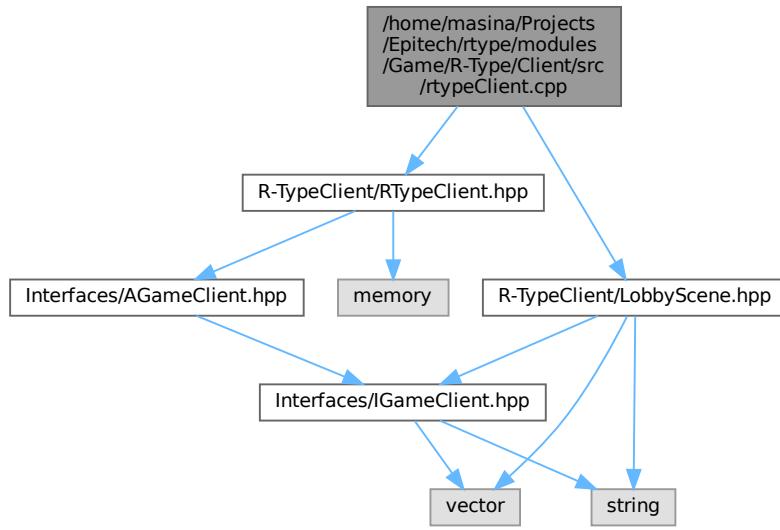
00001 /**
00002  * @file RTypeClient.hpp
00003  * @brief RType client class declaration
00004  * @namespace gme
00005 /**
00006
00007 #pragma once
00008
00009 #include <memory>
00010
00011 #include "Interfaces/AGameClient.hpp"
00012
00013 namespace gme
00014 {
00015
00016 /**
00017  * @class RTypeClient
00018  * @brief Class for the R-Type game
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

```

8.101 /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:



8.102 rtypeClient.cpp

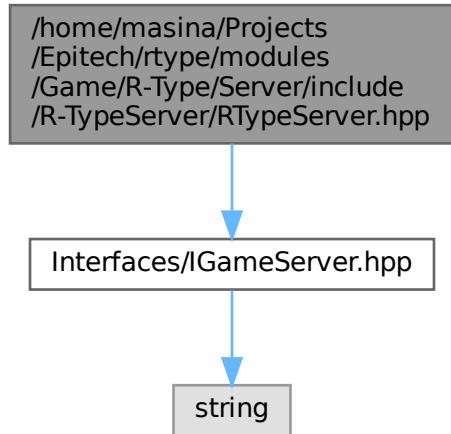
[Go to the documentation of this file.](#)

```
00001 #include "R-TypeClient/RTypClient.hpp"
00002 #include "R-TypeClient/LobbyScene.hpp"
00003
00004 gme::RTypClient::RTypClient() : m_currentScene(std::make_unique<LobbyScene>()) {
    AGameClient::setName("R-Type");
}
00005
00006 void gme::RTypClient::update(const float deltaTime, const unsigned int width, const unsigned int height) {}
```

8.103 /home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Server/include/R-TypeServer/RTypServer.hpp File Reference

RTyp 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](#)

8.103.1 Detailed Description

RType client class declaration.

Definition in file [RTypeServer.hpp](#).

8.104 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 // @class RTypeServer
00016 // @brief Class for the R-Type game
00017 // @namespace gme
00018 /**
00019 class RTypeServer final : public IGameServer
00020 {
00021     public:
00022         RTypeServer() = default;
00023         ~RTypeServer() override = default;
00024
00025         RTypeServer(const RTypeServer &) = delete;
00026         RTypeServer &operator=(const RTypeServer &) = delete;
00027         RTypeServer(RTypeServer &&) = delete;
00028         RTypeServer &operator=(RTypeServer &&) = delete;
00029
00030     private:
00031 };
00032 } // namespace gme
```

8.105 /home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Server/src/rtypeServer.cpp File Reference

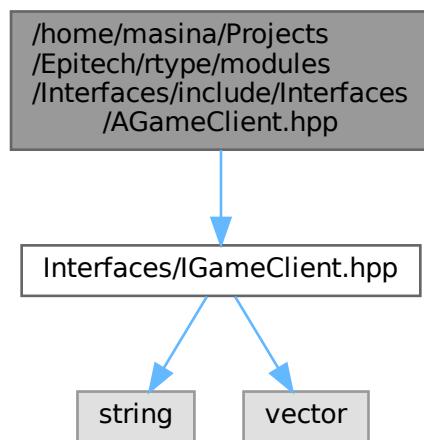
8.106 rtypeServer.cpp

[Go to the documentation of this file.](#)
00001

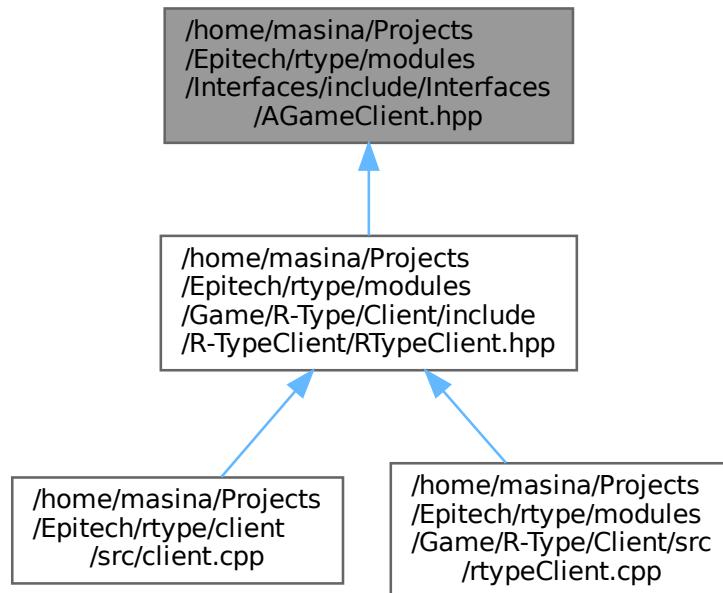
8.107 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/AGameClient.hpp File Reference

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](#)

8.107.1 Detailed Description

This file contains the game abstract class.

Definition in file [AGameClient.hpp](#).

8.108 AGameClient.hpp

[Go to the documentation of this file.](#)

```

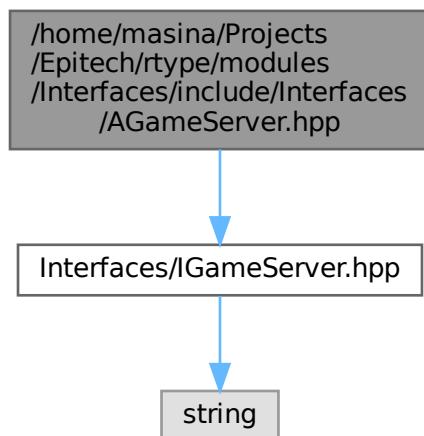
00001 /**
00002 /// @file AGameClient.hpp
00003 /// @brief This file contains the game abstract class
00004 /// @namespace gme
00005 /**
00006
00007 #pragma once
  
```

```
00008
00009 #include "Interfaces/IGameClient.hpp"
00010
00011 namespace gme
00012 {
00013
00014 /**
00015  * @class AGameClient
00016  * @brief Abstraction for the games
00017  * @namespace gme
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
```

8.109 /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](#)

8.109.1 Detailed Description

This file contains the game abstract class.

Definition in file [AGameServer.hpp](#).

8.110 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

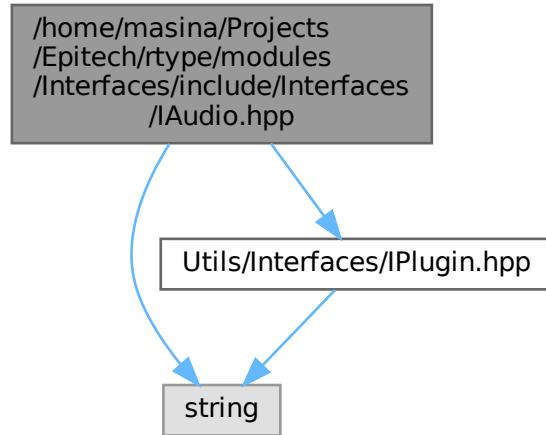
```

8.111 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IAudio.hpp File Reference

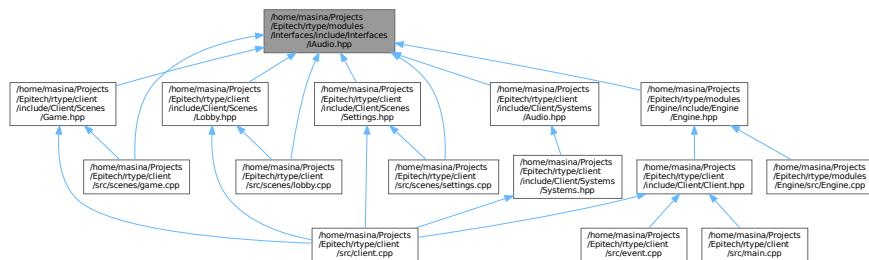
This file contains the Audio interface.

```
#include <string>
#include "Utils/Interfaces/IPlugin.hpp"
```

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` }

8.111.1 Detailed Description

This file contains the Audio interface.

Definition in file [IAudio.hpp](#).

8.112 IAudio.hpp

[Go to the documentation of this file.](#)

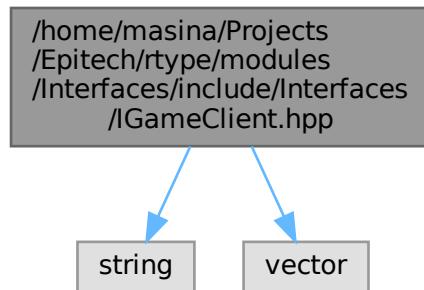
```
00001 /**
00002  * @file IAudio.hpp
00003  * @brief This file contains the Audio interface
00004  * @namespace eng
00005 /**
00006
00007 #pragma once
00008
00009 #include <string>
00010
00011 #include "Utils/Interfaces/IPPlugin.hpp"
00012
00013 namespace eng
00014 {
00015
00016     enum class Status
00017     {
00018         Stopped,
00019         Paused,
00020         Playing
00021     };
00022
00023 /**
00024  * @class IAudio
00025  * @brief Interface for the audio
00026  * @namespace eng
00027 /**
00028 class IAudio : public utl::IPPlugin
00029 {
00030     public:
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
```

8.113 /home/masina/Projects/Epitech/rtype/modules/← Interfaces/include/Interfaces/IGameClient.hpp File Reference

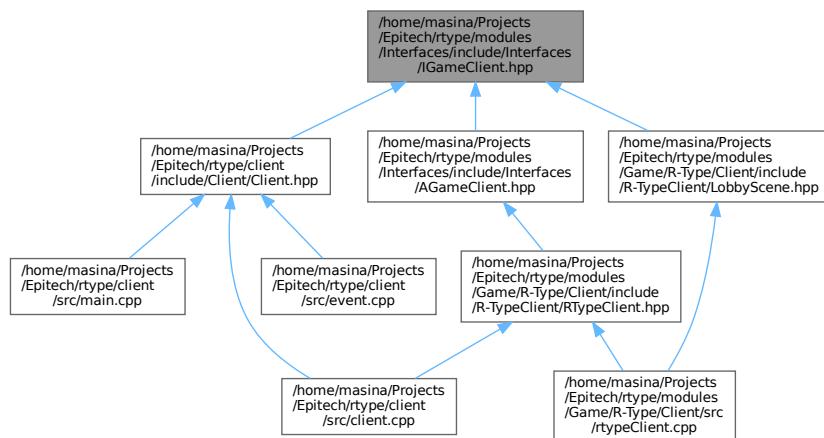
This file contains the Game interface.

```
#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`

8.113.1 Detailed Description

This file contains the Game interface.

Definition in file [IGameClient.hpp](#).

8.114 IGameClient.hpp

[Go to the documentation of this file.](#)

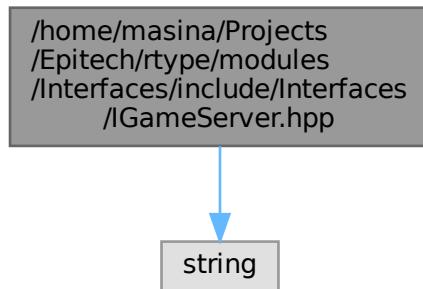
```
00001 /**
00002  * @file IGameClient.hpp
00003  * @brief This file contains the Game interface
00004  * @namespace gme
00005 /**
00006
00007 #pragma once
00008
00009 #include <string>
00010 #include <vector>
00011
00012 namespace gme
00013 {
00014
00015     struct Sprite
00016     { // TODO(bobis33): 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
```

8.115 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IGameServer.hpp File Reference

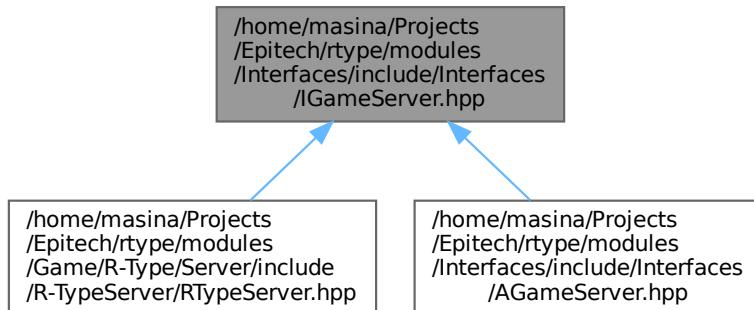
This file contains the Game interface.

```
#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](#)

8.115.1 Detailed Description

This file contains the Game interface.

Definition in file [IGameServer.hpp](#).

8.116 IGameServer.hpp

[Go to the documentation of this file.](#)

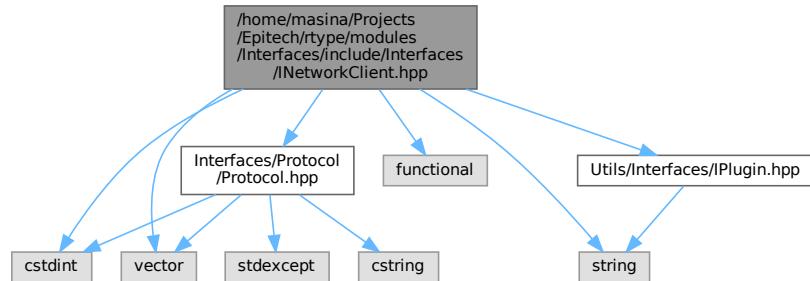
```
00001 /**
00002 /// @file IGameServer.hpp
00003 /// @brief This file contains the Game interface
00004 /// @namespace gme
00005 /**
00006
00007 #pragma once
00008
00009 #include <string>
00010
00011 namespace gme
00012 {
00013
00014 /**
00015 /// @class IGameServer
00016 /// @brief Interface for the games
00017 /// @namespace gme
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
```

8.117 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/INetworkClient.hpp File Reference

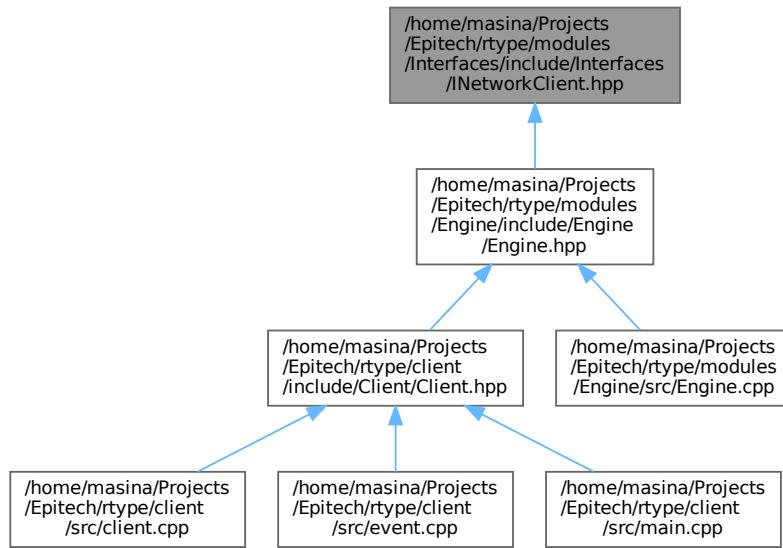
This file contains the client network interface.

```
#include <cstdint>
#include <functional>
#include <string>
#include <vector>
#include "Interfaces/Protocol/Protocol.hpp"
#include "Utils/Interfaces/IPlugin.hpp"
```

Include dependency graph for INetworkClient.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [eng::INetworkClient](#)
Interface for the client network.

Namespaces

- namespace [eng](#)

8.117.1 Detailed Description

This file contains the client network interface.

Definition in file [INetworkClient.hpp](#).

8.118 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 #include <cstdint>
00010 #include <functional>
00011 #include <string>
00012 #include <vector>
  
```

```

00013
00014 #include "Interfaces/Protocol/Protocol.hpp"
00015 #include "Utils/Interfaces/IPlugin.hpp"
00016
00017 namespace eng
00018 {
00019
00020     /**
00021     /// @class INetworkClient
00022     /// @brief Interface for the client network
00023     /// @namespace eng
00024     ///
00025     class INetworkClient : public utl::IPlugin
00026     {
00027         public:
00028             using PacketHandler = std::function<void(const rnp::PacketHeader &, const std::vector<uint8_t> &)>;
00029
00030             virtual ~INetworkClient() = default;
00031
00032             // Connection management
00033             virtual void connect(const std::string &host, uint16_t port) = 0;
00034             virtual void disconnect() = 0;
00035
00036             // Protocol messages
00037             virtual void sendConnect(const std::string &playerName) = 0;
00038             virtual void sendConnectWithCaps(const std::string &playerName, std::uint32_t clientCaps) = 0;
00039             virtual void sendDisconnect() = 0;
00040             virtual void sendDisconnect(rnp::DisconnectReason reason) = 0;
00041             virtual void sendPlayerInput(uint8_t direction, uint8_t shooting) = 0;
00042             virtual void sendPlayerInputAsEvent(std::uint16_t playerId, uint8_t direction, uint8_t shooting,
00043                                                 std::uint32_t clientTimeMs) = 0;
00044             virtual void sendPing() = 0;
00045             virtual void sendPing(std::uint32_t nonce, std::uint32_t sendTimeMs) = 0;
00046             virtual void sendAck(std::uint32_t cumulative, std::uint32_t ackBits) = 0;
00047
00048             // Handler management
00049             virtual void setPacketHandler(rnp::PacketType type, PacketHandler handler) = 0;
00050             virtual void setEventsHandler(std::function<void(const std::vector<rnp::EventRecord> &)> handler) = 0;
00051
00052             // Getters
00053             virtual std::uint32_t getSessionId() const = 0;
00054             virtual std::uint16_t getServerTickRate() const = 0;
00055
00056         private:
00057     }; // class INetworkClient
00058
00059 } // namespace eng

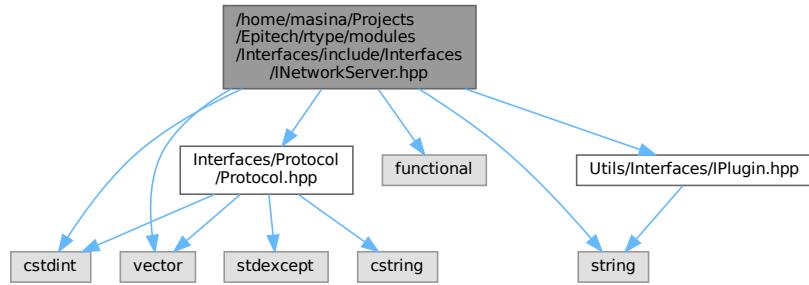
```

8.119 /home/masina/Projects/Epitech/rtype/modules/← Interfaces/include/Interfaces/INetworkServer.hpp File Reference

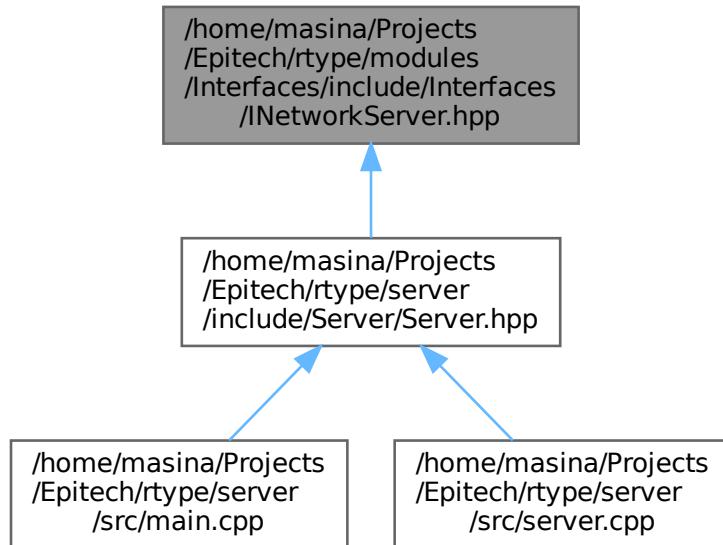
This file contains the server network interface.

```
#include <cstdint>
#include <functional>
#include <string>
#include <vector>
#include "Interfaces/Protocol/Protocol.hpp"
#include "Utils/Interfaces/IPlugin.hpp"
```

Include dependency graph for INetworkServer.hpp:



This graph shows which files directly or indirectly include this file:



Classes

- class [srv::INetworkServer](#)
Interface for the server network.

Namespaces

- namespace [srv](#)

Variables

- `constexpr size_t srv::MAX_CLIENTS = 16`
- `constexpr size_t srv::MAX_IP_LENGTH = 8`
- `constexpr size_t srv::MAX_LEN_RECV_BUFFER = 1024`

8.119.1 Detailed Description

This file contains the server network interface.

Definition in file [INetworkServer.hpp](#).

8.120 INetworkServer.hpp

[Go to the documentation of this file.](#)

```

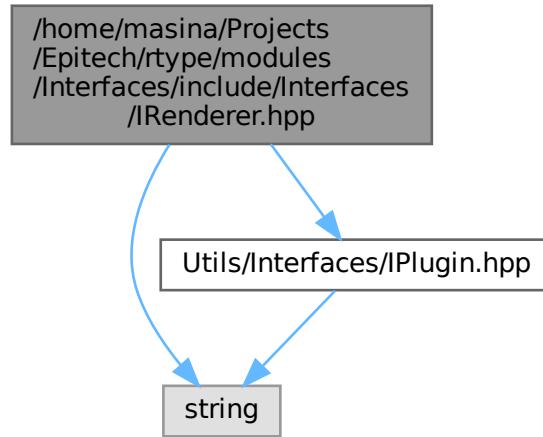
00001 /**
00002 /// @file INetworkServer.hpp
00003 /// @brief This file contains the server network interface
00004 /// @namespace srv
00005 /**
00006
00007 #pragma once
00008
00009 #include <cstdint>
00010 #include <functional>
00011 #include <string>
00012 #include <vector>
00013
00014 #include "Interfaces/Protocol/Protocol.hpp"
00015 #include "Utils/Interfaces/IPlugin.hpp"
00016
00017 namespace srv
00018 {
00019
00020     constexpr size_t MAX_CLIENTS = 16;
00021     constexpr size_t MAX_IP_LENGTH = 8;
00022     constexpr size_t MAX_LEN_RECV_BUFFER = 1024;
00023
00024 /**
00025 /// @class INetworkServer
00026 /// @brief Interface for the server network
00027 /// @namespace srv
00028 /**
00029 class INetworkServer : public utl::IPlugin
00030 {
00031     public:
00032         virtual ~INetworkServer() = default;
00033
00034         virtual void init(const std::string &host, uint16_t port) = 0;
00035         virtual void start() = 0;
00036         virtual void stop() = 0;
00037
00038     // Configuration
00039         virtual void setTickRate(uint16_t tickRate) = 0;
00040         virtual void setServerCapabilities(uint32_t caps) = 0;
00041
00042     private:
00043 }; // class INetworkServer
00044
00045 } // namespace srv

```

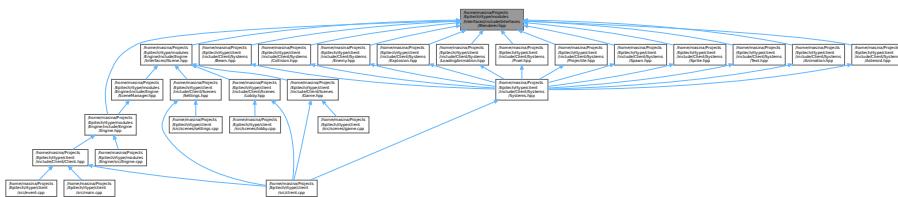
8.121 /home/masina/Projects/Epitech/rtype/modules/← Interfaces/include/Interfaces/IRenderer.hpp File Reference

This file contains the IRenderer class declaration.

```
#include <string>
#include "Utils/Interfaces/IPlugin.hpp"
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::Enter , 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 }

8.121.1 Detailed Description

This file contains the IRenderer class declaration.

Definition in file [IRenderer.hpp](#).

8.122 IRenderer.hpp

[Go to the documentation of this file.](#)

```

00001 /**
00002 /// @file IRenderer.hpp
00003 /// @brief This file contains the IRenderer class declaration
00004 /// @namespace eng
00005 /**
00006
00007 #pragma once
00008
00009 #include <string>
00010
00011 #include "Utils/Interfaces/IPlugin.hpp"
00012
00013 namespace eng
00014 {
00015     struct Color
00016     {
00017         unsigned char r;
00018         unsigned char g;
00019         unsigned char b;
00020         unsigned char a;
00021     };
00022     struct Text
00023     {
00024         std::string font_name;
00025         Color color;
00026         std::string content;
00027         unsigned int size;
00028         float x;
00029         float y;
00030         std::string name;
00031     };
00032
00033     enum class Key
00034     {
00035         Unknown,
00036         Escape,
00037         Enter,
00038         Space,
00039         Up,
00040         Down,
00041         Left,
00042         Right,
00043         A,
00044         B,

```

```

00045     C,
00046     D,
00047     E,
00048     F,
00049     G,
00050     H,
00051     I,
00052     J,
00053     K,
00054     L,
00055     M,
00056     N,
00057     O,
00058     P,
00059     Q,
00060     R,
00061     S,
00062     T,
00063     U,
00064     V,
00065     W,
00066     X,
00067     Y,
00068     Z,
00069     Num0,
00070     Num1,
00071     Num2,
00072     Num3,
00073     Num4,
00074     Num5,
00075     Num6,
00076     Num7,
00077     Num8,
00078     Num9
00079 };
00080 enum class EventType
00081 {
00082     Closed,
00083     KeyPressed,
00084     KeyReleased,
00085     None
00086 };
00087
00088 struct Event
00089 {
00090     EventType type = EventType::None;
00091     Key key = Key::Unknown;
00092 };
00093
00094 struct WindowSize
00095 {
00096     unsigned int width;
00097     unsigned int height;
00098 };
00099
00100 /**
00101 * @class IRenderer
00102 * @brief Interface for the renderer
00103 * @namespace eng
00104 */
00105 class IRenderer : public utl::IPlugin
00106 {
00107
00108     public:
00109         virtual void createWindow(const std::string &title, unsigned int height, unsigned int width,
00110             unsigned int frameLimit, bool fullscreen) = 0;
00111         [[nodiscard]] virtual bool windowIsOpen() const = 0;
00112         virtual void closeWindow() = 0;
00113         virtual void clearWindow(Color color) = 0;
00114         virtual void displayWindow() = 0;
00115         [[nodiscard]] virtual WindowSize getWindowSize() = 0;
00116
00117         [[nodiscard]] virtual bool pollEvent(Event &event) = 0;
00118         virtual void setFrameLimit(unsigned int frameLimit) = 0;
00119
00120         virtual void createFont(const std::string &name, const std::string &path) = 0;
00121         virtual void createText(Text text) = 0;
00122         virtual void drawText(const std::string &name) = 0;
00123         virtual void setTextColor(const std::string &name, const std::string &content) = 0;
00124         virtual void setTextPosition(const std::string &name, float x, float y) = 0;
00125         virtual void setTextColor(const std::string &name, Color color) = 0;
00126
00127         virtual void createTexture(const std::string &name, const std::string &path) = 0;
00128         virtual void createSprite(const std::string &name, const std::string &textureName, float x, float y,
00129             float scale_x = 1, float scale_y = 1, int fx = 0, int fy = 0, int fnx = -1,
00130             int fny = -1) = 0;
00131         virtual void drawSprite(const std::string &name) = 0;

```

```

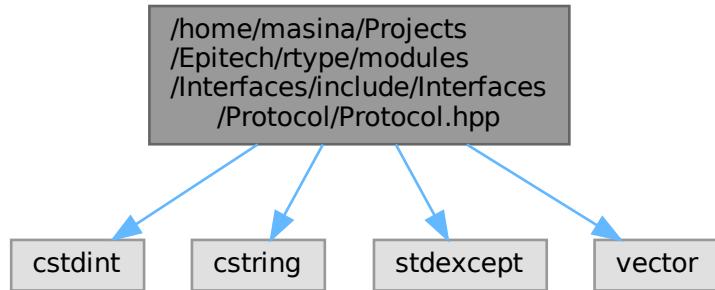
00132     virtual void setSpritePosition(const std::string &name, float x, float y) = 0;
00133     virtual void setSpriteTexture(const std::string &name, const std::string &path) = 0;
00134     virtual void setSpriteScale(const std::string &name, int x, int y) = 0;
00135     virtual void setSpriteFrame(const std::string &name, int fx, int fy, int fnx, int fny) = 0;
00136
00137     virtual void drawPoint(float x, float y, Color color) = 0;
00138
00139     private:
00140 };
00141 // class IRenderer
00142 } // namespace eng

```

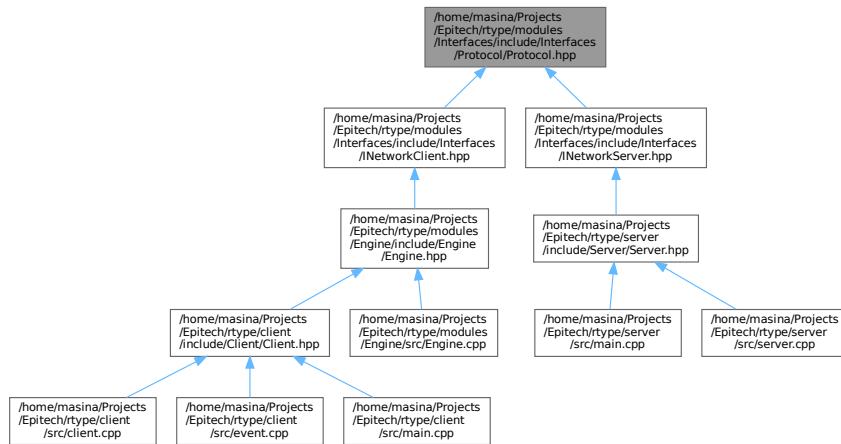
8.123 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/Protocol/Protocol.hpp File Reference

This file contains the network protocol.

```
#include <cstdint>
#include <cstring>
#include <stdexcept>
#include <vector>
Include dependency graph for Protocol.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- **struct rnp::EventRecord**
Event record for ENTITY_EVENT packets (TLV format)
- **struct rnp::PacketHeader**
Packet header according to RNP specification (Big Endian) Total size: 16 bytes.
- **struct rnp::PacketConnect**
CONNECT packet payload.
- **struct rnp::PacketConnectAccept**
CONNECT_ACCEPT packet payload.
- **struct rnp::PacketDisconnect**
DISCONNECT packet payload.
- **struct rnp::EntityState**
Entity state for WORLD_STATE packet.
- **struct rnp::PacketWorldState**
WORLD_STATE packet payload.
- **struct rnp::PacketPingPong**
PING/PONG packet payload.
- **struct rnp::PacketAck**
ACK packet payload.
- **struct rnp::PacketError**
ERROR packet payload.
- **struct rnp::FragmentHeader**
Fragmentation header (when FRAG flag is set)

Namespaces

- namespace **rnp**

Enumerations

- enum class `rnp::PacketType` : std::uint8_t {

 `rnp::CONNECT` = 0x01 , `rnp::DISCONNECT` = 0x02 , `rnp::WORLD_STATE` = 0x03 , `rnp::PING` = 0x04 ,

 `rnp::PONG` = 0x05 , `rnp::PACKET_ERROR` = 0x06 , `rnp::ACK` = 0x07 , `rnp::ENTITY_EVENT` = 0x08 ,

 `rnp::CONNECT_ACCEPT` = 0x09 , `rnp::PLAYER_INPUT` = 0x03 }

 Packet types according to RNP specification.
- enum class `rnp::PacketFlags` : std::uint16_t {

 `rnp::NONE` = 0x0000 , `rnp::ACK_REQ` = 0x0001 , `rnp::RELIABLE` = 0x0002 , `rnp::FRAG` = 0x0004 ,

 `rnp::COMPRESSED` = 0x0008 }

 Packet flags for reliability and fragmentation.
- enum class `rnp::DisconnectReason` : std::uint16_t {

 `rnp::UNSPECIFIED` = 0 , `rnp::CLIENT_REQUEST` = 1 , `rnp::TIMEOUT` = 2 , `rnp::PROTOCOL_ERROR` = 3 ,

 `rnp::SERVER_SHUTDOWN` = 4 , `rnp::SERVER_FULL` = 5 , `rnp::BANNED` = 6 }

 Disconnect reason codes.
- enum class `rnp::ErrorCode` : std::uint16_t { `rnp::INVALID_PAYLOAD` = 1 , `rnp::UNAUTHORIZED_SESSION` = 2 , `rnp::RATE_LIMITED` = 3 , `rnp::INTERNAL_ERROR` = 4 }

 Error codes.
- enum class `rnp::EventType` : std::uint8_t {

 `rnp::SPAWN` = 0x01 , `rnp::DESPAWN` = 0x02 , `rnp::DAMAGE` = 0x03 , `rnp::SCORE` = 0x04 ,

 `rnp::POWERUP` = 0x05 , `rnp::INPUT` = 0x06 , `rnp::CUSTOM` = 0xFF }

 Event types for ENTITY_EVENT packets.
- enum class `rnp::EntityType` : std::uint16_t {

 `rnp::PLAYER` = 0x01 , `rnp::ENEMY` = 0x02 , `rnp::PROJECTILE` = 0x03 , `rnp::POWERUP` = 0x04 ,

 `rnp::OBSTACLE` = 0x05 }

 Entity types for world state.

Functions

- std::vector< std::uint8_t > `rnp::serializeEvents` (const std::vector< `EventRecord` > &events)

 Serialize events in ENTITY_EVENT format (TLV with entity_id) Format per event: type(1) | entity_id(4, BE) | data_len(1) | data(data_len)
- std::vector< `EventRecord` > `rnp::deserializeEvents` (const std::uint8_t *payload, const std::size_t length)

 Deserialize ENTITY_EVENT payload into event records Format per event: type(1) | entity_id(4, BE) | data_len(1) | data(data_len)
- std::vector< uint8_t > `rnp::serializeHeader` (const `PacketHeader` &header)

 Serialize packet header (Big Endian as per RNP spec)
- std::vector< uint8_t > `rnp::serialize` (const `PacketHeader` &header, const uint8_t *payload=nullptr)

 Serialize packet with header and optional payload (Big Endian)
- `PacketHeader rnp::deserializeHeader` (const uint8_t *data, const std::size_t size)

 Deserialize packet header (Big Endian)

Variables

- constexpr std::uint8_t `rnp::PROTOCOL_VERSION` = 1
- constexpr std::size_t `rnp::MAX_PAYLOAD` = 512

8.123.1 Detailed Description

This file contains the network protocol.

Definition in file [Protocol.hpp](#).

8.124 Protocol.hpp

[Go to the documentation of this file.](#)

```

00001 /**
00002  * @file Protocol.hpp
00003  * @brief This file contains the network protocol
00004  * @namespace rnp
00005 /**
00006
00007 #pragma once
00008
00009 #include <cstdint>
00010 #include <cstring>
00011 #include <stdexcept>
00012 #include <vector>
00013
00014 namespace rnp
00015 {
00016
00017     inline constexpr std::uint8_t PROTOCOL_VERSION = 1;
00018     inline constexpr std::size_t MAX_PAYLOAD = 512;
00019
00020 /**
00021  * @brief Packet types according to RNP specification
00022 /**
00023     enum class PacketType : std::uint8_t
00024     {
00025         CONNECT = 0x01,
00026         DISCONNECT = 0x02,
00027         WORLD_STATE = 0x03,
00028         PING = 0x04,
00029         PONG = 0x05,
00030         PACKET_ERROR = 0x06,
00031         ACK = 0x07,
00032         ENTITY_EVENT = 0x08,
00033         CONNECT_ACCEPT = 0x09,
00034         PLAYER_INPUT = 0x03 // Deprecated: use ENTITY_EVENT with INPUT type
00035     };
00036
00037 /**
00038  * @brief Packet flags for reliability and fragmentation
00039 /**
00040     enum class PacketFlags : std::uint16_t
00041     {
00042         NONE = 0x0000,
00043         ACK_REQ = 0x0001,
00044         RELIABLE = 0x0002,
00045         FRAG = 0x0004,
00046         COMPRESSED = 0x0008
00047     };
00048
00049 /**
00050  * @brief Disconnect reason codes
00051 /**
00052     enum class DisconnectReason : std::uint16_t
00053     {
00054         UNSPECIFIED = 0,
00055         CLIENT_REQUEST = 1,
00056         TIMEOUT = 2,
00057         PROTOCOL_ERROR = 3,
00058         SERVER_SHUTDOWN = 4,
00059         SERVER_FULL = 5,
00060         BANNED = 6
00061     };
00062
00063 /**
00064  * @brief Error codes
00065 /**
00066     enum class ErrorCode : std::uint16_t
00067     {
00068         INVALID_PAYLOAD = 1,
00069         UNAUTHORIZED_SESSION = 2,
00070         RATE_LIMITED = 3,

```

```

00071     INTERNAL_ERROR = 4
00072 };
00073
00074 /**
00075  * @brief Event types for ENTITY_EVENT packets
00076 */
00077 enum class EventType : std::uint8_t
00078 {
00079     SPAWN = 0x01,
00080     DESPAWN = 0x02,
00081     DAMAGE = 0x03,
00082     SCORE = 0x04,
00083     POWERUP = 0x05,
00084     INPUT = 0x06,
00085     CUSTOM = 0xFF
00086 };
00087
00088 /**
00089  * @brief Entity types for world state
00090 */
00091 enum class EntityType : std::uint16_t
00092 {
00093     PLAYER = 0x01,
00094     ENEMY = 0x02,
00095     PROJECTILE = 0x03,
00096     POWERUP = 0x04,
00097     OBSTACLE = 0x05
00098 };
00099
00100 /**
00101  * @brief Event record for ENTITY_EVENT packets (TLV format)
00102 */
00103 struct EventRecord
00104 {
00105     EventType type;
00106     std::uint32_t entityId;
00107     std::vector<std::uint8_t> data;
00108 };
00109
00110 /**
00111  * @brief Packet header according to RNP specification (Big Endian)
00112  * Total size: 16 bytes
00113 */
00114 struct PacketHeader
00115 {
00116     std::uint8_t type;      // PacketType
00117     std::uint16_t length;   // Payload length in bytes
00118     std::uint16_t flags;    // PacketFlags bitfield
00119     std::uint16_t reserved; // Must be 0
00120     std::uint32_t sequence; // Per-session, monotonic sequence number
00121     std::uint32_t sessionId; // Server-assigned session ID
00122 };
00123
00124 /**
00125  * @brief CONNECT packet payload
00126 */
00127 struct PacketConnect
00128 {
00129     std::uint8_t nameLen;
00130     char playerName[32];
00131     std::uint32_t clientCaps;
00132 };
00133
00134 /**
00135  * @brief CONNECT_ACCEPT packet payload
00136 */
00137 struct PacketConnectAccept
00138 {
00139     std::uint32_t sessionId;
00140     std::uint16_t tickRateHz;
00141     std::uint16_t mtuPayloadBytes;
00142     std::uint32_t serverCaps;
00143 };
00144
00145 /**
00146  * @brief DISCONNECT packet payload
00147 */
00148 struct PacketDisconnect
00149 {
00150     std::uint16_t reasonCode; // DisconnectReason
00151 };
00152
00153 /**
00154  * @brief Entity state for WORLD_STATE packet
00155 */
00156 struct EntityState
00157 {

```

```

00158     std::uint32_t id;
00159     std::uint16_t type; // EntityType
00160     float x, y;
00161     float vx, vy;
00162     std::uint8_t stateFlags;
00163 };
00164
00165 /**
00166  * @brief WORLD_STATE packet payload
00167 */
00168 struct PacketWorldState
00169 {
00170     std::uint32_t serverTick;
00171     std::uint16_t entityCount;
00172     std::vector<EntityState> entities;
00173 };
00174
00175 /**
00176  * @brief PING/PONG packet payload
00177 */
00178 struct PacketPingPong
00179 {
00180     std::uint32_t nonce;
00181     std::uint32_t sendTimeMs;
00182 };
00183
00184 /**
00185  * @brief ACK packet payload
00186 */
00187 struct PacketAck
00188 {
00189     std::uint32_t cumulativeAck;
00190     std::uint32_t ackBits; // 32-bit SACK window
00191 };
00192
00193 /**
00194  * @brief ERROR packet payload
00195 */
00196 struct PacketError
00197 {
00198     std::uint16_t errorCode; // ErrorCode
00199     std::uint16_t msgLen;
00200     std::string description;
00201 };
00202
00203 /**
00204  * @brief Fragmentation header (when FRAG flag is set)
00205 */
00206 struct FragmentHeader
00207 {
00208     std::uint16_t fragId;
00209     std::uint16_t fragIndex;
00210     std::uint16_t fragCount;
00211 };
00212
00213 /**
00214  * @brief Serialize events in ENTITY_EVENT format (TLV with entity_id)
00215  * Format per event: type(1) | entity_id(4, BE) | data_len(1) | data(data_len)
00216 */
00217 inline std::vector<std::uint8_t> serializeEvents(const std::vector<EventRecord> &events)
00218 {
00219     std::vector<std::uint8_t> payload;
00220     payload.reserve(64);
00221
00222     for (const auto &ev : events)
00223     {
00224         const std::uint8_t dataLen = static_cast<std::uint8_t>(ev.data.size());
00225
00226         // Event type (1 byte)
00227         payload.push_back(static_cast<std::uint8_t>(ev.type()));
00228
00229         // Entity ID (4 bytes, big endian)
00230         payload.push_back(static_cast<std::uint8_t>((ev.entityId >> 24) & 0xFF));
00231         payload.push_back(static_cast<std::uint8_t>((ev.entityId >> 16) & 0xFF));
00232         payload.push_back(static_cast<std::uint8_t>((ev.entityId >> 8) & 0xFF));
00233         payload.push_back(static_cast<std::uint8_t>(ev.entityId & 0xFF));
00234
00235         // Data length (1 byte)
00236         payload.push_back(dataLen);
00237
00238         // Data (dataLen bytes)
00239         payload.insert(payload.end(), ev.data.begin(), ev.data.end());
00240
00241         if (payload.size() > MAX_PAYLOAD)
00242         {
00243             throw std::runtime_error("Events payload exceeds MAX_PAYLOAD");
00244         }
    
```

```

00245     }
00246     return payload;
00247 }
00248
00249 /**
00250 /// @brief Deserialize ENTITY_EVENT payload into event records
00251 /// Format per event: type(1) | entity_id(4, BE) | data_len(1) | data(data_len)
00252 /**
00253 inline std::vector<EventRecord> deserializeEvents(const std::uint8_t *payload, const std::size_t length)
00254 {
00255     std::vector<EventRecord> events;
00256     std::size_t offset = 0;
00257
00258     while (offset < length)
00259     {
00260         if (length - offset < 6) // type(1) + entity_id(4) + data_len(1)
00261         {
00262             throw std::runtime_error("Truncated event header in payload");
00263         }
00264
00265         const EventType type = static_cast<EventType>(payload[offset]);
00266         const std::uint32_t entityId = (static_cast<std::uint32_t>(payload[offset + 1]) « 24) |
00267             (static_cast<std::uint32_t>(payload[offset + 2]) « 16) |
00268             (static_cast<std::uint32_t>(payload[offset + 3]) « 8) |
00269             static_cast<std::uint32_t>(payload[offset + 4]);
00270         const std::uint8_t dataLen = payload[offset + 5];
00271         offset += 6;
00272
00273         if (length - offset < dataLen)
00274         {
00275             throw std::runtime_error("Truncated event data in payload");
00276         }
00277
00278         EventRecord rec{type, entityId, {}};
00279         if (dataLen > 0)
00280         {
00281             rec.data.insert(rec.data.end(), payload + offset, payload + offset + dataLen);
00282         }
00283         events.emplace_back(std::move(rec));
00284         offset += dataLen;
00285     }
00286     return events;
00287 }
00288
00289 /**
00290 /// @brief Serialize packet header (Big Endian as per RNP spec)
00291 /**
00292 inline std::vector<uint8_t> serializeHeader(const PacketHeader &header)
00293 {
00294     std::vector<uint8_t> buffer(16); // Fixed header size
00295
00296     buffer[0] = header.type;
00297
00298     // length (2 bytes, big endian)
00299     buffer[1] = static_cast<uint8_t>((header.length » 8) & 0xFF);
00300     buffer[2] = static_cast<uint8_t>(header.length & 0xFF);
00301
00302     // flags (2 bytes, big endian)
00303     buffer[3] = static_cast<uint8_t>((header.flags » 8) & 0xFF);
00304     buffer[4] = static_cast<uint8_t>(header.flags & 0xFF);
00305
00306     // reserved (2 bytes)
00307     buffer[5] = static_cast<uint8_t>((header.reserved » 8) & 0xFF);
00308     buffer[6] = static_cast<uint8_t>(header.reserved & 0xFF);
00309
00310     // sequence (4 bytes, big endian)
00311     buffer[7] = static_cast<uint8_t>((header.sequence » 24) & 0xFF);
00312     buffer[8] = static_cast<uint8_t>((header.sequence » 16) & 0xFF);
00313     buffer[9] = static_cast<uint8_t>((header.sequence » 8) & 0xFF);
00314     buffer[10] = static_cast<uint8_t>(header.sequence & 0xFF);
00315
00316     // sessionId (4 bytes, big endian)
00317     buffer[11] = static_cast<uint8_t>((header.sessionId » 24) & 0xFF);
00318     buffer[12] = static_cast<uint8_t>((header.sessionId » 16) & 0xFF);
00319     buffer[13] = static_cast<uint8_t>((header.sessionId » 8) & 0xFF);
00320     buffer[14] = static_cast<uint8_t>(header.sessionId & 0xFF);
00321
00322     buffer[15] = 0; // Padding to 16 bytes
00323
00324     return buffer;
00325 }
00326
00327 /**
00328 /// @brief Serialize packet with header and optional payload (Big Endian)
00329 /**
00330 inline std::vector<uint8_t> serialize(const PacketHeader &header, const uint8_t *payload = nullptr)
00331 {

```

```

00332     std::vector<uint8_t> buffer = serializeHeader(header);
00333
00334     if (payload && header.length > 0)
00335     {
00336         buffer.insert(buffer.end(), payload, payload + header.length);
00337     }
00338
00339     return buffer;
00340 }
00341
00342 /**
00343 * @brief Deserialize packet header (Big Endian)
00344 */
00345 inline PacketHeader deserializeHeader(const uint8_t *data, const std::size_t size)
00346 {
00347     if (size < 16)
00348     {
00349         throw std::runtime_error("Buffer too small for header");
00350     }
00351
00352     PacketHeader header;
00353
00354     header.type = data[0];
00355
00356     // length (2 bytes, big endian)
00357     header.length = (static_cast<std::uint16_t>(data[1]) << 8) | static_cast<std::uint16_t>(data[2]);
00358
00359     // flags (2 bytes, big endian)
00360     header.flags = (static_cast<std::uint16_t>(data[3]) << 8) | static_cast<std::uint16_t>(data[4]);
00361
00362     // reserved (2 bytes)
00363     header.reserved = (static_cast<std::uint16_t>(data[5]) << 8) | static_cast<std::uint16_t>(data[6]);
00364
00365     // sequence (4 bytes, big endian)
00366     header.sequence = (static_cast<std::uint32_t>(data[7]) << 24) | (static_cast<std::uint32_t>(data[8]) << 16) |
00367             (static_cast<std::uint32_t>(data[9]) << 8) | static_cast<std::uint32_t>(data[10]);
00368
00369     // sessionId (4 bytes, big endian)
00370     header.sessionId = (static_cast<std::uint32_t>(data[11]) << 24) | (static_cast<std::uint32_t>(data[12]) << 16) |
00371             (static_cast<std::uint32_t>(data[13]) << 8) | static_cast<std::uint32_t>(data[14]);
00372
00373     return header;
00374 }
00375
00376 } // namespace rnp

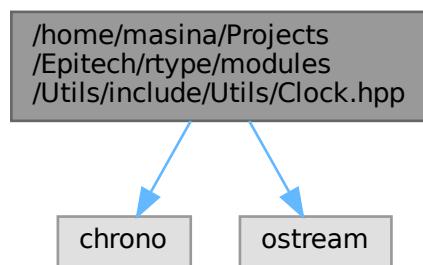
```

8.125 /home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/Clock.hpp File Reference

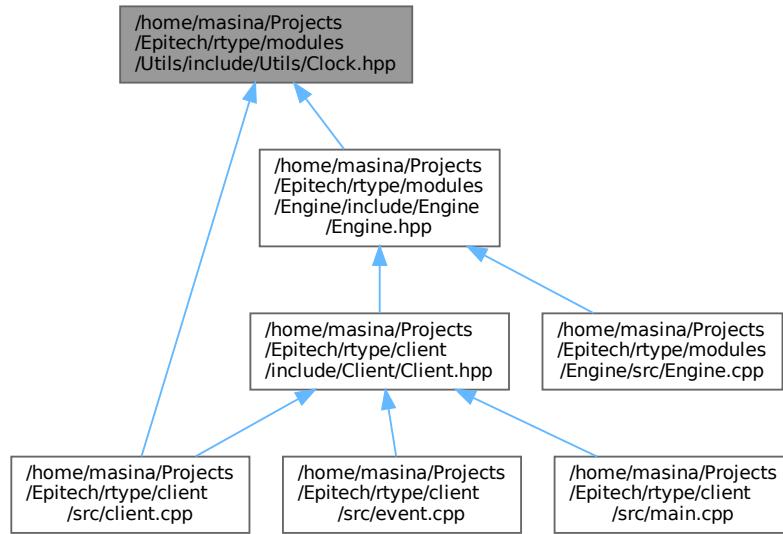
This file contains the Clock class.

```
#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](#)

8.125.1 Detailed Description

This file contains the Clock class.

Definition in file [Clock.hpp](#).

8.126 Clock.hpp

[Go to the documentation of this file.](#)

```

00001 /**
00002 /**
00003 /**
00004 /**
00005 /**
00006
00007 #pragma once
00008
00009 #include <chrono>
00010 #include <iostream>
00011
00012 namespace utl
00013 {
  
```

```

00014
00015  /**
00016  * @class Clock
00017  * @brief Class for clock
00018  * @namespace utl
00019  */
00020 class Clock
00021 {
00022
00023     public:
00024         using TimePoint = std::chrono::time_point<std::chrono::high_resolution_clock>;
00025
00026     explicit Clock(const bool startNow = true) : m_start{startNow ? now() : TimePoint{}}, m_pausedDuration{0} {}
00027     ~Clock() = default;
00028
00029     Clock(const Clock &) = delete;
00030     Clock &operator=(const Clock &) = delete;
00031     Clock(Clock &&) = delete;
00032     Clock &operator=(Clock &&) = delete;
00033
00034     friend std::ostream &operator<<(std::ostream &os, const Clock &clock)
00035     {
00036         os << "Elapsed time: " << clock.getDeltaSeconds() << " seconds";
00037         return os;
00038     }
00039
00040     static TimePoint now() { return std::chrono::high_resolution_clock::now(); }
00041     void restart()
00042     {
00043         m_start = now();
00044         m_pausedDuration = Duration(0);
00045         m_isPaused = false;
00046     }
00047     void pause()
00048     {
00049         if (!m_isPaused)
00050         {
00051             m_pausedTime = now();
00052             m_isPaused = true;
00053         }
00054     }
00055     void resume()
00056     {
00057         if (m_isPaused)
00058         {
00059             m_pausedDuration += now() - m_pausedTime;
00060             m_isPaused = false;
00061         }
00062     }
00063     [[nodiscard]] float getDeltaSeconds() const
00064     {
00065         if (m_isPaused)
00066         {
00067             return std::chrono::duration<float>(m_pausedTime - m_start - m_pausedDuration).count();
00068         }
00069         return std::chrono::duration<float>(now() - m_start - m_pausedDuration).count();
00070     }
00071
00072     template <typename Duration = std::chrono::seconds> [[nodiscard]] auto getElapsed() const
00073     {
00074         return std::chrono::duration_cast<Duration>(now() - m_start - m_pausedDuration);
00075     }
00076
00077     private:
00078         using Duration = std::chrono::high_resolution_clock::duration;
00079
00080         TimePoint m_start;
00081         TimePoint m_pausedTime;
00082         Duration m_pausedDuration;
00083         bool m_isPaused{false};
00084
00085     }; // class Clock
00086
00087 } // namespace utl

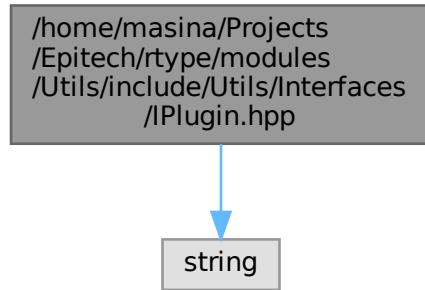
```

8.127 /home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/Interfaces/IPPlugin.hpp File Reference

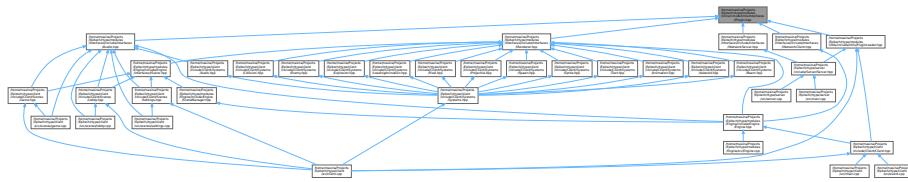
This file contains the plugin interface.

```
#include <string>
```

Include dependency graph for `IPlugin.hpp`:



This graph shows which files directly or indirectly include this file:



Classes

- interface [utl::IPlugin](#)
Interface for plugins.

Namespaces

- namespace [utl](#)

Enumerations

- enum class [utl::PluginType](#) : uint8_t {

 [utl::AUDIO](#) = 0 , [utl::NETWORK_CLIENT](#) = 1 , [utl::NETWORK_SERVER](#) = 1 ,

 [utl::RENDERER](#) = 2 ,

 [utl::UNKNOWN](#) = 255 }

8.127.1 Detailed Description

This file contains the plugin interface.

Definition in file [IPlugin.hpp](#).

8.128 IPlugin.hpp

[Go to the documentation of this file.](#)

```

00001 /**
00002 /// @file IPlugin.hpp
00003 /// @brief This file contains the plugin interface
00004 /// @namespace utl
00005 /**
00006
00007 #pragma once
00008
00009 #include <string>
00010
00011 namespace utl
00012 {
00013
00014     enum class PluginType : uint8_t
00015     {
00016         AUDIO = 0,
00017         NETWORK_CLIENT = 1,
00018         NETWORK_SERVER = 1,
00019         RENDERER = 2,
00020         UNKNOWN = 255
00021     };
00022
00023 /**
00024 /// @interface IPlugin
00025 /// @brief Interface for plugins
00026 /// @namespace utl
00027 /**
00028 class IPlugin
00029 {
00030
00031     public:
00032         virtual ~IPlugin() = default;
00033         [[nodiscard]] virtual const std::string getName() const = 0;
00034         [[nodiscard]] virtual PluginType getType() const = 0;
00035
00036 }; // interface IPlugin
00037 } // namespace utl

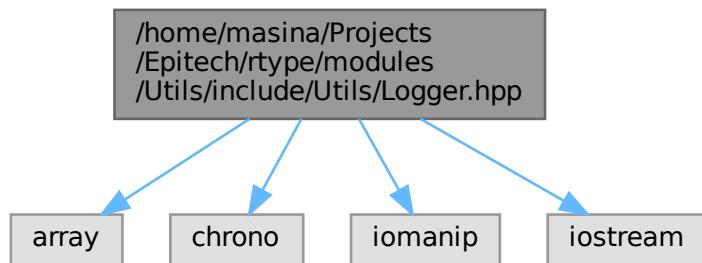
```

8.129 /home/masina/Projects/Epitech/rtype/modules/Utils/include/_UTILS/Logger.hpp File Reference

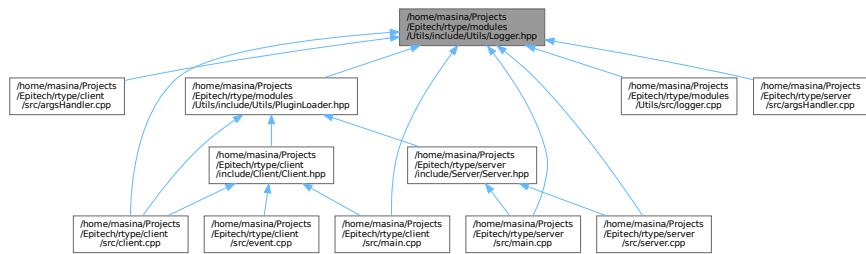
This file contains the Logger class.

```
#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](#)

Class for logging.

Namespaces

- namespace [utl](#)

Enumerations

- enum class [utl::LogLevel](#) : uint8_t { [utl::INFO](#) , [utl::WARNING](#) }

8.129.1 Detailed Description

This file contains the Logger class.

Definition in file [Logger.hpp](#).

8.130 Logger.hpp

[Go to the documentation of this file.](#)

```

00001 /**
00002  * @file Logger.hpp
00003  * @brief This file contains the Logger class
00004  * @namespace utl
00005 /**
00006
00007 #pragma once
00008
00009 #include <array>
00010 #include <chrono>
00011 #include <iomanip>
00012 #include <iostream>
00013
00014 namespace utl
00015 {
00016
00017     enum class LogLevel : uint8_t
00018     {
00019         INFO,
00020         WARNING
00021     };
00022 }
```

```

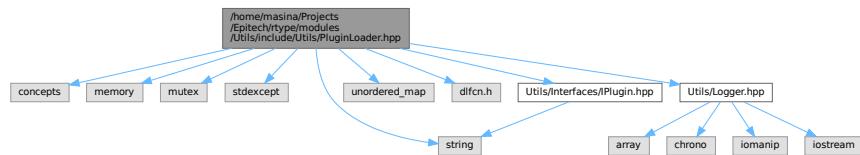
00023 ///////////////////////////////////////////////////////////////////////////////
00024 // @class Logger
00025 // @brief Class for logging
00026 // @namespace utl
00027 /**
00028 class Logger
00029 {
00030
00031     public:
00032         Logger(const Logger &) = delete;
00033         Logger &operator=(const Logger &) = delete;
00034         Logger(Logger &&) = delete;
00035         Logger &operator=(Logger &&) = delete;
00036
00037         static void init();
00038
00039         template <typename Func> static void logExecutionTime(const std::string &message, Func &&func)
00040     {
00041         const auto start = std::chrono::high_resolution_clock::now();
00042         func();
00043         const auto end = std::chrono::high_resolution_clock::now();
00044         const auto duration = std::chrono::duration<float, std::milli>(end - start).count();
00045
00046         std::cout << getColorForDuration(duration)
00047             << formatLogMessage(LogLevel::INFO, message + " took " + std::to_string(duration) + " ms")
00048             << LOG_LEVEL_COLOR[COLOR_RESET];
00049     }
00050
00051         static void log(const std::string &message, const LogLevel &logLevel)
00052     {
00053         std::cout << (logLevel == LogLevel::INFO ? LOG_LEVEL_COLOR[COLOR_INFO] :
00054             LOG_LEVEL_COLOR[COLOR_WARNING])
00055             << formatLogMessage(logLevel, message) << LOG_LEVEL_COLOR[COLOR_RESET];
00056     }
00057
00058     private:
00059         enum ColorIndex : uint8_t
00060     {
00061         COLOR_ERROR,
00062         COLOR_INFO,
00063         COLOR_WARNING,
00064         COLOR_RESET
00065     };
00066
00067         static constexpr std::array<const char *, 4> LOG_LEVEL_COLOR = {
00068             "\033[31m", // ERROR/slow execution
00069             "\033[32m", // INFO/fast execution
00070             "\033[33m", // WARNING/medium execution
00071             "\033[0m\n" // RESET + newline
00072         };
00073
00074         static constexpr std::array<const char *, 2> LOG_LEVEL_STRING = {"INFO", "WARNING"};
00075
00076         Logger() = default;
00077         ~Logger() = default;
00078
00079         [[nodiscard]] static const char *getColorForDuration(const float duration)
00080     {
00081         return duration < 20.0F
00082             ? LOG_LEVEL_COLOR[COLOR_INFO]
00083             : (duration < 90.0F ? LOG_LEVEL_COLOR[COLOR_WARNING] :
00084                 LOG_LEVEL_COLOR[COLOR_ERROR]);
00085     }
00086
00087         [[nodiscard]] static std::string formatLogMessage(LogLevel level, const std::string &message)
00088     {
00089         const auto inTime = std::chrono::system_clock::to_time_t(std::chrono::system_clock::now());
00090         std::ostringstream ss;
00091         ss << "[" << std::put_time(std::localtime(&inTime), "%Y-%m-%d %X") << "] ";
00092         ss << "[" << LOG_LEVEL_STRING[static_cast<uint8_t>(level)] << "] " << message;
00093         return ss.str();
00094     }
00095
00096 }; // class Logger
00097 } // namespace utl

```

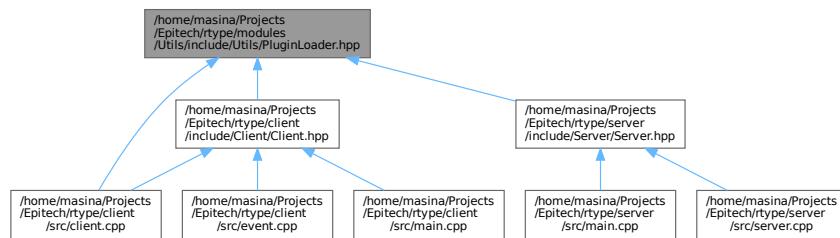
8.131 /home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils/PluginLoader.hpp File Reference

Modern, cross-platform plugin loader.

```
#include <concepts>
#include <memory>
#include <mutex>
#include <stdexcept>
#include <string>
#include <unordered_map>
#include <dlfcn.h>
#include "Utils/Interfaces/IPlugin.hpp"
#include "Utils/Logger.hpp"
Include dependency graph for PluginLoader.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- struct `utl::SharedLib`
Handle to a dynamic library with RAII.
- class `utl::PluginLoader`
Modern, type-safe plugin loader.

Namespaces

- namespace `utl`

Macros

- `#define PLUGINS_EXTENSION ".so"`

TypeDefs

- using `utl::LibHandle`
- using `utl::EntryPointFn = IPlugin *(*)()`

8.131.1 Detailed Description

Modern, cross-platform plugin loader.

Definition in file [PluginLoader.hpp](#).

8.131.2 Macro Definition Documentation

8.131.2.1 PLUGINS_EXTENSION

```
#define PLUGINS_EXTENSION ".so"
```

Definition at line 24 of file [PluginLoader.hpp](#).

8.132 PluginLoader.hpp

[Go to the documentation of this file.](#)

```
00001 /**
00002  * @file PluginLoader.hpp
00003  * @brief Modern, cross-platform plugin loader
00004  * @namespace utl
00005 /**
00006
00007 #pragma once
00008
00009 #include <concepts>
00010 #include <memory>
00011 #include <mutex>
00012 #include <stdexcept>
00013 #include <string>
00014 #include <unordered_map>
00015
00016 #ifdef __WIN32
00017 #include <windows.h>
00018 #define PLUGINS_EXTENSION ".dll"
00019 #else
00020 #include <dlfcn.h>
00021 #if __APPLE__
00022 #define PLUGINS_EXTENSION ".dylib"
00023 #else
00024 #define PLUGINS_EXTENSION ".so"
00025 #endif
00026 #endif
00027
00028 #include "Utils/Interfaces/IPlugin.hpp"
00029 #include "Utils/Logger.hpp"
00030
00031 namespace utl
00032 {
00033
00034     using LibHandle =
00035     #ifdef __WIN32
00036         HMODULE;
00037     #else
00038         void *;
00039     #endif
00040
00041     /// Handle to a dynamic library with RAII
00042     struct SharedLib
00043     {
00044         LibHandle handle = nullptr;
00045
00046         explicit SharedLib(const LibHandle h = nullptr) : handle(h) {}
00047         ~SharedLib() { close(); }
00048
00049         SharedLib(const SharedLib &) = delete;
00050         SharedLib &operator=(const SharedLib &) = delete;
00051         SharedLib(SharedLib &&other) noexcept : handle(other.handle) { other.handle = nullptr; }
00052         SharedLib &operator=(SharedLib &&other) noexcept
00053         {
00054             if (this != &other)
```

```

00055         {
00056             close();
00057             handle = other.handle;
00058             other.handle = nullptr;
00059         }
00060     return *this;
00061 }
00062
00063     void close()
00064     {
00065         if (!handle)
00066             return;
00067 #ifdef _WIN32
00068         FreeLibrary(handle);
00069 #else
00070         dlclose(handle);
00071 #endif
00072         handle = nullptr;
00073     }
00074
00075     explicit operator bool() const { return handle != nullptr; }
00076 };
00077
00078 using EntryPointFn = IPlugin *(*)();
00079
00080 /**
00081  * @class PluginLoader
00082  * @brief Modern, type-safe plugin loader
00083  * @namespace utl
00084  */
00085 class PluginLoader
00086 {
00087     public:
00088         PluginLoader() = default;
00089         ~PluginLoader() = default;
00090
00091         PluginLoader(const PluginLoader &) = delete;
00092         PluginLoader &operator=(const PluginLoader &) = delete;
00093         PluginLoader(PluginLoader &&) = delete;
00094         PluginLoader &operator=(PluginLoader &&) = delete;
00095
00096 /**
00097  * Load a plugin of type T
00098  * @param T Expected plugin interface (must derive from IPlugin)
00099  * @param path Path to the dynamic library
00100  * @return shared_ptr<T> instance
00101  */
00102 template <std::derived_from<IPlugin> T> std::shared_ptr<T> loadPlugin(const std::string &path)
00103 {
00104     std::scoped_lock lock(m_mutex);
00105
00106     if (m_plugins.contains(path))
00107         throw std::runtime_error("Plugin already loaded: " + path);
00108
00109     SharedLib lib = loadLibrary(path);
00110     const EntryPointFn entry = getEntryPoint(lib, path);
00111
00112     std::unique_ptr<IPlugin> plugin(entry());
00113     if (!plugin)
00114         throw std::runtime_error("EntryPoint failed: " + path);
00115
00116     T *typed = dynamic_cast<T *>(plugin.get());
00117     if (!typed)
00118         throw std::runtime_error("Plugin type mismatch: " + path);
00119
00120     auto [it, inserted] = m_plugins.emplace(path, std::move(plugin));
00121     if (!inserted)
00122         throw std::runtime_error("Failed to store plugin: " + path);
00123
00124     m_handles[path] = std::move(lib);
00125
00126     Logger::log("Plugin loaded:\t name: " + it->second->getName() + "\t path: " + path, LogLevel::INFO);
00127
00128     std::shared_ptr<IPlugin> baseShared(it->second.get(), [](IPlugin *){});
00129     return std::shared_ptr<T>(baseShared, typed);
00130 }
00131
00132 private:
00133     std::mutex m_mutex;
00134     std::unordered_map<std::string, SharedLib> m_handles;
00135     std::unordered_map<std::string, std::unique_ptr<IPlugin>> m_plugins;
00136
00137     SharedLib loadLibrary(const std::string &path)
00138     {
00139
00140 #ifdef _WIN32
00141         const LibHandle handle = LoadLibraryA(path.c_str());

```

```

00142     if (!handle)
00143     {
00144         throw std::runtime_error("Cannot load library: " + path);
00145     }
00146 #else
00147     dlerror();
00148     const LibHandle handle = dlopen(path.c_str(), RTLD_LAZY);
00149     if (!handle)
00150     {
00151         const char *error = dlerror();
00152         std::string msg = "Cannot load library: " + path;
00153         if (error)
00154         {
00155             msg += (" (" + std::string(error) + ")");
00156         }
00157         throw std::runtime_error(msg);
00158     }
00159 #endif
00160     return SharedLib(handle);
00161 }
00162
00163     EntryPointFn getEntryPoint(SharedLib &lib, const std::string &path)
00164     {
00165         EntryPointFn entry =
00166 #ifdef _WIN32
00167             reinterpret_cast<EntryPointFn>(GetProcAddress(lib.handle, "entryPoint"));
00168 #else
00169             reinterpret_cast<EntryPointFn>(dlsym(lib.handle, "entryPoint"));
00170 #endif
00171         if (!entry)
00172             throw std::runtime_error("EntryPoint not found in plugin: " + path);
00173         return entry;
00174     }
00175 };
00176
00177 } // namespace utl

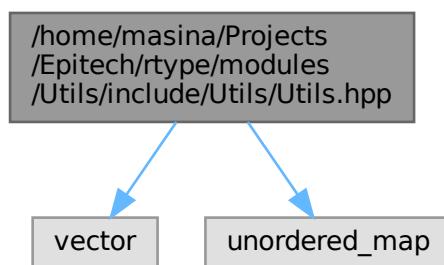
```

8.133 /home/masina/Projects/Epitech/rtype/modules/Utils/include/Utils.hpp File Reference

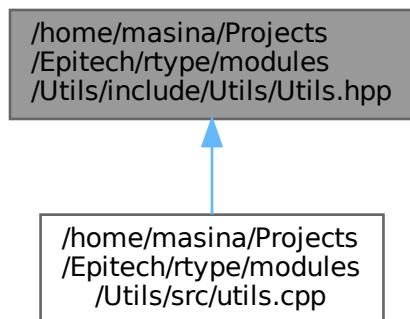
This file contains utility functions.

```
#include <vector>
#include <unordered_map>
```

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)
- std::unordered_map< std::string, std::string > [utl::getEnvMap](#) (const char *const *env)

8.133.1 Detailed Description

This file contains utility functions.

Definition in file [Utils.hpp](#).

8.134 Utils.hpp

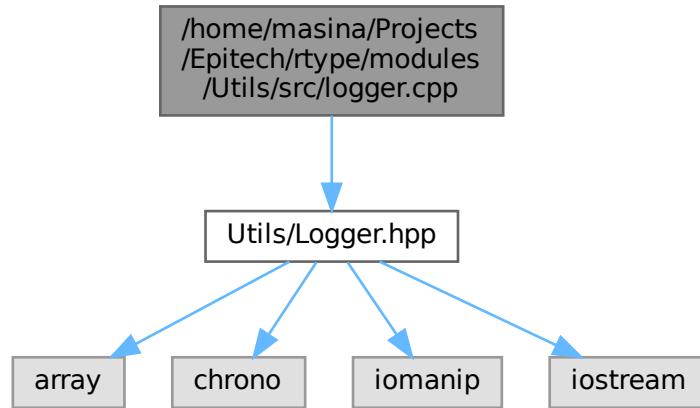
[Go to the documentation of this file.](#)

```

00001 /**
00002 /// @file Utils.hpp
00003 /// @brief This file contains utility functions
00004 /// @namespace utl
00005 /**
00006
00007 #pragma once
00008
00009 #include <vector>
00010 #include <unordered_map>
00011
00012 namespace utl
00013 {
00014     [[nodiscard]] std::vector<char> readFile(const std::string &filename);
00015     [[nodiscard]] std::unordered_map<std::string, std::string> getEnvMap(const char *const *env);
00016
00017 } // namespace utl
  
```

8.135 /home/masina/Projects/Epitech/rtype/modules/Utils/src/logger.cpp File Reference

```
#include "Utils/Logger.hpp"
Include dependency graph for logger.cpp:
```



8.136 logger.cpp

[Go to the documentation of this file.](#)

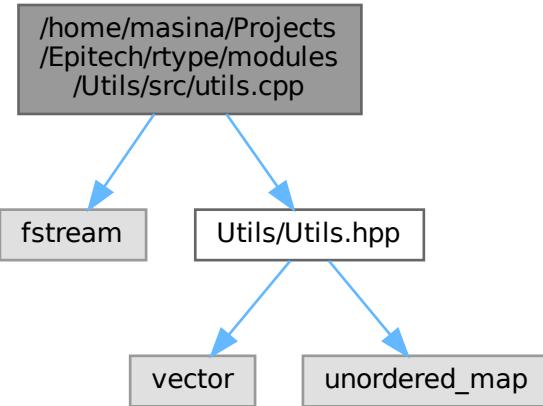
```

00001 #ifndef _WIN32
00002 #include <windows.h>
00003 #endif
00004
00005 #include "Utils/Logger.hpp"
00006
00007 void utl::Logger::init()
00008 {
00009 #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 }
```

8.137 /home/masina/Projects/Epitech/rtype/modules/Utils/src/utils.cpp File Reference

```
#include <fstream>
#include "Utils/Utils.hpp"
```

Include dependency graph for utils.cpp:



8.138 utils.cpp

[Go to the documentation of this file.](#)

```

00001 #include <fstream>
00002
00003 #ifdef _WIN32
00004 #include <windows.h>
00005 #endif
00006
00007 #include "Utils/Utils.hpp"
00008
00009 std::vector<char> utl::readFile(const std::string &filename)
00010 {
00011     std::ifstream file(filename, std::ios::binary | std::ios::ate);
00012     if (!file.is_open())
00013     {
00014         throw std::runtime_error("failed to open file " + filename);
00015     }
00016     const long int fileSize = file.tellg();
00017     if (fileSize <= 0)
00018     {
00019         throw std::runtime_error("file " + filename + " is empty");
00020     }
00021     std::vector<char> buffer(static_cast<long unsigned int>(fileSize));
00022     file.seekg(0, std::ios::beg);
00023     if (!file.read(buffer.data(), fileSize))
00024     {
00025         throw std::runtime_error("failed to read file " + filename);
00026     }
00027     return buffer;
00028 }
00029
00030 std::unordered_map<std::string, std::string> utl::getEnvMap(const char *const *env)
00031 {
00032     std::unordered_map<std::string, std::string> cpyEnv;
00033
00034 #ifdef _WIN32
00035     LPCH envStrings = GetEnvironmentStringsA();
00036     if (!envStrings)
00037     {
00038         return cpyEnv;
00039     }
00040
00041     for (LPCH var = envStrings; *var; var += std::strlen(var) + 1)
00042     {
00043         std::string entry(var);
00044         if (const auto pos = entry.find('='); pos != std::string::npos)
  
```

```

00045     {
00046         cpyEnv.emplace(entry.substr(0, pos), entry.substr(pos + 1));
00047     }
00048 }
00049
00050     FreeEnvironmentStringsA(envStrings);
00051 #else
00052     for (const char *const *current = env; (current != nullptr) && (*current != nullptr); ++current)
00053     {
00054         std::string entry(*current);
00055         if (const auto pos = entry.find('='); pos != std::string::npos)
00056         {
00057             cpyEnv.emplace(entry.substr(0, pos), entry.substr(pos + 1));
00058         }
00059     }
00060 #endif
00061
00062     return cpyEnv;
00063 }
00064

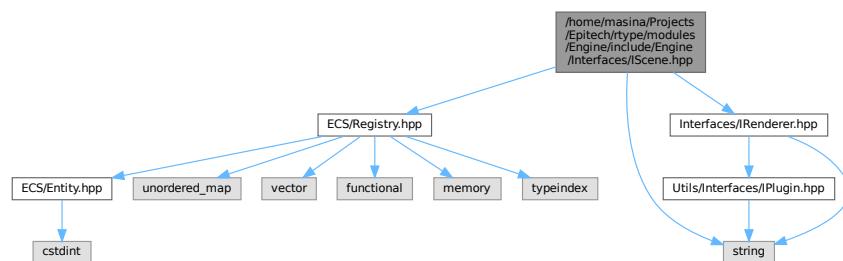
```

8.139 /home/masina/Projects/Epitech/rtype/README.md File Reference

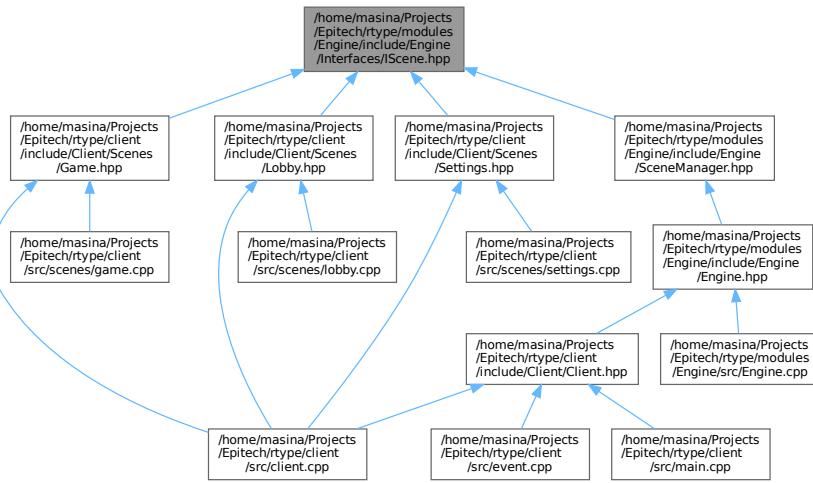
8.140 /home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/Interfaces/IScene.hpp File Reference

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

8.140.1 Detailed Description

This file contains the `IScene` class.

Definition in file [IScene.hpp](#).

8.141 IScene.hpp

[Go to the documentation of this file.](#)

```

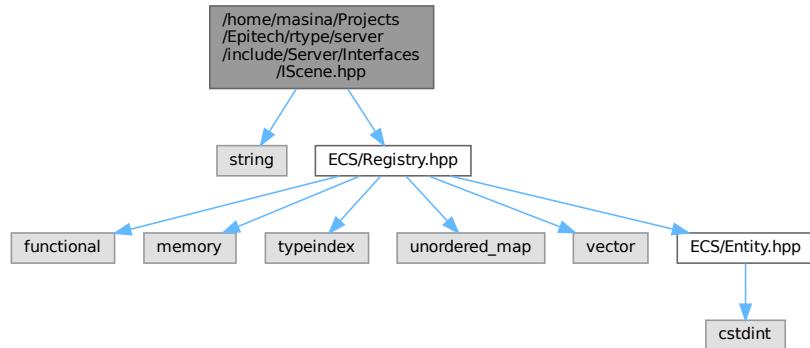
00001 /**
00002  * @file IScene.hpp
00003  * @brief This file contains the IScene class
00004  * @namespace eng
00005 /**
00006
00007 #pragma once
00008
00009 #include <string>
00010
00011 #include "ECS/Registry.hpp"
00012 #include "Interfaces/IRenderer.hpp"
00013
00014 namespace eng
00015 {
00016
00017     using id = unsigned int;
00018
00019 /**
00020  * @class IScene
00021  * @brief interface class for scene
00022  * @namespace eng
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  * @class AScene
00042  * @brief Class for scene
00043  * @namespace eng
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 };
00068 // class AScene
00069 } // namespace eng

```

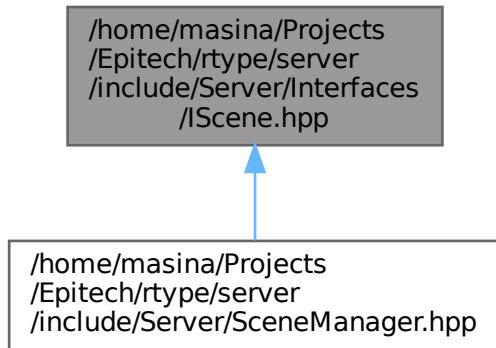
8.142 /home/masina/Projects/Epitech/rtype/server/include/Server/Interfaces/IScene.hpp File Reference

This file contains the IScene class.

```
#include <string>
#include "ECS/Registry.hpp"
Include dependency graph for IScene.hpp:
```



This graph shows which files directly or indirectly include this file:



Classes

- class `srv::IScene`
interface class for scene
- class `srv::AScene`
Class for scene.

Namespaces

- namespace `SRV`

Typedefs

- using `srv::id` = unsigned int

8.142.1 Detailed Description

This file contains the IScene class.

Definition in file [IScene.hpp](#).

8.143 IScene.hpp

[Go to the documentation of this file.](#)

```

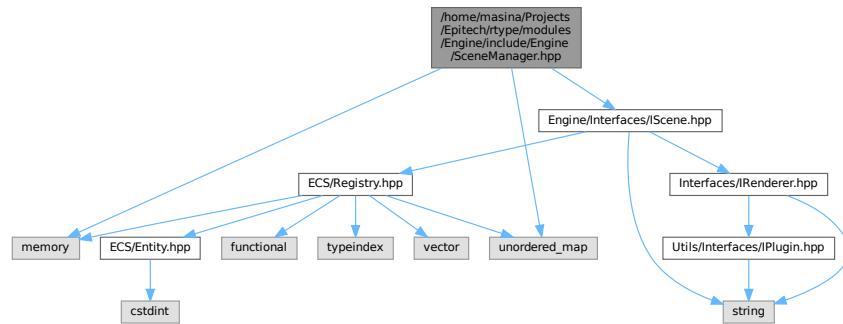
00001 /**
00002  * @file IScene.hpp
00003  * @brief This file contains the IScene class
00004  * @namespace srv
00005 /**
00006
00007 #pragma once
00008
00009 #include <string>
00010
00011 #include "ECS/Registry.hpp"
00012
00013 namespace srv
00014 {
00015
00016     using id = unsigned int;
00017
00018 /**
00019  * @class IScene
00020  * @brief interface class for scene
00021  * @namespace srv
00022 /**
00023 class IScene
00024 {
00025     public:
00026         virtual ~IScene() = default;
00027
00028     [[nodiscard]] virtual std::string &getName() = 0;
00029     [[nodiscard]] virtual id getId() const = 0;
00030     [[nodiscard]] virtual ecs::Registry &getRegistry() = 0;
00031
00032     virtual void setName(const std::string &newName) = 0;
00033
00034     virtual void update(float dt) = 0;
00035 }; // class IScene
00036
00037 /**
00038  * @class AScene
00039  * @brief Class for scene
00040  * @namespace srv
00041 /**
00042 class AScene : public IScene
00043 {
00044     public:
00045         AScene() : m_id(s_nextId++) {}
00046         ~AScene() override = default;
00047
00048         AScene(const AScene &other) = delete;
00049         AScene(AScene &&other) = delete;
00050         AScene &operator=(const AScene &other) = delete;
00051         AScene &operator=(AScene &&other) = delete;
00052
00053     [[nodiscard]] std::string &getName() override { return m_name; }
00054     [[nodiscard]] id getId() const override { return m_id; }
00055     [[nodiscard]] ecs::Registry &getRegistry() override { return m_registry; }
00056
00057     void setName(const std::string &newName) override { m_name = newName; }
00058
00059     private:
00060         std::string m_name = "default_name";
00061         id m_id = 1;
00062         ecs::Registry m_registry;
00063         inline static id s_nextId = 1;
00064 }; // class AScene
00065
00066 } // namespace srv

```

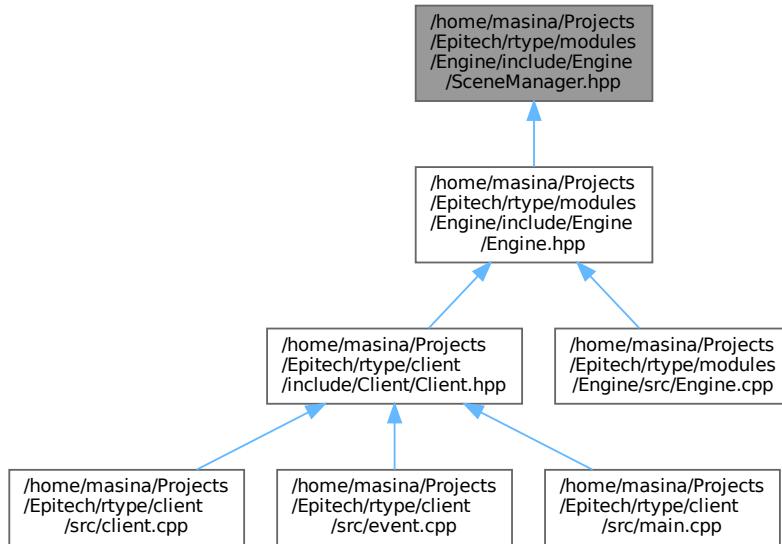
8.144 /home/masina/Projects/Epitech/rtype/modules/Engine/include/Engine/SceneManager.hpp File Reference

This file contains the SceneManager class declaration.

```
#include <memory>
#include <unordered_map>
#include "Engine/Interfaces/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

8.144.1 Detailed Description

This file contains the SceneManager class declaration.

Definition in file [SceneManager.hpp](#).

8.145 SceneManager.hpp

[Go to the documentation of this file.](#)

```

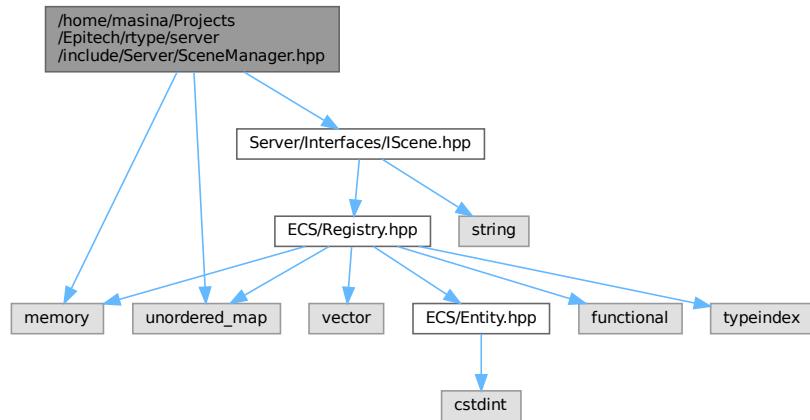
00001 /**
00002  * @file SceneManager.hpp
00003  * @brief This file contains the SceneManager class declaration
00004  * @namespace eng
00005 /**
00006
00007 #pragma once
00008
00009 #include <memory>
00010 #include <unordered_map>
00011
00012 #include "Engine/Interfaces/IScene.hpp"
00013
00014 namespace 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_currentScenId); }
00036     void switchToScene(const id sceneId) { m_currentScenId = 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_currentScenId = 1;
00058 }; // class SceneManager
00059 } // namespace eng

```

8.146 /home/masina/Projects/Epitech/rtype/server/include/Server/SceneManager.hpp File Reference

This file contains the SceneManager class declaration.

```
#include <memory>
#include <unordered_map>
#include "Server/Interfaces/IScene.hpp"
Include dependency graph for SceneManager.hpp:
```



Classes

- class [srv::SceneManager](#)
Class for managing scenes.

Namespaces

- namespace [srv](#)

8.146.1 Detailed Description

This file contains the SceneManager class declaration.

Definition in file [SceneManager.hpp](#).

8.147 SceneManager.hpp

[Go to the documentation of this file.](#)

```

00001 /**
00002  /// @file SceneManager.hpp
00003  /// @brief This file contains the SceneManager class declaration
00004  /// @namespace srv
00005 /**
00006
00007 #pragma once
00008
00009 #include <memory>
00010 #include <unordered_map>
00011
00012 #include "Server/Interfaces/IScene.hpp"
00013
00014 namespace srv
00015 {
00016
00017 /**
00018  /// @class SceneManager
00019  /// @brief Class for managing scenes
00020  /// @namespace srv
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&... defns) {
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     //     (defns.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 srv

```

8.148 /home/masina/Projects/Epitech/rtype/server/include/Server/← Scenes/Level_0.hpp File Reference

This file contains the level_0 definitions.

Namespaces

- namespace **srv**

8.148.1 Detailed Description

This file contains the level_0 definitions.

Definition in file [Level_0.hpp](#).

8.149 Level_0.hpp

[Go to the documentation of this file.](#)

```
00001 /**
00002 /// @file Level_0.hpp
00003 /// @brief This file contains the level_0 definitions
00004 /// @namespace srv
00005 /**
00006
00007 #pragma once
00008
00009 namespace srv
00010 {
00011
00012 } // namespace srv
```

8.150 /home/masina/Projects/Epitech/rtype/server/include/Server/← Scenes/Level_1.hpp File Reference

This file contains the level_0 definitions.

Namespaces

- namespace [srv](#)

8.150.1 Detailed Description

This file contains the level_0 definitions.

Definition in file [Level_1.hpp](#).

8.151 Level_1.hpp

[Go to the documentation of this file.](#)

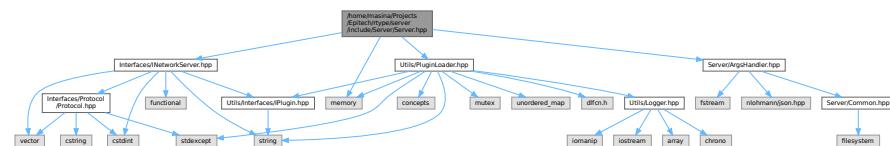
```
00001 /**
00002 /// @file Level_1.hpp
00003 /// @brief This file contains the level_0 definitions
00004 /// @namespace srv
00005 /**
00006
00007 #pragma once
00008
00009 namespace srv
00010 {
00011
00012 } // namespace srv
```

8.152 /home/masina/Projects/Epitech/rtype/server/include/Server/Server.hpp File Reference

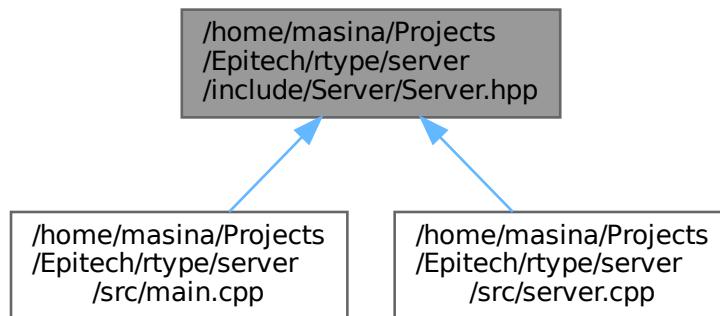
This file contains the Server class declaration.

```
#include <memory>
#include "Interfaces/INetworkServer.hpp"
#include "Server/ArgsHandler.hpp"
#include "Utils/PluginLoader.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](#)

8.152.1 Detailed Description

This file contains the Server class declaration.

Definition in file [Server.hpp](#).

8.153 Server.hpp

[Go to the documentation of this file.](#)

```

00001 /**
00002 /// @file Server.hpp
00003 /// @brief This file contains the Server class declaration
00004 /// @namespace srv
00005 /**
00006
00007 #pragma once
00008
00009 #include <memory>
00010
00011 #include "Interfaces/INetworkServer.hpp"
00012 #include "Server/ArgsHandler.hpp"
00013 #include "Utils/PluginLoader.hpp"
00014
00015 namespace srv
00016 {
00017
00018 /**
00019 /// @class Server
00020 /// @brief Class for the server
00021 /// @namespace srv
00022 /**
00023 class Server
00024 {
00025
00026     public:
00027         explicit Server(const ArgsConfig &config);
00028         ~Server() = default;
00029
00030         Server(const Server &) = delete;
00031         Server &operator=(const Server &) = delete;
00032         Server(Server &&) = delete;
00033         Server &operator=(Server &&) = delete;
00034
00035         void run() const;
00036
00037     private:
00038         std::unique_ptr<util::PluginLoader> m_pluginLoader;
00039         std::shared_ptr<INetworkServer> m_network;
00040     }; // class Server
00041
00042 } // namespace srv

```

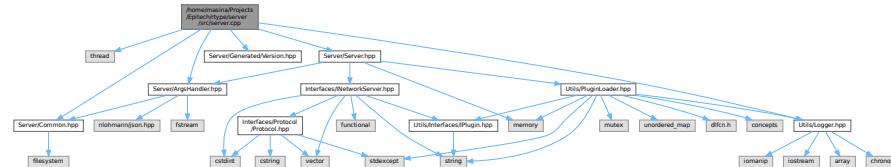
8.154 /home/masina/Projects/Epitech/rtype/server/src/server.cpp File Reference

```

#include <thread>
#include "Server/ArgsHandler.hpp"
#include "Server/Common.hpp"
#include "Server/Generated/Version.hpp"
#include "Server/Server.hpp"
#include "Utils/Logger.hpp"

```

Include dependency graph for server.cpp:



8.155 server.cpp

[Go to the documentation of this file.](#)

```
00001 #include <thread>
00002
00003 #include "Server/ArgsHandler.hpp"
00004 #include "Server/Common.hpp"
00005 #include "Server/Generated/Version.hpp"
00006 #include "Server/Server.hpp"
00007 #include "Utils/Logger.hpp"
00008
00009 srv::Server::Server(const ArgsConfig &config)
00010   : m_pluginLoader(std::make_unique<utl::PluginLoader>()),
00011     m_network(m_pluginLoader->loadPlugin<INetworkServer>(!config.network_lib_path.empty() ?
00012       config.network_lib_path : Path::Plugin::PLUGINS_NETWORK_ASIO_SERVER.string())))
00013 {
00014   utl::Logger::log("PROJECT INFO:", utl::LogLevel::INFO);
00015   std::cout « "\tName: " PROJECT_NAME "\n"
00016             "\tVersion: " PROJECT_VERSION "\n"
00017             "\tBuild type: " BUILD_TYPE "\n"
00018             "\tGit tag: " GIT_TAG "\n"
00019             "\tGit commit hash: " GIT_COMMIT_HASH "\n";
00020   m_network->init(config.host, config.port);
00021 }
00022
00023 void srv::Server::run() const
00024 {
00025   m_network->start();
00026   for (;;)
00027   {
00028     std::this_thread::sleep_for(std::chrono::seconds(1));
00029   }
00030 }
```


Index

/home/masina/Projects/Epitech/rtype/README.md
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
529
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
377, 378
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
381, 383
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
383, 385
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
388, 391
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
393, 395
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
397, 398
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
399, 401
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
401, 403
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
403, 404
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
405, 406
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
406, 408
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
408, 410
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
411, 412
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
413, 414
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
416, 417
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
419, 420
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
422, 423
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
424, 425
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
426, 427
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
428, 430
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
433, 434
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
435, 437
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
437, 438
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
440, 441
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
442, 443
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
456
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
444, 446
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
450, 451
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
452
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
453, 454
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
457, 459
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
463, 464
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
467
/home/masina/Projects/Epitech/rtype/client/include/Client/System/Client.h
469
/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.h
472, 473
/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.h
475, 476
/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.h
476, 477
/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.h
478, 479
/home/masina/Projects/Epitech/rtype/modules/ECS/src/registry.h
481
/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.h
481, 482
/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.h
529, 531
/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.h
534, 535
/home/masina/Projects/Epitech/rtype/modules/ECS/include/ECS/Component.h
484
/home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Client/include/R-TypeClient/LobbyScene.hpp
484, 486
/home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Client/include/R-TypeClient/LobbyScene.hpp
486, 488
/home/masina/Projects/Epitech/rtype/modules/Game/R-Type/Client/include/R-TypeClient/Spawn.hpp
488

/home/masina/Projects/Epitech/rtype/modules/Game/R-
 Type/Client/src/rtypeClient.cpp, 488, 448, 449
 489 /home/masina/Projects/Epitech/rtype/server/src/main.cpp,
 /home/masina/Projects/Epitech/rtype/modules/Game/R- 454, 455
 Type/Server/include/R-TypeServer/RTypes.h, 450, 541
 489, 490
 /home/masina/Projects/Epitech/rtype/modules/Game/AGameClient
 Type/Server/src/rtypeServer.cpp, 491 gme::AGameClient, 62
 /home/masina/Projects/Epitech/rtype/modules/Interfaces/AGameServer.hpp,
 491, 492 gme::AGameServer, 65
 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/AGameServer.hpp,
 493, 494 eng::AScene, 86
 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IAudio.hpp,
 494, 496 ~AnimationSystem
 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IGameClient.hpp,
 496, 498 ~ArgsHandler
 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IGameServer.hpp,
 499, 500 srv::ArgsHandler, 82
 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/INetworkClient.hpp,
 500, 501 cli::AsteroidSystem, 100
 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/INetworkServer.hpp,
 502, 504 cli::AudioSystem, 109
 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/IRenderer.hpp,
 504, 506 cli::BeamSystem, 116
 /home/masina/Projects/Epitech/rtype/modules/Interfaces/include/Interfaces/Protocol.hpp,
 508, 511 cli::Client, 119
 /home/masina/Projects/Epitech/rtype/modules/Utils/Clock.hpp,
 515, 516 utl::Clock, 124
 /home/masina/Projects/Epitech/rtype/modules/Utils/CollisionSystem.hpp,
 517, 519 cli::CollisionSystem, 131
 /home/masina/Projects/Epitech/rtype/modules/Utils/EnemySystem.hpp,
 519, 520 cli::EnemySystem, 148
 /home/masina/Projects/Epitech/rtype/modules/Utils/Engine.hpp,
 521, 523 eng::Engine, 152
 /home/masina/Projects/Epitech/rtype/modules/Utils/ExplosionSystem.hpp,
 525, 526 cli::ExplosionSystem, 172
 /home/masina/Projects/Epitech/rtype/modules/Utils/GameLogger.cpp,
 527 cli::Game, 181
 /home/masina/Projects/Epitech/rtype/modules/Utils/IGameClient.hpp,
 527, 528 gme::IGameClient, 193
 /home/masina/Projects/Epitech/rtype/server/include/ISceneHandler.hpp,
 379, 381 gme::IGameServer, 196
 /home/masina/Projects/Epitech/rtype/server/include/IServer/ISocketClient.hpp,
 386, 388 eng::INetworkClient, 199
 /home/masina/Projects/Epitech/rtype/server/include/IServer/ISocketGenerated.hpp,
 395, 397 srv::INetworkServer, 204
 /home/masina/Projects/Epitech/rtype/server/include/ISystem/IScene.hpp,
 531, 533 utl::IPlugin, 206
 /home/masina/Projects/Epitech/rtype/server/include/ISystem/SceneManager.hpp,
 536, 537 ecs::Registry::IPool, 207
 /home/masina/Projects/Epitech/rtype/server/include/ISystem/Scenes/Level_0.hpp,
 537, 538 eng::IScene, 218
 /home/masina/Projects/Epitech/rtype/server/include/ISystem/Scenes/Level_1.hpp,
 538 srv::IScene, 223
 /home/masina/Projects/Epitech/rtype/server/include/ISystem/Server.hpp,
 539, 540 eng::ISystem, 225
 /home/masina/Projects/Epitech/rtype/server/include/ISystem/LoadingAnimationSystem.hpp,
 439, 440 cli::LoadingAnimationSystem, 231

~Lobby
 cli::Lobby, 235
 ~Logger
 utl::Logger, 244
 ~PixelSystem
 cli::PixelSystem, 264
 ~PlayerDirectionSystem
 cli::PlayerDirectionSystem, 270
 ~PluginLoader
 utl::PluginLoader, 273
 ~ProjectileManager
 cli::ProjectileManager, 284
 ~ProjectileSystem
 cli::ProjectileSystem, 290
 ~RTTypeClient
 gme::RTTypeClient, 308
 ~RTTypeServer
 gme::RTTypeServer, 312
 ~Registry
 ecs::Registry, 297
 ~SceneManager
 eng::SceneManager, 317
 srv::SceneManager, 321
 ~Server
 srv::Server, 325
 ~Settings
 cli::Settings, 329
 ~SharedLib
 utl::SharedLib, 332
 ~SpawnSystem
 cli::SpawnSystem, 337
 ~SpriteSystem
 cli::SpriteSystem, 349
 ~TextSystem
 cli::TextSystem, 358
 ~WeaponSystem
 cli::WeaponSystem, 369

A

eng, 46
 a
 ecs::Color, 139
 eng::Color, 140
 gme::Sprite, 344
 ACK
 rnp, 51
 ACK_REQ
 rnp, 51
 ackBits
 rnp::PacketAck, 250
 add
 ecs::Registry::Pool< T >, 279
 addComponent
 ecs::Registry, 298
 addEntity
 gme::LobbyScene, 240
 addScene
 eng::SceneManager, 317
 srv::SceneManager, 321

addSystem
 eng::Engine, 153
ANIMATION_DURATION
 cli::GameConfig::Asteroid::Small, 24
 cli::GameConfig::Enemy::Easy, 27
 cli::GameConfig::Explosion, 29
 cli::GameConfig::LoadingAnimation, 32
 cli::GameConfig::Projectile::Supercharged, 36
ANIMATION_FRAMES
 cli::GameConfig::Asteroid::Small, 24
 cli::GameConfig::Enemy::Easy, 27
 cli::GameConfig::Explosion, 29
 cli::GameConfig::LoadingAnimation, 32
 cli::GameConfig::Projectile::Supercharged, 36
 AnimationSystem
 cli::AnimationSystem, 71
APP_EXTENSION
 argsHandler.cpp, 445, 448
 ArgsHandler
 cli::ArgsHandler, 79
 srv::ArgsHandler, 82
 argsHandler.cpp
 APP_EXTENSION, 445, 448
 HELP_MESSAGE, 445, 448
 VERSION_MESSAGE, 445, 448
AScene
 eng::AScene, 86
 srv::AScene, 91
ASTEROID_LARGE_RADIUS
 cli::GameConfig::Hitbox, 30
ASTEROID_MEDIUM_RADIUS
 cli::GameConfig::Hitbox, 30
ASTEROID_SMALL_RADIUS
 cli::GameConfig::Hitbox, 31
 asteroidRect
 cli::SpriteRect, 42
 AsteroidSystem
 cli::AsteroidSystem, 100
AUDIO
 utl, 58
AUDIO_BATTLE_THEME
 cli::Path::Audio, 39
AUDIO_COIN
 cli::Path::Audio, 39
 audio_lib_path
 cli::ArgsConfig, 74
AUDIO_TITLE
 cli::Path::Audio, 39
 AudioSystem
 cli::AudioSystem, 109

B

eng, 46
 b
 ecs::Color, 139
 eng::Color, 140
 gme::Sprite, 344
BANNED
 rnp, 49

BAR_HEIGHT
 cli::GameConfig::Beam, 25

BAR_WIDTH
 cli::GameConfig::Beam, 25

BAR_X
 cli::GameConfig::Beam, 26

BAR_Y
 cli::GameConfig::Beam, 26

BASIC
 ecs::Projectile, 282

BeamSystem
 cli::BeamSystem, 116

BLUE
 game.cpp, 458

BLUE_SECOND
 game.cpp, 458

build
 ecs::Registry::EntityBuilder, 159

BUILD_TYPE
 Version.hpp, 393, 396

C
 eng, 46

CHARGE_RATE
 cli::GameConfig::Beam, 26

CHARGE_TIME
 cli::GameConfig::Projectile::Supercharged, 36

checkCircularCollision
 cli::CollisionSystem, 132

clearWindow
 eng::IRenderer, 210

cli, 17
 json, 18

cli::AnimationSystem, 69
 ~AnimationSystem, 71
 AnimationSystem, 71
 operator=, 71
 update, 71

cli::ArgsConfig, 72
 audio_lib_path, 74
 exit, 74
 frameLimit, 74
 fromFile, 74
 fullscreen, 74
 height, 75
 host, 75
 network_lib_path, 75
 port, 75
 renderer_lib_path, 75
 width, 75

cli::ArgsHandler, 78
 ~ArgsHandler, 79
 ArgsHandler, 79
 operator=, 79
 ParseArgs, 80
 ParseEnv, 80

cli::AsteroidSystem, 98
 ~AsteroidSystem, 100
 AsteroidSystem, 100

 isEnabled, 100
 m_renderer, 102
 operator=, 100, 101
 setEnable, 101
 update, 101

cli::AudioSystem, 106
 ~AudioSystem, 109
 AudioSystem, 109
 m_audio, 110
 operator=, 110
 update, 110

cli::BeamSystem, 113
 ~BeamSystem, 116
 BeamSystem, 116
 m_renderer, 117
 operator=, 116
 update, 117

cli::Client, 118
 ~Client, 119
 Client, 119, 120
 handleEvents, 120
 m_engine, 121
 m_game, 121
 m_keysPressed, 121
 m_pluginLoader, 121
 operator=, 120
 run, 120

cli::CollisionSystem, 128
 ~CollisionSystem, 131
 checkCircularCollision, 132
 CollisionSystem, 131
 createExplosion, 132
 isEnabled, 133
 m_renderer, 136
 operator=, 133
 removeAsteroid, 133
 removeEnemy, 134
 removeProjectile, 135
 setEnable, 135
 update, 135

cli::Config, 18

cli::Config::Audio, 18
 DEFAULT_AUDIO_MUTED, 18
 DEFAULT_AUDIO_VOLUME, 18

cli::Config::Network, 19
 DEFAULT_NETWORK_HOST, 19
 DEFAULT_NETWORK_PORT, 19

cli::Config::Window, 19
 DEFAULT_WINDOW_FRAME_LIMIT, 19
 DEFAULT_WINDOW_FULLSCREEN, 19
 DEFAULT_WINDOW_HEIGHT, 20
 DEFAULT_WINDOW_WIDTH, 20

cli::EnemySystem, 146
 ~EnemySystem, 148
 EnemySystem, 148
 isEnabled, 148
 m_renderer, 150
 operator=, 148, 149

setEnable, 149
update, 149
cli::EnvConfig, 162
cli::ExplosionSystem, 169
~ExplosionSystem, 172
ExplosionSystem, 172
isEnabled, 173
m_renderer, 174
operator=, 173
removeExplosion, 173
setEnabled, 173
update, 174
cli::Game, 178
~Game, 181
event, 182
Game, 181, 182
m_asteroidCounterEntity, 183
m_audio, 183
m_enemyCounterEntity, 183
m_fpsEntity, 183
m_keysPressed, 183
m_playerEntity, 183
operator=, 182
update, 182
cli::GameConfig, 20
cli::GameConfig::Animation, 20
FRAME_DURATION, 20
cli::GameConfig::Asteroid, 21
cli::GameConfig::Asteroid::Large, 21
HEALTH, 21
ROTATION_SPEED, 21
SCALE, 21
SPAWN_RATE, 21
SPEED, 21
SPRITE_HEIGHT, 22
SPRITE_WIDTH, 22
cli::GameConfig::Asteroid::Medium, 22
HEALTH, 22
ROTATION_SPEED, 22
SCALE, 22
SPAWN_RATE, 23
SPEED, 23
SPRITE_HEIGHT, 23
SPRITE_WIDTH, 23
cli::GameConfig::Asteroid::Small, 23
ANIMATION_DURATION, 24
ANIMATION_FRAMES, 24
FRAMES_PER_ROW, 24
HEALTH, 24
ROTATION_SPEED, 24
SCALE, 24
SPAWN_RATE, 24
SPEED, 25
SPRITE_HEIGHT, 25
SPRITE_WIDTH, 25
cli::GameConfig::Beam, 25
BAR_HEIGHT, 25
BAR_WIDTH, 25
BAR_X, 26
BAR_Y, 26
CHARGE_RATE, 26
MAX_CHARGE, 26
cli::GameConfig::Enemy, 26
cli::GameConfig::Enemy::Easy, 27
ANIMATION_DURATION, 27
ANIMATION_FRAMES, 27
DAMAGE, 27
FRAMES_PER_ROW, 27
HEALTH, 27
SCALE, 28
SHOOT_COOLDOWN, 28
SPAWN_RATE, 28
SPEED, 28
SPRITE_HEIGHT, 28
SPRITE_WIDTH, 28
cli::GameConfig::Explosion, 29
ANIMATION_DURATION, 29
ANIMATION_FRAMES, 29
FRAMES_PER_ROW, 29
LIFETIME, 29
SCALE, 30
SPRITE_HEIGHT, 30
SPRITE_WIDTH, 30
cli::GameConfig::Hitbox, 30
ASTEROID_LARGE_RADIUS, 30
ASTEROID_MEDIUM_RADIUS, 30
ASTEROID_SMALL_RADIUS, 31
ENEMY_RADIUS, 31
PLAYER_RADIUS, 31
PROJECTILE_BASIC_RADIUS, 31
PROJECTILE_SUPERCHARGED_RADIUS, 31
cli::GameConfig::LoadingAnimation, 32
ANIMATION_DURATION, 32
ANIMATION_FRAMES, 32
OFFSET_X, 32
OFFSET_Y, 32
SPRITE_HEIGHT, 32
SPRITE_WIDTH, 32
cli::GameConfig::Player, 33
DIAGONAL_SPEED_MULTIPLIER, 33
FRAMES_PER_ROW, 33
SCALE, 33
SPEED, 33
SPRITE_HEIGHT, 33
SPRITE_WIDTH, 34
TOTAL_FRAMES, 34
cli::GameConfig::Projectile, 34
cli::GameConfig::Projectile::Basic, 34
DAMAGE, 34
FIRE_COOLDOWN, 34
LIFETIME, 35
SCALE, 35
SPEED, 35
SPRITE_HEIGHT, 35
SPRITE_WIDTH, 35

cli::GameConfig::Projectile::Supercharged, 35
 ANIMATION_DURATION, 36
 ANIMATION_FRAMES, 36
 CHARGE_TIME, 36
 DAMAGE, 36
 FIRE_COOLDOWN, 36
 LIFETIME, 36
 SCALE, 36
 SPEED, 36
 SPRITE_HEIGHT, 37
 SPRITE_WIDTH, 37
 cli::GameConfig::Screen, 37
 HEIGHT, 37
 MAX_Y, 37
 MIN_Y, 37
 REMOVE_MAX_Y, 38
 REMOVE_MIN_Y, 38
 REMOVE_X, 38
 SPAWN_X, 38
 WIDTH, 38
 cli::LoadingAnimationSystem, 229
 ~LoadingAnimationSystem, 231
 LoadingAnimationSystem, 231
 m_renderer, 232
 operator=, 231
 update, 232
 cli::Lobby, 233
 ~Lobby, 235
 event, 236
 Lobby, 235, 236
 m_audio, 237
 m_fpsEntity, 237
 m_keysPressed, 237
 m_menuOptions, 237
 m_selectedIndex, 237
 onOptionSelected, 237
 operator=, 236
 update, 236
 cli::Path, 39
 cli::Path::Audio, 39
 AUDIO_BATTLE_THEME, 39
 AUDIO_COIN, 39
 AUDIO_TITLE, 39
 cli::Path::Font, 39
 FONTS_RTYPE, 40
 cli::Path::Plugin, 40
 PLUGIN_AUDIO_SFML, 40
 PLUGIN_NETWORK_ASIO_CLIENT, 40
 PLUGIN_RENDERER_SFML, 40
 cli::Path::Texture, 41
 TEXTURE_ASTEROID, 41
 TEXTURE_ENEMY_EASY, 41
 TEXTURE_EXPLOSION, 41
 TEXTURE_PLAYER, 41
 TEXTURE_SHOOT, 41
 TEXTURE_SHOOT_CHARGED, 42
 TEXTURE_SHOOT_LOADING, 42
 cli::PixelSystem, 262
 ~PixelSystem, 264
 m_renderer, 265
 operator=, 264
 PixelSystem, 264
 update, 265
 cli::PlayerDirectionSystem, 268
 ~PlayerDirectionSystem, 270
 operator=, 270
 PlayerDirectionSystem, 270
 update, 270
 cli::ProjectileManager, 283
 ~ProjectileManager, 284
 createBasicProjectile, 284
 createProjectile, 285
 createSuperchargedProjectile, 286
 operator=, 287
 ProjectileManager, 284
 cli::ProjectileSystem, 288
 ~ProjectileSystem, 290
 operator=, 290
 ProjectileSystem, 290
 update, 290
 cli::Settings, 327
 ~Settings, 329
 event, 330
 m_audio, 331
 m_keysPressed, 331
 onLeave, 331
 operator=, 330
 Settings, 329, 330
 update, 330
 cli::SpawnSystem, 334
 ~SpawnSystem, 337
 isEnabled, 338
 m_asteroidSpawnTimer, 341
 m_enemySpawnTimer, 341
 m_renderer, 341
 m_waveTimer, 341
 operator=, 338
 setEnable, 338
 spawnAsteroid, 338
 spawnEnemy, 339
 SpawnSystem, 337
 spawnWave, 340
 update, 340
 cli::SpriteRect, 42
 asteroidRect, 42
 enemyRect, 42
 cli::SpriteRect::Rect, 291
 h, 292
 w, 292
 x, 292
 y, 292
 cli::SpriteRect::SpriteSheet, 43
 cli::SpriteRect::SpriteSheet::Asteroid, 93
 frameH, 94
 frameW, 94
 marginX, 94

marginY, 94
spacingX, 94
spacingY, 95
totalFrames, 95
cli::SpriteRect::SpriteSheet::Enemy, 141
 frameH, 141
 frameW, 141
 marginX, 142
 marginY, 142
 spacingX, 142
 spacingY, 142
 totalFrames, 142
cli::SpriteSystem, 346
 ~SpriteSystem, 349
 m_renderer, 350
 operator=, 349
 SpriteSystem, 349
 update, 350
cli::TextSystem, 355
 ~TextSystem, 358
 m_renderer, 359
 operator=, 359
 TextSystem, 358
 update, 359
cli::WeaponSystem, 366
 ~WeaponSystem, 369
 hideLoadingAnimation, 370
 m_fireCooldown, 375
 m_isCharging, 375
 operator=, 371
 reset, 371
 showLoadingAnimation, 371
tryFireBasic, 372
tryFireSupercharged, 373
update, 374
WeaponSystem, 369, 370
Client
 cli::Client, 119, 120
client.cpp
 DARK, 451
CLIENT_REQUEST
 rnp, 49
clientCaps
 rnp::PacketConnect, 251
Clock
 utl::Clock, 124
close
 utl::SharedLib, 333
Closed
 eng, 45
closeWindow
 eng::IRenderer, 210
CollisionSystem
 cli::CollisionSystem, 131
color
 eng::Text, 354
COLOR_ERROR
 utl::Logger, 243
COLOR_INFO
 utl::Logger, 243
COLOR_RESET
 utl::Logger, 243
COLOR_WARNING
 utl::Logger, 243
ColorIndex
 utl::Logger, 243
COMPRESSED
 rnp, 51
CONNECT
 rnp, 51
connect
 eng::INetworkClient, 199
CONNECT_ACCEPT
 rnp, 51
content
 ecs::Text, 353
 eng::Text, 354
createAudio
 eng::IAudio, 188
createBasicProjectile
 cli::ProjectileManager, 284
createEntity
 ecs::Registry, 298
createExplosion
 cli::CollisionSystem, 132
createFont
 eng::IRenderer, 210
createProjectile
 cli::ProjectileManager, 285
createSprite
 eng::IRenderer, 210
createSuperchargedProjectile
 cli::ProjectileManager, 286
createText
 eng::IRenderer, 211
createTexture
 eng::IRenderer, 211
createWindow
 eng::IRenderer, 211
cumulativeAck
 rnp::PacketAck, 250
current_charge
 ecs::BeamCharge, 113
current_frame
 ecs::Animation, 68
 ecs::Explosion, 168
 ecs::LoadingAnimation, 228
current_lifetime
 ecs::Explosion, 168
 ecs::Projectile, 282
current_time
 ecs::Animation, 68
 ecs::Explosion, 168
 ecs::LoadingAnimation, 228
CUSTOM
 rnp, 50

D
 eng, 46
DAMAGE
 cli::GameConfig::Enemy::Easy, 27
 cli::GameConfig::Projectile::Basic, 34
 cli::GameConfig::Projectile::Supercharged, 36
 rnp, 50
damage
 ecs::Enemy, 145
 ecs::Projectile, 282
DARK
 client.cpp, 451
data
 ecs::Registry::Pool< T >, 280
 rnp::EventRecord, 165
DEFAULT
 eng, 47
DEFAULT_AUDIO_MUTED
 cli::Config::Audio, 18
DEFAULT_AUDIO_VOLUME
 cli::Config::Audio, 18
DEFAULT_MAX_CLIENT
 srv::Config::Network, 55
DEFAULT_NETWORK_HOST
 cli::Config::Network, 19
 srv::Config::Network, 55
DEFAULT_NETWORK_PORT
 cli::Config::Network, 19
 srv::Config::Network, 55
DEFAULT_TICK_RATE
 srv::Game, 56
DEFAULT_UPDATE_INTERVAL
 srv::Game, 56
DEFAULT_WINDOW_FRAME_LIMIT
 cli::Config::Window, 19
DEFAULT_WINDOW_FULLSCREEN
 cli::Config::Window, 19
DEFAULT_WINDOW_HEIGHT
 cli::Config::Window, 20
DEFAULT_WINDOW_WIDTH
 cli::Config::Window, 20
description
 rnp::PacketError, 255
deserializeEvents
 rnp, 51
deserializeHeader
 rnp, 51
DESPAWN
 rnp, 50
DIAGONAL_SPEED_MULTIPLIER
 cli::GameConfig::Player, 33
DISCONNECT
 rnp, 51
disconnect
 eng::INetworkClient, 199
DisconnectReason
 rnp, 49
displayWindow

eng::IRenderer, 211
Down
 eng, 46
drawPoint
 eng::IRenderer, 212
drawSprite
 eng::IRenderer, 212
drawText
 eng::IRenderer, 212
Duration
 utl::Clock, 123

E
 eng, 46
ecs, 43
Entity, 44
 INVALID_ENTITY, 44
ecs::Animation, 66
 current_frame, 68
 current_time, 68
 frame_duration, 68
 frame_height, 68
 frame_width, 68
 frames_per_row, 68
 total_frames, 68
ecs::Asteroid, 95
 health, 97
 LARGE, 97
 MEDIUM, 97
 rotation_speed, 97
 Size, 97
 size, 97
 SMALL, 97
ecs::Audio, 104
 loop, 106
 path, 106
 play, 106
 volume, 106
ecs::BeamCharge, 111
 current_charge, 113
 max_charge, 113
ecs::Color, 136
 a, 139
 b, 139
 g, 139
 r, 139
ecs::Enemy, 143
 damage, 145
 health, 145
 last_shot_time, 145
 max_health, 145
 shoot_cooldown, 145
 speed, 145
ecs::Explosion, 166
 current_frame, 168
 current_lifetime, 168
 current_time, 168
 frame_duration, 168
 frame_height, 168

frame_width, 168
frames_per_row, 169
lifetime, 169
total_frames, 169
ecs::Font, 175
path, 177
ecs::Hitbox, 184
radius, 186
ecs::IComponent, 190
id, 191
ecs::LoadingAnimation, 226
current_frame, 228
current_time, 228
frame_duration, 228
frame_height, 228
frame_width, 228
frames_per_row, 228
total_frames, 228
ecs::Mob, 248
ecs::Pixel, 260
ecs::Player, 266
is_cli, 268
ecs::Projectile, 280
BASIC, 282
current_lifetime, 282
damage, 282
lifetime, 282
SUPERCHARGED, 282
Type, 282
type, 282
ecs::Rect, 293
pos_x, 295
pos_y, 295
size_x, 295
size_y, 295
ecs::Registry, 295
~Registry, 297
addComponent, 298
createEntity, 298
getAll, 299
getComponent, 300
getPool, 301
hasComponent, 302
m_components, 305
m_entities, 305
m_lastEntity, 305
m_onComponentAddedCallbacks, 305
onComponentAdded, 303
operator=, 303
Registry, 297
removeComponent, 304
ecs::Registry::EntityBuilder, 157
build, 159
EntityBuilder, 159
m_entity, 160
m_registry, 160
with, 159
ecs::Registry::IPool, 206
~IPool, 207
remove, 207
ecs::Registry::Pool< T >, 277
add, 279
data, 280
get, 279
has, 279
remove, 279
ecs::Scale, 313
x, 315
y, 315
ecs::Text, 351
content, 353
font_size, 353
ecs::Texture, 360
path, 362
ecs::Transform, 362
rotation, 364
x, 364
y, 364
ecs::Velocity, 364
x, 366
y, 366
ENEMY
rnp, 50
ENEMY_RADIUS
cli::GameConfig::Hitbox, 31
enemyRect
cli::SpriteRect, 42
EnemySystem
cli::EnemySystem, 148
eng, 44
A, 46
B, 46
C, 46
Closed, 45
D, 46
DEFAULT, 47
Down, 46
E, 46
Enter, 46
Escape, 46
EventType, 45
F, 46
G, 46
H, 46
I, 46
id, 45
J, 46
K, 46
Key, 45
KeyPressed, 45
KeyReleased, 45
L, 46
Left, 46
M, 46
N, 46
None, 45

Num0, 46
 Num1, 46
 Num2, 46
 Num3, 46
 Num4, 46
 Num5, 46
 Num6, 46
 Num7, 46
 Num8, 47
 Num9, 47
 O, 46
 P, 46
 Paused, 47
 Playing, 47
 Q, 46
 R, 46
 Right, 46
 RUN, 47
 S, 46
 Space, 46
 State, 47
 Status, 47
 STOP, 47
 Stopped, 47
 T, 46
 U, 46
 Unknown, 46
 Up, 46
 V, 46
 W, 46
 X, 46
 Y, 46
 Z, 46
 eng::AScene, 84
 ~AScene, 86
 AScene, 86
 getId, 87
 getName, 87
 getRegistry, 87
 m_id, 88
 m_name, 88
 m_registry, 88
 operator=, 87
 s_nextId, 88
 setName, 87
 eng::ASystem, 102
 isEnable, 103
 m_isEnable, 103
 setEnable, 103
 eng::Color, 139
 a, 140
 b, 140
 g, 140
 r, 140
 eng::Engine, 150
 ~Engine, 152
 addSystem, 153
 Engine, 152, 153
 getAudio, 153
 getClock, 153
 getNetwork, 154
 getRenderer, 154
 getSceneManager, 154
 getState, 154
 m_audio, 156
 m_clock, 156
 m_network, 156
 m_renderer, 156
 m_sceneManager, 156
 m_state, 157
 m_systems, 157
 operator=, 155
 render, 155
 setState, 155
 stop, 155
 updateSystems, 156
 eng::Event, 163
 key, 164
 type, 164
 eng::IAudio, 186
 createAudio, 188
 isPlaying, 188
 playAudio, 188
 setLoop, 188
 setVolume, 189
 stopAudio, 189
 eng::INetworkClient, 196
 ~INetworkClient, 199
 connect, 199
 disconnect, 199
 getServerTickRate, 200
 getSessionId, 200
 PacketHandler, 199
 sendAck, 200
 sendConnect, 200
 sendConnectWithCaps, 200
 sendDisconnect, 200
 sendPing, 200
 sendPlayerInput, 201
 sendPlayerInputAsEvent, 201
 setEventsHandler, 201
 setPacketHandler, 201
 eng::IRenderer, 208
 clearWindow, 210
 closeWindow, 210
 createFont, 210
 createSprite, 210
 createText, 211
 createTexture, 211
 createWindow, 211
 displayWindow, 211
 drawPoint, 212
 drawSprite, 212
 drawText, 212
 getWindowSize, 213
 pollEvent, 213

setFrameLimit, 213
setSpriteFrame, 213
setSpritePosition, 213
setSpriteScale, 214
setSpriteTexture, 214
setTextColor, 214
setTextContent, 215
setTextPosition, 215
windowIsOpen, 215
eng::IScene, 216
 ~IScene, 218
 event, 218
 getId, 218
 getName, 219
 getRegistry, 219
 setName, 219
 update, 219
eng::ISystem, 224
 ~ISystem, 225
 isEnabled, 225
 setEnabled, 225
 update, 225
eng::SceneManager, 315
 ~SceneManager, 317
 addScene, 317
 getCurrentScene, 317
 getScene, 318
 m_currentSceneId, 318
 m_scenes, 318
 operator=, 318
 SceneManager, 317
 switchToScene, 318
eng::Text, 353
 color, 354
 content, 354
 font_name, 354
 name, 354
 size, 354
 x, 354
 y, 355
eng::WindowSize, 376
 height, 376
 width, 376
Engine
 eng::Engine, 152, 153
Enter
 eng, 46
entities
 rnp::PacketWorldState, 260
Entity
 ecs, 44
ENTITY_EVENT
 rnp, 51
EntityBuilder
 ecs::Registry::EntityBuilder, 159
entityCount
 rnp::PacketWorldState, 260
entityId
 rnp::EventRecord, 165
EntityType
 rnp, 49
EntryPointFn
 utl, 57
ErrorCode
 rnp, 50
errorCode
 rnp::PacketError, 255
Escape
 eng, 46
event
 cli::Game, 182
 cli::Lobby, 236
 cli::Settings, 330
 eng::IScene, 218
EventType
 eng, 45
 rnp, 50
exit
 cli::ArgsConfig, 74
 srv::ArgsConfig, 77
ExplosionSystem
 cli::ExplosionSystem, 172

F
 eng, 46
FIRE_COOLDOWN
 cli::GameConfig::Projectile::Basic, 34
 cli::GameConfig::Projectile::Supercharged, 36
flags
 rnp::PacketHeader, 256
font_name
 eng::Text, 354
font_size
 ecs::Text, 353
FONTS_RTYPE
 cli::Path::Font, 40
formatLogMessage
 utl::Logger, 244
FRAG
 rnp, 51
fragCount
 rnp::FragmentHeader, 178
fragId
 rnp::FragmentHeader, 178
fragIndex
 rnp::FragmentHeader, 178
FRAME_DURATION
 cli::GameConfig::Animation, 20
frame_duration
 ecs::Animation, 68
 ecs::Explosion, 168
 ecs::LoadingAnimation, 228
frame_height
 ecs::Animation, 68
 ecs::Explosion, 168
 ecs::LoadingAnimation, 228
frame_width

ecs::Animation, 68
 ecs::Explosion, 168
 ecs::LoadingAnimation, 228
 frameH
 cli::SpriteRect::SpriteSheet::Asteroid, 94
 cli::SpriteRect::SpriteSheet::Enemy, 141
 frameLimit
 cli::ArgsConfig, 74
 FRAMES_PER_ROW
 cli::GameConfig::Asteroid::Small, 24
 cli::GameConfig::Enemy::Easy, 27
 cli::GameConfig::Explosion, 29
 cli::GameConfig::Player, 33
 frames_per_row
 ecs::Animation, 68
 ecs::Explosion, 169
 ecs::LoadingAnimation, 228
 frameW
 cli::SpriteRect::SpriteSheet::Asteroid, 94
 cli::SpriteRect::SpriteSheet::Enemy, 141
 fromFile
 cli::ArgsConfig, 74
 srv::ArgsConfig, 77
 fullscreen
 cli::ArgsConfig, 74

G
 eng, 46

g
 ecs::Color, 139
 eng::Color, 140
 gme::Sprite, 344
 Game
 cli::Game, 181, 182
 game.cpp
 BLUE, 458
 BLUE_SECOND, 458
 GREEN, 458
 PURPLE, 458
 WHITE, 458
 WHITE_TRANS, 458
 YELLOW, 459
 get
 ecs::Registry::Pool< T >, 279
 getAll
 ecs::Registry, 299
 getAudio
 eng::Engine, 153
 getClock
 eng::Engine, 153
 getColorForDuration
 utl::Logger, 244
 getComponent
 ecs::Registry, 300
 getCurrentScene
 eng::SceneManager, 317
 gme::IGameClient, 193
 gme::RTTypeClient, 309
 srv::SceneManager, 321

 getDeltaSeconds
 utl::Clock, 124
 getElapsed
 utl::Clock, 124
 getEntities
 gme::IScene, 221
 gme::LobbyScene, 240
 getEntitiesMutable
 gme::IScene, 221
 gme::LobbyScene, 240
 getEntryPoint
 utl::PluginLoader, 274
 getEnvMap
 utl, 58
 getId
 eng::AScene, 87
 eng::IScene, 218
 srv::AScene, 92
 srv::IScene, 223
 getName
 eng::AScene, 87
 eng::IScene, 219
 gme::AGameClient, 62
 gme::AGameServer, 65
 gme::IGameClient, 193
 gme::IGameServer, 196
 gme::IScene, 221
 gme::LobbyScene, 241
 srv::AScene, 92
 srv::IScene, 223
 utl::IPlugin, 206
 getNetwork
 eng::Engine, 154
 getPool
 ecs::Registry, 301
 getRegistry
 eng::AScene, 87
 eng::IScene, 219
 srv::AScene, 92
 srv::IScene, 224
 getRenderer
 eng::Engine, 154
 getScene
 eng::SceneManager, 318
 srv::SceneManager, 322
 getSceneManager
 eng::Engine, 154
 getServerTickRate
 eng::INetworkClient, 200
 getSessionId
 eng::INetworkClient, 200
 getState
 eng::Engine, 154
 getType
 utl::IPlugin, 206
 getWindowSize
 eng::IRenderer, 213
 GIT_COMMIT_HASH

Version.hpp, 393, 396
GIT_TAG
 Version.hpp, 394, 396
gme, 47
gme::AGameClient, 59
 ~AGameClient, 62
 getName, 62
 m_name, 63
 setName, 62
gme::AGameServer, 63
 ~AGameServer, 65
 getName, 65
 m_name, 65
 setName, 65
gme::IGameClient, 191
 ~IGameClient, 193
 getCurrentScene, 193
 getName, 193
 setName, 194
 update, 194
gme::IGameServer, 194
 ~IGameServer, 196
 getName, 196
 setName, 196
gme::IScene, 220
 ~IScene, 221
 getEntities, 221
 getEntitiesMutable, 221
 getName, 221
gme::LobbyScene, 238
 addEntity, 240
 getEntities, 240
 getEntitiesMutable, 240
 getName, 241
 LobbyScene, 240
 m_entities, 241
 m_name, 241
gme::RTypClient, 306
 ~RTypClient, 308
 getCurrentScene, 309
 m_currentScene, 309
 operator=, 309
 RTypClient, 308
 update, 309
gme::RTypServer, 310
 ~RTypServer, 312
 operator=, 312
 RTypServer, 312
gme::Sprite, 342
 a, 344
 b, 344
 g, 344
 id, 344
 pos_x, 344
 pos_y, 344
 r, 345
 scale_x, 345
 scale_y, 345
text_rect_fx, 345
text_rect_fy, 345
text_rect_x, 345
text_rect_y, 345
texture_path, 345
type, 346
v_x, 346
v_y, 346
GREEN
 game.cpp, 458

H
 eng, 46
h
 cli::SpriteRect::Rect, 292
handle
 utl::SharedLib, 334
handleEvents
 cli::Client, 120
has
 ecs::Registry::Pool< T >, 279
hasComponent
 ecs::Registry, 302
HEALTH
 cli::GameConfig::Asteroid::Large, 21
 cli::GameConfig::Asteroid::Medium, 22
 cli::GameConfig::Asteroid::Small, 24
 cli::GameConfig::Enemy::Easy, 27
health
 ecs::Asteroid, 97
 ecs::Enemy, 145
HEIGHT
 cli::GameConfig::Screen, 37
height
 cli::ArgsConfig, 75
 eng::WindowSize, 376
HELP_MESSAGE
 argsHandler.cpp, 445, 448
hideLoadingAnimation
 cli::WeaponSystem, 370
host
 cli::ArgsConfig, 75
 srv::ArgsConfig, 77

I
 eng, 46
id
 ecs::IComponent, 191
 eng, 45
 gme::Sprite, 344
 rnp:: EntityState, 161
 srv, 54
INFO
 utl, 58
init
 srv::INetworkServer, 204
 utl::Logger, 245
INPUT
 rnp, 50

INTERNAL_ERROR
 rnp, 50

INVALID_ENTITY
 ecs, 44

INVALID_PAYLOAD
 rnp, 50

is_cli
 ecs::Player, 268

isEnabled
 cli::AsteroidSystem, 100
 cli::CollisionSystem, 133
 cli::EnemySystem, 148
 cli::ExplosionSystem, 173
 cli::SpawnSystem, 338
 eng::ASystem, 103
 eng::ISystem, 225

isPlaying
 eng::IAudio, 188

J
 eng, 46

json
 cli, 18
 srv, 54

K
 eng, 46

Key
 eng, 45

key
 eng::Event, 164

KeyPressed
 eng, 45

KeyReleased
 eng, 45

L
 eng, 46

LARGE
 ecs::Asteroid, 97

last_shot_time
 ecs::Enemy, 145

Left
 eng, 46

length
 rnp::PacketHeader, 256

LibHandle
 utl, 57

LIFETIME
 cli::GameConfig::Explosion, 29
 cli::GameConfig::Projectile::Basic, 35
 cli::GameConfig::Projectile::Supercharged, 36

lifetime
 ecs::Explosion, 169
 ecs::Projectile, 282

LoadingAnimationSystem
 cli::LoadingAnimationSystem, 231

loadLibrary
 utl::PluginLoader, 274

loadPlugin
 utl::PluginLoader, 274

Lobby
 cli::Lobby, 235, 236

lobby.cpp
 WHITE, 464

LobbyScene
 gme::LobbyScene, 240

log
 utl::Logger, 245

LOG_LEVEL_COLOR
 utl::Logger, 247

LOG_LEVEL_STRING
 utl::Logger, 247

logExecutionTime
 utl::Logger, 246

Logger
 utl::Logger, 243, 244

LogLevel
 utl, 57

loop
 ecs::Audio, 106

M
 eng, 46

m_asteroidCounterEntity
 cli::Game, 183

m_asteroidSpawnTimer
 cli::SpawnSystem, 341

m_audio
 cli::AudioSystem, 110
 cli::Game, 183
 cli::Lobby, 237
 cli::Settings, 331
 eng::Engine, 156

m_clock
 eng::Engine, 156

m_components
 ecs::Registry, 305

m_currentScene
 gme::RTTypeClient, 309

m_currentScenId
 eng::SceneManager, 318
 srv::SceneManager, 322

m_enemyCounterEntity
 cli::Game, 183

m_enemySpawnTimer
 cli::SpawnSystem, 341

m_engine
 cli::Client, 121

m_entities
 ecs::Registry, 305
 gme::LobbyScene, 241

m_entity
 ecs::Registry::EntityBuilder, 160

m_fireCooldown
 cli::WeaponSystem, 375

m_fpsEntity
 cli::Game, 183

cli::Lobby, 237
m_game
 cli::Client, 121
m_handles
 utl::PluginLoader, 276
m_id
 eng::AScene, 88
 srv::AScene, 93
m_isCharging
 cli::WeaponSystem, 375
m_isEnable
 eng::ASystem, 103
m_isPaused
 utl::Clock, 127
m_keysPressed
 cli::Client, 121
 cli::Game, 183
 cli::Lobby, 237
 cli::Settings, 331
m_lastEntity
 ecs::Registry, 305
m_menuOptions
 cli::Lobby, 237
m_mutex
 utl::PluginLoader, 276
m_name
 eng::AScene, 88
 gme::AGameClient, 63
 gme::AGameServer, 65
 gme::LobbyScene, 241
 srv::AScene, 93
m_network
 eng::Engine, 156
 srv::Server, 326
m_onComponentAddedCallbacks
 ecs::Registry, 305
m_pausedDuration
 utl::Clock, 127
m_pausedTime
 utl::Clock, 127
M_PI
 PlayerDirection.hpp, 429
m_playerEntity
 cli::Game, 183
m_pluginLoader
 cli::Client, 121
 srv::Server, 326
m_plugins
 utl::PluginLoader, 276
m_registry
 ecs::Registry::EntityBuilder, 160
 eng::AScene, 88
 srv::AScene, 93
m_renderer
 cli::AsteroidSystem, 102
 cli::BeamSystem, 117
 cli::CollisionSystem, 136
 cli::EnemySystem, 150
cli::ExplosionSystem, 174
cli::LoadingAnimationSystem, 232
cli::PixelSystem, 265
cli::SpawnSystem, 341
cli::SpriteSystem, 350
cli::TextSystem, 359
 eng::Engine, 156
m_sceneManager
 eng::Engine, 156
m_scenes
 eng::SceneManager, 318
 srv::SceneManager, 322
m_selectedIndex
 cli::Lobby, 237
m_start
 utl::Clock, 128
m_state
 eng::Engine, 157
m_systems
 eng::Engine, 157
m_waveTimer
 cli::SpawnSystem, 341
main
 main.cpp, 453, 455
main.cpp
 main, 453, 455
marginX
 cli::SpriteRect::SpriteSheet::Asteroid, 94
 cli::SpriteRect::SpriteSheet::Enemy, 142
marginY
 cli::SpriteRect::SpriteSheet::Asteroid, 94
 cli::SpriteRect::SpriteSheet::Enemy, 142
MAX_CHARGE
 cli::GameConfig::Beam, 26
max_charge
 ecs::BeamCharge, 113
MAX_CLIENTS
 srv, 54
max_health
 ecs::Enemy, 145
MAX_IP_LENGTH
 srv, 54
MAX_LEN_RECV_BUFFER
 srv, 55
MAX_PAYLOAD
 rnp, 53
MAX_Y
 cli::GameConfig::Screen, 37
MEDIUM
 ecs::Asteroid, 97
MIN_Y
 cli::GameConfig::Screen, 37
msgLen
 rnp::PacketError, 255
mtuPayloadBytes
 rnp::PacketConnectAccept, 253
N
 eng, 46

name
 eng::Text, 354
 nameLen
 rnp::PacketConnect, 251
 NETWORK_CLIENT
 utl, 58
 network_lib_path
 cli::ArgsConfig, 75
 srv::ArgsConfig, 77
 NETWORK_SERVER
 utl, 58
 nonce
 rnp::PacketPingPong, 258
 NONE
 rnp, 51
 None
 eng, 45
 now
 utl::Clock, 125
 Num0
 eng, 46
 Num1
 eng, 46
 Num2
 eng, 46
 Num3
 eng, 46
 Num4
 eng, 46
 Num5
 eng, 46
 Num6
 eng, 46
 Num7
 eng, 46
 Num8
 eng, 47
 Num9
 eng, 47

O

 eng, 46

OBSTACLE

 rnp, 50

OFFSET_X

 cli::GameConfig::LoadingAnimation, 32

OFFSET_Y

 cli::GameConfig::LoadingAnimation, 32

onComponentAdded

 ecs::Registry, 303

onLeave

 cli::Settings, 331

onOptionSelected

 cli::Lobby, 237

operator bool

 utl::SharedLib, 333

operator<<

 utl::Clock, 127

operator=

 cli::AnimationSystem, 71
 cli::ArgsHandler, 79
 cli::AsteroidSystem, 100, 101
 cli::AudioSystem, 110
 cli::BeamSystem, 116
 cli::Client, 120
 cli::CollisionSystem, 133
 cli::EnemySystem, 148, 149
 cli::ExplosionSystem, 173
 cli::Game, 182
 cli::LoadingAnimationSystem, 231
 cli::Lobby, 236
 cli::PixelSystem, 264
 cli::PlayerDirectionSystem, 270
 cli::ProjectileManager, 287
 cli::ProjectileSystem, 290
 cli::Settings, 330
 cli::SpawnSystem, 338
 cli::SpriteSystem, 349
 cli::TextSystem, 359
 cli::WeaponSystem, 371
 ecs::Registry, 303
 eng::AScene, 87
 eng::Engine, 155
 eng::SceneManager, 318
 gme::RTYPEClient, 309
 gme::RTYPEServer, 312
 srv::ArgsHandler, 82
 srv::AScene, 92
 srv::SceneManager, 322
 srv::Server, 326
 utl::Clock, 125, 126
 utl::Logger, 247
 utl::PluginLoader, 275
 utl::SharedLib, 333

P

 eng, 46

PACKET_ERROR

 rnp, 51

PacketFlags

 rnp, 50

PacketHandler

 eng::INetworkClient, 199

PacketType

 rnp, 51

ParseArgs

 cli::ArgsHandler, 80
 srv::ArgsHandler, 82

ParseEnv

 cli::ArgsHandler, 80
 srv::ArgsHandler, 83

path

 ecs::Audio, 106
 ecs::Font, 177
 ecs::Texture, 362

pause

 utl::Clock, 126

Paused

eng, 47
PING
 rnp, 51
PixelSystem
 cli::PixelSystem, 264
play
 ecs::Audio, 106
playAudio
 eng::IAudio, 188
PLAYER
 rnp, 50
PLAYER_INPUT
 rnp, 51
PLAYER_RADIUS
 cli::GameConfig::Hitbox, 31
PlayerDirection.hpp
 M_PI, 429
PlayerDirectionSystem
 cli::PlayerDirectionSystem, 270
playerName
 rnp::PacketConnect, 251
Playing
 eng, 47
PLUGIN_AUDIO_SFML
 cli::Path::Plugin, 40
PLUGIN_NETWORK_ASIO_CLIENT
 cli::Path::Plugin, 40
PLUGIN_RENDERER_SFML
 cli::Path::Plugin, 40
PluginLoader
 utl::PluginLoader, 273
PluginLoader.hpp
 PLUGINS_EXTENSION, 523
PLUGINS_EXTENSION
 PluginLoader.hpp, 523
PLUGINS_NETWORK_ASIO_SERVER
 srv::Path::Plugin, 56
PluginType
 utl, 58
pollEvent
 eng::IRenderer, 213
PONG
 rnp, 51
port
 cli::ArgsConfig, 75
 srv::ArgsConfig, 78
pos_x
 ecs::Rect, 295
 gme::Sprite, 344
pos_y
 ecs::Rect, 295
 gme::Sprite, 344
POWERUP
 rnp, 50
PROJECT_NAME
 Version.hpp, 394, 396
PROJECT_VERSION
 Version.hpp, 394, 396
PROJECT_VERSION_MAJOR
 Version.hpp, 394, 396
PROJECT_VERSION_MINOR
 Version.hpp, 394, 396
PROJECT_VERSION_PATCH
 Version.hpp, 394, 396
PROJECTILE
 rnp, 50
PROJECTILE_BASIC_RADIUS
 cli::GameConfig::Hitbox, 31
PROJECTILE_SUPERCHARGED_RADIUS
 cli::GameConfig::Hitbox, 31
ProjectileManager
 cli::ProjectileManager, 284
ProjectileSystem
 cli::ProjectileSystem, 290
PROTOCOL_ERROR
 rnp, 49
PROTOCOL_VERSION
 rnp, 53
PURPLE
 game.cpp, 458

Q
 eng, 46

R
 eng, 46
r
 ecs::Color, 139
 eng::Color, 140
 gme::Sprite, 345
r-type, 1
radius
 ecs::Hitbox, 186
RATE_LIMITED
 rnp, 50
readFile
 utl, 58
reasonCode
 rnp::PacketDisconnect, 254
Registry
 ecs::Registry, 297
RELIABLE
 rnp, 51
remove
 ecs::Registry::IPool, 207
 ecs::Registry::Pool< T >, 279
REMOVE_MAX_Y
 cli::GameConfig::Screen, 38
REMOVE_MIN_Y
 cli::GameConfig::Screen, 38
REMOVE_X
 cli::GameConfig::Screen, 38
removeAsteroid
 cli::CollisionSystem, 133
removeComponent
 ecs::Registry, 304
removeEnemy

cli::CollisionSystem, 134
 removeExplosion
 cli::ExplosionSystem, 173
 removeProjectile
 cli::CollisionSystem, 135
 render
 eng::Engine, 155
 RENDERER
 utl, 58
 renderer_lib_path
 cli::ArgsConfig, 75
 reserved
 rnp::PacketHeader, 257
 reset
 cli::WeaponSystem, 371
 restart
 utl::Clock, 126
 resume
 utl::Clock, 126
 Right
 eng, 46
 rnp, 48
 ACK, 51
 ACK_REQ, 51
 BANNED, 49
 CLIENT_REQUEST, 49
 COMPRESSED, 51
 CONNECT, 51
 CONNECT_ACCEPT, 51
 CUSTOM, 50
 DAMAGE, 50
 deserializeEvents, 51
 deserializeHeader, 51
 DESPAWN, 50
 DISCONNECT, 51
 DisconnectReason, 49
 ENEMY, 50
 ENTITY_EVENT, 51
 EntityType, 49
 ErrorCode, 50
 EventType, 50
 FRAG, 51
 INPUT, 50
 INTERNAL_ERROR, 50
 INVALID_PAYLOAD, 50
 MAX_PAYLOAD, 53
 NONE, 51
 OBSTACLE, 50
 PACKET_ERROR, 51
 PacketFlags, 50
 PacketType, 51
 PING, 51
 PLAYER, 50
 PLAYER_INPUT, 51
 PONG, 51
 POWERUP, 50
 PROJECTILE, 50
 PROTOCOL_ERROR, 49
 PROTOCOL_VERSION, 53
 RATE_LIMITED, 50
 RELIABLE, 51
 SCORE, 50
 serialize, 52
 serializeEvents, 52
 serializeHeader, 52
 SERVER_FULL, 49
 SERVER_SHUTDOWN, 49
 SPAWN, 50
 TIMEOUT, 49
 UNAUTHORIZED_SESSION, 50
 UNSPECIFIED, 49
 WORLD_STATE, 51
 rnp::EntityState, 160
 id, 161
 stateFlags, 161
 type, 161
 vx, 161
 vy, 161
 x, 161
 y, 162
 rnp::EventRecord, 164
 data, 165
 entityId, 165
 type, 165
 rnp::FragmentHeader, 177
 fragCount, 178
 fragId, 178
 fragIndex, 178
 rnp::PacketAck, 250
 ackBits, 250
 cumulativeAck, 250
 rnp::PacketConnect, 251
 clientCaps, 251
 nameLen, 251
 playerName, 251
 rnp::PacketConnectAccept, 252
 mtuPayloadBytes, 253
 serverCaps, 253
 sessionId, 253
 tickRateHz, 253
 rnp::PacketDisconnect, 253
 reasonCode, 254
 rnp::PacketError, 254
 description, 255
 errorCode, 255
 msgLen, 255
 rnp::PacketHeader, 256
 flags, 256
 length, 256
 reserved, 257
 sequence, 257
 sessionId, 257
 type, 257
 rnp::PacketPingPong, 258
 nonce, 258
 sendTimeMs, 258

rnp::PacketWorldState, 259
 entities, 260
 entityCount, 260
 serverTick, 260
rotation
 ecs::Transform, 364
ROTATION_SPEED
 cli::GameConfig::Asteroid::Large, 21
 cli::GameConfig::Asteroid::Medium, 22
 cli::GameConfig::Asteroid::Small, 24
rotation_speed
 ecs::Asteroid, 97
RTTypeClient
 gme::RTTypeClient, 308
RTTypeServer
 gme::RTTypeServer, 312
RUN
 eng, 47
run
 cli::Client, 120
 srv::Server, 326

S
 eng, 46
s_nextId
 eng::AScene, 88
 srv::AScene, 93
SCALE
 cli::GameConfig::Asteroid::Large, 21
 cli::GameConfig::Asteroid::Medium, 22
 cli::GameConfig::Asteroid::Small, 24
 cli::GameConfig::Enemy::Easy, 28
 cli::GameConfig::Explosion, 30
 cli::GameConfig::Player, 33
 cli::GameConfig::Projectile::Basic, 35
 cli::GameConfig::Projectile::Supercharged, 36
scale_x
 gme::Sprite, 345
scale_y
 gme::Sprite, 345
SceneManager
 eng::SceneManager, 317
 srv::SceneManager, 321
SCORE
 rnp, 50
sendAck
 eng::INetworkClient, 200
sendConnect
 eng::INetworkClient, 200
sendConnectWithCaps
 eng::INetworkClient, 200
sendDisconnect
 eng::INetworkClient, 200
sendPing
 eng::INetworkClient, 200
sendPlayerInput
 eng::INetworkClient, 201
sendPlayerInputAsEvent
 eng::INetworkClient, 201
sendTimeMs
 rnp::PacketPingPong, 258
sequence
 rnp::PacketHeader, 257
serialize
 rnp, 52
serializeEvents
 rnp, 52
serializeHeader
 rnp, 52
Server
 srv::Server, 325
SERVER_FULL
 rnp, 49
SERVER_SHUTDOWN
 rnp, 49
serverCaps
 rnp::PacketConnectAccept, 253
serverTick
 rnp::PacketWorldState, 260
sessionId
 rnp::PacketConnectAccept, 253
 rnp::PacketHeader, 257
setEnable
 cli::AsteroidSystem, 101
 cli::CollisionSystem, 135
 cli::EnemySystem, 149
 cli::ExplosionSystem, 173
 cli::SpawnSystem, 338
 eng::ASystem, 103
 eng::ISystem, 225
setEventsHandler
 eng::INetworkClient, 201
setFrameLimit
 eng::IRenderer, 213
setLoop
 eng::IAudio, 188
setName
 eng::AScene, 87
 eng::IScene, 219
 gme::AGameClient, 62
 gme::AGameServer, 65
 gme::IGameClient, 194
 gme::IGameServer, 196
 srv::AScene, 92
 srv::IScene, 224
setPacketHandler
 eng::INetworkClient, 201
setServerCapabilities
 srv::INetworkServer, 204
setSpriteFrame
 eng::IRenderer, 213
setSpritePosition
 eng::IRenderer, 213
setSpriteScale
 eng::IRenderer, 214
setSpriteTexture
 eng::IRenderer, 214

setState
 eng::Engine, 155
 setTextColor
 eng::IRenderer, 214
 setTextContent
 eng::IRenderer, 215
 setTextPosition
 eng::IRenderer, 215
 setTickRate
 srv::INetworkServer, 204
 Settings
 cli::Settings, 329, 330
 settings.cpp
 WHITE, 467
 setVolume
 eng::IAudio, 189
 SharedLib
 utl::SharedLib, 332, 333
 SHOOT_COOLDOWN
 cli::GameConfig::Enemy::Easy, 28
 shoot_cooldown
 ecs::Enemy, 145
 showLoadingAnimation
 cli::WeaponSystem, 371
 Size
 ecs::Asteroid, 97
 size
 ecs::Asteroid, 97
 eng::Text, 354
 size_x
 ecs::Rect, 295
 size_y
 ecs::Rect, 295
 SMALL
 ecs::Asteroid, 97
 Space
 eng, 46
 spacingX
 cli::SpriteRect::SpriteSheet::Asteroid, 94
 cli::SpriteRect::SpriteSheet::Enemy, 142
 spacingY
 cli::SpriteRect::SpriteSheet::Asteroid, 95
 cli::SpriteRect::SpriteSheet::Enemy, 142
 SPAWN
 rnp, 50
 SPAWN_RATE
 cli::GameConfig::Asteroid::Large, 21
 cli::GameConfig::Asteroid::Medium, 23
 cli::GameConfig::Asteroid::Small, 24
 cli::GameConfig::Enemy::Easy, 28
 SPAWN_X
 cli::GameConfig::Screen, 38
 spawnAsteroid
 cli::SpawnSystem, 338
 spawnEnemy
 cli::SpawnSystem, 339
 SpawnSystem
 cli::SpawnSystem, 337
 spawnWave
 cli::SpawnSystem, 340
 SPEED
 cli::GameConfig::Asteroid::Large, 21
 cli::GameConfig::Asteroid::Medium, 23
 cli::GameConfig::Asteroid::Small, 25
 cli::GameConfig::Enemy::Easy, 28
 cli::GameConfig::Player, 33
 cli::GameConfig::Projectile::Basic, 35
 cli::GameConfig::Projectile::Supercharged, 36
 speed
 ecs::Enemy, 145
 SPRITE_HEIGHT
 cli::GameConfig::Asteroid::Large, 22
 cli::GameConfig::Asteroid::Medium, 23
 cli::GameConfig::Asteroid::Small, 25
 cli::GameConfig::Enemy::Easy, 28
 cli::GameConfig::Explosion, 30
 cli::GameConfig::LoadingAnimation, 32
 cli::GameConfig::Player, 33
 cli::GameConfig::Projectile::Basic, 35
 cli::GameConfig::Projectile::Supercharged, 37
 SPRITE_WIDTH
 cli::GameConfig::Asteroid::Large, 22
 cli::GameConfig::Asteroid::Medium, 23
 cli::GameConfig::Asteroid::Small, 25
 cli::GameConfig::Enemy::Easy, 28
 cli::GameConfig::Explosion, 30
 cli::GameConfig::LoadingAnimation, 32
 cli::GameConfig::Player, 34
 cli::GameConfig::Projectile::Basic, 35
 cli::GameConfig::Projectile::Supercharged, 37
 SpriteSystem
 cli::SpriteSystem, 349
 srv, 53
 id, 54
 json, 54
 MAX_CLIENTS, 54
 MAX_IP_LENGTH, 54
 MAX_LEN_RECV_BUFFER, 55
 srv::ArgsConfig, 76
 exit, 77
 fromFile, 77
 host, 77
 network_lib_path, 77
 port, 78
 srv::ArgsHandler, 81
 ~ArgsHandler, 82
 ArgsHandler, 82
 operator=, 82
 ParseArgs, 82
 ParseEnv, 83
 srv::AScene, 89
 ~AScene, 91
 AScene, 91
 getId, 92
 getName, 92
 getRegistry, 92

m_id, 93
 m_name, 93
 m_registry, 93
 operator=, 92
 s_nextId, 93
 setName, 92
 srv::Config, 55
 srv::Config::Network, 55
 DEFAULT_MAX_CLIENT, 55
 DEFAULT_NETWORK_HOST, 55
 DEFAULT_NETWORK_PORT, 55
 srv::EnvConfig, 163
 srv::Game, 56
 DEFAULT_TICK_RATE, 56
 DEFAULT_UPDATE_INTERVAL, 56
 srv::INetworkServer, 201
 ~INetworkServer, 204
 init, 204
 setServerCapabilities, 204
 setTickRate, 204
 start, 204
 stop, 204
 srv::IScene, 222
 ~IScene, 223
 getId, 223
 getName, 223
 getRegistry, 224
 setName, 224
 update, 224
 srv::Path, 56
 srv::Path::Plugin, 56
 PLUGINS_NETWORK_ASIO_SERVER, 56
 srv::SceneManager, 319
 ~SceneManager, 321
 addScene, 321
 getCurrentScene, 321
 getScene, 322
 m_currentSceneId, 322
 m_scenes, 322
 operator=, 322
 SceneManager, 321
 switchToScene, 322
 srv::Server, 323
 ~Server, 325
 m_network, 326
 m_pluginLoader, 326
 operator=, 326
 run, 326
 Server, 325
 start
 srv::INetworkServer, 204
 State
 eng, 47
 stateFlags
 rnp:: EntityState, 161
 Status
 eng, 47
 STOP

 eng, 47
stop
 eng::Engine, 155
 srv::INetworkServer, 204
stopAudio
 eng::IAudio, 189
Stopped
 eng, 47
SUPERCHARGED
 ecs::Projectile, 282
switchToScene
 eng::SceneManager, 318
 srv::SceneManager, 322

T

 eng, 46
text_rect_fx
 gme::Sprite, 345
text_rect_fy
 gme::Sprite, 345
text_rect_x
 gme::Sprite, 345
text_rect_y
 gme::Sprite, 345
TextSystem
 cli::TextSystem, 358
TEXTURE_ASTEROID
 cli::Path::Texture, 41
TEXTURE_ENEMY_EASY
 cli::Path::Texture, 41
TEXTURE_EXPLOSION
 cli::Path::Texture, 41
texture_path
 gme::Sprite, 345
TEXTURE_PLAYER
 cli::Path::Texture, 41
TEXTURE_SHOOT
 cli::Path::Texture, 41
TEXTURE_SHOOT_CHARGED
 cli::Path::Texture, 42
TEXTURE_SHOOT_LOADING
 cli::Path::Texture, 42
tickRateHz
 rnp::PacketConnectAccept, 253
TIMEOUT
 rnp, 49
TimePoint
 utl::Clock, 123
TOTAL_FRAMES
 cli::GameConfig::Player, 34
total_frames
 ecs::Animation, 68
 ecs::Explosion, 169
 ecs::LoadingAnimation, 228
totalFrames
 cli::SpriteRect::SpriteSheet::Asteroid, 95
 cli::SpriteRect::SpriteSheet::Enemy, 142
tryFireBasic
 cli::WeaponSystem, 372

tryFireSupercharged
 cli::WeaponSystem, 373

Type
 ecs::Projectile, 282

type
 ecs::Projectile, 282
 eng::Event, 164
 gme::Sprite, 346
 rnp:: EntityState, 161
 rnp::EventRecord, 165
 rnp::PacketHeader, 257

U
 eng, 46

UNAUTHORIZED_SESSION
 rnp, 50

UNKNOWN
 utl, 58

Unknown
 eng, 46

UNSPECIFIED
 rnp, 49

Up
 eng, 46

update
 cli::AnimationSystem, 71
 cli::AsteroidSystem, 101
 cli::AudioSystem, 110
 cli::BeamSystem, 117
 cli::CollisionSystem, 135
 cli::EnemySystem, 149
 cli::ExplosionSystem, 174
 cli::Game, 182
 cli::LoadingAnimationSystem, 232
 cli::Lobby, 236
 cli::PixelSystem, 265
 cli::PlayerDirectionSystem, 270
 cli::ProjectileSystem, 290
 cli::Settings, 330
 cli::SpawnSystem, 340
 cli::SpriteSystem, 350
 cli::TextSystem, 359
 cli::WeaponSystem, 374
 eng::IScene, 219
 eng::ISystem, 225
 gme::IGameClient, 194
 gme::RTTypeClient, 309
 srv::IScene, 224

updateSystems
 eng::Engine, 156

utl, 57
 AUDIO, 58
 EntryPointFn, 57
 getEnvMap, 58
 INFO, 58
 LibHandle, 57
 LogLevel, 57
 NETWORK_CLIENT, 58
 NETWORK_SERVER, 58

PluginType, 58
 readFile, 58
 RENDERER, 58
 UNKNOWN, 58
 WARNING, 58

utl::Clock, 122
 ~Clock, 124
 Clock, 124
 Duration, 123
 getDeltaSeconds, 124
 getElapsed, 124
 m_isPaused, 127
 m_pausedDuration, 127
 m_pausedTime, 127
 m_start, 128
 now, 125
 operator<<, 127
 operator=, 125, 126
 pause, 126
 restart, 126
 resume, 126
 TimePoint, 123

utl::IPlugin, 205
 ~IPlugin, 206
 getName, 206
 getType, 206

utl::Logger, 242
 ~Logger, 244
 COLOR_ERROR, 243
 COLOR_INFO, 243
 COLOR_RESET, 243
 COLOR_WARNING, 243
 ColorIndex, 243
 formatLogMessage, 244
 getColorForDuration, 244
 init, 245
 log, 245
 LOG_LEVEL_COLOR, 247
 LOG_LEVEL_STRING, 247
 logExecutionTime, 246
 Logger, 243, 244
 operator=, 247

utl::PluginLoader, 271
 ~PluginLoader, 273
 getEntryPoint, 274
 loadLibrary, 274
 loadPlugin, 274
 m_handles, 276
 m_mutex, 276
 m_plugins, 276
 operator=, 275
 PluginLoader, 273

utl::SharedLib, 331
 ~SharedLib, 332
 close, 333
 handle, 334
 operator bool, 333
 operator=, 333

SharedLib, 332, 333

V
 eng, 46

v_x
 gme::Sprite, 346

v_y
 gme::Sprite, 346

Version.hpp
 BUILD_TYPE, 393, 396
 GIT_COMMIT_HASH, 393, 396
 GIT_TAG, 394, 396
 PROJECT_NAME, 394, 396
 PROJECT_VERSION, 394, 396
 PROJECT_VERSION_MAJOR, 394, 396
 PROJECT_VERSION_MINOR, 394, 396
 PROJECT_VERSION_PATCH, 394, 396

VERSION_MESSAGE
 argsHandler.cpp, 445, 448

volume
 ecs::Audio, 106

vx
 rnp::EntityState, 161

vy
 rnp::EntityState, 161

W
 eng, 46

w
 cli::SpriteRect::Rect, 292

WARNING
 utl, 58

WeaponSystem
 cli::WeaponSystem, 369, 370

WHITE
 game.cpp, 458
 lobby.cpp, 464
 settings.cpp, 467

WHITE_TRANS
 game.cpp, 458

WIDTH
 cli::GameConfig::Screen, 38

width
 cli::ArgsConfig, 75
 eng::WindowSize, 376

windowIsOpen
 eng::IRenderer, 215

with
 ecs::Registry::EntityBuilder, 159

WORLD_STATE
 rnp, 51

X
 eng, 46

x
 cli::SpriteRect::Rect, 292
 ecs::Scale, 315
 ecs::Transform, 364
 ecs::Velocity, 366