

## R-Type architecture

Generated by Doxygen 1.13.2



<b>1 Hierarchical Index</b>	<b>1</b>
1.1 Class Hierarchy	1
<b>2 Class Index</b>	<b>5</b>
2.1 Class List	5
<b>3 File Index</b>	<b>9</b>
3.1 File List	9
<b>4 Class Documentation</b>	<b>15</b>
4.1 ecs::AComponent Class Reference	15
4.2 ecs::AComponentArray< T > Class Template Reference	16
4.2.1 Member Function Documentation	16
4.2.1.1 getMaxEntityId()	16
4.2.1.2 removeComponents()	16
4.2.1.3 removeOneComponent()	16
4.3 ActionFactory Class Reference	17
4.4 err::AError Class Reference	17
4.4.1 Member Function Documentation	18
4.4.1.1 getCode()	18
4.4.1.2 getDetails()	18
4.4.1.3 getType()	18
4.4.1.4 what()	18
4.5 ui::AFocusableElement Class Reference	19
4.5.1 Member Function Documentation	20
4.5.1.1 canBeFocused()	20
4.5.1.2 handleInput()	21
4.5.1.3 isFocused()	21
4.5.1.4 onActivated()	21
4.5.1.5 onFocusGained()	21
4.5.1.6 onFocusLost()	21
4.5.1.7 setFocused()	21
4.6 gsm::AGameState Class Reference	22
4.6.1 Member Function Documentation	23
4.6.1.1 addSystem() [1/2]	23
4.6.1.2 addSystem() [2/2]	23
4.6.1.3 enter() [1/2]	23
4.6.1.4 enter() [2/2]	23
4.6.1.5 exit() [1/2]	23
4.6.1.6 exit() [2/2]	23
4.6.1.7 getSystems() [1/2]	24
4.6.1.8 getSystems() [2/2]	24
4.6.1.9 update() [1/2]	24

4.6.1.10 update() [2/2]	24
4.7 gsm::AGameStateMachine Class Reference	24
4.7.1 Member Function Documentation	25
4.7.1.1 changeState() [1/2]	25
4.7.1.2 changeState() [2/2]	25
4.7.1.3 popState() [1/2]	25
4.7.1.4 popState() [2/2]	25
4.7.1.5 pushState() [1/2]	26
4.7.1.6 pushState() [2/2]	26
4.7.1.7 requestStateChange()	26
4.7.1.8 requestStatePop()	26
4.7.1.9 requestStatePush()	26
4.7.1.10 update() [1/2]	26
4.7.1.11 update() [2/2]	27
4.8 ecs::AIMovementPatternComponent Class Reference	27
4.9 ecs::AIMovementSystem Class Reference	28
4.9.1 Member Function Documentation	28
4.9.1.1 update()	28
4.10 ecs::AIMoverTag Struct Reference	29
4.11 ecs::AIShooterTag Struct Reference	29
4.12 ecs::AIShootingSystem Class Reference	29
4.12.1 Member Function Documentation	30
4.12.1.1 update()	30
4.13 ecs::AnimationClip Struct Reference	30
4.14 ecs::AnimationComponent Class Reference	31
4.15 ecs::AnimationCondition Struct Reference	32
4.16 ecs::AnimationConditionFactory Class Reference	32
4.17 ecs::AnimationRenderingSystem Class Reference	33
4.17.1 Member Function Documentation	33
4.17.1.1 update()	33
4.18 APrefab Class Reference	33
4.18.1 Member Function Documentation	34
4.18.1.1 instantiate() [1/2]	34
4.18.1.2 instantiate() [2/2]	34
4.19 ecs::ASystem Class Reference	34
4.19.1 Member Function Documentation	35
4.19.1.1 updateSystem()	35
4.20 ecs::ASystemManager Class Reference	36
4.20.1 Member Function Documentation	36
4.20.1.1 addSystem()	36
4.20.1.2 removeSystem()	36
4.20.1.3 updateAllSystems()	36

4.21 ui::Background Class Reference . . . . .	37
4.21.1 Member Function Documentation . . . . .	38
4.21.1.1 render() . . . . .	38
4.21.1.2 update() . . . . .	38
4.22 ecs::BackGroundMusicTag Class Reference . . . . .	39
4.23 gsm::BootState Class Reference . . . . .	39
4.23.1 Member Function Documentation . . . . .	40
4.23.1.1 enter() [1/2] . . . . .	40
4.23.1.2 enter() [2/2] . . . . .	40
4.23.1.3 exit() . . . . .	40
4.23.1.4 update() . . . . .	40
4.24 ui::Button Class Reference . . . . .	41
4.24.1 Member Function Documentation . . . . .	43
4.24.1.1 render() . . . . .	43
4.25 math::Chrono Class Reference . . . . .	44
4.26 err::ClientError Class Reference . . . . .	44
4.26.1 Member Function Documentation . . . . .	45
4.26.1.1 getType() . . . . .	45
4.27 ClientNetwork Class Reference . . . . .	45
4.28 err::ClientNetworkError Class Reference . . . . .	48
4.28.1 Member Function Documentation . . . . .	49
4.28.1.1 getType() . . . . .	49
4.29 ecs::ColliderComponent Class Reference . . . . .	49
4.30 ecs::CollisionRule Struct Reference . . . . .	50
4.31 ecs::CollisionRules Class Reference . . . . .	50
4.32 ecs::CollisionRulesData Struct Reference . . . . .	51
4.33 ecs::CollisionRulesParser Class Reference . . . . .	51
4.34 rserv::ComponentDeltaTracker Class Reference . . . . .	52
4.35 rserv::ComponentSerializer Class Reference . . . . .	52
4.36 ComposantParser Class Reference . . . . .	53
4.37 ecs::ControllableTag Class Reference . . . . .	54
4.38 Core Class Reference . . . . .	54
4.39 ecs::DamageComponent Class Reference . . . . .	55
4.40 ecs::DamageIntentComponent Class Reference . . . . .	56
4.41 ecs::DeathIntentComponent Class Reference . . . . .	56
4.42 ecs::DeathSystem Class Reference . . . . .	57
4.42.1 Member Function Documentation . . . . .	57
4.42.1.1 update() . . . . .	57
4.43 debug::Debug Class Reference . . . . .	58
4.44 gsm::DevState Class Reference . . . . .	58
4.44.1 Member Function Documentation . . . . .	59
4.44.1.1 enter() . . . . .	59

4.44.1.2 exit()	59
4.44.1.3 update()	59
4.45 DLoader< T > Class Template Reference	60
4.45.1 Member Function Documentation	60
4.45.1.1 Close()	60
4.45.1.2 Error()	60
4.45.1.3 getHandler()	60
4.45.1.4 Open()	61
4.45.1.5 Symbol()	61
4.46 ecs::EnnemyProjectileTag Class Reference	61
4.47 ecs::EntityCreationContext Struct Reference	61
4.48 ecs::EntityFactory Class Reference	62
4.48.1 Member Function Documentation	63
4.48.1.1 createEntity()	63
4.48.1.2 getNextNetworkId()	63
4.48.1.3 setNextNetworkId()	63
4.49 EntityParser Class Reference	63
4.50 EntityPrefabManager Class Reference	64
4.51 rserv::EntitySnapshot Struct Reference	64
4.52 Field Struct Reference	65
4.53 FieldValue Struct Reference	65
4.54 math::FRect Class Reference	65
4.55 gsm::GameEndState Class Reference	66
4.55.1 Member Function Documentation	67
4.55.1.1 enter()	67
4.56 gsm::GameStateMachine Class Reference	67
4.56.1 Member Function Documentation	68
4.56.1.1 requestStateChange()	68
4.56.1.2 requestStatePop()	68
4.56.1.3 requestStatePush()	68
4.57 ecs::GameZoneColliderTag Class Reference	68
4.58 ecs::GameZoneComponent Class Reference	69
4.59 ecs::GameZoneRenderingSystem Class Reference	69
4.59.1 Member Function Documentation	70
4.59.1.1 update()	70
4.60 ecs::GameZoneViewSystem Class Reference	70
4.60.1 Member Function Documentation	71
4.60.1.1 update()	71
4.61 ecs::GraphicalInputProvider Class Reference	71
4.61.1 Member Function Documentation	72
4.61.1.1 getActionAxis()	72
4.61.1.2 getAxisValue()	72

4.61.1.3 getInputMapping()	72
4.61.1.4 isActionPressed()	72
4.62 ecs::Group< Components > Class Template Reference	73
4.63 ecs::HealthBarComponent Class Reference	73
4.64 ecs::HealthBarRenderingSystem Class Reference	73
4.64.1 Member Function Documentation	74
4.64.1.1 update()	74
4.65 ecs::HealthComponent Class Reference	74
4.66 ecs::HealthSystem Class Reference	75
4.66.1 Member Function Documentation	75
4.66.1.1 update()	75
4.67 ecs::HitboxRenderComponent Class Reference	76
4.68 ecs::HitboxRenderingSystem Class Reference	76
4.68.1 Member Function Documentation	77
4.68.1.1 update()	77
4.69 ecs::IComponent Class Reference	77
4.70 ecs::IComponentArray Class Reference	78
4.71 ecs::IEntityFactory Class Reference	78
4.72 err::IError Class Reference	79
4.73 ui::IFocusable Class Reference	79
4.74 gsm::IGameState Class Reference	80
4.75 gsm::IGameStateMachine Class Reference	80
4.76 ecs::IInputProvider Class Reference	81
4.77 ILoader Class Reference	81
4.78 gsm::InGameState Class Reference	82
4.78.1 Member Function Documentation	83
4.78.1.1 enter() [1/2]	83
4.78.1.2 enter() [2/2]	83
4.78.1.3 exit()	83
4.78.1.4 update()	84
4.79 ecs::InputIntentComponent Class Reference	84
4.80 ecs::InputMapping Struct Reference	85
4.81 ecs::InputMappingManager Class Reference	85
4.82 ecs::InputNormalizer Class Reference	86
4.83 ecs::InputToVelocitySystem Class Reference	86
4.83.1 Member Function Documentation	86
4.83.1.1 update()	86
4.84 ecs::InteractionConfigComponent Class Reference	87
4.85 ecs::InteractionMapping Struct Reference	87
4.86 ecs::InteractionSystem Class Reference	88
4.86.1 Member Function Documentation	88
4.86.1.1 update()	88

4.87 IPrefab Class Reference . . . . .	88
4.88 rserv::IServer Class Reference . . . . .	89
4.89 ecs::ISystem Class Reference . . . . .	90
4.90 ecs::ISystemManager Class Reference . . . . .	90
4.91 ecs::View< Components >::Iterator Class Reference . . . . .	91
4.92 ui::Background::Layer Struct Reference . . . . .	91
4.93 ui::LayoutConfig Struct Reference . . . . .	92
4.94 err::LibairiesLoadError Class Reference . . . . .	92
4.94.1 Member Function Documentation . . . . .	93
4.94.1.1 getType() . . . . .	93
4.95 ecs::LifetimeComponent Class Reference . . . . .	93
4.96 ecs::LifetimeSystem Class Reference . . . . .	94
4.96.1 Member Function Documentation . . . . .	94
4.96.1.1 update() . . . . .	94
4.97 gsm::LoadingState Class Reference . . . . .	94
4.97.1 Member Function Documentation . . . . .	95
4.97.1.1 enter() [1/2] . . . . .	95
4.97.1.2 enter() [2/2] . . . . .	95
4.97.1.3 exit() . . . . .	96
4.97.1.4 update() . . . . .	96
4.98 gsm::LobbyBrowserState Class Reference . . . . .	96
4.98.1 Member Function Documentation . . . . .	97
4.98.1.1 enter() . . . . .	97
4.98.1.2 exit() . . . . .	97
4.98.1.3 update() . . . . .	97
4.99 gsm::LobbyState Class Reference . . . . .	98
4.99.1 Member Function Documentation . . . . .	99
4.99.1.1 enter() [1/2] . . . . .	99
4.99.1.2 enter() [2/2] . . . . .	99
4.99.1.3 exit() . . . . .	99
4.99.1.4 update() [1/2] . . . . .	99
4.99.1.5 update() [2/2] . . . . .	99
4.100 ecs::LocalPlayerTag Class Reference . . . . .	100
4.101 gsm::MainMenuState Class Reference . . . . .	100
4.101.1 Member Function Documentation . . . . .	101
4.101.1.1 enter() . . . . .	101
4.101.1.2 exit() . . . . .	101
4.101.1.3 update() . . . . .	102
4.102 MapParser Class Reference . . . . .	102
4.103 ecs::MobTag Class Reference . . . . .	103
4.104 MouseClickInfo Struct Reference . . . . .	103
4.105 MouseInputHandler Class Reference . . . . .	103



4.106	<a href="#">ecs::MovementInputSystem Class Reference</a>	104
4.106.1	<a href="#">Member Function Documentation</a>	104
4.106.1.1	<a href="#">update()</a>	104
4.107	<a href="#">ecs::MovementIntentComponent Class Reference</a>	105
4.108	<a href="#">ecs::MovementSystem Class Reference</a>	105
4.108.1	<a href="#">Member Function Documentation</a>	106
4.108.1.1	<a href="#">update()</a>	106
4.109	<a href="#">ecs::MultiShotPattern Struct Reference</a>	106
4.110	<a href="#">ecs::MusicComponent Class Reference</a>	107
4.111	<a href="#">ecs::MusicIntentComponent Class Reference</a>	108
4.112	<a href="#">ecs::MusicSystem Class Reference</a>	108
4.112.1	<a href="#">Member Function Documentation</a>	109
4.112.1.1	<a href="#">update()</a>	109
4.113	<a href="#">NetworkEvent Struct Reference</a>	109
4.114	<a href="#">ecs::NetworkHealthState Struct Reference</a>	109
4.115	<a href="#">ecs::NetworkIdComponent Class Reference</a>	110
4.116	<a href="#">ecs::NetworkInterpolationSystem Class Reference</a>	110
4.116.1	<a href="#">Member Function Documentation</a>	111
4.116.1.1	<a href="#">update()</a>	111
4.117	<a href="#">ecs::NetworkStateComponent Class Reference</a>	111
4.118	<a href="#">ecs::NetworkTransformState Struct Reference</a>	112
4.119	<a href="#">ecs::ObstacleTag Class Reference</a>	112
4.120	<a href="#">ecs::OutOfBoundsSystem Class Reference</a>	113
4.120.1	<a href="#">Member Function Documentation</a>	113
4.120.1.1	<a href="#">update()</a>	113
4.121	<a href="#">ecs::OwnerComponent Class Reference</a>	114
4.122	<a href="#">err::PacketError Class Reference</a>	114
4.122.1	<a href="#">Member Function Documentation</a>	115
4.122.1.1	<a href="#">getType()</a>	115
4.123	<a href="#">ecs::ParallaxComponent Class Reference</a>	115
4.124	<a href="#">ecs::ParallaxLayer Struct Reference</a>	116
4.125	<a href="#">ecs::ParallaxRenderingSystem Class Reference</a>	117
4.125.1	<a href="#">Member Function Documentation</a>	117
4.125.1.1	<a href="#">update()</a>	117
4.126	<a href="#">ParsedEntityPrefab Class Reference</a>	118
4.126.1	<a href="#">Member Function Documentation</a>	118
4.126.1.1	<a href="#">instantiate() [1/2]</a>	118
4.126.1.2	<a href="#">instantiate() [2/2]</a>	118
4.127	<a href="#">Parser Class Reference</a>	119
4.128	<a href="#">err::ParserError Class Reference</a>	119
4.128.1	<a href="#">Member Function Documentation</a>	120
4.128.1.1	<a href="#">getType()</a>	120

4.129 gsm::PauseState Class Reference	120
4.129.1 Member Function Documentation	121
4.129.1.1 enter()	121
4.129.1.2 exit()	121
4.129.1.3 update()	122
4.130 ecs::PlayerProjectileTag Class Reference	122
4.131 ecs::PlayerTag Class Reference	122
4.132 ecs::ProjectilePassThroughTag Class Reference	123
4.133 ecs::ProjectilePrefabComponent Class Reference	123
4.134 ecs::RectangleRenderComponent Class Reference	124
4.135 ecs::RectangleRenderingSystem Class Reference	124
4.135.1 Member Function Documentation	125
4.135.1.1 update()	125
4.136 ecs::Registry Class Reference	125
4.137 ecs::RemappableKeyBinding Struct Reference	126
4.138 ResourceManager Class Reference	126
4.139 gsm::ResultsState Class Reference	127
4.139.1 Member Function Documentation	127
4.139.1.1 enter()	127
4.139.1.2 exit()	128
4.139.1.3 update()	128
4.140 ecs::ScoreComponent Class Reference	128
4.141 gsm::ScoreFeedback Struct Reference	129
4.142 ScoreIntentComponent Class Reference	129
4.143 ecs::ScoreSystem Class Reference	129
4.143.1 Member Function Documentation	130
4.143.1.1 update()	130
4.144 ecs::ScoreValueComponent Class Reference	130
4.145 rserv::Server Class Reference	131
4.145.1 Member Function Documentation	133
4.145.1.1 getClientCount()	133
4.145.1.2 getConfig()	133
4.145.1.3 getConnectedClientEndpoints()	133
4.145.1.4 getConnectedClients()	133
4.145.1.5 getEventQueue()	133
4.145.1.6 getNetwork()	134
4.145.1.7 getPort()	134
4.145.1.8 getState()	134
4.145.1.9 hasEvents()	134
4.145.1.10 init()	134
4.145.1.11 onClientConnected()	134
4.145.1.12 onClientDisconnected()	134

4.145.1.13 onPacketReceived()	135
4.145.1.14 operator int()	135
4.145.1.15 processConnections()	135
4.145.1.16 processDisconnections()	135
4.145.1.17 processEndOfGame()	135
4.145.1.18 processEvents()	135
4.145.1.19 processIncomingPackets()	135
4.145.1.20 setConfig()	136
4.145.1.21 setNetwork()	136
4.145.1.22 setPort()	136
4.145.1.23 setState()	136
4.145.1.24 start()	136
4.145.1.25 stop()	136
4.146 rserv::ServerConfig Class Reference	137
4.147 err::ServerError Class Reference	137
4.147.1 Member Function Documentation	138
4.147.1.1 getType()	138
4.148 ecs::ServerInputProvider Class Reference	138
4.148.1 Member Function Documentation	139
4.148.1.1 getActionAxis()	139
4.148.1.2 getAxisValue()	140
4.148.1.3 getInputMapping()	140
4.148.1.4 isActionPressed()	140
4.149 ecs::ServerMovementInputSystem Class Reference	140
4.149.1 Member Function Documentation	141
4.149.1.1 update()	141
4.150 ecs::ServerShootInputSystem Class Reference	141
4.150.1 Member Function Documentation	142
4.150.1.1 update()	142
4.151 SettingsConfig Class Reference	142
4.152 SettingsManager Class Reference	143
4.153 gsm::SettingsState Class Reference	144
4.153.1 Member Function Documentation	146
4.153.1.1 enter()	146
4.153.1.2 exit()	146
4.153.1.3 update()	147
4.154 ecs::ShooterTag Class Reference	147
4.155 ecs::ShootingStatsComponent Class Reference	147
4.156 ecs::ShootingSystem Class Reference	148
4.156.1 Member Function Documentation	149
4.156.1.1 update()	149
4.157 ecs::ShootInputSystem Class Reference	149

4.157.1 Member Function Documentation	150
4.157.1.1 update()	150
4.158 ecs::ShootIntentComponent Class Reference	150
4.159 gsm::ShutdownState Class Reference	151
4.159.1 Member Function Documentation	152
4.159.1.1 enter()	152
4.160 Signal Class Reference	152
4.161 ui::Slider Class Reference	152
4.161.1 Member Function Documentation	155
4.161.1.1 handleInput()	155
4.161.1.2 onActivated()	156
4.161.1.3 onNavigateLeft()	156
4.161.1.4 onNavigateRight()	156
4.161.1.5 render()	156
4.162 ecs::SoundIntentComponent Class Reference	156
4.163 ecs::SoundSystem Class Reference	157
4.163.1 Member Function Documentation	157
4.163.1.1 update()	157
4.164 ecs::SpatialGrid Class Reference	158
4.165 ecs::SpawnIntentComponent Class Reference	158
4.166 ecs::SpawnSystem Class Reference	159
4.166.1 Member Function Documentation	160
4.166.1.1 update()	160
4.167 ecs::SpeedComponent Class Reference	160
4.168 ecs::SpriteComponent Class Reference	161
4.169 ecs::SpriteRenderingSystem Class Reference	161
4.169.1 Member Function Documentation	162
4.169.1.1 update()	162
4.170 ecs::SystemManager Class Reference	162
4.171 TagRegistry Class Reference	163
4.172 ui::Text Class Reference	163
4.172.1 Member Function Documentation	165
4.172.1.1 render()	165
4.172.1.2 setScale()	165
4.172.1.3 update()	165
4.173 ecs::TextComponent Class Reference	166
4.174 ui::TextInput Class Reference	166
4.174.1 Member Function Documentation	169
4.174.1.1 handleInput()	169
4.174.1.2 render()	169
4.174.1.3 update()	169
4.175 ecs::TextRenderingSystem Class Reference	170

4.175.1 Member Function Documentation	170
4.175.1.1 update()	170
4.176 ui::ToggleSwitch Class Reference	171
4.176.1 Member Function Documentation	173
4.176.1.1 containsPoint()	173
4.176.1.2 handleInput()	173
4.176.1.3 render()	174
4.177 ecs::TransformComponent Class Reference	174
4.178 ecs::Transition Struct Reference	175
4.179 ecs::TriggerIntentComponent Class Reference	175
4.180 ecs::TriggerSystem Class Reference	176
4.180.1 Member Function Documentation	176
4.180.1.1 update()	176
4.181 ui::UIElement Class Reference	177
4.182 ui::UILayout Class Reference	178
4.182.1 Member Function Documentation	180
4.182.1.1 render()	180
4.182.1.2 setScale()	180
4.182.1.3 update()	180
4.183 ui::UIManager Class Reference	181
4.184 ui::UINavigationController Class Reference	182
4.185 Utils Class Reference	182
4.186 math::Vector2f Class Reference	183
4.187 ecs::VelocityComponent Class Reference	183
4.188 ecs::View< Components > Class Template Reference	184
<b>5 File Documentation</b>	<b>185</b>
5.1 ClientNetwork.hpp	185
5.2 colors.hpp	187
5.3 AnimationComponent.hpp	188
5.4 HealthBarComponent.hpp	190
5.5 HitboxRenderComponent.hpp	190
5.6 MusicComponent.hpp	191
5.7 ParallaxComponent.hpp	191
5.8 RectangleRenderComponent.hpp	193
5.9 SpriteComponent.hpp	193
5.10 TextComponent.hpp	194
5.11 BackGroundMusicTag.hpp	194
5.12 MusicIntentComponent.hpp	195
5.13 SoundIntentComponent.hpp	195
5.14 constants.hpp	196
5.15 constants.hpp	197

5.16 Core.hpp	200
5.17 Core.hpp	200
5.18 AGameStateMachine.hpp	201
5.19 AGameStateMachine.hpp	202
5.20 GameStateMachine.hpp	202
5.21 GameStateMachine.hpp	202
5.22 AGameState.hpp	202
5.23 AGameState.hpp	203
5.24 BootState.hpp	203
5.25 BootState.hpp	204
5.26 DevState.hpp	204
5.27 InGameState.hpp	205
5.28 InGameState.hpp	206
5.29 LoadingState.hpp	206
5.30 LoadingState.hpp	206
5.31 LobbyState.hpp	207
5.32 LobbyState.hpp	207
5.33 LobbyBrowserState.hpp	208
5.34 MainMenuState.hpp	208
5.35 PauseState.hpp	209
5.36 ResultsState.hpp	209
5.37 SettingsState.hpp	210
5.38 GraphicalInputProvider.hpp	211
5.39 initResourcesManager.hpp	212
5.40 initResourcesManager.hpp	212
5.41 MouseInputHandler.hpp	213
5.42 NetworkStateComponent.hpp	213
5.43 DefaultPacketHandlers.hpp	214
5.44 DefaultPacketHandlers.hpp	215
5.45 DefaultPacketHandlers.hpp	215
5.46 SettingsConfig.hpp	215
5.47 SettingsManager.hpp	216
5.48 MusicSystem.hpp	217
5.49 SoundSystem.hpp	217
5.50 MovementInputSystem.hpp	218
5.51 ShootInputSystem.hpp	218
5.52 NetworkInterpolationSystem.hpp	219
5.53 AnimationRenderingSystem.hpp	219
5.54 GameZoneRenderingSystem.hpp	219
5.55 GameZoneViewSystem.hpp	220
5.56 HealthBarRenderingSystem.hpp	220
5.57 HitboxRenderingSystem.hpp	221

5.58 ParallaxRenderingSystem.hpp . . . . .	221
5.59 RectangleRenderingSystem.hpp . . . . .	222
5.60 SpriteRenderingSystem.hpp . . . . .	222
5.61 TextRenderingSystem.hpp . . . . .	222
5.62 AFocusableElement.hpp . . . . .	223
5.63 IFocusable.hpp . . . . .	224
5.64 UILayout.hpp . . . . .	224
5.65 Background.hpp . . . . .	225
5.66 UIElement.hpp . . . . .	226
5.67 Button.hpp . . . . .	227
5.68 Slider.hpp . . . . .	228
5.69 TextInput.hpp . . . . .	229
5.70 ToggleSwitch.hpp . . . . .	230
5.71 Text.hpp . . . . .	231
5.72 UIManager.hpp . . . . .	231
5.73 UINavigationManager.hpp . . . . .	232
5.74 Utils.hpp . . . . .	233
5.75 Utils.hpp . . . . .	234
5.76 CollisionRules.hpp . . . . .	234
5.77 CollisionRulesData.hpp . . . . .	235
5.78 AComponent.hpp . . . . .	235
5.79 IComponent.hpp . . . . .	235
5.80 AIMovementPatternComponent.hpp . . . . .	236
5.81 ColliderComponent.hpp . . . . .	237
5.82 DamageComponent.hpp . . . . .	237
5.83 GameZoneComponent.hpp . . . . .	238
5.84 HealthComponent.hpp . . . . .	238
5.85 InteractionConfigComponent.hpp . . . . .	239
5.86 LifetimeComponent.hpp . . . . .	239
5.87 NetworkIdComponent.hpp . . . . .	240
5.88 OwnerComponent.hpp . . . . .	240
5.89 ProjectilePrefabComponent.hpp . . . . .	241
5.90 ScoreComponent.hpp . . . . .	241
5.91 ScoreValueComponent.hpp . . . . .	241
5.92 ShootingStatsComponent.hpp . . . . .	242
5.93 SpeedComponent.hpp . . . . .	243
5.94 TransformComponent.hpp . . . . .	243
5.95 VelocityComponent.hpp . . . . .	244
5.96 AIMoverTag.hpp . . . . .	244
5.97 AShooterTag.hpp . . . . .	244
5.98 ControllableTag.hpp . . . . .	245
5.99 EnemyProjectileTag.hpp . . . . .	245

5.100 GameZoneColliderTag.hpp . . . . .	245
5.101 LocalPlayerTag.hpp . . . . .	246
5.102 MobTag.hpp . . . . .	246
5.103 ObstacleTag.hpp . . . . .	246
5.104 PlayerProjectileTag.hpp . . . . .	247
5.105 PlayerTag.hpp . . . . .	247
5.106 ProjectilePassThroughTag.hpp . . . . .	248
5.107 ShooterTag.hpp . . . . .	248
5.108 DamageIntentComponent.hpp . . . . .	248
5.109 DeathIntentComponent.hpp . . . . .	249
5.110 InputIntentComponent.hpp . . . . .	249
5.111 MovementIntentComponent.hpp . . . . .	250
5.112 ScoreIntentComponent.hpp . . . . .	250
5.113 ShootIntentComponent.hpp . . . . .	250
5.114 SpawnIntentComponent.hpp . . . . .	251
5.115 TriggerIntentComponent.hpp . . . . .	252
5.116 debug.hpp . . . . .	252
5.117 DLLoader.hpp . . . . .	253
5.118 ILoader.hpp . . . . .	254
5.119 LoaderType.hpp . . . . .	255
5.120 AComponentArray.hpp . . . . .	255
5.121 IComponentArray.hpp . . . . .	256
5.122 Entity.hpp . . . . .	256
5.123 EntityCreationContext.hpp . . . . .	257
5.124 EntityFactory.hpp . . . . .	257
5.125 IEntityFactory.hpp . . . . .	258
5.126 Registry.hpp . . . . .	258
5.127 View.hpp . . . . .	259
5.128 AError.hpp . . . . .	260
5.129 ClientError.hpp . . . . .	260
5.130 ClientNetworkError.hpp . . . . .	261
5.131 IError.hpp . . . . .	261
5.132 LibrariesLoadError.hpp . . . . .	262
5.133 PacketError.hpp . . . . .	262
5.134 ParserError.hpp . . . . .	263
5.135 ServerError.hpp . . . . .	263
5.136 IGameState.hpp . . . . .	263
5.137 IGameStateMachine.hpp . . . . .	264
5.138 IInputProvider.hpp . . . . .	264
5.139 InputAction.hpp . . . . .	265
5.140 InputMapping.hpp . . . . .	265
5.141 InputMappingManager.hpp . . . . .	266



5.142 GameStateHandlers.hpp	267
5.143 AnimationConditionFactory.hpp	267
5.144 CollisionRulesParser.hpp	268
5.145 ComposantParser.hpp	268
5.146 EntityParser.hpp	268
5.147 MapParser.hpp	269
5.148 Parser.hpp	270
5.149 ParserParam.hpp	271
5.150 APrefab.hpp	272
5.151 EntityPrefabManager.hpp	272
5.152 IPrefab.hpp	273
5.153 ParsedEntityPrefab.hpp	273
5.154 ResourceManager.hpp	274
5.155 Signal.hpp	274
5.156 SpatialGrid.hpp	275
5.157 AIMovementSystem.hpp	276
5.158 AIShootingSystem.hpp	277
5.159 ASystem.hpp	277
5.160 ISystem.hpp	277
5.161 OutOfBoundsSystem.hpp	278
5.162 DeathSystem.hpp	278
5.163 HealthSystem.hpp	279
5.164 InputNormalizer.hpp	279
5.165 ActionFactory.hpp	280
5.166 InteractionSystem.hpp	280
5.167 TagRegistry.hpp	281
5.168 TriggerSystem.hpp	281
5.169 LifetimeSystem.hpp	282
5.170 InputToVelocitySystem.hpp	283
5.171 MovementSystem.hpp	283
5.172 ScoreSystem.hpp	284
5.173 ShootingSystem.hpp	284
5.174 SpawnSystem.hpp	285
5.175 ASystemManager.hpp	286
5.176 ISystemManager.hpp	286
5.177 SystemManager.hpp	286
5.178 translationToECS.hpp	287
5.179 Chrono.hpp	287
5.180 FRect.hpp	288
5.181 Vector2f.hpp	288
5.182 Constants.hpp	289
5.183 ComponentDeltaTracker.hpp	289

5.184 ComponentSerializer.hpp . . . . .	290
5.185 GameEndState.hpp . . . . .	291
5.186 ShutdownState.hpp . . . . .	291
5.187 ServerInputProvider.hpp . . . . .	292
5.188 IServer.hpp . . . . .	292
5.189 Server.hpp . . . . .	293
5.190 ServerConfig.hpp . . . . .	295
5.191 ServerMovementInputSystem.hpp . . . . .	296
5.192 ServerShootInputSystem.hpp . . . . .	296
<b>Index</b>	<b>299</b>

# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

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

ActionFactory . . . . .	17
ecs::AnimationClip . . . . .	30
ecs::AnimationCondition . . . . .	32
ecs::AnimationConditionFactory . . . . .	32
math::Chrono . . . . .	44
ClientNetwork . . . . .	45
ecs::CollisionRule . . . . .	50
ecs::CollisionRules . . . . .	50
ecs::CollisionRulesData . . . . .	51
ecs::CollisionRulesParser . . . . .	51
rserve::ComponentDeltaTracker . . . . .	52
rserve::ComponentSerializer . . . . .	52
ComposantParser . . . . .	53
Core . . . . .	54
debug::Debug . . . . .	58
std::enable_shared_from_this	
ecs::Registry . . . . .	125
ui::UIElement . . . . .	177
ui::AFocusableElement . . . . .	19
ui::Button . . . . .	41
ui::Slider . . . . .	152
ui::TextInput . . . . .	166
ui::ToggleSwitch . . . . .	171
ui::Background . . . . .	37
ui::Text . . . . .	163
ui::UILayout . . . . .	178
ecs::EntityCreationContext . . . . .	61
EntityParser . . . . .	63
EntityPrefabManager . . . . .	64
rserve::EntitySnapshot . . . . .	64
std::exception	
err::IError . . . . .	79
err::AError . . . . .	17
err::ClientError . . . . .	44
err::ClientNetworkError . . . . .	48

err::LibrairiesLoadError . . . . .	92
err::PacketError . . . . .	114
err::ParserError . . . . .	119
err::ServerError . . . . .	137
Field . . . . .	65
FieldValueVariant	
FieldValue . . . . .	65
math::FRect . . . . .	65
ecs::Group< Components > . . . . .	73
ecs::IComponent . . . . .	77
ecs::AComponent . . . . .	15
ecs::AIMovementPatternComponent . . . . .	27
ecs::AIMoverTag . . . . .	29
ecs::AIShooterTag . . . . .	29
ecs::AnimationComponent . . . . .	31
ecs::BackGroundMusicTag . . . . .	39
ecs::ColliderComponent . . . . .	49
ecs::ControllableTag . . . . .	54
ecs::DamageComponent . . . . .	55
ecs::DamageIntentComponent . . . . .	56
ecs::DeathIntentComponent . . . . .	56
ecs::EnemyProjectileTag . . . . .	61
ecs::GameZoneColliderTag . . . . .	68
ecs::GameZoneComponent . . . . .	69
ecs::HealthBarComponent . . . . .	73
ecs::HealthComponent . . . . .	74
ecs::HitboxRenderComponent . . . . .	76
ecs::InputIntentComponent . . . . .	84
ecs::InteractionConfigComponent . . . . .	87
ecs::LifetimeComponent . . . . .	93
ecs::LocalPlayerTag . . . . .	100
ecs::MobTag . . . . .	103
ecs::MovementIntentComponent . . . . .	105
ecs::MusicComponent . . . . .	107
ecs::MusicIntentComponent . . . . .	108
ecs::NetworkIdComponent . . . . .	110
ecs::NetworkStateComponent . . . . .	111
ecs::ObstacleTag . . . . .	112
ecs::OwnerComponent . . . . .	114
ecs::ParallaxComponent . . . . .	115
ecs::PlayerProjectileTag . . . . .	122
ecs::PlayerTag . . . . .	122
ecs::ProjectilePassThroughTag . . . . .	123
ecs::ProjectilePrefabComponent . . . . .	123
ecs::RectangleRenderComponent . . . . .	124
ecs::ScoreComponent . . . . .	128
ecs::ScoreValueComponent . . . . .	130
ecs::ShootIntentComponent . . . . .	150
ecs::ShooterTag . . . . .	147
ecs::ShootingStatsComponent . . . . .	147
ecs::SoundIntentComponent . . . . .	156
ecs::SpawnIntentComponent . . . . .	158
ecs::SpeedComponent . . . . .	160
ecs::SpriteComponent . . . . .	161
ecs::TextComponent . . . . .	166
ecs::TransformComponent . . . . .	174
ecs::TriggerIntentComponent . . . . .	175
ecs::VelocityComponent . . . . .	183

ecs::IComponentArray . . . . .	78
ecs::AComponentArray< T > . . . . .	16
ecs::IEntityFactory . . . . .	78
ecs::EntityFactory . . . . .	62
ui::IFocusable . . . . .	79
ui::AFocusableElement . . . . .	19
gsm::IGameState . . . . .	80
gsm::AGameState . . . . .	22
gsm::BootState . . . . .	39
gsm::BootState . . . . .	39
gsm::DevState . . . . .	58
gsm::GameEndState . . . . .	66
gsm::InGameState . . . . .	82
gsm::InGameState . . . . .	82
gsm::LoadingState . . . . .	94
gsm::LoadingState . . . . .	94
gsm::LobbyBrowserState . . . . .	96
gsm::LobbyState . . . . .	98
gsm::LobbyState . . . . .	98
gsm::MainMenuState . . . . .	100
gsm::PauseState . . . . .	120
gsm::ResultsState . . . . .	127
gsm::SettingsState . . . . .	144
gsm::ShutdownState . . . . .	151
gsm::AGameState . . . . .	22
gsm::IGameStateMachine . . . . .	80
gsm::AGameStateMachine . . . . .	24
gsm::GameStateMachine . . . . .	67
gsm::GameStateMachine . . . . .	67
gsm::AGameStateMachine . . . . .	24
ecs::IInputProvider . . . . .	81
ecs::GraphicalInputProvider . . . . .	71
ecs::ServerInputProvider . . . . .	138
ILoader . . . . .	81
DLLoader< createNetworkLib_t > . . . . .	60
DLLoader< createBuffer_t > . . . . .	60
DLLoader< createPacket_t > . . . . .	60
DLLoader< T > . . . . .	60
ecs::InputMapping . . . . .	85
ecs::InputMappingManager . . . . .	85
ecs::InputNormalizer . . . . .	86
ecs::InteractionMapping . . . . .	87
IPrefab . . . . .	88
APrefab . . . . .	33
ParsedEntityPrefab . . . . .	118
rserv::IServer . . . . .	89
rserv::Server . . . . .	131
ecs::ISystem . . . . .	90
ecs::ASystem . . . . .	34
ecs::AIMovementSystem . . . . .	28
ecs::AIShootingSystem . . . . .	29
ecs::AnimationRenderingSystem . . . . .	33
ecs::DeathSystem . . . . .	57
ecs::GameZoneRenderingSystem . . . . .	69
ecs::GameZoneViewSystem . . . . .	70

ecs::HealthBarRenderingSystem . . . . .	73
ecs::HealthSystem . . . . .	75
ecs::HitboxRenderingSystem . . . . .	76
ecs::InputToVelocitySystem . . . . .	86
ecs::InteractionSystem . . . . .	88
ecs::LifetimeSystem . . . . .	94
ecs::MovementInputSystem . . . . .	104
ecs::MovementSystem . . . . .	105
ecs::MusicSystem . . . . .	108
ecs::NetworkInterpolationSystem . . . . .	110
ecs::OutOfBoundsSystem . . . . .	113
ecs::ParallaxRenderingSystem . . . . .	117
ecs::RectangleRenderingSystem . . . . .	124
ecs::ScoreSystem . . . . .	129
ecs::ServerMovementInputSystem . . . . .	140
ecs::ServerShootInputSystem . . . . .	141
ecs::ShootInputSystem . . . . .	149
ecs::ShootingSystem . . . . .	148
ecs::SoundSystem . . . . .	157
ecs::SpawnSystem . . . . .	159
ecs::SpriteRenderingSystem . . . . .	161
ecs::TextRenderingSystem . . . . .	170
ecs::TriggerSystem . . . . .	176
ecs::ISystemManager . . . . .	90
ecs::ASystemManager . . . . .	36
ecs::SystemManager . . . . .	162
ecs::View< Components >::Iterator . . . . .	91
ui::Background::Layer . . . . .	91
ui::LayoutConfig . . . . .	92
MapParser . . . . .	102
MouseClickedInfo . . . . .	103
MouseInputHandler . . . . .	103
ecs::MultiShotPattern . . . . .	106
NetworkEvent . . . . .	109
ecs::NetworkHealthState . . . . .	109
ecs::NetworkTransformState . . . . .	112
ecs::ParallaxLayer . . . . .	116
Parser . . . . .	119
ecs::RemappableKeyBinding . . . . .	126
ResourceManager . . . . .	126
gsm::ScoreFeedback . . . . .	129
ScoreIntentComponent . . . . .	129
rserve::ServerConfig . . . . .	137
SettingsConfig . . . . .	142
SettingsManager . . . . .	143
Signal . . . . .	152
ecs::SpatialGrid . . . . .	158
TagRegistry . . . . .	163
ecs::Transition . . . . .	175
ui::UIManager . . . . .	181
ui::UINavigationManager . . . . .	182
Utils . . . . .	182
math::Vector2f . . . . .	183
ecs::View< Components > . . . . .	184

## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">ecs::AComponent</a>	15
<a href="#">ecs::AComponentArray&lt; T &gt;</a>	16
<a href="#">ActionFactory</a>	17
<a href="#">err::AError</a>	17
<a href="#">ui::AFocusableElement</a>	19
<a href="#">gsm::AGameState</a>	22
<a href="#">gsm::AGameStateMachine</a>	24
<a href="#">ecs::AIMovementPatternComponent</a>	27
<a href="#">ecs::AIMovementSystem</a>	28
<a href="#">ecs::AIMoverTag</a>	29
<a href="#">ecs::AIShooterTag</a>	29
<a href="#">ecs::AIShootingSystem</a>	29
<a href="#">ecs::AnimationClip</a>	30
<a href="#">ecs::AnimationComponent</a>	31
<a href="#">ecs::AnimationCondition</a>	32
<a href="#">ecs::AnimationConditionFactory</a>	32
<a href="#">ecs::AnimationRenderingSystem</a>	33
<a href="#">APrefab</a>	33
<a href="#">ecs::ASystem</a>	34
<a href="#">ecs::ASystemManager</a>	36
<a href="#">ui::Background</a>	37
<a href="#">ecs::BackGroundMusicTag</a>	39
<a href="#">gsm::BootState</a>	39
<a href="#">ui::Button</a>	41
<a href="#">math::Chrono</a>	44
<a href="#">err::ClientError</a>	44
<a href="#">ClientNetwork</a>	45
<a href="#">err::ClientNetworkError</a>	48
<a href="#">ecs::ColliderComponent</a>	49
<a href="#">ecs::CollisionRule</a>	50
<a href="#">ecs::CollisionRules</a>	50
<a href="#">ecs::CollisionRulesData</a>	51
<a href="#">ecs::CollisionRulesParser</a>	51
<a href="#">rserv::ComponentDeltaTracker</a>	52
<a href="#">rserv::ComponentSerializer</a>	52

<a href="#">ComposantParser</a>	53
<a href="#">ecs::ControllableTag</a>	54
<a href="#">Core</a>	54
<a href="#">ecs::DamageComponent</a>	55
<a href="#">ecs::DamageIntentComponent</a>	56
<a href="#">ecs::DeathIntentComponent</a>	56
<a href="#">ecs::DeathSystem</a>	57
<a href="#">debug::Debug</a>	58
<a href="#">gsm::DevState</a>	58
<a href="#">DLLoader&lt; T &gt;</a>	60
<a href="#">ecs::EnnemyProjectileTag</a>	61
<a href="#">ecs::EntityCreationContext</a>	61
<a href="#">ecs::EntityFactory</a>	62
<a href="#">EntityParser</a>	63
<a href="#">EntityPrefabManager</a>	64
<a href="#">rserv::EntitySnapshot</a>	64
<a href="#">Field</a>	65
<a href="#">FieldValue</a>	65
<a href="#">math::FRect</a>	65
<a href="#">gsm::GameEndState</a>	66
<a href="#">gsm::GameStateMachine</a>	67
<a href="#">ecs::GameZoneColliderTag</a>	68
<a href="#">ecs::GameZoneComponent</a>	69
<a href="#">ecs::GameZoneRenderingSystem</a>	69
<a href="#">ecs::GameZoneViewSystem</a>	70
<a href="#">ecs::GraphicalInputProvider</a>	71
<a href="#">ecs::Group&lt; Components &gt;</a>	73
<a href="#">ecs::HealthBarComponent</a>	73
<a href="#">ecs::HealthBarRenderingSystem</a>	73
<a href="#">ecs::HealthComponent</a>	74
<a href="#">ecs::HealthSystem</a>	75
<a href="#">ecs::HitboxRenderComponent</a>	76
<a href="#">ecs::HitboxRenderingSystem</a>	76
<a href="#">ecs::IComponent</a>	77
<a href="#">ecs::IComponentArray</a>	78
<a href="#">ecs::IEntityFactory</a>	78
<a href="#">err::IError</a>	79
<a href="#">ui::IFocusable</a>	79
<a href="#">gsm::IGameState</a>	80
<a href="#">gsm::IGameStateMachine</a>	80
<a href="#">ecs::IInputProvider</a>	81
<a href="#">ILoader</a>	81
<a href="#">gsm::InGameState</a>	82
<a href="#">ecs::InputIntentComponent</a>	84
<a href="#">ecs::InputMapping</a>	85
<a href="#">ecs::InputMappingManager</a>	85
<a href="#">ecs::InputNormalizer</a>	86
<a href="#">ecs::InputToVelocitySystem</a>	86
<a href="#">ecs::InteractionConfigComponent</a>	87
<a href="#">ecs::InteractionMapping</a>	87
<a href="#">ecs::InteractionSystem</a>	88
<a href="#">IPrefab</a>	88
<a href="#">rserv::IServer</a>	89
<a href="#">ecs::ISystem</a>	90
<a href="#">ecs::ISystemManager</a>	90
<a href="#">ecs::View&lt; Components &gt;::Iterator</a>	91
<a href="#">ui::Background::Layer</a>	91
<a href="#">ui::LayoutConfig</a>	92



err::LibrairiesLoadError	92
ecs::LifetimeComponent	93
ecs::LifetimeSystem	94
gsm::LoadingState	94
gsm::LobbyBrowserState	96
gsm::LobbyState	98
ecs::LocalPlayerTag	100
gsm::MainMenuState	100
MapParser	102
ecs::MobTag	103
MouseClickedInfo	103
MouseInputHandler	103
ecs::MovementInputSystem	104
ecs::MovementIntentComponent	105
ecs::MovementSystem	105
ecs::MultiShotPattern	106
ecs::MusicComponent	107
ecs::MusicIntentComponent	108
ecs::MusicSystem	108
NetworkEvent	109
ecs::NetworkHealthState	109
ecs::NetworkIdComponent	110
ecs::NetworkInterpolationSystem	110
ecs::NetworkStateComponent	111
ecs::NetworkTransformState	112
ecs::ObstacleTag	112
ecs::OutOfBoundsSystem	113
ecs::OwnerComponent	114
err::PacketError	114
ecs::ParallaxComponent	115
ecs::ParallaxLayer	116
ecs::ParallaxRenderingSystem	117
ParsedEntityPrefab	118
Parser	119
err::ParserError	119
gsm::PauseState	120
ecs::PlayerProjectileTag	122
ecs::PlayerTag	122
ecs::ProjectilePassThroughTag	123
ecs::ProjectilePrefabComponent	123
ecs::RectangleRenderComponent	124
ecs::RectangleRenderingSystem	124
ecs::Registry	125
ecs::RemappableKeyBinding	126
ResourceManager	126
gsm::ResultsState	127
ecs::ScoreComponent	128
gsm::ScoreFeedback	129
ScoreIntentComponent	129
ecs::ScoreSystem	129
ecs::ScoreValueComponent	130
rserv::Server	131
rserv::ServerConfig	137
err::ServerError	137
ecs::ServerInputProvider	138
ecs::ServerMovementInputSystem	140
ecs::ServerShootInputSystem	141
SettingsConfig	142

SettingsManager	143
gsm::SettingsState	144
ecs::ShooterTag	147
ecs::ShootingStatsComponent	147
ecs::ShootingSystem	148
ecs::ShootInputSystem	149
ecs::ShootIntentComponent	150
gsm::ShutdownState	151
Signal	152
ui::Slider	152
ecs::SoundIntentComponent	156
ecs::SoundSystem	157
ecs::SpatialGrid	158
ecs::SpawnIntentComponent	158
ecs::SpawnSystem	159
ecs::SpeedComponent	160
ecs::SpriteComponent	161
ecs::SpriteRenderingSystem	161
ecs::SystemManager	162
TagRegistry	163
ui::Text	163
ecs::TextComponent	166
ui::TextInput	166
ecs::TextRenderingSystem	170
ui::ToggleSwitch	171
ecs::TransformComponent	174
ecs::Transition	175
ecs::TriggerIntentComponent	175
ecs::TriggerSystem	176
ui::UIElement	177
ui::UILayout	178
ui::UIManager	181
ui::UINavigationController	182
Utils	182
math::Vector2f	183
ecs::VelocityComponent	183
ecs::View< Components >	184

## Chapter 3

# File Index

### 3.1 File List

Here is a list of all documented files with brief descriptions:

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ClientNetwork.hpp	185
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/colors.hpp	187
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/constants.hpp	196
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/Core.hpp	200
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/SettingsConfig.hpp	215
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/SettingsManager.hpp	216
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/Utils.hpp	233
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/AnimationComponent.hpp	188
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/HealthBarComponent.hpp	190
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/HitboxRenderComponent.hpp	190
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/MusicComponent.hpp	191
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/ParallaxComponent.hpp	191
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/RectangleRenderComponent.hpp	193
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/SpriteComponent.hpp	193
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/TextComponent.hpp	194
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/tags/BackGroundMusicTag.hpp	194
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/temporary/MusicIntentComponent.hpp	195
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/temporary/SoundIntentComponent.hpp	195
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/machine/AGameStateMachine.hpp	201
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/machine/GameStateMachine.hpp	202
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/base/AGameState.hpp	202
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Boot/BootState.hpp	203

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Dev/DevState.hpp	204
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/InGame/InGameState.hpp	205
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Loading/LoadingState.hpp	206
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Lobby/LobbyState.hpp	207
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/LobbyBrowser/LobbyBrowserState.hpp	208
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/MainMenu/MainMenuState.hpp	208
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Pause/PauseState.hpp	209
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Results/ResultsState.hpp	209
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Settings/SettingsState.hpp	210
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/initResourcesManager/GraphicalInputProvider.hpp	211
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/initResourcesManager/initResourcesManager.hpp	212
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/input/MouseInputHandler.hpp	213
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/interpolation/NetworkStateComponent.hpp	213
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/packet/DefaultPacketHandlers.hpp	214
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/audio/MusicSystem.hpp	217
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/audio/SoundSystem.hpp	217
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/input/MovementInputSystem.hpp	218
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/input/ShootInputSystem.hpp	218
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/network/NetworkInterpolationSystem.hpp	219
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/AnimationRenderingSystem.hpp	219
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/GameZoneRenderingSystem.hpp	219
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/GameZoneViewSystem.hpp	220
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/HealthBarRenderingSystem.hpp	220
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/HitboxRenderingSystem.hpp	221
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/ParallaxRenderingSystem.hpp	221
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/RectangleRenderingSystem.hpp	222
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/SpriteRenderingSystem.hpp	222
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/TextRenderingSystem.hpp	222
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/core/AFocusableElement.hpp	223
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/core/IFocusable.hpp	224
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/core/UILayout.hpp	224
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/Background.hpp	225
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/Text.hpp	231
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/base/UIElement.hpp	226
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/focusable/Button.hpp	227
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/focusable/Slider.hpp	228

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/focusable/TextInput.hpp	229
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/focusable/ToggleSwitch.hpp	230
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/manager/UIManager.hpp	231
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/navigation/UINavigationManager.hpp	232
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/constants.hpp	197
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/debug.hpp	252
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/translationToECS.hpp	287
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/CollisionRules/CollisionRules.hpp	234
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/CollisionRules/CollisionRulesData.hpp	235
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/base/AComponent.hpp	235
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/base/IComponent.hpp	235
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/AIMovementPatternComponent.hpp	236
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/ColliderComponent.hpp	237
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/DamageComponent.hpp	237
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/GameZoneComponent.hpp	238
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/HealthComponent.hpp	238
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/InteractionConfigComponent.hpp	239
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/LifetimeComponent.hpp	239
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/NetworkIdComponent.hpp	240
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/OwnerComponent.hpp	240
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/ProjectilePrefabComponent.hpp	241
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/ScoreComponent.hpp	241
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/ScoreValueComponent.hpp	241
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/ShootingStatsComponent.hpp	242
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/SpeedComponent.hpp	243
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/TransformComponent.hpp	243
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/VelocityComponent.hpp	244
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/tags/AIMoverTag.hpp	244
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/tags/AIShooterTag.hpp	244
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/tags/ControllableTag.hpp	245
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/tags/EnemyProjectileTag.hpp	245
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/tags/GameZoneColliderTag.hpp	245
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/tags/LocalPlayerTag.hpp	246

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/tags/MobTag.hpp . .	246
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/tags/ObstacleTag.hpp	246
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/tags/PlayerProjectileTag.hpp	247
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/tags/PlayerTag.hpp .	247
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/tags/ProjectilePassThroughTag.hpp	248
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/tags/ShooterTag.hpp	248
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/temporary/DamageIntentComponent.hpp	248
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/temporary/DeathIntentComponent.hpp	249
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/temporary/InputIntentComponent.hpp	249
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/temporary/MovementIntentComponent.hpp	250
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/temporary/ScoreIntentComponent.hpp	250
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/temporary/ShootIntentComponent.hpp	250
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/temporary/SpawnIntentComponent.hpp	251
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/temporary/TriggerIntentComponent.hpp	252
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/DLLoader/DLLoader.hpp . . . . .	253
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/DLLoader/ILoader.hpp . . . . .	254
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/DLLoader/LoaderType.hpp . . . . .	255
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/entity/Entity.hpp . . . . .	256
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/entity/EntityCreationContext.hpp	257
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/entity/componentArray/AComponentArray.hpp	255
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/entity/componentArray/IComponentArray.hpp	256
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/entity/factory/EntityFactory.hpp	257
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/entity/factory/IEntityFactory.hpp	258
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/entity/registry/Registry.hpp .	258
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/view/View.hpp . . . . .	259
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/AError.hpp . . . . .	260
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/ClientError.hpp . . . . .	260
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/ClientNetworkError.hpp . .	261
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/LError.hpp . . . . .	261
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/LibrairiesLoadError.hpp . .	262
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/PacketError.hpp . . . . .	262
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/ParserError.hpp . . . . .	263
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/ServerError.hpp . . . . .	263
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/gsm/IGameState.hpp . . . . .	263
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/gsm/IGameStateMachine.hpp . .	264
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/InputMapping/IInputProvider.hpp .	264
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/InputMapping/InputAction.hpp . .	265
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/InputMapping/InputMapping.hpp .	265
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/InputMapping/InputMappingManager.hpp	266
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/packet/DefaultPacketHandlers.hpp	215
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/packet/GameStateHandlers.hpp .	267
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/CollisionRulesParser.hpp .	268

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/Parser.hpp . . . . .	270
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/ParserParam.hpp . . . . .	271
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/Animation/AnimationConditionFactory.hpp 267	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/ComposantParser/ComposantParser.hpp 268	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/EntityParser/EntityParser.hpp 268	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/MapParser/MapParser.hpp . . . . .	269
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Prefab/APrefab.hpp . . . . .	272
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Prefab/IPrefab.hpp . . . . .	273
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Prefab/ParsedEntityPrefab.hpp . . . . .	273
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Prefab/entityPrefabManager/EntityPrefabManager.hpp 272	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/resourceManager/ResourceManager.hpp 274	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Signal/Signal.hpp . . . . .	274
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/SpatialGrid/SpatialGrid.hpp . . . . .	275
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/ai/AIMovementSystem.hpp 276	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/ai/AIShootingSystem.hpp . . . . .	277
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/base/ASystem.hpp . . . . .	277
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/base/ISystem.hpp . . . . .	277
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/bounds/OutOfBoundsSystem.hpp 278	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/death/DeathSystem.hpp . . . . .	278
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/health/HealthSystem.hpp . . . . .	279
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/input/InputNormalizer.hpp . . . . .	279
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/interactions/ActionFactory.hpp 280	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/interactions/InteractionSystem.hpp 280	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/interactions/TagRegistry.hpp 281	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/interactions/TriggerSystem.hpp 281	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/lifetime/LifetimeSystem.hpp 282	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/movement/InputToVelocitySystem.hpp 283	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/movement/MovementSystem.hpp 283	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/score/ScoreSystem.hpp . . . . .	284
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/shooting/ShootingSystem.hpp 284	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/spawn/SpawnSystem.hpp . . . . .	285
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/systemManager/ASystemManager.hpp 286	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/systemManager/ISystemManager.hpp 286	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/systemManager/SystemManager.hpp 286	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/types/Chrono.hpp . . . . .	287
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/types/FRect.hpp . . . . .	288
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/types/Vector2f.hpp . . . . .	288
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/Constants.hpp . . . . .	289
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/Core.hpp . . . . .	200
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/IServer.hpp . . . . .	292

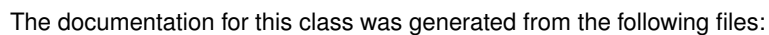


/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/Server.hpp . . . . .	293
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/ServerConfig.hpp . . . . .	295
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/Utils.hpp . . . . .	234
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/deltaTracker/ComponentDeltaTracker.hpp 289	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/deltaTracker/ComponentSerializer.hpp 290	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/machine/AGameStateMachine.hpp 202	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/machine/GameStateMachine.hpp 202	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/AGameState.hpp . . . .	203
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/scenes/Boot/BootState.hpp 204	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/scenes/GameEnd/GameEndState.hpp 291	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/scenes/InGame/InGameState.hpp 206	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/scenes/Loading/LoadingState.hpp 206	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/scenes/Lobby/LobbyState.hpp 207	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/scenes/Shutdown/ShutdownState.hpp 291	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/initResourcesManager/initResourcesManager.hpp 212	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/initResourcesManager/ServerInputProvider.hpp 292	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/packet/DefaultPacketHandlers.hpp .	215
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/systems/input/ServerMovementInputSystem.hpp 296	
/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/systems/input/ServerShootInputSystem.hpp 296	



## Class Documentation

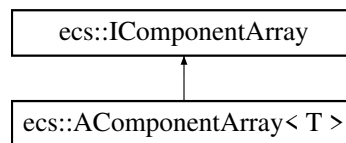
Inheritance diagram for `ecs::AComponent`:



- Generated by Doxygen

## 4.2 ecs::AComponentArray< T > Class Template Reference

Inheritance diagram for ecs::AComponentArray< T >:



### Public Member Functions

- void **add** (Entity entityId, std::shared\_ptr< T > component)
- std::shared\_ptr< T > **get** (Entity entityId) const
- std::vector< std::shared\_ptr< T > > **getAll** (Entity entityId) const
- void **removeComponents** (Entity entityId) override
- void **removeOneComponent** (Entity entityId) override
- bool **has** (Entity entityId) const
- Entity **getMaxEntityId** () const override

### Private Attributes

- std::vector< std::vector< std::shared\_ptr< T > > > **\_components**

### 4.2.1 Member Function Documentation

#### 4.2.1.1 getMaxEntityId()

```
template<typename T>
Entity ecs::AComponentArray< T >::getMaxEntityId () const [override], [virtual]
```

Implements [ecs::IComponentArray](#).

#### 4.2.1.2 removeComponents()

```
template<typename T>
void ecs::AComponentArray< T >::removeComponents (
    Entity entityId) [override], [virtual]
```

Implements [ecs::IComponentArray](#).

#### 4.2.1.3 removeOneComponent()

```
template<typename T>
void ecs::AComponentArray< T >::removeOneComponent (
    Entity entityId) [override], [virtual]
```

Implements [ecs::IComponentArray](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/entity/componentArray/AComponentArray.hpp

## 4.3 ActionFactory Class Reference

### Public Types

- using **ActionFunction** = std::function<void(std::shared\_ptr<ecs::Registry>, ecs::Entity, ecs::Entity)>

### Public Member Functions

- void **registerAction** (const std::string &actionId, ActionFunction action)
- void **executeAction** (const std::string &actionId, std::shared\_ptr< ecs::Registry > registry, ecs::Entity self, ecs::Entity other) const
- bool **hasAction** (const std::string &actionId) const

### Static Public Member Functions

- static const [ActionFactory](#) & **getInstance** ()

### Private Member Functions

- **ActionFactory** (const [ActionFactory](#) &)=delete
- [ActionFactory](#) & **operator=** (const [ActionFactory](#) &)=delete
- void **initializeConditions** ()

### Private Attributes

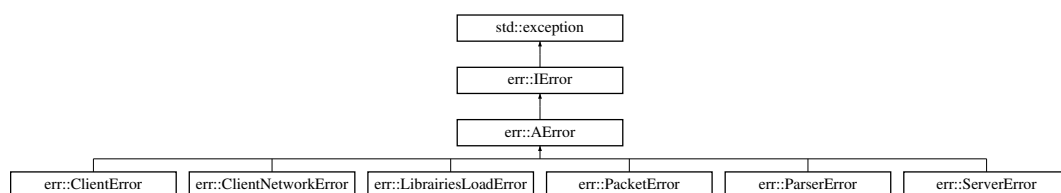
- std::unordered\_map< std::string, ActionFunction > **\_actions**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/interactions/ActionFactory.↔  
hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/interactions/ActionFactory.↔  
cpp

## 4.4 err::AError Class Reference

Inheritance diagram for err::AError:



## Public Member Functions

- **AError** (const std::string &message, int code=0)
- const char \* [what](#) () const noexcept override
- int [getCode](#) () const noexcept override
- std::string [getDetails](#) () const noexcept override
- virtual std::string [getType](#) () const noexcept override=0

## Protected Attributes

- std::string **m\_message**
- int **m\_code**

## 4.4.1 Member Function Documentation

### 4.4.1.1 [getCode\(\)](#)

```
int err::AError::getCode () const [override], [virtual], [noexcept]
```

Implements [err::IError](#).

### 4.4.1.2 [getDetails\(\)](#)

```
std::string err::AError::getDetails () const [override], [virtual], [noexcept]
```

Implements [err::IError](#).

### 4.4.1.3 [getType\(\)](#)

```
virtual std::string err::AError::getType () const [override], [pure virtual], [noexcept]
```

Implements [err::IError](#).

### 4.4.1.4 [what\(\)](#)

```
const char * err::AError::what () const [override], [virtual], [noexcept]
```

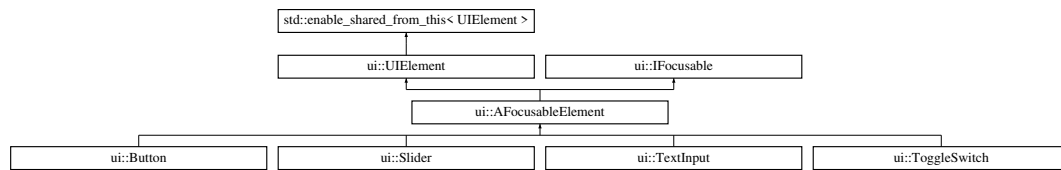
Implements [err::IError](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/AError.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/AError.cpp

## 4.5 ui::AFocusableElement Class Reference

Inheritance diagram for ui::AFocusableElement:



### Public Member Functions

- **AFocusableElement** (std::shared\_ptr< [ResourceManager](#) > resourceManager)
- virtual void **setFocused** (bool focused) override
- virtual bool **isFocused** () const override
- virtual bool **canBeFocused** () const override
- virtual void **onFocusGained** () override
- virtual void **onFocusLost** () override
- virtual void **onActivated** () override
- void **setOnFocusGained** (std::function< void()> callback)
- void **setOnFocusLost** (std::function< void()> callback)
- void **setOnActivated** (std::function< void()> callback)
- virtual void **handleInput** (const [math::Vector2f](#) &mousePos, bool mousePressed) override

### Public Member Functions inherited from ui::UIElement

- **UIElement** (std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void **setPosition** (const [math::Vector2f](#) &position)
- void **setSize** (const [math::Vector2f](#) &size)
- [math::Vector2f](#) **getPosition** () const
- [math::Vector2f](#) **getSize** () const
- [math::Vector2f](#) **getAbsolutePosition** () const
- [math::Vector2f](#) **getAbsoluteSize** () const
- void **setVisible** (bool visible)
- bool **isVisible** () const
- void **setState** (UIState state)
- UIState **getState** () const
- virtual void **setScale** (UIScale scale)
- UIScale **getScale** () const
- void **setParent** (std::weak\_ptr< [UIElement](#) > parent)
- std::shared\_ptr< [UIElement](#) > **getParent** () const
- void **addChild** (std::shared\_ptr< [UIElement](#) > child)
- void **removeChild** (std::shared\_ptr< [UIElement](#) > child)
- const std::vector< std::shared\_ptr< [UIElement](#) > > & **getChildren** () const
- virtual bool **containsPoint** (const [math::Vector2f](#) &point) const
- void **setOnClick** (std::function< void()> callback)
- void **setOnHover** (std::function< void()> callback)
- void **setOnRelease** (std::function< void()> callback)
- virtual void **render** ()
- virtual void **update** (float deltaTime)

## Public Member Functions inherited from [ui::IFocusable](#)

- virtual bool **onNavigateLeft** ()
- virtual bool **onNavigateRight** ()

## Protected Member Functions

- virtual void **onFocusStateChanged** (bool focused)

## Protected Member Functions inherited from [ui::UIElement](#)

- std::pair< int, int > **getWindowSize** () const
- std::pair< int, int > **getLogicalSize** () const
- float **getScaleFactor** () const

## Protected Attributes

- bool **\_focused** = false
- bool **\_pressedInside** = false
- bool **\_wasPressed** = false
- std::function< void()> **\_onFocusGained**
- std::function< void()> **\_onFocusLost**
- std::function< void()> **\_onActivated**

## Protected Attributes inherited from [ui::UIElement](#)

- std::weak\_ptr< [ResourceManager](#) > **\_resourceManager**
- [math::Vector2f](#) **\_position**
- [math::Vector2f](#) **\_size**
- bool **\_visible** = true
- UIState **\_state** = UIState::Normal
- UIScale **\_scale** = UIScale::Normal
- std::weak\_ptr< [UIElement](#) > **\_parent**
- std::vector< std::shared\_ptr< [UIElement](#) > > **\_children**
- bool **\_pressedInside** = false
- bool **\_wasPressed** = false
- std::function< void()> **\_onClick**
- std::function< void()> **\_onHover**
- std::function< void()> **\_onRelease**

## 4.5.1 Member Function Documentation

### 4.5.1.1 canBeFocused()

```
bool ui::AFocusableElement::canBeFocused () const [override], [virtual]
```

Implements [ui::IFocusable](#).

#### 4.5.1.2 handleInput()

```
void ui::AFocusableElement::handleInput (
    const math::Vector2f & mousePos,
    bool mousePressed) [override], [virtual]
```

Reimplemented from [ui::UIElement](#).

#### 4.5.1.3 isFocused()

```
bool ui::AFocusableElement::isFocused () const [override], [virtual]
```

Implements [ui::IFocusable](#).

#### 4.5.1.4 onActivated()

```
void ui::AFocusableElement::onActivated () [override], [virtual]
```

Implements [ui::IFocusable](#).

#### 4.5.1.5 onFocusGained()

```
void ui::AFocusableElement::onFocusGained () [override], [virtual]
```

Implements [ui::IFocusable](#).

#### 4.5.1.6 onFocusLost()

```
void ui::AFocusableElement::onFocusLost () [override], [virtual]
```

Implements [ui::IFocusable](#).

#### 4.5.1.7 setFocused()

```
void ui::AFocusableElement::setFocused (
    bool focused) [override], [virtual]
```

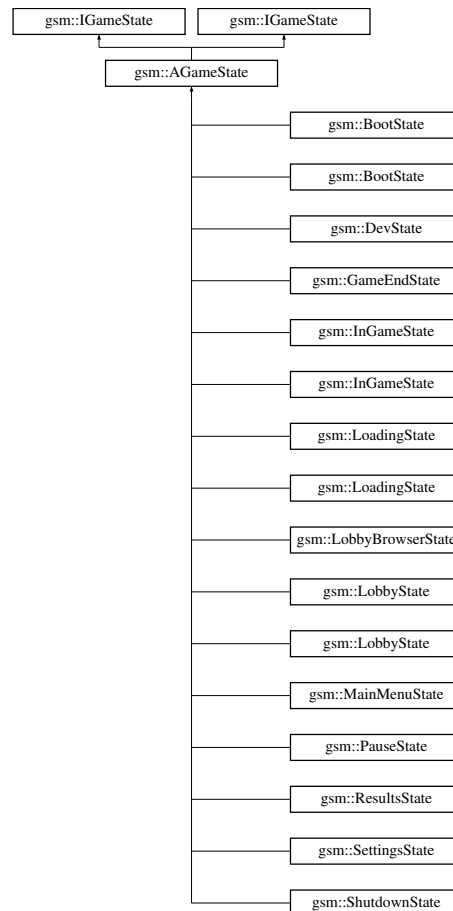
Implements [ui::IFocusable](#).

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

- [/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/core/AFocusableElement.hpp](#)
- [/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/core/AFocusableElement.cpp](#)

## 4.6 gsm::AGameState Class Reference

Inheritance diagram for gsm::AGameState:



### Public Member Functions

- **AGameState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void [enter](#) () override
- void [update](#) (float deltaTime) override
- void [exit](#) () override
- std::vector< std::shared\_ptr< [ecs::ISystem](#) > > [getSystems](#) () const override
- **AGameState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void [enter](#) () override
- void [update](#) (float deltaTime) override
- void [exit](#) () override
- std::vector< std::shared\_ptr< [ecs::ISystem](#) > > [getSystems](#) () const override

### Protected Member Functions

- void [addSystem](#) (std::shared\_ptr< [ecs::ISystem](#) > system) override
- void [addSystem](#) (std::shared\_ptr< [ecs::ISystem](#) > system) override



### Protected Attributes

- `std::weak_ptr< IGameStateMachine > _gsm`
- `std::shared_ptr< ResourceManager > _resourceManager`
- `std::vector< std::shared_ptr< ecs::ISystem > > _systems`

## 4.6.1 Member Function Documentation

### 4.6.1.1 addSystem() [1/2]

```
void gsm::AGameState::addSystem (  
    std::shared_ptr< ecs::ISystem > system) [override], [protected], [virtual]
```

Implements [gsm::IGameState](#).

### 4.6.1.2 addSystem() [2/2]

```
void gsm::AGameState::addSystem (  
    std::shared_ptr< ecs::ISystem > system) [override], [protected], [virtual]
```

Implements [gsm::IGameState](#).

### 4.6.1.3 enter() [1/2]

```
void gsm::AGameState::enter () [override], [virtual]
```

Implements [gsm::IGameState](#).

### 4.6.1.4 enter() [2/2]

```
void gsm::AGameState::enter () [override], [virtual]
```

Implements [gsm::IGameState](#).

### 4.6.1.5 exit() [1/2]

```
void gsm::AGameState::exit () [override], [virtual]
```

Implements [gsm::IGameState](#).

### 4.6.1.6 exit() [2/2]

```
void gsm::AGameState::exit () [override], [virtual]
```

Implements [gsm::IGameState](#).

**4.6.1.7 getSystems() [1/2]**

```
std::vector< std::shared_ptr< ecs::ISystem > > gsm::AGameState::getSystems () const [override],
[virtual]
```

Implements [gsm::IGameState](#).

**4.6.1.8 getSystems() [2/2]**

```
std::vector< std::shared_ptr< ecs::ISystem > > gsm::AGameState::getSystems () const [override],
[virtual]
```

Implements [gsm::IGameState](#).

**4.6.1.9 update() [1/2]**

```
void gsm::AGameState::update (
    float deltaTime) [override], [virtual]
```

Implements [gsm::IGameState](#).

**4.6.1.10 update() [2/2]**

```
void gsm::AGameState::update (
    float deltaTime) [override], [virtual]
```

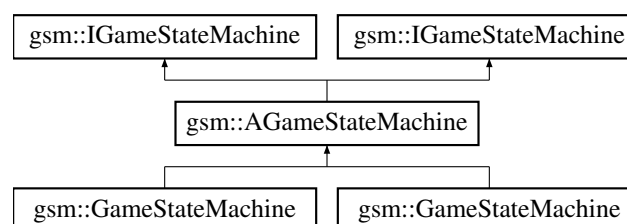
Implements [gsm::IGameState](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/base/AGameState.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/AGameState.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/base/AGameState.cpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/AGameState.cpp

**4.7 gsm::AGameStateMachine Class Reference**

Inheritance diagram for gsm::AGameStateMachine:



## Public Member Functions

- void [changeState](#) (std::shared\_ptr< [IGameState](#) > newState) override
- void [pushState](#) (std::shared\_ptr< [IGameState](#) > newState) override
- void [popState](#) () override
- void [requestStateChange](#) (std::shared\_ptr< [IGameState](#) > newState) override
- void [requestStatePush](#) (std::shared\_ptr< [IGameState](#) > newState) override
- void [requestStatePop](#) () override
- void [update](#) (float deltaTime) override
- void [changeState](#) (std::shared\_ptr< [IGameState](#) > newState) override
- void [pushState](#) (std::shared\_ptr< [IGameState](#) > newState) override
- void [popState](#) () override
- void [update](#) (float deltaTime) override

## Protected Attributes

- std::stack< std::shared\_ptr< [IGameState](#) > > [\\_states](#)
- std::shared\_ptr< [IGameState](#) > [\\_pendingChangeState](#)
- std::shared\_ptr< [IGameState](#) > [\\_pendingPushState](#)
- bool [\\_pendingPopState](#) = false

## 4.7.1 Member Function Documentation

### 4.7.1.1 [changeState\(\)](#) [1/2]

```
void gsm::AGameStateMachine::changeState (
    std::shared_ptr< IGameState > newState) [override], [virtual]
```

Implements [gsm::IGameStateMachine](#).

### 4.7.1.2 [changeState\(\)](#) [2/2]

```
void gsm::AGameStateMachine::changeState (
    std::shared_ptr< IGameState > newState) [override], [virtual]
```

Implements [gsm::IGameStateMachine](#).

### 4.7.1.3 [popState\(\)](#) [1/2]

```
void gsm::AGameStateMachine::popState () [override], [virtual]
```

Implements [gsm::IGameStateMachine](#).

### 4.7.1.4 [popState\(\)](#) [2/2]

```
void gsm::AGameStateMachine::popState () [override], [virtual]
```

Implements [gsm::IGameStateMachine](#).

#### 4.7.1.5 pushState() [1/2]

```
void gsm::AGameStateMachine::pushState (  
    std::shared_ptr< IGameState > newState) [override], [virtual]
```

Implements [gsm::IGameStateMachine](#).

#### 4.7.1.6 pushState() [2/2]

```
void gsm::AGameStateMachine::pushState (  
    std::shared_ptr< IGameState > newState) [override], [virtual]
```

Implements [gsm::IGameStateMachine](#).

#### 4.7.1.7 requestStateChange()

```
void gsm::AGameStateMachine::requestStateChange (  
    std::shared_ptr< IGameState > newState) [override], [virtual]
```

Implements [gsm::IGameStateMachine](#).

#### 4.7.1.8 requestStatePop()

```
void gsm::AGameStateMachine::requestStatePop () [override], [virtual]
```

Implements [gsm::IGameStateMachine](#).

#### 4.7.1.9 requestStatePush()

```
void gsm::AGameStateMachine::requestStatePush (  
    std::shared_ptr< IGameState > newState) [override], [virtual]
```

Implements [gsm::IGameStateMachine](#).

#### 4.7.1.10 update() [1/2]

```
void gsm::AGameStateMachine::update (  
    float deltaTime) [override], [virtual]
```

Implements [gsm::IGameStateMachine](#).

## 4.7.1.11 update() [2/2]

```
void gsm::AGameStateMachine::update (
    float deltaTime) [override], [virtual]
```

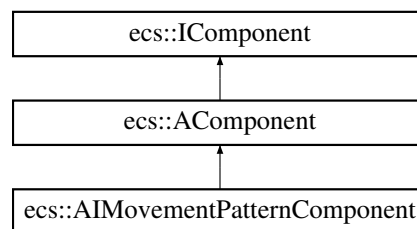
Implements [gsm::IGameStateMachine](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/machine/AGameStateMachine.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/machine/AGameStateMachine.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/machine/AGameStateMachine.cpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/machine/AGameStateMachine.cpp

## 4.8 ecs::AIMovementPatternComponent Class Reference

Inheritance diagram for ecs::AIMovementPatternComponent:



## Public Member Functions

- **AIMovementPatternComponent** (AIMovementPattern p=AIMovementPattern::STRAIGHT\_LINE)
- AIMovementPattern **getPattern** () const
- float **getZigzagAmplitude** () const
- float **getZigzagFrequency** () const
- float **getDetectionRange** () const
- float **getVerticalDeadzone** () const
- float **getTimer** () const
- void **setPattern** (AIMovementPattern p)
- void **setZigzagAmplitude** (float amplitude)
- void **setZigzagFrequency** (float frequency)
- void **setDetectionRange** (float range)
- void **setVerticalDeadzone** (float deadzone)
- void **setTimer** (float t)

## Public Attributes

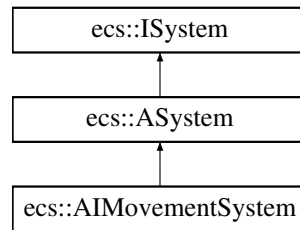
- AIMovementPattern **pattern**
- float **zigzagAmplitude** = constants::DEFAULT\_ZIGZAG\_AMPLITUDE
- float **zigzagFrequency** = constants::DEFAULT\_ZIGZAG\_FREQUENCY
- float **detectionRange** = constants::DEFAULT\_DETECTION\_RANGE
- float **verticalDeadzone** = constants::DEFAULT\_VERTICAL\_DEADZONE
- float **timer** = constants::DEFAULT\_TIMER

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/AIMovementPatternComponent.hpp

## 4.9 ecs::AIMovementSystem Class Reference

Inheritance diagram for ecs::AIMovementSystem:



### Public Member Functions

- void [update](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

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

- void [updateSystem](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

### Private Member Functions

- void [executeStraightLine](#) (std::shared\_ptr< [AIMovementPatternComponent](#) > movement, std::shared\_ptr< [TransformComponent](#) > transform, std::shared\_ptr< [InputIntentComponent](#) > inputIntent)
- void [executeZigzag](#) (std::shared\_ptr< [AIMovementPatternComponent](#) > movement, std::shared\_ptr< [TransformComponent](#) > transform, std::shared\_ptr< [InputIntentComponent](#) > inputIntent, float deltaTime)
- void [executeVerticalMirror](#) (std::shared\_ptr< [Registry](#) > registry, std::shared\_ptr< [AIMovementPatternComponent](#) > movement, std::shared\_ptr< [TransformComponent](#) > transform, std::shared\_ptr< [InputIntentComponent](#) > inputIntent)
- void [executeFollowRight](#) (std::shared\_ptr< [AIMovementPatternComponent](#) > movement, std::shared\_ptr< [TransformComponent](#) > transform, std::shared\_ptr< [InputIntentComponent](#) > inputIntent, float deltaTime)
- std::optional< size\_t > [findNearestPlayer](#) (std::shared\_ptr< [Registry](#) > registry, const [math::Vector2f](#) &position, float range)

## 4.9.1 Member Function Documentation

### 4.9.1.1 update()

```

void ecs::AIMovementSystem::update (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [virtual]
  
```

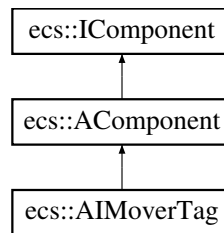
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/ai/AIMovementSystem.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/ai/AIMovementSystem.cpp

## 4.10 ecs::AIMoverTag Struct Reference

Inheritance diagram for ecs::AIMoverTag:

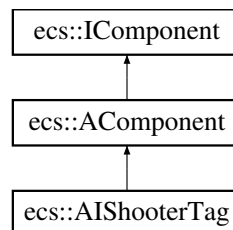


The documentation for this struct was generated from the following file:

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/tags/AIMoverTag.hpp

## 4.11 ecs::AIShooterTag Struct Reference

Inheritance diagram for ecs::AIShooterTag:

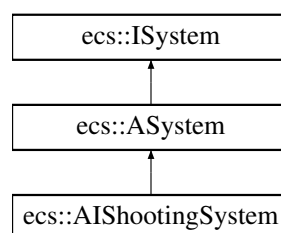


The documentation for this struct was generated from the following file:

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/tags/AIShooterTag.hpp

## 4.12 ecs::AIShootingSystem Class Reference

Inheritance diagram for ecs::AIShootingSystem:



## Public Member Functions

- void [update](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

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

- void [updateSystem](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

### 4.12.1 Member Function Documentation

#### 4.12.1.1 update()

```
void ecs::AIShootingSystem::update (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [virtual]
```

Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/ai/AIShootingSystem.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/ai/AIShootingSystem.cpp

## 4.13 [ecs::AnimationClip](#) Struct Reference

### Public Attributes

- std::string **texturePath**
- float **frameWidth**
- float **frameHeight**
- int **frameCount**
- float **startWidth**
- float **startHeight**
- float **speed**
- bool **loop**

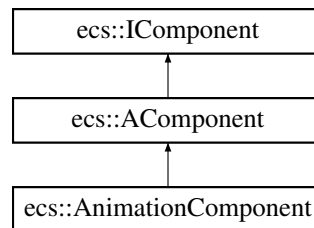
The documentation for this struct was generated from the following file:

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/Animation↵  
Component.hpp



## 4.14 ecs::AnimationComponent Class Reference

Inheritance diagram for ecs::AnimationComponent:



### Public Member Functions

- void **addState** (const std::string &name, std::shared\_ptr< [AnimationClip](#) > clip)
- void **addTransition** (const std::string &from, const std::string &to, const std::vector< [AnimationCondition](#) > &conditions, bool playRewind=false)
- void **setCurrentState** (const std::string &state)
- const std::string & **getCurrentState** () const
- float **getTimer** () const
- void **setTimer** (float timer)
- bool **isPlaying** () const
- void **setPlaying** (bool playing)
- bool **isPlayingRewind** () const
- void **setPlayingRewind** (bool rewind)
- int **getRewindStartFrame** () const
- void **setRewindStartFrame** (int frame)
- std::shared\_ptr< const [AnimationClip](#) > **getCurrentClip** () const
- const std::vector< [Transition](#) > & **getTransitions** () const
- int **getCurrentFrame** () const
- void **setCurrentFrame** (int frame)
- const [math::FRect](#) & **getFrameRect** () const
- void **setFrameRect** (const [math::FRect](#) &rect)
- bool **isValid** () const
- bool **isAnimationFinished** () const
- void **setStateJustChanged** (bool changed)
- bool **getStateJustChanged** () const
- void **setMinAnimationTime** (float time)
- float **getMinAnimationTime** () const

### Private Attributes

- std::unordered\_map< std::string, std::shared\_ptr< [AnimationClip](#) > > **\_states**
- std::vector< [Transition](#) > **\_transitions**
- std::string **\_currentState**
- float **\_timer**
- bool **\_isPlaying**
- bool **\_playRewind**
- int **\_currentFrame**
- int **\_rewindStartFrame**
- [math::FRect](#) **\_frameRect**
- bool **\_stateJustChanged** = false
- float **\_minAnimationTime** = 0.0f

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/Animation↔  
Component.hpp

## 4.15 ecs::AnimationCondition Struct Reference

### Public Attributes

- std::string **param**
- bool **equals**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/Animation↵  
Component.hpp

## 4.16 ecs::AnimationConditionFactory Class Reference

### Public Types

- using **ConditionFunction** = std::function<bool(std::shared\_ptr<Registry>, Entity)>

### Public Member Functions

- void **registerCondition** (const std::string &name, ConditionFunction condition)
- bool **evaluateCondition** (const std::string &name, std::shared\_ptr< Registry > registry, Entity entity) const
- bool **hasCondition** (const std::string &name) const
- void **unregisterCondition** (const std::string &name)
- void **clearConditions** ()

### Static Public Member Functions

- static const [AnimationConditionFactory](#) & **getInstance** ()
- static bool **getConditionValue** (const std::string &param, std::shared\_ptr< [Registry](#) > registry, Entity entity)

### Private Member Functions

- void **initializeConditions** ()
- **AnimationConditionFactory** (const [AnimationConditionFactory](#) &)=delete
- [AnimationConditionFactory](#) & **operator=** (const [AnimationConditionFactory](#) &)=delete

### Private Attributes

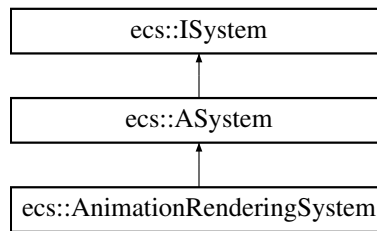
- std::unordered\_map< std::string, ConditionFunction > **\_conditions**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/Animation/AnimationCondition↵  
Factory.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/Animation/AnimationCondition↵  
Factory.cpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/Animation/AnimationConditions↵  
Registry.cpp

## 4.17 ecs::AnimationRenderingSystem Class Reference

Inheritance diagram for ecs::AnimationRenderingSystem:



### Protected Member Functions

- void [update](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

### Additional Inherited Members

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

- void [updateSystem](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

### 4.17.1 Member Function Documentation

#### 4.17.1.1 update()

```

void ecs::AnimationRenderingSystem::update (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [protected], [virtual]
  
```

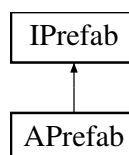
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/AnimationRendering↵  
System.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/AnimationRendering↵  
System.cpp

## 4.18 APrefab Class Reference

Inheritance diagram for APrefab:



## Public Member Functions

- `ecs::Entity` [instantiate](#) (`const std::shared_ptr< ecs::Registry > &registry`, `const std::shared_ptr< ecs::IEntityFactory > &factory`, `const ecs::EntityCreationContext &context=ecs::EntityCreationContext::forLocalClient()`) `override`
- `ecs::Entity` [instantiate](#) (`const std::shared_ptr< ecs::Registry > &registry`) `override`

## 4.18.1 Member Function Documentation

### 4.18.1.1 [instantiate\(\)](#) [1/2]

```
ecs::Entity APrefab::instantiate (
    const std::shared_ptr< ecs::Registry > & registry) [override], [virtual]
```

Implements [IPrefab](#).

### 4.18.1.2 [instantiate\(\)](#) [2/2]

```
ecs::Entity APrefab::instantiate (
    const std::shared_ptr< ecs::Registry > & registry,
    const std::shared_ptr< ecs::IEntityFactory > & factory,
    const ecs::EntityCreationContext & context = ecs::EntityCreationContext::forLocalClient\(\))
[override], [virtual]
```

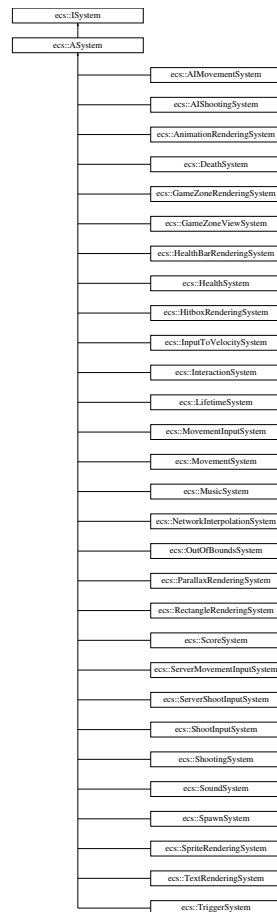
Implements [IPrefab](#).

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

- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Prefab/APrefab.hpp`
- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Prefab/APrefab.cpp`

## 4.19 `ecs::ASystem` Class Reference

Inheritance diagram for `ecs::ASystem`:



## Public Member Functions

- void **updateSystem** (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

## Protected Member Functions

- virtual void **update** (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime)=0

## 4.19.1 Member Function Documentation

### 4.19.1.1 updateSystem()

```

void ecs::ASystem::updateSystem (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [virtual]

```

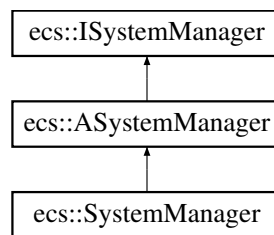
Implements [ecs::ISystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/base/ASystem.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/base/ASystem.cpp

## 4.20 ecs::ASystemManager Class Reference

Inheritance diagram for ecs::ASystemManager:



### Public Member Functions

- void [updateAllSystems](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override
- void [addSystem](#) (std::shared\_ptr< [ISystem](#) > system) override
- void [removeSystem](#) (std::shared\_ptr< [ISystem](#) > system) override

### Private Attributes

- std::vector< std::shared\_ptr< [ISystem](#) > > **\_systems**

### 4.20.1 Member Function Documentation

#### 4.20.1.1 addSystem()

```
void ecs::ASystemManager::addSystem (
    std::shared_ptr< ISystem > system) [override], [virtual]
```

Implements [ecs::ISystemManager](#).

#### 4.20.1.2 removeSystem()

```
void ecs::ASystemManager::removeSystem (
    std::shared_ptr< ISystem > system) [override], [virtual]
```

Implements [ecs::ISystemManager](#).

#### 4.20.1.3 updateAllSystems()

```
void ecs::ASystemManager::updateAllSystems (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [virtual]
```

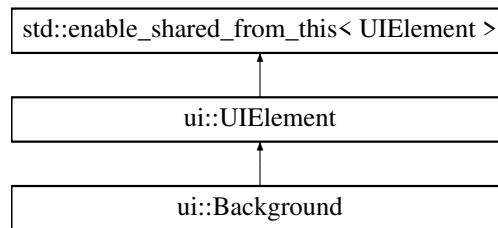
Implements [ecs::ISystemManager](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/systemManager/ASystemManager.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/systemManager/ASystemManager.cpp

## 4.21 ui::Background Class Reference

Inheritance diagram for ui::Background:



### Classes

- struct [Layer](#)

### Public Member Functions

- **Background** (std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void [render](#) () override
- void [update](#) (float deltaTime) override
- void **addLayer** (const std::string &texturePath, float speedX, float speedY=0.0f, const [math::Vector2f](#) &sourceSize=[math::Vector2f](#)(constants::MAX\_WIDTH, constants::MAX\_HEIGHT))

### Public Member Functions inherited from ui::UIElement

- **UIElement** (std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void **setPosition** (const [math::Vector2f](#) &position)
- void **setSize** (const [math::Vector2f](#) &size)
- [math::Vector2f](#) **getPosition** () const
- [math::Vector2f](#) **getSize** () const
- [math::Vector2f](#) **getAbsolutePosition** () const
- [math::Vector2f](#) **getAbsoluteSize** () const
- void **setVisible** (bool visible)
- bool **isVisible** () const
- void **setState** (UIState state)
- UIState **getState** () const
- virtual void **setScale** (UIScale scale)
- UIScale **getScale** () const
- void **setParent** (std::weak\_ptr< [UIElement](#) > parent)
- std::shared\_ptr< [UIElement](#) > **getParent** () const
- void **addChild** (std::shared\_ptr< [UIElement](#) > child)
- void **removeChild** (std::shared\_ptr< [UIElement](#) > child)
- const std::vector< std::shared\_ptr< [UIElement](#) > > & **getChildren** () const
- virtual void **handleInput** (const [math::Vector2f](#) &mousePos, bool mousePressed)
- virtual bool **containsPoint** (const [math::Vector2f](#) &point) const
- void **setOnClick** (std::function< void()> callback)
- void **setOnHover** (std::function< void()> callback)
- void **setOnRelease** (std::function< void()> callback)

### Private Member Functions

- float **calculateScale** (const [Layer](#) &layer, float screenWidth)

### Private Attributes

- std::vector< [Layer](#) > **\_layers**

### Additional Inherited Members

### Protected Member Functions inherited from [ui::UIElement](#)

- std::pair< int, int > **getWindowSize** () const
- std::pair< int, int > **getLogicalSize** () const
- float **getScaleFactor** () const

### Protected Attributes inherited from [ui::UIElement](#)

- std::weak\_ptr< [ResourceManager](#) > **\_resourceManager**
- [math::Vector2f](#) **\_position**
- [math::Vector2f](#) **\_size**
- bool **\_visible** = true
- UIState **\_state** = UIState::Normal
- UIScale **\_scale** = UIScale::Normal
- std::weak\_ptr< [UIElement](#) > **\_parent**
- std::vector< std::shared\_ptr< [UIElement](#) > > **\_children**
- bool **\_pressedInside** = false
- bool **\_wasPressed** = false
- std::function< void()> **\_onClick**
- std::function< void()> **\_onHover**
- std::function< void()> **\_onRelease**

## 4.21.1 Member Function Documentation

### 4.21.1.1 render()

```
void ui::Background::render () [override], [virtual]
```

Reimplemented from [ui::UIElement](#).

### 4.21.1.2 update()

```
void ui::Background::update (
    float deltaTime) [override], [virtual]
```

Reimplemented from [ui::UIElement](#).

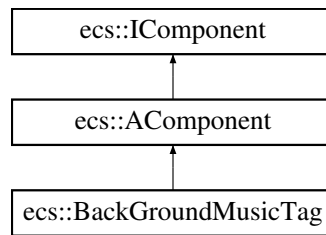
The documentation for this class was generated from the following files:

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/Background.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/Background.cpp



## 4.22 ecs::BackGroundMusicTag Class Reference

Inheritance diagram for ecs::BackGroundMusicTag:

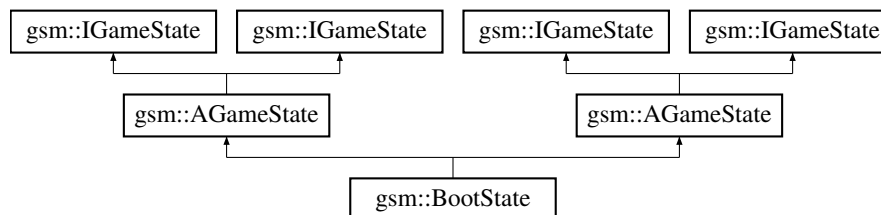


The documentation for this class was generated from the following file:

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/tags/BackGroundMusicTag.hpp

## 4.23 gsm::BootState Class Reference

Inheritance diagram for gsm::BootState:



### Public Member Functions

- **BootState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void [enter](#) () override
- void [update](#) (float deltaTime) override
- void [exit](#) () override
- **BootState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void [enter](#) () override

### Public Member Functions inherited from [gsm::AGameState](#)

- **AGameState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- std::vector< std::shared\_ptr< [ecs::ISystem](#) > > [getSystems](#) () const override
- **AGameState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- std::vector< std::shared\_ptr< [ecs::ISystem](#) > > [getSystems](#) () const override

## Additional Inherited Members

### Protected Member Functions inherited from [gsm::AGameState](#)

- void [addSystem](#) (std::shared\_ptr< [ecs::ISystem](#) > system) override
- void [addSystem](#) (std::shared\_ptr< [ecs::ISystem](#) > system) override

### Protected Attributes inherited from [gsm::AGameState](#)

- std::weak\_ptr< [IGameStateMachine](#) > [\\_gsm](#)
- std::shared\_ptr< [ResourceManager](#) > [\\_resourceManager](#)
- std::vector< std::shared\_ptr< [ecs::ISystem](#) > > [\\_systems](#)

## 4.23.1 Member Function Documentation

### 4.23.1.1 [enter\(\)](#) [1/2]

```
void gsm::BootState::enter () [override], [virtual]
```

Reimplemented from [gsm::AGameState](#).

### 4.23.1.2 [enter\(\)](#) [2/2]

```
void gsm::BootState::enter () [override], [virtual]
```

Reimplemented from [gsm::AGameState](#).

### 4.23.1.3 [exit\(\)](#)

```
void gsm::BootState::exit () [override], [virtual]
```

Reimplemented from [gsm::AGameState](#).

### 4.23.1.4 [update\(\)](#)

```
void gsm::BootState::update (
    float deltaTime) [override], [virtual]
```

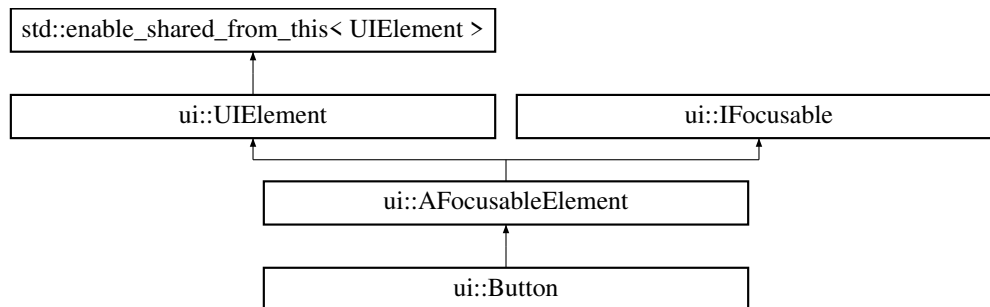
Reimplemented from [gsm::AGameState](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Boot/BootState.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/scenes/Boot/BootState.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Boot/BootState.cpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/scenes/Boot/BootState.cpp

## 4.24 ui::Button Class Reference

Inheritance diagram for ui::Button:



### Public Member Functions

- **Button** (std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void **setText** (const std::string &text)
- const std::string & **getText** () const
- void **setTextColor** (const gfx::color\_t &color)
- void **setFontPath** (const std::string &fontPath)
- void **setNormalColor** (const gfx::color\_t &color)
- void **setHoveredColor** (const gfx::color\_t &color)
- void **setPressedColor** (const gfx::color\_t &color)
- void **setDisabledColor** (const gfx::color\_t &color)
- void **setFocusedColor** (const gfx::color\_t &color)
- void **setBaseFontSize** (size\_t fontSize)
- size\_t **getBaseFontSize** () const
- virtual void **render** () override

### Public Member Functions inherited from [ui::AFocusableElement](#)

- **AFocusableElement** (std::shared\_ptr< [ResourceManager](#) > resourceManager)
- virtual void **setFocused** (bool focused) override
- virtual bool **isFocused** () const override
- virtual bool **canBeFocused** () const override
- virtual void **onFocusGained** () override
- virtual void **onFocusLost** () override
- virtual void **onActivated** () override
- void **setOnFocusGained** (std::function< void()> callback)
- void **setOnFocusLost** (std::function< void()> callback)
- void **setOnActivated** (std::function< void()> callback)
- virtual void **handleInput** (const [math::Vector2f](#) &mousePos, bool mousePressed) override

## Public Member Functions inherited from [ui::UIElement](#)

- **UIElement** (std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void **setPosition** (const [math::Vector2f](#) &position)
- void **setSize** (const [math::Vector2f](#) &size)
- [math::Vector2f](#) **getPosition** () const
- [math::Vector2f](#) **getSize** () const
- [math::Vector2f](#) **getAbsolutePosition** () const
- [math::Vector2f](#) **getAbsoluteSize** () const
- void **setVisible** (bool visible)
- bool **isVisible** () const
- void **setState** (UIState state)
- UIState **getState** () const
- virtual void **setScale** (UIScale scale)
- UIScale **getScale** () const
- void **setParent** (std::weak\_ptr< [UIElement](#) > parent)
- std::shared\_ptr< [UIElement](#) > **getParent** () const
- void **addChild** (std::shared\_ptr< [UIElement](#) > child)
- void **removeChild** (std::shared\_ptr< [UIElement](#) > child)
- const std::vector< std::shared\_ptr< [UIElement](#) > > & **getChildren** () const
- virtual bool **containsPoint** (const [math::Vector2f](#) &point) const
- void **setOnClick** (std::function< void()> callback)
- void **setOnHover** (std::function< void()> callback)
- void **setOnRelease** (std::function< void()> callback)
- virtual void **update** (float deltaTime)

## Public Member Functions inherited from [ui::IFocusable](#)

- virtual bool **onNavigateLeft** ()
- virtual bool **onNavigateRight** ()

## Private Member Functions

- [gfx::color\\_t](#) **getCurrentColor** () const
- [size\\_t](#) **getFontSize** () const

## Private Attributes

- std::string **\_text**
- [gfx::color\\_t](#) **\_textColor** = colors::UI\_TEXT
- std::string **\_fontPath** = "assets/fonts/arial.ttf"
- [gfx::color\\_t](#) **\_normalColor** = colors::BUTTON\_PRIMARY
- [gfx::color\\_t](#) **\_hoveredColor** = colors::BUTTON\_PRIMARY\_HOVER
- [gfx::color\\_t](#) **\_pressedColor** = colors::BUTTON\_PRIMARY\_PRESSED
- [gfx::color\\_t](#) **\_disabledColor** = colors::UI\_DISABLED
- [gfx::color\\_t](#) **\_focusedColor** = colors::UI\_FOCUSED
- [size\\_t](#) **\_baseFontSize** = constants::BUTTON\_FONT\_SIZE\_BASE

## Additional Inherited Members

## Protected Member Functions inherited from [ui::AFocusableElement](#)

- virtual void **onFocusStateChanged** (bool focused)

### Protected Member Functions inherited from [ui::UIElement](#)

- `std::pair< int, int > getWindowSize () const`
- `std::pair< int, int > getLogicalSize () const`
- `float getScaleFactor () const`

### Protected Attributes inherited from [ui::AFocusableElement](#)

- `bool _focused = false`
- `bool _pressedInside = false`
- `bool _wasPressed = false`
- `std::function< void()> _onFocusGained`
- `std::function< void()> _onFocusLost`
- `std::function< void()> _onActivated`

### Protected Attributes inherited from [ui::UIElement](#)

- `std::weak_ptr< ResourceManager > _resourceManager`
- `math::Vector2f _position`
- `math::Vector2f _size`
- `bool _visible = true`
- `UIState _state = UIState::Normal`
- `UIScale _scale = UIScale::Normal`
- `std::weak_ptr< UIElement > _parent`
- `std::vector< std::shared_ptr< UIElement > > _children`
- `bool _pressedInside = false`
- `bool _wasPressed = false`
- `std::function< void()> _onClick`
- `std::function< void()> _onHover`
- `std::function< void()> _onRelease`

## 4.24.1 Member Function Documentation

### 4.24.1.1 `render()`

```
void ui::Button::render () [override], [virtual]
```

Reimplemented from [ui::UIElement](#).

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

- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/focusable/Button.hpp`
- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/focusable/Button.cpp`

## 4.25 math::Chrono Class Reference

### Public Member Functions

- void **start** ()
- void **stop** ()
- void **reset** ()
- float **getElapsedSeconds** () const
- float **getElapsedMilliseconds** () const
- bool **isRunning** () const

### Private Attributes

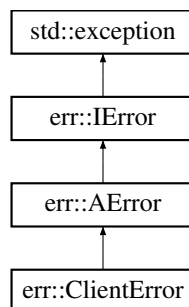
- std::chrono::high\_resolution\_clock::time\_point **\_startTime**
- std::chrono::high\_resolution\_clock::time\_point **\_stopTime**
- bool **\_isRunning**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/types/Chrono.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/types/Chrono.cpp

## 4.26 err::ClientError Class Reference

Inheritance diagram for err::ClientError:



### Public Types

- enum **ErrorCode** {  
**UNKNOWN** = 2000 , **CONNECTION\_FAILED** = 2001 , **DISCONNECTED** = 2002 , **TIMEOUT** = 2003 ,  
**NOT\_INITIALIZED** = 2004 , **CAN\_NOT\_OPEN\_FILE** = 2005 }

### Public Member Functions

- **ClientError** (const std::string &message, ErrorCode code=UNKNOWN)
- std::string **getType** () const noexcept override

## Public Member Functions inherited from [err::AError](#)

- **AError** (const std::string &message, int code=0)
- const char \* **what** () const noexcept override
- int **getCode** () const noexcept override
- std::string **getDetails** () const noexcept override

## Additional Inherited Members

## Protected Attributes inherited from [err::AError](#)

- std::string **m\_message**
- int **m\_code**

### 4.26.1 Member Function Documentation

#### 4.26.1.1 **getType()**

```
std::string err::ClientError::getType () const [override], [virtual], [noexcept]
```

Implements [err::AError](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/ClientError.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/ClientError.cpp

## 4.27 ClientNetwork Class Reference

### Public Member Functions

- void **init** ()
- void **start** ()
- void **stop** ()
- void **connect** ()
- uint16\_t **getPort** () const
- void **setPort** (int port)
- std::string **getIp** () const
- void **setIp** (const std::string &ip)
- std::shared\_ptr< net::INetwork > **getNetwork** () const
- void **setDebugMode** (bool isDebug)
- bool **isDebugMode** () const
- void **loadNetworkLibrary** ()
- void **loadBufferLibrary** ()
- void **loadPacketLibrary** ()
- void **sendConnectionData** (std::vector< uint8\_t > packet)
- std::string **getName** () const
- void **setName** (const std::string &name)
- uint8\_t **getIdClient** () const

- void **setIdClient** (uint8\_t idClient)
- net::ConnectionState **getConnectionState** () const
- void **eventPacket** (const constants::EventType &eventType, double depth)
- void **disconnectionPacket** ()
- void **connectionPacket** ()
- void **sendReady** ()
- void **sendWhoAml** ()
- void **addToEventQueue** (const [NetworkEvent](#) &event)
- bool **isConnected** () const
- bool **isReady** () const
- void **setResourceManager** (std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void **setGameStateMachine** (std::shared\_ptr< [gsm::IGameStateMachine](#) > gsm)
- std::shared\_ptr< [gsm::IGameStateMachine](#) > **getGameStateMachine** () const
- void **redoServerEndpoint** ()

### Public Attributes

- std::atomic< bool > **\_isConnected**
- std::atomic< bool > **\_ready**

### Protected Member Functions

- std::pair< int, std::chrono::steady\_clock::time\_point > **tryConnection** (const int maxRetries, int retryCount, std::chrono::steady\_clock::time\_point lastRetryTime)
- void **handlePacketType** (uint8\_t type)

### Private Types

- typedef void(ClientNetwork::\* **PacketHandler**) ()
- typedef size\_t(ClientNetwork::\* **ComponentParser**) (const std::vector< uint64\_t > &, size\_t, ecs::Entity)

### Private Member Functions

- void **handleNoOp** ()
- void **handleConnectionAcceptation** ()
- void **handleGameState** ()
- void **handleEndMap** ()
- void **handleEndGame** ()
- void **handleCanStart** ()
- void **handleEntitySpawn** ()
- void **handleEntityDeath** ()
- void **handleWhoAml** ()
- ecs::Entity **findOrCreateNetworkEntity** (std::shared\_ptr< [ecs::Registry](#) > registry, size\_t networkId)
- size\_t **parsePlayerTagComponent** (const std::vector< uint64\_t > &payload, size\_t index, ecs::Entity entityId)
- size\_t **parseTransformComponent** (const std::vector< uint64\_t > &payload, size\_t index, ecs::Entity entityId)
- size\_t **parseSpeedComponent** (const std::vector< uint64\_t > &payload, size\_t index, ecs::Entity entityId)
- size\_t **parseHealthComponent** (const std::vector< uint64\_t > &payload, size\_t index, ecs::Entity entityId)
- size\_t **parseColliderComponent** (const std::vector< uint64\_t > &payload, size\_t index, ecs::Entity entityId)



- `size_t parseShootingStatsComponent` (const std::vector< uint64\_t > &payload, size\_t index, ecs::Entity entityId)
- `size_t parseScoreComponent` (const std::vector< uint64\_t > &payload, size\_t index, ecs::Entity entityId)
- `size_t parseAIMovementPatternComponent` (const std::vector< uint64\_t > &payload, size\_t index, ecs::Entity entityId)
- `size_t parseDamageComponent` (const std::vector< uint64\_t > &payload, size\_t index, ecs::Entity entityId)
- `size_t parseLifetimeComponent` (const std::vector< uint64\_t > &payload, size\_t index, ecs::Entity entityId)
- `size_t parseVelocityComponent` (const std::vector< uint64\_t > &payload, size\_t index, ecs::Entity entityId)
- `size_t parseAIMoverTagComponent` (const std::vector< uint64\_t > &payload, size\_t index, ecs::Entity entityId)
- `size_t parseAIShooterTagComponent` (const std::vector< uint64\_t > &payload, size\_t index, ecs::Entity entityId)
- `size_t parseControllableTagComponent` (const std::vector< uint64\_t > &payload, size\_t index, ecs::Entity entityId)
- `size_t parseEnemyProjectileTagComponent` (const std::vector< uint64\_t > &payload, size\_t index, ecs::Entity entityId)
- `size_t parseGameZoneColliderTagComponent` (const std::vector< uint64\_t > &payload, size\_t index, ecs::Entity entityId)
- `size_t parseMobTagComponent` (const std::vector< uint64\_t > &payload, size\_t index, ecs::Entity entityId)
- `size_t parseObstacleTagComponent` (const std::vector< uint64\_t > &payload, size\_t index, ecs::Entity entityId)
- `size_t parsePlayerProjectileTagComponent` (const std::vector< uint64\_t > &payload, size\_t index, ecs::Entity entityId)
- `size_t parseScoreTagComponent` (const std::vector< uint64\_t > &payload, size\_t index, ecs::Entity entityId)
- `size_t parseShooterTagComponent` (const std::vector< uint64\_t > &payload, size\_t index, ecs::Entity entityId)
- `size_t parseProjectilePassThroughTagComponent` (const std::vector< uint64\_t > &payload, size\_t index, ecs::Entity entityId)
- `size_t parseProjectilePrefabComponent` (const std::vector< uint64\_t > &payload, size\_t index, ecs::Entity entityId)
- `size_t parseNetworkIdComponent` (const std::vector< uint64\_t > &payload, size\_t index, ecs::Entity entityId)
- `size_t parseGameZoneComponent` (const std::vector< uint64\_t > &payload, size\_t index, ecs::Entity entityId)

### Private Attributes

- PacketHandler `_packetHandlers` [constants::MAX\_INDEX\_PACKET\_TYPE]
- std::map< uint64\_t, ComponentParser > `_componentParsers`
- `DLLoader`< createNetworkLib\_t > `_networkloader`
- `DLLoader`< createBuffer\_t > `_bufferloader`
- `DLLoader`< createPacket\_t > `_packetloader`
- std::shared\_ptr< net::INetwork > `_network`
- std::shared\_ptr< IBuffer > `_receptionBuffer`
- std::shared\_ptr< IBuffer > `_sendBuffer`
- std::shared\_ptr< pm::IPacketManager > `_packet`
- std::shared\_ptr< [ResourceManager](#) > `_resourceManager`
- std::shared\_ptr< [gsm::IGameStateMachine](#) > `_gsm`
- uint32\_t `_sequenceNumber`
- uint16\_t `_port`
- std::string `_ip`
- std::string `_name`
- std::vector< std::string > `_clientNames`
- bool `_isDebug`

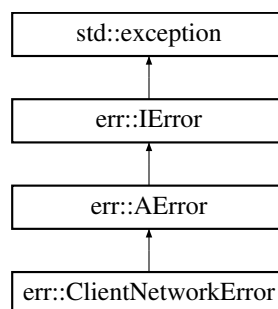
- `uint8_t _idClient`
- `asio::ip::udp::endpoint _serverEndpoint`
- `std::queue< NetworkEvent > _eventQueue`
- `std::mutex _queueMutex`
- `std::condition_variable _queueCond`
- `bool _shouldConnect`

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

- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ClientNetwork.hpp`
- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ClientGameStateConversions.cpp`
- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ClientLibLoading.cpp`
- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ClientNetwork.cpp`
- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ClientReceivedPacket.cpp`
- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ClientSentPacket.cpp`

## 4.28 `err::ClientNetworkError` Class Reference

Inheritance diagram for `err::ClientNetworkError`:



### Public Types

- `enum ErrorCode {`  
`UNKNOWN = 1000 , CONNECTION_FAILED = 1001 , TIMEOUT = 1002 , INVALID_REQUEST = 1003 ,`  
`INTERNAL_ERROR = 1004 , LIBRARY_LOAD_FAILED = 1005 , CONFIG_ERROR = 1006 }`

### Public Member Functions

- `ClientNetworkError` (const std::string &message, ErrorCode code=UNKNOWN)
- `std::string getType ()` const noexcept override

### Public Member Functions inherited from `err::AError`

- `AError` (const std::string &message, int code=0)
- `const char * what ()` const noexcept override
- `int getCode ()` const noexcept override
- `std::string getDetails ()` const noexcept override

## Additional Inherited Members

## Protected Attributes inherited from [err::AError](#)

- `std::string m_message`
- `int m_code`

### 4.28.1 Member Function Documentation

#### 4.28.1.1 `getType()`

```
std::string err::ClientNetworkError::getType () const [override], [virtual], [noexcept]
```

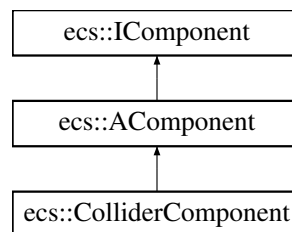
Implements [err::AError](#).

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

- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/ClientNetworkError.hpp`
- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/ClientNetworkError.cpp`

## 4.29 ecs::ColliderComponent Class Reference

Inheritance diagram for `ecs::ColliderComponent`:



## Public Member Functions

- **ColliderComponent** (`math::Vector2f` offset=`math::Vector2f`(0.0f, 0.0f), `math::Vector2f` size=`math::Vector2f`(0.0f, 0.0f), `CollisionType` type=`CollisionType::Solid`)
- `math::Vector2f` **getOffset** () const
- void **setOffset** (`math::Vector2f` offset)
- `math::Vector2f` **getSize** () const
- void **setSize** (`math::Vector2f` size)
- `CollisionType` **getType** () const
- void **setType** (`CollisionType` type)
- `math::FRect` **getHitbox** (`math::Vector2f` entityPosition, `math::Vector2f` scale=`math::Vector2f`(1.0f, 1.0f)) const
- `math::FRect` **getScaledHitbox** (`math::Vector2f` entityPosition, `math::Vector2f` scale) const

### Private Attributes

- [math::Vector2f](#) **\_offset**
- [math::Vector2f](#) **\_size**
- `CollisionType` **\_type**

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

- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/Collider↔  
Component.hpp`

## 4.30 `ecs::CollisionRule` Struct Reference

### Public Attributes

- `std::vector< std::string >` **groupA**
- `std::vector< std::string >` **groupB**

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

- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/CollisionRules/CollisionRulesData.↔  
hpp`

## 4.31 `ecs::CollisionRules` Class Reference

### Public Member Functions

- `bool` **canCollide** (`CollisionType` type, `const std::vector< std::string > &tagsA`, `const std::vector< std::string > &tagsB`) `const`

### Static Public Member Functions

- `static const` [CollisionRules](#) & **getInstance** ()
- `static void` **initWithData** (`const` [CollisionRulesData](#) &data)

### Private Member Functions

- [CollisionRules](#) (`const` [CollisionRules](#) &)=delete
- [CollisionRules](#) & **operator=** (`const` [CollisionRules](#) &)=delete
- `const std::vector<` [CollisionRule](#) `> &` **getAllowRules** (`CollisionType` type) `const`
- `bool` **entityMatchesGroup** (`const std::vector< std::string > &entityTags`, `const std::vector< std::string > &group`) `const`
- `bool` **ruleMatches** (`const` [CollisionRule](#) &rule, `const std::vector< std::string > &tagsA`, `const std::vector< std::string > &tagsB`) `const`

### Private Attributes

- std::shared\_ptr< std::vector< [CollisionRule](#) > > **\_solidAllowRules**
- std::shared\_ptr< std::vector< [CollisionRule](#) > > **\_triggerAllowRules**
- std::shared\_ptr< std::vector< [CollisionRule](#) > > **\_pushAllowRules**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/CollisionRules/CollisionRules.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/CollisionRules/CollisionRules.cpp

## 4.32 ecs::CollisionRulesData Struct Reference

### Public Attributes

- std::shared\_ptr< std::vector< [CollisionRule](#) > > **solidAllowRules**
- std::shared\_ptr< std::vector< [CollisionRule](#) > > **triggerAllowRules**
- std::shared\_ptr< std::vector< [CollisionRule](#) > > **pushAllowRules**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/CollisionRules/CollisionRulesData.↵  
hpp

## 4.33 ecs::CollisionRulesParser Class Reference

### Static Public Member Functions

- static [CollisionRulesData](#) **parseFromFile** (const std::string &filePath)
- static [CollisionRulesData](#) **parseFromJsonString** (const std::string &jsonString)

### Static Private Member Functions

- static void **parseRulesForType** (const nlohmann::json &typeJson, std::shared\_ptr< std::vector< [CollisionRule](#) > > allowRules)

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/CollisionRulesParser.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/CollisionRulesParser.cpp

## 4.34 rserv::ComponentDeltaTracker Class Reference

### Public Member Functions

- `std::vector< uint64_t > createEntityDelta` (uint8\_t clientId, uint32\_t entityId, const [EntitySnapshot](#) &currentSnapshot)
- `std::vector< uint64_t > createMultiEntityDelta` (uint8\_t clientId, const std::vector< [EntitySnapshot](#) > &entities)
- `EntitySnapshot applyDelta` (uint8\_t clientId, const std::vector< uint64\_t > &deltaPayload)
- `void clearClientCache` (uint8\_t clientId)
- `void clearEntityCache` (uint8\_t clientId, uint32\_t entityId)
- `void clearAllCaches` ()

### Private Member Functions

- `std::vector< uint64_t > serializeFullSnapshot` (uint32\_t entityId, const [EntitySnapshot](#) &snapshot)
- `std::vector< uint64_t > serializeDelta` (uint32\_t entityId, uint32\_t changedMask, const std::map< uint8\_t, std::vector< uint64\_t > > &changedComponents)

### Private Attributes

- `std::unordered_map< uint8_t, std::unordered_map< uint32_t, EntitySnapshot > > _clientEntityCache`

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/deltaTracker/ComponentDeltaTracker.↵  
hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/deltaTracker/ComponentDeltaTracker.↵  
cpp

## 4.35 rserv::ComponentSerializer Class Reference

### Static Public Member Functions

- `static std::vector< uint64_t > serializePosition` (uint32\_t x, uint32\_t y)
- `static void deserializePosition` (const std::vector< uint64\_t > &data, uint32\_t &x, uint32\_t &y)
- `static std::vector< uint64_t > serializeVelocity` (int32\_t vx, int32\_t vy)
- `static void deserializeVelocity` (const std::vector< uint64\_t > &data, int32\_t &vx, int32\_t &vy)
- `static std::vector< uint64_t > serializeHealth` (uint32\_t current, uint32\_t max)
- `static void deserializeHealth` (const std::vector< uint64\_t > &data, uint32\_t &current, uint32\_t &max)
- `static std::vector< uint64_t > serializeCollider` (uint32\_t x, uint32\_t y, uint32\_t width, uint32\_t height, uint32\_t rotation)
- `static std::vector< uint64_t > serializeShootingStats` (uint32\_t fireRate, uint32\_t damage, uint32\_t lastShot)
- `static std::vector< uint64_t > serializeScore` (uint64\_t score)
- `static std::vector< uint64_t > serializeDamage` (uint32\_t damage)
- `static std::vector< uint64_t > serializeLifetime` (uint64\_t lifetime)
- `static std::vector< uint64_t > serializeSpeed` (uint64\_t speed)
- `static std::vector< uint64_t > serializeAIMovementPattern` (uint32\_t patternId)
- `static std::vector< uint64_t > serializeNetworkId` (uint32\_t networkId)
- `static std::vector< uint64_t > serializeGameZone` (uint32\_t x, uint32\_t y, uint32\_t width, uint32\_t height)

- static [EntitySnapshot](#) **createSnapshotFromComponents** (uint32\_t entityId, const std::vector< uint64\_t > &componentData)
- static std::vector< uint64\_t > **snapshotToComponentData** (const [EntitySnapshot](#) &snapshot)
- static bool **isTagComponent** (uint8\_t component)
- static bool **isOneParamComponent** (uint8\_t component)

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/deltaTracker/ComponentSerializer.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/deltaTracker/ComponentSerializer.cpp

## 4.36 ComposantParser Class Reference

### Public Types

- using **ShouldParseComponentCallback** = std::function<bool(const std::map<std::string, std::shared\_ptr<[FieldValue](#)>>>&)>

### Public Member Functions

- **ComposantParser** (std::shared\_ptr< const std::map< std::string, std::pair< std::type\_index, std::vector< [Field](#) > > > > componentDefinitions, const std::map< std::type\_index, ComponentCreator > &componentCreators, const ShouldParseComponentCallback &shouldParseCallback=nullptr)
- std::pair< std::shared\_ptr< [ecs::IComponent](#) >, std::type\_index > **parseComponent** (const std::string &componentName, const nlohmann::json &componentData)

### Private Member Functions

- std::shared\_ptr< [FieldValue](#) > **parseFieldValue** (const nlohmann::json &jsonValue, FieldType type)

### Private Attributes

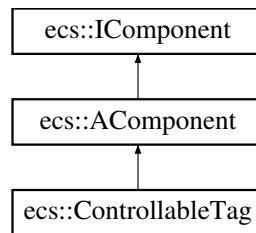
- std::shared\_ptr< const std::map< std::string, std::pair< std::type\_index, std::vector< [Field](#) > > > > **\_componentDefinitions**
- const std::map< std::type\_index, ComponentCreator > & **\_componentCreators**
- ShouldParseComponentCallback **\_shouldParseCallback**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/ComposantParser/ComposantParser.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/ComposantParser/ComposantParser.cpp

## 4.37 ecs::ControllableTag Class Reference

Inheritance diagram for ecs::ControllableTag:



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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/tags/ControllableTag.hpp

## 4.38 Core Class Reference

### Public Member Functions

- void **initFirstScene** ()
- void **run** ()
- void **startNetwork** ()
- std::shared\_ptr< [ClientNetwork](#) > **getNetwork** ()
- void **init** ()
- void **loop** ()
- std::shared\_ptr< [rserv::Server](#) > **getServer** () const
- std::shared\_ptr< [rserv::ServerConfig](#) > **getConfig** () const
- std::shared\_ptr< [ResourceManager](#) > **getResourceManager** () const
- std::shared\_ptr< [ecs::Registry](#) > **getRegistry** () const
- std::shared\_ptr< [Parser](#) > **getParser** () const
- std::shared\_ptr< [ecs::ISystemManager](#) > **getSystemsManager** () const
- std::shared\_ptr< [gsm::GameStateMachine](#) > **getGameStateMachine** () const
- void **processServerEvents** ()

### Private Member Functions

- void **initNetwork** ()
- void **initLibraries** ()
- void **networkLoop** ()



**Private Attributes**

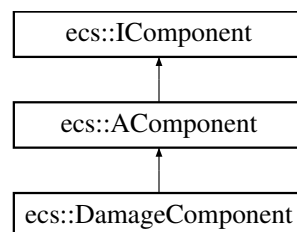
- std::shared\_ptr< [DLLoader](#)< gfx::createWindow\_t > > **\_windowLoader**
- std::shared\_ptr< [DLLoader](#)< gfx::createEvent\_t > > **\_eventLoader**
- std::shared\_ptr< [DLLoader](#)< gfx::createAudio\_t > > **\_audioLoader**
- std::shared\_ptr< [ResourceManager](#) > **\_resourceManager**
- std::shared\_ptr< [gsm::GameStateMachine](#) > **\_gsm**
- std::shared\_ptr< [ecs::Registry](#) > **\_registry**
- std::shared\_ptr< [ClientNetwork](#) > **\_clientNetwork**
- std::shared\_ptr< [Parser](#) > **\_parser**
- std::thread **\_networkThread**
- std::shared\_ptr< [Utils](#) > **\_utils**
- std::shared\_ptr< [rserv::Server](#) > **\_server**
- std::thread **\_serverThread**
- std::shared\_ptr< [ecs::ISystemManager](#) > **\_systemsManager**
- std::shared\_ptr< [ecs::ServerInputProvider](#) > **\_inputProvider**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/Core.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/Core.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/Core.cpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/Core.cpp

## 4.39 ecs::DamageComponent Class Reference

Inheritance diagram for ecs::DamageComponent:

**Public Member Functions**

- **DamageComponent** (float damage=0.0f)
- float **getDamage** () const
- void **setDamage** (float damage)

**Private Attributes**

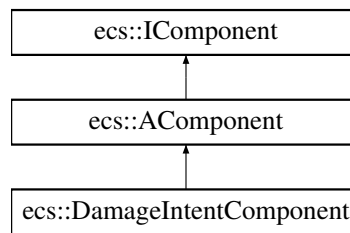
- float **\_damage**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/DamageComponent.hpp

## 4.40 ecs::DamageIntentComponent Class Reference

Inheritance diagram for ecs::DamageIntentComponent:



### Public Member Functions

- **DamageIntentComponent** (float damages=0.0f, ecs::Entity source=0)
- float **getDamages** ()
- void **setDamages** (float damages)
- ecs::Entity **getSource** () const
- void **setSource** (ecs::Entity source)

### Private Attributes

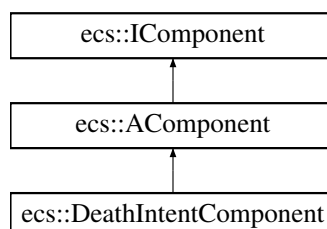
- float **\_damages**
- ecs::Entity **\_source**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/temporary/DamageIntentComponent.hpp

## 4.41 ecs::DeathIntentComponent Class Reference

Inheritance diagram for ecs::DeathIntentComponent:



### Public Member Functions

- **DeathIntentComponent** (ecs::Entity source=0)
- ecs::Entity **getSource** () const
- void **setSource** (ecs::Entity source)

**Private Attributes**

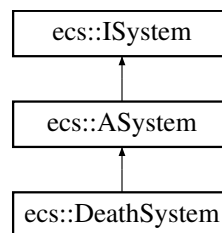
- ecs::Entity **\_source**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/temporary/DeathIntentComponent.hpp

## 4.42 ecs::DeathSystem Class Reference

Inheritance diagram for ecs::DeathSystem:

**Public Member Functions**

- void **update** (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

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

- void **updateSystem** (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

**Private Member Functions**

- [math::Vector2f](#) **getFirstHitboxCenter** (std::shared\_ptr< [Registry](#) > registry, ecs::Entity entity)
- void **spawnExplosionAtMobCenter** (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, ecs::Entity mobEntity, const std::string &prefabName)

### 4.42.1 Member Function Documentation

#### 4.42.1.1 update()

```

void ecs::DeathSystem::update (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [virtual]
  
```

Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/death/DeathSystem.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/death/DeathSystem.cpp

## 4.43 debug::Debug Class Reference

### Static Public Member Functions

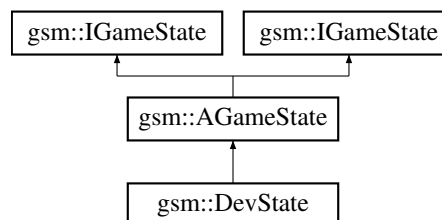
- static void **printDebug** (const bool isDebug, const std::string &message, debugType type, debugLevel level)

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/debug.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/debug.cpp

## 4.44 gsm::DevState Class Reference

Inheritance diagram for gsm::DevState:



### Public Member Functions

- **DevState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void **enter** () override
- void **update** (float deltaTime) override
- void **exit** () override

### Public Member Functions inherited from gsm::AGameState

- **AGameState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- std::vector< std::shared\_ptr< [ecs::ISystem](#) > > **getSystems** () const override
- **AGameState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- std::vector< std::shared\_ptr< [ecs::ISystem](#) > > **getSystems** () const override

### Private Member Functions

- void **renderHUD** ()
- void **drawHealthHUD** (std::shared\_ptr< gfx::IWindow > window, float health, float maxHealth)
- void **drawScoreHUD** (std::shared\_ptr< gfx::IWindow > window, int score)

### Private Attributes

- std::shared\_ptr< [ecs::Registry](#) > **\_registry**
- std::shared\_ptr< [EntityPrefabManager](#) > **\_prefabManager**
- std::shared\_ptr< [Parser](#) > **\_parser**
- int **\_previousScore**
- int **\_previousHealth**
- std::vector< [ScoreFeedback](#) > **\_scoreFeedbacks**
- std::vector< [ScoreFeedback](#) > **\_healthFeedbacks**

### Additional Inherited Members

### Protected Member Functions inherited from [gsm::AGameState](#)

- void [addSystem](#) (std::shared\_ptr< [ecs::ISystem](#) > system) override
- void [addSystem](#) (std::shared\_ptr< [ecs::ISystem](#) > system) override

### Protected Attributes inherited from [gsm::AGameState](#)

- std::weak\_ptr< [IGameStateMachine](#) > **\_gsm**
- std::shared\_ptr< [ResourceManager](#) > **\_resourceManager**
- std::vector< std::shared\_ptr< [ecs::ISystem](#) > > **\_systems**

## 4.44.1 Member Function Documentation

### 4.44.1.1 enter()

```
void gsm::DevState::enter () [override], [virtual]
```

Reimplemented from [gsm::AGameState](#).

### 4.44.1.2 exit()

```
void gsm::DevState::exit () [override], [virtual]
```

Reimplemented from [gsm::AGameState](#).

### 4.44.1.3 update()

```
void gsm::DevState::update (
    float deltaTime) [override], [virtual]
```

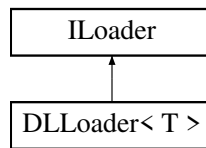
Reimplemented from [gsm::AGameState](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Dev/DevState.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Dev/DevState.cpp

## 4.45 DLLoader< T > Class Template Reference

Inheritance diagram for DLLoader< T >:



### Public Member Functions

- void \* [getHandler](#) () const override
- void \* [Open](#) (const char \*path, int flag=RTLD\_LAZY) override
- void \* [Symbol](#) (const char \*symbolName) override
- T [getSymbol](#) (const char \*symbolName)
- int [Close](#) () override
- const char \* [Error](#) () override

### Private Attributes

- void \* [\\_handler](#) = nullptr

### 4.45.1 Member Function Documentation

#### 4.45.1.1 Close()

```
template<typename T>
int DLLoader< T >::Close () [inline], [override], [virtual]
```

Implements [ILoader](#).

#### 4.45.1.2 Error()

```
template<typename T>
const char * DLLoader< T >::Error () [inline], [override], [virtual]
```

Implements [ILoader](#).

#### 4.45.1.3 getHandler()

```
template<typename T>
void * DLLoader< T >::getHandler () const [inline], [override], [virtual]
```

Implements [ILoader](#).

## 4.45.1.4 Open()

```
template<typename T>
void * DLLoader< T >::Open (
    const char * path,
    int flag = RTLD_LAZY) [inline], [override], [virtual]
```

Implements [ILoader](#).

## 4.45.1.5 Symbol()

```
template<typename T>
void * DLLoader< T >::Symbol (
    const char * symbolName) [inline], [override], [virtual]
```

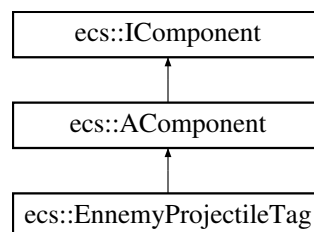
Implements [ILoader](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/DLLoader/DLLoader.hpp

## 4.46 ecs::EnemyProjectileTag Class Reference

Inheritance diagram for ecs::EnemyProjectileTag:



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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/tags/EnemyProjectileTag.hpp

## 4.47 ecs::EntityCreationContext Struct Reference

## Public Member Functions

- bool **shouldHaveNetworkId** () const

### Static Public Member Functions

- static [EntityCreationContext](#) **forServer** (std::optional< size\_t > existingId=std::nullopt)
- static [EntityCreationContext](#) **forNetworkSync** (size\_t serverNetworkId)
- static [EntityCreationContext](#) **forLocalClient** ()

### Public Attributes

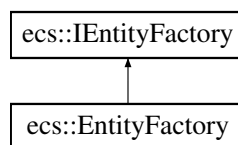
- EntityCreationOrigin **origin** = EntityCreationOrigin::CLIENT\_LOCAL
- std::optional< size\_t > **networkId** = std::nullopt

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/entity/EntityCreationContext.hpp

## 4.48 ecs::EntityFactory Class Reference

Inheritance diagram for ecs::EntityFactory:



### Public Member Functions

- **EntityFactory** (size\_t startingNetworkId=1)
- Entity [createEntity](#) (const std::shared\_ptr< [Registry](#) > &registry, const [EntityCreationContext](#) &context=EntityCreationContext::forLocalClient()) override
- size\_t [getNextNetworkId](#) () const override
- void [setNextNetworkId](#) (size\_t nextId) override

### Private Member Functions

- size\_t [resolveNetworkId](#) (const [EntityCreationContext](#) &context)

### Private Attributes

- std::atomic< size\_t > **\_nextNetworkId**
- std::atomic< size\_t > **\_nextLocalId**



## 4.48.1 Member Function Documentation

### 4.48.1.1 createEntity()

```
Entity ecs::EntityFactory::createEntity (
    const std::shared_ptr< Registry > & registry,
    const EntityCreationContext & context = EntityCreationContext::forLocalClient())
[override], [virtual]
```

Implements [ecs::IEntityFactory](#).

### 4.48.1.2 getNextNetworkId()

```
size_t ecs::EntityFactory::getNextNetworkId () const [override], [virtual]
```

Implements [ecs::IEntityFactory](#).

### 4.48.1.3 setNextNetworkId()

```
void ecs::EntityFactory::setNextNetworkId (
    size_t nextId) [override], [virtual]
```

Implements [ecs::IEntityFactory](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/entity/factory/EntityFactory.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/entity/factory/EntityFactory.cpp

## 4.49 EntityParser Class Reference

### Public Types

- using **ShouldParseComponentCallback** = ComposantParser::ShouldParseComponentCallback

### Public Member Functions

- **EntityParser** (std::shared\_ptr< const std::map< std::string, std::pair< std::type\_index, std::vector< [Field](#) > > > componentDefinitions, const std::map< std::type\_index, ComponentCreator > &componentCreators, const std::map< std::type\_index, ComponentAdder > &componentAdders, const ShouldParseComponent↵ Callback &shouldParseCallback=nullptr)
- std::shared\_ptr< [IPrefab](#) > **parseEntity** (const std::string &filePath)

### Private Attributes

- [ComposantParser](#) **\_composantParser**
- `const std::map< std::type_index, ComponentAdder > & _componentAdders`
- `ShouldParseComponentCallback _shouldParseCallback`

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

- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/EntityParser/EntityParser.hpp`
- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/EntityParser/EntityParser.cpp`

## 4.50 EntityPrefabManager Class Reference

### Public Member Functions

- `void registerPrefab (const std::string &name, const std::shared_ptr< IPrefab > &prefab)`
- `std::shared_ptr< IPrefab > getPrefab (const std::string &name) const`
- `ecs::Entity createEntityFromPrefab (const std::string &prefabName, const std::shared_ptr< ecs::Registry > &registry, const ecs::EntityCreationContext &context)`
- `ecs::Entity createEntityFromPrefab (const std::string &prefabName, const std::shared_ptr< ecs::Registry > &registry)`
- `bool hasPrefab (const std::string &name) const`
- `void deletePrefab (const std::string &name)`
- `void clearPrefabs ()`
- `std::shared_ptr< ecs::IEntityFactory > getEntityFactory () const`
- `void setEntityFactory (std::shared_ptr< ecs::IEntityFactory > factory)`
- `void setOnEntityCreated (std::function< void(ecs::Entity, const std::string &)> callback)`

### Private Attributes

- `std::map< std::string, std::shared_ptr< IPrefab > > _prefabs`
- `std::shared_ptr< ecs::IEntityFactory > _entityFactory`
- `std::function< void(ecs::Entity, const std::string &)> _onEntityCreated`

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

- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Prefab/entityPrefabManager/Entity↔PrefabManager.hpp`
- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Prefab/entityPrefabManager/Entity↔PrefabManager.cpp`

## 4.51 rserv::EntitySnapshot Struct Reference

### Public Attributes

- `uint32_t entityId`
- `uint32_t componentMask`
- `std::map< uint8_t, std::vector< uint64_t > > components`

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

- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/deltaTracker/ComponentDeltaTracker.↔hpp`

## 4.52 Field Struct Reference

### Public Member Functions

- **Field** (std::string n, FieldType t, bool opt=false, std::shared\_ptr< [FieldValue](#) > def=nullptr)

### Public Attributes

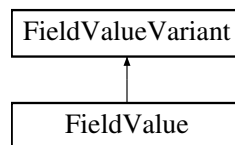
- std::string **name** = ""
- FieldType **type**
- bool **optional** = false
- std::shared\_ptr< [FieldValue](#) > **defaultValue** = nullptr

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/ParserParam.hpp

## 4.53 FieldValue Struct Reference

Inheritance diagram for FieldValue:



### Public Member Functions

- template<typename T>  
**FieldValue** (T &&value)

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/ParserParam.hpp

## 4.54 math::FRect Class Reference

### Public Member Functions

- **FRect** (float left, float top, float width, float height)
- **FRect** ([FRect](#) const &other)
- float **getLeft** () const
- void **setLeft** (float left)
- float **getTop** () const
- void **setTop** (float top)
- float **getWidth** () const
- void **setWidth** (float width)
- float **getHeight** () const
- void **setHeight** (float height)
- bool **contains** (float x, float y) const
- bool **intersects** ([FRect](#) const &other) const
- bool **intersects** ([FRect](#) const &other, [FRect](#) &intersection) const
- [FRect](#) & **operator=** ([FRect](#) const &other)
- bool **operator==** ([FRect](#) const &other) const
- bool **operator!=** ([FRect](#) const &other) const

### Private Attributes

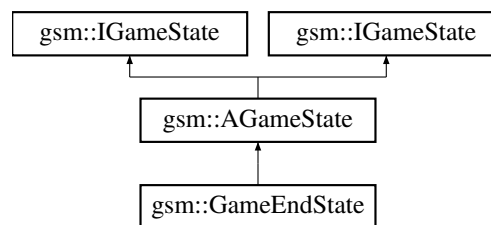
- float **left**
- float **top**
- float **width**
- float **height**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/types/FRect.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/types/FRect.cpp

## 4.55 `gsm::GameEndState` Class Reference

Inheritance diagram for `gsm::GameEndState`:



### Public Member Functions

- **GameEndState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void [enter](#) () override

### Public Member Functions inherited from `gsm::AGameState`

- **AGameState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void [update](#) (float deltaTime) override
- void [exit](#) () override
- std::vector< std::shared\_ptr< [ecs::ISystem](#) > > [getSystems](#) () const override
- **AGameState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void [update](#) (float deltaTime) override
- void [exit](#) () override
- std::vector< std::shared\_ptr< [ecs::ISystem](#) > > [getSystems](#) () const override

### Additional Inherited Members

### Protected Member Functions inherited from `gsm::AGameState`

- void [addSystem](#) (std::shared\_ptr< [ecs::ISystem](#) > system) override
- void [addSystem](#) (std::shared\_ptr< [ecs::ISystem](#) > system) override

## Protected Attributes inherited from `gsm::AGameState`

- `std::weak_ptr< IGameStateMachine > _gsm`
- `std::shared_ptr< ResourceManager > _resourceManager`
- `std::vector< std::shared_ptr< ecs::ISystem > > _systems`

### 4.55.1 Member Function Documentation

#### 4.55.1.1 `enter()`

```
void gsm::GameEndState::enter () [override], [virtual]
```

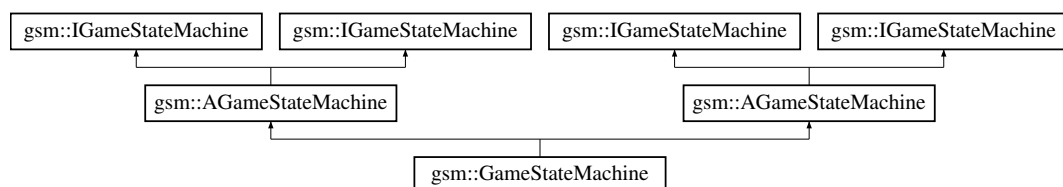
Reimplemented from `gsm::AGameState`.

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

- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/scenes/GameEnd/GameEndState.hpp`
- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/scenes/GameEnd/GameEndState.cpp`

## 4.56 `gsm::GameStateMachine` Class Reference

Inheritance diagram for `gsm::GameStateMachine`:



### Public Member Functions

- void `requestStateChange` (`std::shared_ptr< IGameState > newState`) override
- void `requestStatePush` (`std::shared_ptr< IGameState > newState`) override
- void `requestStatePop` () override

### Public Member Functions inherited from `gsm::AGameStateMachine`

- void `changeState` (`std::shared_ptr< IGameState > newState`) override
- void `pushState` (`std::shared_ptr< IGameState > newState`) override
- void `popState` () override
- void `update` (`float deltaTime`) override
- void `changeState` (`std::shared_ptr< IGameState > newState`) override
- void `pushState` (`std::shared_ptr< IGameState > newState`) override
- void `popState` () override
- void `update` (`float deltaTime`) override

## Additional Inherited Members

### Protected Attributes inherited from [gsm::AGameStateMachine](#)

- `std::stack< std::shared_ptr< IGameState > > _states`
- `std::shared_ptr< IGameState > _pendingChangeState`
- `std::shared_ptr< IGameState > _pendingPushState`
- `bool _pendingPopState = false`

## 4.56.1 Member Function Documentation

### 4.56.1.1 requestStateChange()

```
void gsm::GameStateMachine::requestStateChange (
    std::shared_ptr< IGameState > newState) [override], [virtual]
```

Reimplemented from [gsm::AGameStateMachine](#).

### 4.56.1.2 requestStatePop()

```
void gsm::GameStateMachine::requestStatePop () [override], [virtual]
```

Reimplemented from [gsm::AGameStateMachine](#).

### 4.56.1.3 requestStatePush()

```
void gsm::GameStateMachine::requestStatePush (
    std::shared_ptr< IGameState > newState) [override], [virtual]
```

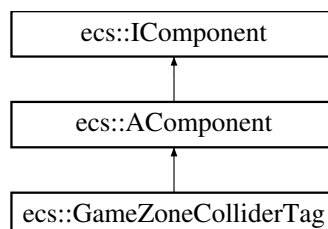
Reimplemented from [gsm::AGameStateMachine](#).

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

- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/machine/GameStateMachine.hpp`
- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/machine/GameStateMachine.hpp`
- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/machine/GameStateMachine.cpp`
- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/machine/GameStateMachine.cpp`

## 4.57 ecs::GameZoneColliderTag Class Reference

Inheritance diagram for `ecs::GameZoneColliderTag`:

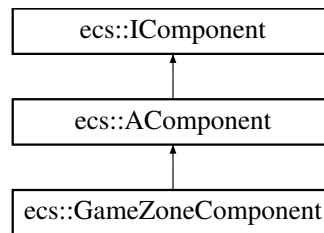


The documentation for this class was generated from the following file:

- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/tags/GameZoneColliderTag.hpp`

## 4.58 ecs::GameZoneComponent Class Reference

Inheritance diagram for ecs::GameZoneComponent:



### Public Member Functions

- **GameZoneComponent** ([math::FRect](#) zone=[math::FRect](#)(0.0f, 0.0f, constants::MAX\_WIDTH, constants::MAX\_HEIGHT))
- [math::FRect](#) **getZone** () const
- void **setZone** ([math::FRect](#) zone)

### Private Attributes

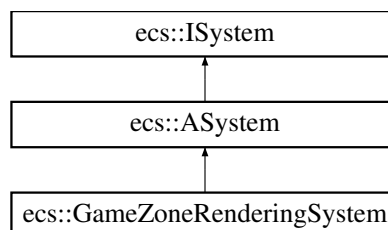
- [math::FRect](#) **\_zone**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/GameZoneComponent.hpp

## 4.59 ecs::GameZoneRenderingSystem Class Reference

Inheritance diagram for ecs::GameZoneRenderingSystem:



### Protected Member Functions

- void **update** (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

## Additional Inherited Members

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

- void [updateSystem](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

### 4.59.1 Member Function Documentation

#### 4.59.1.1 update()

```
void ecs::GameZoneRenderingSystem::update (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [protected], [virtual]
```

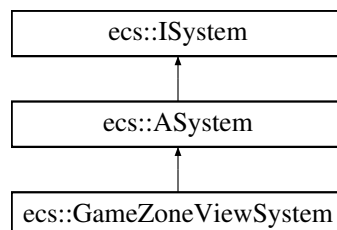
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/GameZoneRenderingSystem.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/GameZoneRenderingSystem.cpp

## 4.60 [ecs::GameZoneViewSystem](#) Class Reference

Inheritance diagram for [ecs::GameZoneViewSystem](#):



## Protected Member Functions

- void [update](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

## Additional Inherited Members

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

- void [updateSystem](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override



## 4.60.1 Member Function Documentation

### 4.60.1.1 update()

```
void ecs::GameZoneViewSystem::update (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [protected], [virtual]
```

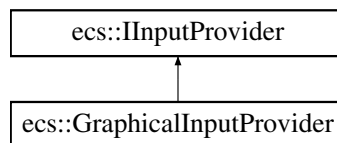
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/GameZoneView↵  
System.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/GameZoneView↵  
System.cpp

## 4.61 ecs::GraphicalInputProvider Class Reference

Inheritance diagram for ecs::GraphicalInputProvider:



### Public Member Functions

- **GraphicalInputProvider** (std::shared\_ptr< gfx::IEvent > eventSystem, std::shared\_ptr< [InputMappingManager](#) > mappingManager)
- float [getAxisValue](#) (event\_t axis, size\_t clientID=0) override
- bool [isActionPressed](#) (InputAction action, size\_t clientID=0) override
- float [getActionAxis](#) (InputAction action, size\_t clientID=0) override
- [InputMapping](#) [getInputMapping](#) (size\_t clientID=0) const override
- void **setToggleMode** (bool enabled)
- bool **isToggleMode** () const

### Private Attributes

- std::shared\_ptr< gfx::IEvent > **\_eventSystem**
- std::shared\_ptr< [InputMappingManager](#) > **\_mappingManager**
- bool **\_toggleMode**
- std::map< InputAction, bool > **\_toggledStates**
- std::map< InputAction, bool > **\_lastKeyState**
- std::map< std::pair< InputAction, gfx::EventType >, bool > **\_keyPressedState**
- std::map< std::pair< InputAction, gfx::EventType >, bool > **\_toggledKeyStates**
- std::map< std::pair< InputAction, gfx::EventType >, int > **\_lastToggleFrame**
- int **\_currentFrame**

## Additional Inherited Members

## Public Types inherited from [ecs::IInputProvider](#)

- using **event\_t** = gfx::EventType

## 4.61.1 Member Function Documentation

### 4.61.1.1 `getActionAxis()`

```
float ecs::GraphicalInputProvider::getActionAxis (
    InputAction action,
    size_t clientID = 0) [override], [virtual]
```

Implements [ecs::IInputProvider](#).

### 4.61.1.2 `getAxisValue()`

```
float ecs::GraphicalInputProvider::getAxisValue (
    event_t axis,
    size_t clientID = 0) [override], [virtual]
```

Implements [ecs::IInputProvider](#).

### 4.61.1.3 `getInputMapping()`

```
InputMapping ecs::GraphicalInputProvider::getInputMapping (
    size_t clientID = 0) const [override], [virtual]
```

Implements [ecs::IInputProvider](#).

### 4.61.1.4 `isActionPressed()`

```
bool ecs::GraphicalInputProvider::isActionPressed (
    InputAction action,
    size_t clientID = 0) [override], [virtual]
```

Implements [ecs::IInputProvider](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/initResourcesManager/GraphicalInputProvider.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/initResourcesManager/GraphicalInputProvider.cpp

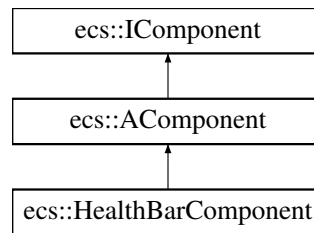
## 4.62 ecs::Group< Components > Class Template Reference

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/entity/registry/Registry.hpp

## 4.63 ecs::HealthBarComponent Class Reference

Inheritance diagram for ecs::HealthBarComponent:

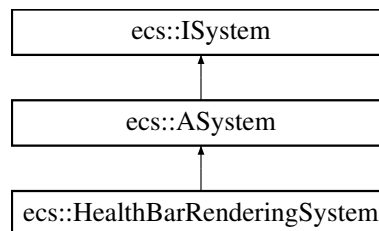


The documentation for this class was generated from the following file:

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/HealthBarComponent.hpp

## 4.64 ecs::HealthBarRenderingSystem Class Reference

Inheritance diagram for ecs::HealthBarRenderingSystem:



### Protected Member Functions

- void [update](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

### Additional Inherited Members

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

- void [updateSystem](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

### 4.64.1 Member Function Documentation

#### 4.64.1.1 update()

```
void ecs::HealthBarRenderingSystem::update (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [protected], [virtual]
```

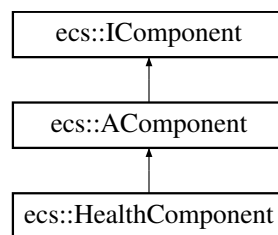
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/HealthBarRendering↵  
System.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/HealthBarRendering↵  
System.cpp

## 4.65 ecs::HealthComponent Class Reference

Inheritance diagram for ecs::HealthComponent:



### Public Member Functions

- **HealthComponent** (float health=100)
- float **getHealth** () const
- void **setHealth** (float health)
- void **decreaseHealth** (float quantity)
- float **getBaseHealth** () const
- void **setBaseHealth** (float health)
- ecs::Entity **getLastDamageSource** () const
- void **setLastDamageSource** (ecs::Entity source)

### Private Attributes

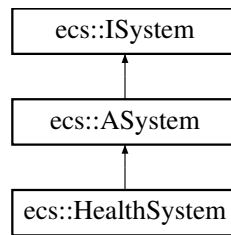
- float **\_health**
- float **\_baseHealth**
- ecs::Entity **\_lastDamageSource**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/Health↵  
Component.hpp

## 4.66 ecs::HealthSystem Class Reference

Inheritance diagram for ecs::HealthSystem:



### Public Member Functions

- void [update](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

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

- void [updateSystem](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

### Private Member Functions

- void [\\_handleDamageUpdates](#) (std::shared\_ptr< [Registry](#) > registry)
- void [\\_handleHealthUpdates](#) (std::shared\_ptr< [Registry](#) > registry)

### 4.66.1 Member Function Documentation

#### 4.66.1.1 update()

```
void ecs::HealthSystem::update (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [virtual]
```

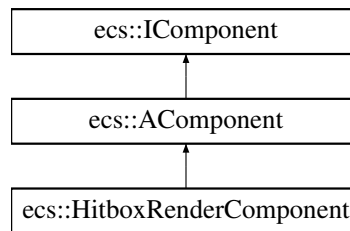
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/health/HealthSystem.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/health/HealthSystem.cpp

## 4.67 ecs::HitboxRenderComponent Class Reference

Inheritance diagram for ecs::HitboxRenderComponent:



### Public Member Functions

- **HitboxRenderComponent** (gfx::color\_t color, float outlineThickness=1.0f)
- const gfx::color\_t & **getColor** () const
- void **setColor** (const gfx::color\_t &color)
- float **getOutlineThickness** () const
- void **setOutlineThickness** (float thickness)

### Private Attributes

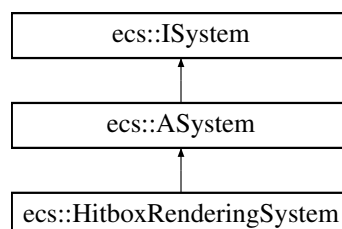
- gfx::color\_t **\_color**
- float **\_outlineThickness**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/HitboxRenderComponent.hpp↔

## 4.68 ecs::HitboxRenderingSystem Class Reference

Inheritance diagram for ecs::HitboxRenderingSystem:



### Protected Member Functions

- void **update** (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

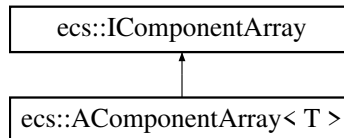


The documentation for this class was generated from the following file:

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/base/IComponent.hpp

## 4.70 ecs::IComponentArray Class Reference

Inheritance diagram for ecs::IComponentArray:



### Public Member Functions

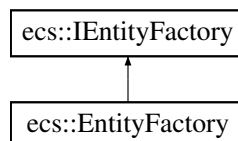
- virtual Entity **getMaxEntityId** () const =0
- virtual void **removeComponents** (Entity entityId)=0
- virtual void **removeOneComponent** (Entity entityId)=0

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/entity/componentArray/IComponentArray.hpp

## 4.71 ecs::IEntityFactory Class Reference

Inheritance diagram for ecs::IEntityFactory:



### Public Member Functions

- virtual Entity **createEntity** (const std::shared\_ptr< Registry > &registry, const EntityCreationContext &context=EntityCreationContext::forLocalClient())=0
- virtual size\_t **getNextNetworkId** () const =0
- virtual void **setNextNetworkId** (size\_t nextId)=0

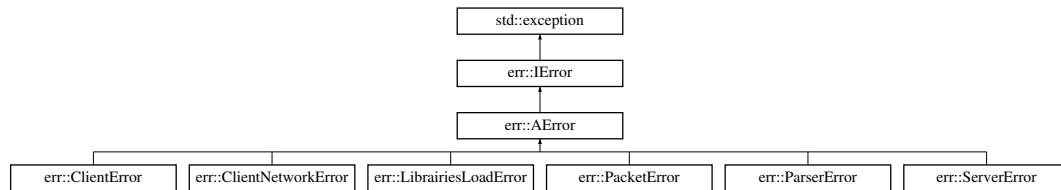
The documentation for this class was generated from the following file:

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/entity/factory/IEntityFactory.hpp



## 4.72 err::IError Class Reference

Inheritance diagram for err::IError:



### Public Member Functions

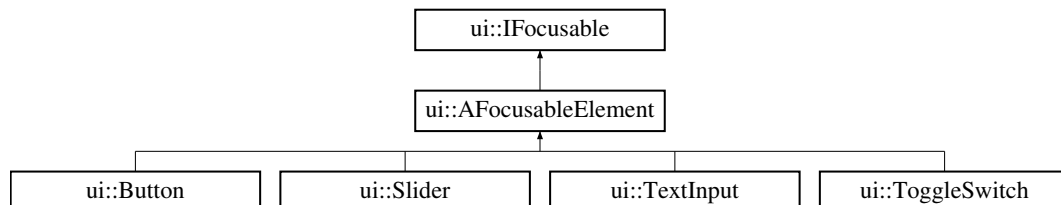
- virtual const char \* **what** () const noexcept override=0
- virtual int **getCode** () const noexcept=0
- virtual std::string **getType** () const noexcept=0
- virtual std::string **getDetails** () const noexcept=0

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/IError.hpp

## 4.73 ui::IFocusable Class Reference

Inheritance diagram for ui::IFocusable:



### Public Member Functions

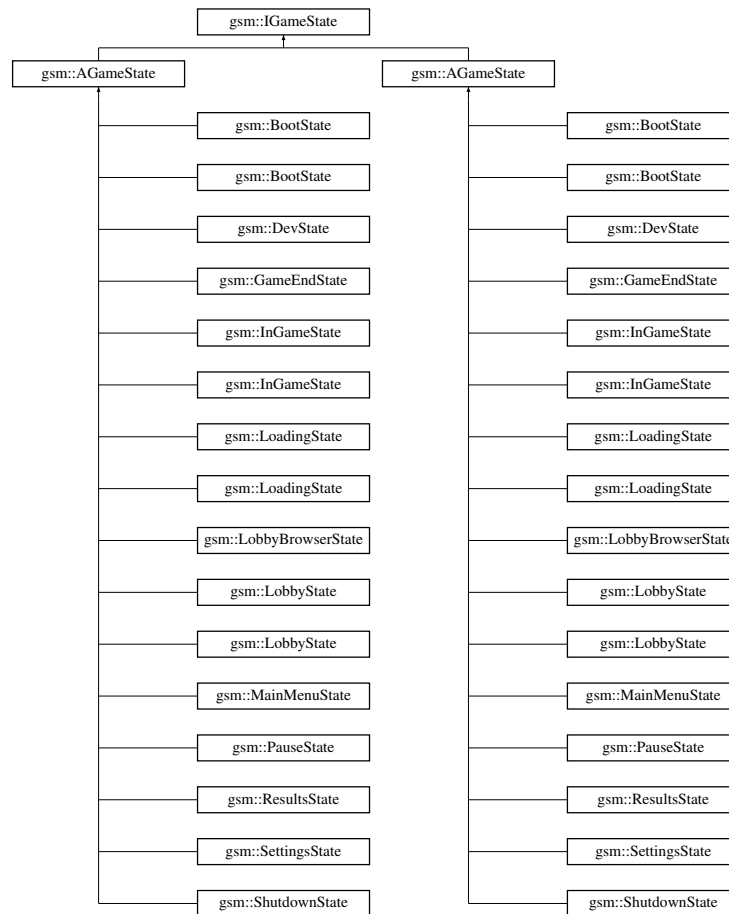
- virtual void **setFocused** (bool focused)=0
- virtual bool **isFocused** () const =0
- virtual bool **canBeFocused** () const =0
- virtual void **onFocusGained** ()=0
- virtual void **onFocusLost** ()=0
- virtual void **onActivated** ()=0
- virtual bool **onNavigateLeft** ()
- virtual bool **onNavigateRight** ()

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/core/IFocusable.hpp

## 4.74 gsm::IGameState Class Reference

Inheritance diagram for gsm::IGameState:



### Public Member Functions

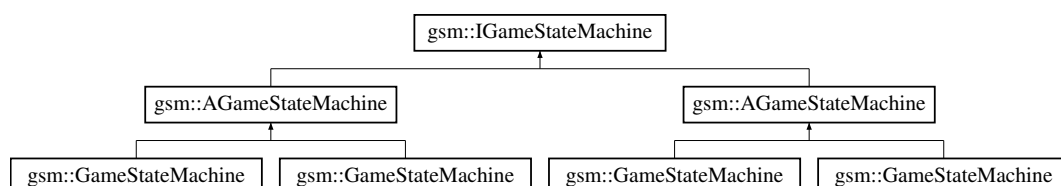
- virtual void **enter** ()=0
- virtual void **update** (float deltaTime)=0
- virtual void **exit** ()=0
- virtual void **addSystem** (std::shared\_ptr< [ecs::ISystem](#) > system)=0
- virtual std::vector< std::shared\_ptr< [ecs::ISystem](#) > > **getSystems** () const =0

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/gsm/IGameState.hpp

## 4.75 gsm::IGameStateMachine Class Reference

Inheritance diagram for gsm::IGameStateMachine:



**Public Member Functions**

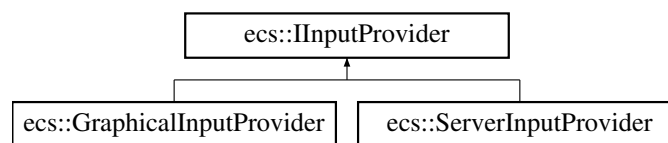
- virtual void **changeState** (std::shared\_ptr< [IGameState](#) > newState)=0
- virtual void **pushState** (std::shared\_ptr< [IGameState](#) > newState)=0
- virtual void **popState** ()=0
- virtual void **requestStateChange** (std::shared\_ptr< [IGameState](#) > newState)=0
- virtual void **requestStatePush** (std::shared\_ptr< [IGameState](#) > newState)=0
- virtual void **requestStatePop** ()=0
- virtual void **update** (float deltaTime)=0

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/gsm/IGameStateMachine.hpp

## 4.76 ecs::IInputProvider Class Reference

Inheritance diagram for ecs::IInputProvider:

**Public Types**

- using **event\_t** = gfx::EventType

**Public Member Functions**

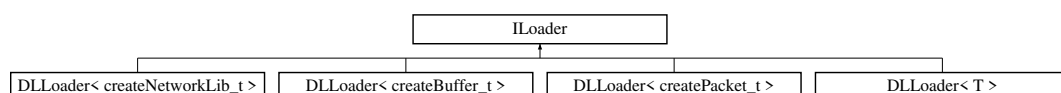
- virtual float **getAxisValue** (event\_t axis, size\_t clientID=0)=0
- virtual bool **isActionPressed** (InputAction action, size\_t clientID=0)=0
- virtual float **getActionAxis** (InputAction action, size\_t clientID=0)=0
- virtual [InputMapping](#) **getInputMapping** (size\_t clientID=0) const =0

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/InputMapping/IInputProvider.hpp

## 4.77 ILoader Class Reference

Inheritance diagram for ILoader:



### Public Member Functions

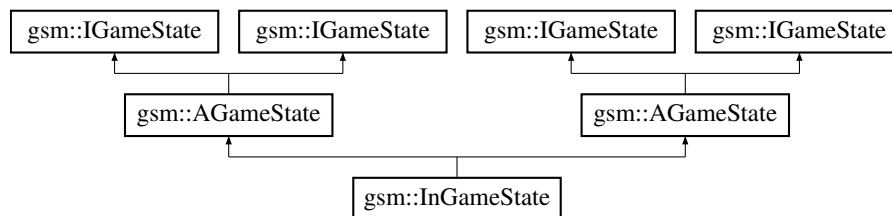
- virtual void \* **Open** (const char \*path, int flag)=0
- virtual void \* **Symbol** (const char \*symbolName)=0
- virtual int **Close** ()=0
- virtual const char \* **Error** ()=0
- virtual void \* **getHandler** () const =0

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/DLLoader/ILoader.hpp

## 4.78 gsm::InGameState Class Reference

Inheritance diagram for gsm::InGameState:



### Public Member Functions

- **InGameState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void [enter](#) () override
- void [update](#) (float deltaTime) override
- void [exit](#) () override
- **InGameState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void [enter](#) () override

### Public Member Functions inherited from gsm::AGameState

- **AGameState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- std::vector< std::shared\_ptr< [ecs::ISystem](#) > > [getSystems](#) () const override
- **AGameState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- std::vector< std::shared\_ptr< [ecs::ISystem](#) > > [getSystems](#) () const override

### Private Member Functions

- void **renderHUD** ()
- void **drawHealthHUD** (std::shared\_ptr< [gfx::IWindow](#) > window, float health, float maxHealth)
- void **drawScoreHUD** (std::shared\_ptr< [gfx::IWindow](#) > window, int score)

### Private Attributes

- `std::shared_ptr< ecs::Registry > _registry`
- `std::shared_ptr< EntityPrefabManager > _prefabManager`
- `std::shared_ptr< Parser > _parser`
- `int _previousScore`
- `int _previousHealth`
- `std::vector< ScoreFeedback > _scoreFeedbacks`
- `std::vector< ScoreFeedback > _healthFeedbacks`

### Additional Inherited Members

### Protected Member Functions inherited from [gsm::AGameState](#)

- `void addSystem (std::shared_ptr< ecs::ISystem > system) override`
- `void addSystem (std::shared_ptr< ecs::ISystem > system) override`

### Protected Attributes inherited from [gsm::AGameState](#)

- `std::weak_ptr< IGameStateMachine > _gsm`
- `std::shared_ptr< ResourceManager > _resourceManager`
- `std::vector< std::shared_ptr< ecs::ISystem > > _systems`

## 4.78.1 Member Function Documentation

### 4.78.1.1 `enter()` [1/2]

```
void gsm::InGameState::enter () [override], [virtual]
```

Reimplemented from [gsm::AGameState](#).

### 4.78.1.2 `enter()` [2/2]

```
void gsm::InGameState::enter () [override], [virtual]
```

Reimplemented from [gsm::AGameState](#).

### 4.78.1.3 `exit()`

```
void gsm::InGameState::exit () [override], [virtual]
```

Reimplemented from [gsm::AGameState](#).

#### 4.78.1.4 update()

```
void gsm::InGameState::update (
    float deltaTime) [override], [virtual]
```

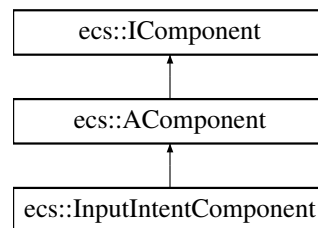
Reimplemented from [gsm::AGameState](#).

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

- [/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/InGame/InGameState.hpp](#)
- [/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/scenes/InGame/InGameState.hpp](#)
- [/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/InGame/InGameState.cpp](#)
- [/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/scenes/InGame/InGameState.cpp](#)

## 4.79 ecs::InputIntentComponent Class Reference

Inheritance diagram for `ecs::InputIntentComponent`:



### Public Member Functions

- **InputIntentComponent** (const [math::Vector2f](#) &direction=[math::Vector2f](#)(0.0f, 0.0f))
- [math::Vector2f](#) **getDirection** () const
- void **setDirection** (const [math::Vector2f](#) &direction)

### Private Attributes

- [math::Vector2f](#) **\_direction**

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

- [/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/temporary/InputIntentComponent.hpp](#)

## 4.80 ecs::InputMapping Struct Reference

### Public Member Functions

- std::map< InputAction, std::map< gfx::EventType, float > > **getAllMappings** () const

### Public Attributes

- std::map< RemappableAction, [RemappableKeyBinding](#) > **remappableKeys**
- std::map< InputAction, std::map< gfx::EventType, float > > **fixedMappings**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/InputMapping/InputMapping.hpp

## 4.81 ecs::InputMappingManager Class Reference

### Public Member Functions

- void **loadDefault** ()
- void **setMapping** (const [InputMapping](#) &mapping)
- const [InputMapping](#) & **getMapping** () const
- [InputMapping](#) & **getMutableMapping** ()
- gfx::EventType **getKeyForRemappableAction** (RemappableAction action, bool getPrimary=true) const
- void **remapKey** (RemappableAction action, gfx::EventType newKey, bool setPrimary)
- bool **isKeyboardKey** (gfx::EventType eventType)

### Static Public Member Functions

- static std::string **eventTypeToString** (gfx::EventType eventType)
- static gfx::EventType **stringToEventType** (const std::string &str)
- static std::string **remappableActionToString** (RemappableAction action)
- static RemappableAction **stringToRemappableAction** (const std::string &str)

### Private Attributes

- [InputMapping](#) **\_mapping**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/InputMapping/InputMappingManager.↔  
hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/InputMapping/InputMappingManager.↔  
cpp

## 4.82 ecs::InputNormalizer Class Reference

### Static Public Member Functions

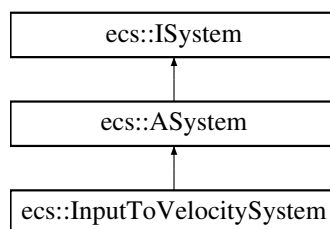
- static [math::Vector2f](#) **normalizeDirection** (const [math::Vector2f](#) &direction)
- static [math::Vector2f](#) **normalizeAnalogInput** (float rawX, float rawY)

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/input/InputNormalizer.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/input/InputNormalizer.cpp

## 4.83 ecs::InputToVelocitySystem Class Reference

Inheritance diagram for `ecs::InputToVelocitySystem`:



### Public Member Functions

- void **update** (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

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

- void **updateSystem** (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

### 4.83.1 Member Function Documentation

#### 4.83.1.1 update()

```

void ecs::InputToVelocitySystem::update (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [virtual]
  
```

Implements [ecs::ASystem](#).

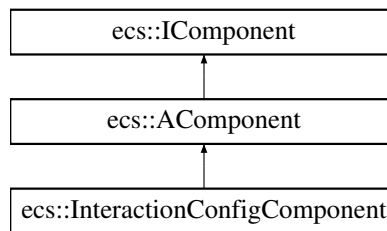
The documentation for this class was generated from the following files:

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/movement/InputToVelocity↔System.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/movement/InputToVelocity↔System.cpp



## 4.84 ecs::InteractionConfigComponent Class Reference

Inheritance diagram for ecs::InteractionConfigComponent:



### Public Member Functions

- **InteractionConfigComponent** (const std::vector< [InteractionMapping](#) > &mappings)
- const std::vector< [InteractionMapping](#) > & **getMappings** () const
- void **setMappings** (const std::vector< [InteractionMapping](#) > &mappings)
- void **addMapping** (const [InteractionMapping](#) &mapping)

### Private Attributes

- std::vector< [InteractionMapping](#) > **\_mappings**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/Interaction↵  
ConfigComponent.hpp

## 4.85 ecs::InteractionMapping Struct Reference

### Public Attributes

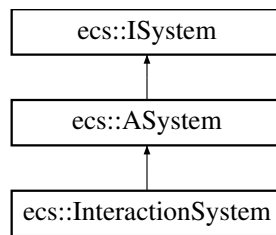
- std::vector< std::string > **targetTags**
- std::vector< std::string > **actionsToOther**
- std::vector< std::string > **actionsToSelf**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/Interaction↵  
ConfigComponent.hpp

## 4.86 ecs::InteractionSystem Class Reference

Inheritance diagram for ecs::InteractionSystem:



### Public Member Functions

- void [update](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

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

- void [updateSystem](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

### 4.86.1 Member Function Documentation

#### 4.86.1.1 update()

```

void ecs::InteractionSystem::update (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [virtual]
  
```

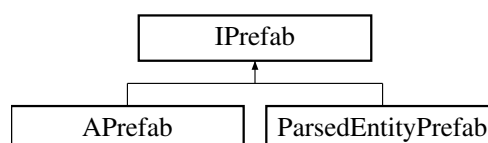
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/interactions/Interaction↵System.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/interactions/Interaction↵System.cpp

## 4.87 IPrefab Class Reference

Inheritance diagram for IPrefab:



### Public Member Functions

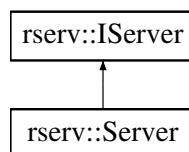
- virtual ecs::Entity **instantiate** (const std::shared\_ptr< [ecs::Registry](#) > &registry, const std::shared\_ptr< [ecs::IEntityFactory](#) > &factory, const [ecs::EntityCreationContext](#) &context=ecs::EntityCreationContext::forLocalClient())=0
- virtual ecs::Entity **instantiate** (const std::shared\_ptr< [ecs::Registry](#) > &registry)=0

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Prefab/IPrefab.hpp

## 4.88 rserv::IServer Class Reference

Inheritance diagram for rserv::IServer:



### Public Member Functions

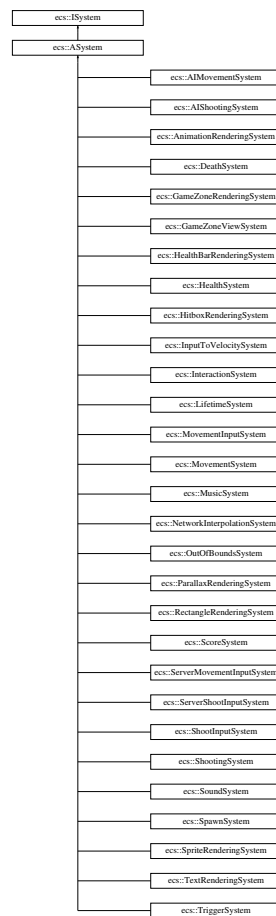
- virtual void **init** ()=0
- virtual void **start** ()=0
- virtual void **stop** ()=0
- virtual void **setConfig** (std::shared\_ptr< [ServerConfig](#) > config)=0
- virtual std::shared\_ptr< [ServerConfig](#) > **getConfig** () const =0
- virtual uint16\_t **getPort** () const =0
- virtual void **setPort** (uint16\_t port)=0
- virtual int **getState** () const =0
- virtual void **setState** (int state)=0
- virtual **operator int** () const noexcept=0
- virtual std::shared\_ptr< net::INetwork > **getNetwork** () const =0
- virtual void **setNetwork** (std::shared\_ptr< net::INetwork > network)=0
- virtual void **onClientConnected** (uint8\_t idClient)=0
- virtual void **onClientDisconnected** (uint8\_t idClient)=0
- virtual void **onPacketReceived** (uint8\_t idClient, const pm::IPacketManager &packet)=0
- virtual void **processIncomingPackets** ()=0
- virtual bool **processConnections** (std::pair< asio::ip::udp::endpoint, std::vector< uint8\_t > > client)=0
- virtual bool **processDisconnections** (uint8\_t idClient)=0
- virtual bool **processEvents** (uint8\_t idClient)=0
- virtual bool **processEndOfGame** (uint8\_t idClient)=0
- virtual std::vector< uint8\_t > **getConnectedClients** () const =0
- virtual std::vector< asio::ip::udp::endpoint > **getConnectedClientEndpoints** () const =0
- virtual size\_t **getClientCount** () const =0
- virtual std::shared\_ptr< std::queue< std::tuple< uint8\_t, constants::EventType, double > > > **getEventQueue** ()=0
- virtual bool **hasEvents** () const =0

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/IServer.hpp

## 4.89 ecs::ISystem Class Reference

Inheritance diagram for ecs::ISystem:



### Public Member Functions

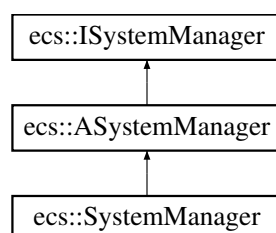
- virtual void **updateSystem** (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime)=0

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/base/ISystem.hpp

## 4.90 ecs::ISystemManager Class Reference

Inheritance diagram for ecs::ISystemManager:



**Public Member Functions**

- virtual void **updateAllSystems** (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime)=0
- virtual void **addSystem** (std::shared\_ptr< [ISystem](#) > system)=0
- virtual void **removeSystem** (std::shared\_ptr< [ISystem](#) > system)=0

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/systemManager/ISystemManager.hpp

**4.91 ecs::View< Components >::Iterator Class Reference****Public Member Functions**

- **Iterator** (std::shared\_ptr< [Registry](#) > registry, size\_t entityId, size\_t maxEntityId)
- bool **operator!=** (const [Iterator](#) &other) const
- [Iterator](#) & **operator++** ()
- size\_t **operator\*** () const

**Private Member Functions**

- bool **hasAllComponents** () const

**Private Attributes**

- std::shared\_ptr< [Registry](#) > **\_registry**
- size\_t **\_entityId**
- size\_t **\_maxEntityId**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/view/View.hpp

**4.92 ui::Background::Layer Struct Reference****Public Attributes**

- std::string **texturePath**
- float **speedX**
- float **speedY**
- [math::Vector2f](#) **sourceSize**
- float **offsetX** = 0.0f
- float **offsetY** = 0.0f

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/Background.hpp

## 4.93 ui::LayoutConfig Struct Reference

### Public Attributes

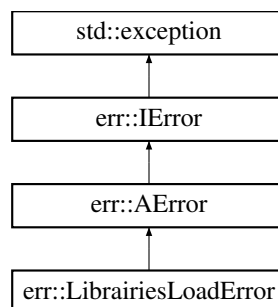
- LayoutDirection **direction** = LayoutDirection::Vertical
- LayoutAlignment **alignment** = LayoutAlignment::Start
- float **spacing** = 0.0f
- [math::Vector2f](#) **padding** = [math::Vector2f](#)(0.0f, 0.0f)
- [math::Vector2f](#) **offset** = [math::Vector2f](#)(0.0f, 0.0f)
- bool **autoResize** = false
- AnchorX **anchorX** = AnchorX::None
- AnchorY **anchorY** = AnchorY::None

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/core/UILayout.hpp

## 4.94 err::LibrairiesLoadError Class Reference

Inheritance diagram for err::LibrairiesLoadError:



### Public Types

- enum **ErrorCode** { **UNKNOWN** = 1000 , **LIBRARY\_NOT\_FOUND** = 1001 , **SYMBOL\_NOT\_FOUND** = 1002 }

### Public Member Functions

- **LibrairiesLoadError** (const std::string &message, ErrorCode code=UNKNOWN)
- std::string [getType](#) () const noexcept override

### Public Member Functions inherited from [err::AError](#)

- **AError** (const std::string &message, int code=0)
- const char \* [what](#) () const noexcept override
- int [getCode](#) () const noexcept override
- std::string [getDetails](#) () const noexcept override

## Additional Inherited Members

## Protected Attributes inherited from [err::AError](#)

- std::string **m\_message**
- int **m\_code**

## 4.94.1 Member Function Documentation

### 4.94.1.1 getType()

```
std::string err::LibrairiesLoadError::getType () const [override], [virtual], [noexcept]
```

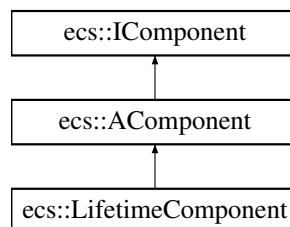
Implements [err::AError](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/LibrairiesLoadError.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/LibrairiesLoadError.cpp

## 4.95 ecs::LifetimeComponent Class Reference

Inheritance diagram for ecs::LifetimeComponent:



## Public Member Functions

- **LifetimeComponent** (float lifetime=0.0f)
- float **getLifetime** () const
- void **setLifetime** (float lifetime)

## Private Attributes

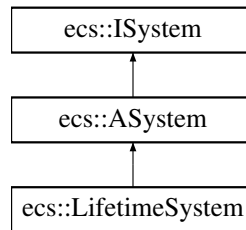
- float **\_lifetime**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/Lifetime↔Component.hpp

## 4.96 ecs::LifetimeSystem Class Reference

Inheritance diagram for ecs::LifetimeSystem:



### Public Member Functions

- void `update` (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

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

- void `updateSystem` (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

### 4.96.1 Member Function Documentation

#### 4.96.1.1 update()

```

void ecs::LifetimeSystem::update (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [virtual]
  
```

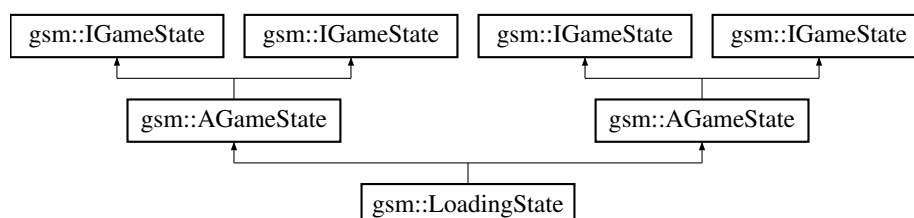
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/lifetime/LifetimeSystem.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/lifetime/LifetimeSystem.cpp

## 4.97 gsm::LoadingState Class Reference

Inheritance diagram for gsm::LoadingState:





## Public Member Functions

- **LoadingState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void [enter](#) () override
- void [update](#) (float deltaTime) override
- void [exit](#) () override
- **LoadingState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void [enter](#) () override

## Public Member Functions inherited from [gsm::AGameState](#)

- **AGameState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- std::vector< std::shared\_ptr< [ecs::ISystem](#) > > [getSystems](#) () const override
- **AGameState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- std::vector< std::shared\_ptr< [ecs::ISystem](#) > > [getSystems](#) () const override

## Additional Inherited Members

## Protected Member Functions inherited from [gsm::AGameState](#)

- void [addSystem](#) (std::shared\_ptr< [ecs::ISystem](#) > system) override
- void [addSystem](#) (std::shared\_ptr< [ecs::ISystem](#) > system) override

## Protected Attributes inherited from [gsm::AGameState](#)

- std::weak\_ptr< [IGameStateMachine](#) > [\\_gsm](#)
- std::shared\_ptr< [ResourceManager](#) > [\\_resourceManager](#)
- std::vector< std::shared\_ptr< [ecs::ISystem](#) > > [\\_systems](#)

### 4.97.1 Member Function Documentation

#### 4.97.1.1 [enter\(\)](#) [1/2]

```
void gsm::LoadingState::enter () [override], [virtual]
```

Reimplemented from [gsm::AGameState](#).

#### 4.97.1.2 [enter\(\)](#) [2/2]

```
void gsm::LoadingState::enter () [override], [virtual]
```

Reimplemented from [gsm::AGameState](#).

#### 4.97.1.3 exit()

```
void gsm::LoadingState::exit () [override], [virtual]
```

Reimplemented from [gsm::AGameState](#).

#### 4.97.1.4 update()

```
void gsm::LoadingState::update (
    float deltaTime) [override], [virtual]
```

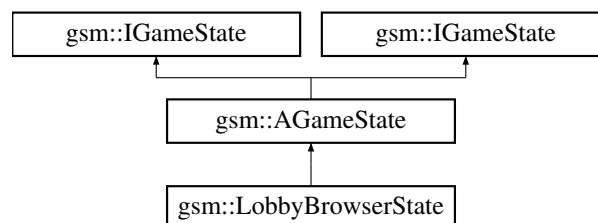
Reimplemented from [gsm::AGameState](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Loading/Loading↔State.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/scenes/Loading/Loading↔State.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Loading/Loading↔State.cpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/scenes/Loading/Loading↔State.cpp

## 4.98 gsm::LobbyBrowserState Class Reference

Inheritance diagram for gsm::LobbyBrowserState:



### Public Member Functions

- **LobbyBrowserState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void [enter](#) () override
- void [update](#) (float deltaTime) override
- void [exit](#) () override

### Public Member Functions inherited from [gsm::AGameState](#)

- **AGameState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- std::vector< std::shared\_ptr< [ecs::ISystem](#) > > [getSystems](#) () const override
- **AGameState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- std::vector< std::shared\_ptr< [ecs::ISystem](#) > > [getSystems](#) () const override

### Additional Inherited Members

### Protected Member Functions inherited from `gsm::AGameState`

- void `addSystem` (`std::shared_ptr< ecs::ISystem >` system) override
- void `addSystem` (`std::shared_ptr< ecs::ISystem >` system) override

### Protected Attributes inherited from `gsm::AGameState`

- `std::weak_ptr< IGameStateMachine > _gsm`
- `std::shared_ptr< ResourceManager > _resourceManager`
- `std::vector< std::shared_ptr< ecs::ISystem > > _systems`

## 4.98.1 Member Function Documentation

### 4.98.1.1 `enter()`

```
void gsm::LobbyBrowserState::enter () [override], [virtual]
```

Reimplemented from `gsm::AGameState`.

### 4.98.1.2 `exit()`

```
void gsm::LobbyBrowserState::exit () [override], [virtual]
```

Reimplemented from `gsm::AGameState`.

### 4.98.1.3 `update()`

```
void gsm::LobbyBrowserState::update (
    float deltaTime) [override], [virtual]
```

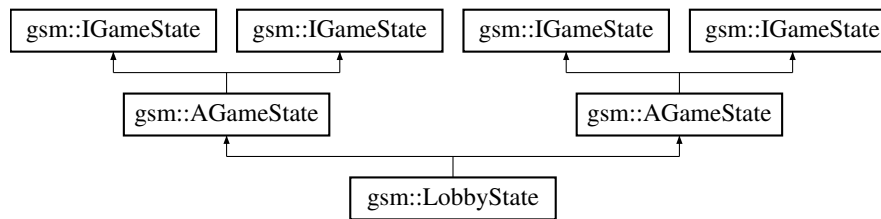
Reimplemented from `gsm::AGameState`.

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

- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/LobbyBrowser/LobbyBrowserState.hpp`
- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/LobbyBrowser/LobbyBrowserState.cpp`

## 4.99 `gsm::LobbyState` Class Reference

Inheritance diagram for `gsm::LobbyState`:



### Public Member Functions

- **LobbyState** (`std::shared_ptr< IGameStateMachine > gsm`, `std::shared_ptr< ResourceManager > resourceManager`)
- void `enter` () override
- void `update` (float deltaTime) override
- void `exit` () override
- **LobbyState** (`std::shared_ptr< IGameStateMachine > gsm`, `std::shared_ptr< ResourceManager > resourceManager`)
- void `enter` () override
- void `update` (float deltaTime) override

### Public Member Functions inherited from `gsm::AGameState`

- **AGameState** (`std::shared_ptr< IGameStateMachine > gsm`, `std::shared_ptr< ResourceManager > resourceManager`)
- `std::vector< std::shared_ptr< ecs::ISystem > > getSystems` () const override
- **AGameState** (`std::shared_ptr< IGameStateMachine > gsm`, `std::shared_ptr< ResourceManager > resourceManager`)
- `std::vector< std::shared_ptr< ecs::ISystem > > getSystems` () const override

### Additional Inherited Members

### Protected Member Functions inherited from `gsm::AGameState`

- void `addSystem` (`std::shared_ptr< ecs::ISystem > system`) override
- void `addSystem` (`std::shared_ptr< ecs::ISystem > system`) override

### Protected Attributes inherited from `gsm::AGameState`

- `std::weak_ptr< IGameStateMachine > _gsm`
- `std::shared_ptr< ResourceManager > _resourceManager`
- `std::vector< std::shared_ptr< ecs::ISystem > > _systems`

## 4.99.1 Member Function Documentation

### 4.99.1.1 enter() [1/2]

```
void gsm::LobbyState::enter () [override], [virtual]
```

Reimplemented from [gsm::AGameState](#).

### 4.99.1.2 enter() [2/2]

```
void gsm::LobbyState::enter () [override], [virtual]
```

Reimplemented from [gsm::AGameState](#).

### 4.99.1.3 exit()

```
void gsm::LobbyState::exit () [override], [virtual]
```

Reimplemented from [gsm::AGameState](#).

### 4.99.1.4 update() [1/2]

```
void gsm::LobbyState::update (  
    float deltaTime) [override], [virtual]
```

Reimplemented from [gsm::AGameState](#).

### 4.99.1.5 update() [2/2]

```
void gsm::LobbyState::update (  
    float deltaTime) [override], [virtual]
```

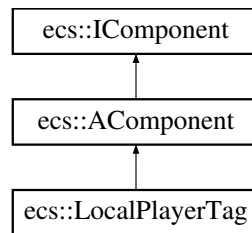
Reimplemented from [gsm::AGameState](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Lobby/LobbyState.↵  
hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/scenes/Lobby/LobbyState.↵  
hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Lobby/LobbyState.cpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/scenes/Lobby/LobbyState.↵  
cpp

## 4.100 ecs::LocalPlayerTag Class Reference

Inheritance diagram for ecs::LocalPlayerTag:

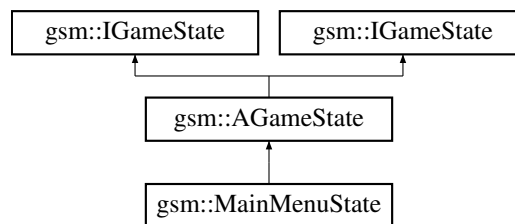


The documentation for this class was generated from the following file:

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/tags/LocalPlayerTag.hpp

## 4.101 gsm::MainMenuState Class Reference

Inheritance diagram for gsm::MainMenuState:



### Public Member Functions

- **MainMenuState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void [enter](#) () override
- void [update](#) (float deltaTime) override
- void [exit](#) () override

### Public Member Functions inherited from [gsm::AGameState](#)

- **AGameState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- std::vector< std::shared\_ptr< [ecs::ISystem](#) > > [getSystems](#) () const override
- **AGameState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- std::vector< std::shared\_ptr< [ecs::ISystem](#) > > [getSystems](#) () const override

### Private Member Functions

- void **renderUI** ()
- void **updatePlayButtonText** ()

### Private Attributes

- std::unique\_ptr< [MouseInputHandler](#) > **\_mouseHandler**
- std::shared\_ptr< [ui::Button](#) > **\_playButton**
- std::shared\_ptr< [ui::Button](#) > **\_settingsButton**
- std::shared\_ptr< [ui::Button](#) > **\_quitButton**
- std::shared\_ptr< [ui::Button](#) > **\_connectButton**
- std::unique\_ptr< [ui::UIManager](#) > **\_uiManager**
- std::shared\_ptr< [ui::UILayout](#) > **\_leftLayout**
- std::shared\_ptr< [ui::UILayout](#) > **\_mainMenuLayout**
- std::shared\_ptr< [ui::UILayout](#) > **\_rightLayout**
- std::shared\_ptr< [ui::Button](#) > **\_devButton**
- std::shared\_ptr< [ui::TextInput](#) > **\_ipInput**
- std::shared\_ptr< [ui::TextInput](#) > **\_portInput**
- std::shared\_ptr< [ui::Background](#) > **\_background**

### Additional Inherited Members

### Protected Member Functions inherited from [gsm::AGameState](#)

- void [addSystem](#) (std::shared\_ptr< [ecs::ISystem](#) > system) override
- void [addSystem](#) (std::shared\_ptr< [ecs::ISystem](#) > system) override

### Protected Attributes inherited from [gsm::AGameState](#)

- std::weak\_ptr< [IGameStateMachine](#) > **\_gsm**
- std::shared\_ptr< [ResourceManager](#) > **\_resourceManager**
- std::vector< std::shared\_ptr< [ecs::ISystem](#) > > **\_systems**

## 4.101.1 Member Function Documentation

### 4.101.1.1 enter()

```
void gsm::MainMenuState::enter () [override], [virtual]
```

Reimplemented from [gsm::AGameState](#).

### 4.101.1.2 exit()

```
void gsm::MainMenuState::exit () [override], [virtual]
```

Reimplemented from [gsm::AGameState](#).

#### 4.101.1.3 update()

```
void gsm::MainMenuState::update (
    float deltaTime) [override], [virtual]
```

Reimplemented from [gsm::AGameState](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/MainMenu/MainMenuState.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/MainMenu/MainMenuState.cpp

## 4.102 MapParser Class Reference

### Public Member Functions

- **MapParser** (std::shared\_ptr< [EntityPrefabManager](#) > prefabManager, std::shared\_ptr< [ecs::Registry](#) > registry)
- void **parseMapFromFile** (const std::string &filePath)
- void **parseMap** (const nlohmann::json &mapJson)
- void **generateMapEntities** ()
- nlohmann::json **getMapJson** () const
- void **setMapJson** (const nlohmann::json &mapJson)
- void **setCreationContext** (const [ecs::EntityCreationContext](#) &context)
- [ecs::EntityCreationContext](#) **getCreationContext** () const

### Private Member Functions

- void **createBackgroundEntity** (const std::string &entityName)
- void **createMusicEntity** (const std::string &prefabName)
- void **createGameZoneEntity** (float scrollSpeed)
- void **parsePowerUps** (const nlohmann::json &powerUps)
- void **parseObstacles** (const nlohmann::json &obstacles)
- void **parseWaves** (const nlohmann::json &waves)
- std::vector< float > **getPositionsFromDistrib** (int count, const nlohmann::json &distribution, float limit)
- [ecs::Entity](#) **createEntityFromPrefab** (const std::string &prefabName, float x, float y)

### Private Attributes

- std::shared\_ptr< [EntityPrefabManager](#) > **\_prefabManager**
- std::shared\_ptr< [ecs::Registry](#) > **\_registry**
- [ecs::EntityCreationContext](#) **\_creationContext**
- nlohmann::json **\_mapJson**

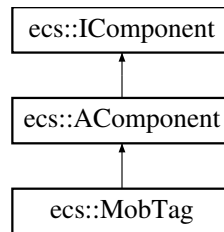
The documentation for this class was generated from the following files:

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/MapParser/MapParser.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/MapParser/MapParser.cpp



## 4.103 ecs::MobTag Class Reference

Inheritance diagram for ecs::MobTag:



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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/tags/MobTag.hpp

## 4.104 MouseButtonInfo Struct Reference

### Public Attributes

- [math::Vector2f](#) **position**
- constants::MouseButton **button**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/input/MouseInputHandler.hpp

## 4.105 MouseInputHandler Class Reference

### Public Member Functions

- **MouseInputHandler** (std::shared\_ptr< [ResourceManager](#) > resourceManager)
- std::optional< [MouseButtonInfo](#) > **pollMouseClicked** ()
- [math::Vector2f](#) **getMousePosition** () const
- [math::Vector2f](#) **getWorldMousePosition** () const
- [math::Vector2f](#) **getNormalizedMousePosition** () const
- bool **isMouseButtonPressed** (int button) const

### Private Attributes

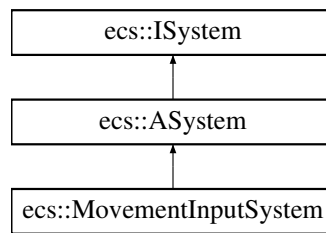
- std::weak\_ptr< [ResourceManager](#) > **\_resourceManager**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/input/MouseInputHandler.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/input/MouseInputHandler.cpp

## 4.106 ecs::MovementInputSystem Class Reference

Inheritance diagram for ecs::MovementInputSystem:



### Public Member Functions

- void [update](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

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

- void [updateSystem](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

### Private Member Functions

- [math::Vector2f](#) [getMovementDirection](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager) const
- void [updateInputIntent](#) (std::shared\_ptr< [Registry](#) > registry, Entity entityId, const [math::Vector2f](#) &direction)
- [math::Vector2f](#) [getAnalogStickInput](#) (std::shared\_ptr< [IInputProvider](#) > inputProvider) const
- void [sendAxisEvents](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, const [math::Vector2f](#) &direction)

### Private Attributes

- bool [\\_wasMovingLastFrame](#) = false

## 4.106.1 Member Function Documentation

### 4.106.1.1 update()

```

void ecs::MovementInputSystem::update (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [virtual]
  
```

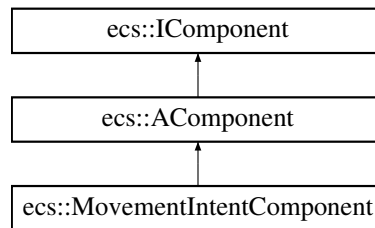
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/input/MovementInputSystem.↔hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/input/MovementInputSystem.↔cpp

## 4.107 ecs::MovementIntentComponent Class Reference

Inheritance diagram for ecs::MovementIntentComponent:



### Public Member Functions

- **MovementIntentComponent** (const [math::Vector2f](#) &direction=[math::Vector2f](#)(0.0f, 0.0f), bool active=false)
- [math::Vector2f](#) **getDirection** () const
- void **setDirection** (const [math::Vector2f](#) &direction)
- bool **isActive** () const
- void **setActive** (bool active)

### Private Attributes

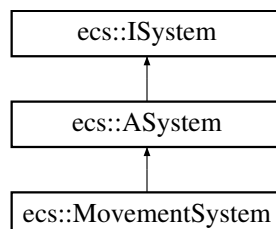
- [math::Vector2f](#) **\_direction**
- bool **\_active**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/temporary/MovementIntentComponent.hpp

## 4.108 ecs::MovementSystem Class Reference

Inheritance diagram for ecs::MovementSystem:



### Public Member Functions

- void **update** (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

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

- void [updateSystem](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

## Private Member Functions

- void [buildSpatialGrid](#) (std::shared\_ptr< [Registry](#) > registry)
- bool [checkCollision](#) (std::shared\_ptr< [Registry](#) > registry, size\_t entityId, [math::Vector2f](#) newPos)
- [math::Vector2f](#) [calculateSmoothMovement](#) (std::shared\_ptr< [Registry](#) > registry, size\_t entityId, [math::Vector2f](#) startPos, [math::Vector2f](#) desiredPos)
- [math::Vector2f](#) [calculateSlidingMovement](#) (std::shared\_ptr< [Registry](#) > registry, size\_t entityId, [math::Vector2f](#) basePos, [math::Vector2f](#) desiredPos)
- [math::Vector2f](#) [calculateSmoothSlidingPosition](#) (std::shared\_ptr< [Registry](#) > registry, size\_t entityId, [math::Vector2f](#) startPos, [math::Vector2f](#) desiredPos)
- void [handlePushCollision](#) (std::shared\_ptr< [Registry](#) > registry, size\_t entityId, [math::Vector2f](#) finalPos, float deltaTime)
- bool [shouldCollide](#) (std::shared\_ptr< [Registry](#) > registry, size\_t entityA, const [ColliderComponent](#) &colliderA, size\_t entityB)
- bool [checkCollisionWithBoundaries](#) (std::shared\_ptr< [Registry](#) > registry, size\_t entityId, [math::Vector2f](#) newPos)

## Private Attributes

- [SpatialGrid](#) [\\_spatialGrid](#)
- std::vector< Entity > [\\_boundaryEntities](#)

## 4.108.1 Member Function Documentation

### 4.108.1.1 update()

```
void ecs::MovementSystem::update (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [virtual]
```

Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/movement/Movement↵System.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/movement/Movement↵System.cpp

## 4.109 ecs::MultiShotPattern Struct Reference

### Public Member Functions

- **MultiShotPattern** (int count, float spread, float offset)

**Public Attributes**

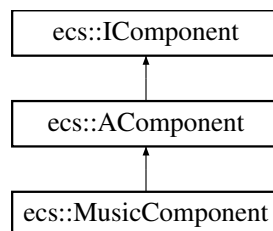
- int **shotCount** = 1
- float **angleSpread** = 0.0f
- float **offsetDistance** = 0.0f

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/ShootingStatsComponent.hpp↔

**4.110 ecs::MusicComponent Class Reference**

Inheritance diagram for ecs::MusicComponent:

**Public Member Functions**

- **MusicComponent** (std::string musicFile="", MusicState initialState=STOPPED, float volume=100.0f, bool loop=false)
- void **playMusic** ()
- void **pauseMusic** ()
- void **stopMusic** ()
- bool **isPlaying** () const
- MusicState **getState** () const
- void **playNewMusic** (const std::string &musicFile)
- std::string **getCurrentMusic** () const
- void **setCurrentMusic** (const std::string &musicFile)
- float **getVolume** () const
- void **setVolume** (float volume)
- bool **isLooping** () const
- void **setLoop** (bool loop)

**Private Attributes**

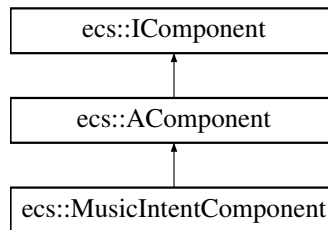
- std::string **\_currentMusic**
- MusicState **\_state**
- float **\_volume**
- bool **\_loop**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/MusicComponent.hpp↔

### 4.111 ecs::MusicIntentComponent Class Reference

Inheritance diagram for ecs::MusicIntentComponent:



#### Public Member Functions

- **MusicIntentComponent** (MusicAction action=PLAY, const std::string &musicPath="", float volume=100.0f)
- MusicAction **getAction** () const
- void **setAction** (MusicAction action)
- std::string **getMusicPath** () const
- void **setMusicPath** (const std::string &musicPath)
- float **getVolume** () const
- void **setVolume** (float volume)

#### Private Attributes

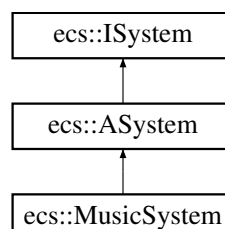
- MusicAction **\_action**
- std::string **\_musicPath**
- float **\_volume**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/temporary/MusicIntentComponent.hpp↵

### 4.112 ecs::MusicSystem Class Reference

Inheritance diagram for ecs::MusicSystem:



#### Protected Member Functions

- void **update** (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

### Additional Inherited Members

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

- void [updateSystem](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

### 4.112.1 Member Function Documentation

#### 4.112.1.1 update()

```
void ecs::MusicSystem::update (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [protected], [virtual]
```

Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/audio/MusicSystem.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/audio/MusicSystem.cpp

## 4.113 NetworkEvent Struct Reference

### Public Attributes

- constants::EventType **eventType**
- double **depth**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ClientNetwork.hpp

## 4.114 ecs::NetworkHealthState Struct Reference

### Public Attributes

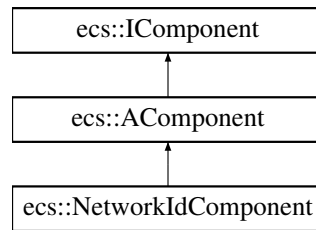
- uint32\_t **health**
- uint32\_t **baseHealth**
- std::chrono::steady\_clock::time\_point **timestamp**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/interpolation/NetworkStateComponent.↔hpp

## 4.115 ecs::NetworkIdComponent Class Reference

Inheritance diagram for ecs::NetworkIdComponent:



### Public Member Functions

- **NetworkIdComponent** (size\_t networkId=0)
- size\_t **getNetworkId** () const
- void **setNetworkId** (size\_t networkId)

### Private Attributes

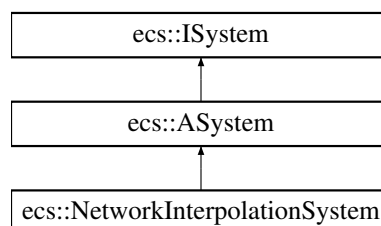
- size\_t **\_networkId**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/NetworkIdComponent.hpp

## 4.116 ecs::NetworkInterpolationSystem Class Reference

Inheritance diagram for ecs::NetworkInterpolationSystem:



### Public Member Functions

- void **update** (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

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

- void **updateSystem** (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override



## Private Member Functions

- void **interpolateTransform** (std::shared\_ptr< [NetworkStateComponent](#) > networkState, std::shared\_ptr< [TransformComponent](#) > transform)

## 4.116.1 Member Function Documentation

### 4.116.1.1 update()

```
void ecs::NetworkInterpolationSystem::update (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [virtual]
```

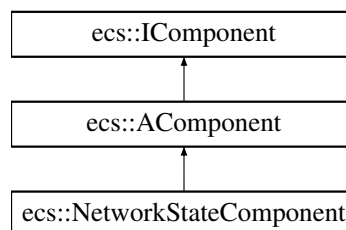
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/network/NetworkInterpolation↵  
System.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/network/NetworkInterpolation↵  
System.cpp

## 4.117 ecs::NetworkStateComponent Class Reference

Inheritance diagram for ecs::NetworkStateComponent:



## Public Member Functions

- void **setCurrentTransform** (const [math::Vector2f](#) &pos, float rot, const [math::Vector2f](#) &scale)
- bool **hasTransform** () const
- const [NetworkTransformState](#) & **getPreviousTransform** () const
- const [NetworkTransformState](#) & **getCurrentTransform** () const
- void **setCurrentHealth** (uint32\_t health, uint32\_t baseHealth)
- bool **hasHealth** () const
- const [NetworkHealthState](#) & **getPreviousHealth** () const
- const [NetworkHealthState](#) & **getCurrentHealth** () const
- void **setInterpolationTime** (float time)
- float **getInterpolationTime** () const
- float **getTransformInterpolationFactor** () const

**Private Attributes**

- [NetworkTransformState](#) **\_previousTransform**
- [NetworkTransformState](#) **\_currentTransform**
- bool **\_hasTransform**
- [NetworkHealthState](#) **\_previousHealth**
- [NetworkHealthState](#) **\_currentHealth**
- bool **\_hasHealth**
- float **\_interpolationTime**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/interpolation/NetworkStateComponent.↔  
hpp

**4.118 ecs::NetworkTransformState Struct Reference****Public Attributes**

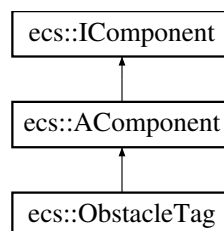
- [math::Vector2f](#) **position**
- float **rotation**
- [math::Vector2f](#) **scale**
- std::chrono::steady\_clock::time\_point **timestamp**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/interpolation/NetworkStateComponent.↔  
hpp

**4.119 ecs::ObstacleTag Class Reference**

Inheritance diagram for ecs::ObstacleTag:

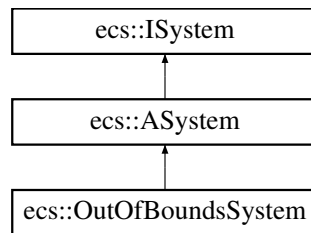


The documentation for this class was generated from the following file:

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/tags/ObstacleTag.hpp

## 4.120 ecs::OutOfBoundsSystem Class Reference

Inheritance diagram for ecs::OutOfBoundsSystem:



### Public Member Functions

- void `update` (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

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

- void `updateSystem` (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

### Private Attributes

- float `_margin`

### 4.120.1 Member Function Documentation

#### 4.120.1.1 update()

```
void ecs::OutOfBoundsSystem::update (  
    std::shared_ptr< ResourceManager > resourceManager,  
    std::shared_ptr< Registry > registry,  
    float deltaTime) [override], [virtual]
```

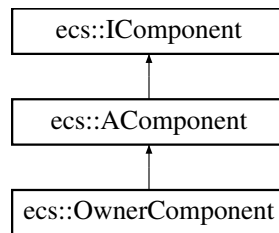
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/bounds/OutOfBounds↵  
System.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/bounds/OutOfBounds↵  
System.cpp

## 4.121 ecs::OwnerComponent Class Reference

Inheritance diagram for ecs::OwnerComponent:



### Public Member Functions

- **OwnerComponent** (ecs::Entity owner=0)
- ecs::Entity **getOwner** () const
- void **setOwner** (ecs::Entity owner)

### Private Attributes

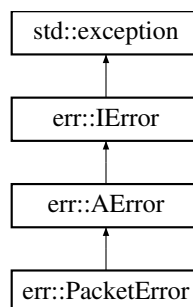
- ecs::Entity **\_owner**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/Owner↔  
Component.hpp

## 4.122 err::PacketError Class Reference

Inheritance diagram for err::PacketError:



### Public Types

- enum **ErrorCode** { **UNKNOWN** = 1000 , **SERIALIZER\_ATTRIBUTION\_FAILED** = 1001 , **STRING\_↔  
FORMATTING\_ERROR** = 1002 }

### Public Member Functions

- **PacketError** (const std::string &message, ErrorCode code=UNKNOWN)
- std::string [getType](#) () const noexcept override

### Public Member Functions inherited from [err::AError](#)

- **AError** (const std::string &message, int code=0)
- const char \* [what](#) () const noexcept override
- int [getCode](#) () const noexcept override
- std::string [getDetails](#) () const noexcept override

### Additional Inherited Members

### Protected Attributes inherited from [err::AError](#)

- std::string **m\_message**
- int **m\_code**

## 4.122.1 Member Function Documentation

### 4.122.1.1 [getType\(\)](#)

```
std::string err::PacketError::getType () const [override], [virtual], [noexcept]
```

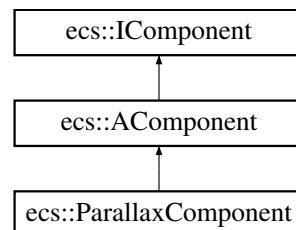
Implements [err::AError](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/PacketError.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/PacketError.cpp

## 4.123 ecs::ParallaxComponent Class Reference

Inheritance diagram for ecs::ParallaxComponent:



### Public Member Functions

- float **getBaseScrollSpeed** () const
- const [math::Vector2f](#) & **getDirection** () const
- const std::vector< [ParallaxLayer](#) > & **getLayers** () const
- void **setBaseScrollSpeed** (float speed)
- void **setDirection** (const [math::Vector2f](#) &direction)
- void **addLayer** (const [ParallaxLayer](#) &layer)
- void **clearLayers** ()
- void **updateLayerOffsets** (const [math::Vector2f](#) &direction, float baseSpeed, float deltaTime)
- size\_t **getLayerCount** () const
- void **sortLayersByZIndex** ()

### Private Attributes

- float **\_baseScrollSpeed**
- [math::Vector2f](#) **\_direction**
- std::vector< [ParallaxLayer](#) > **\_layers**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/ParallaxComponent.↵  
hpp

## 4.124 ecs::ParallaxLayer Struct Reference

### Public Attributes

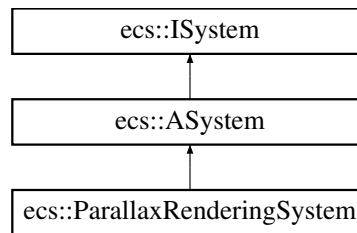
- std::string **name**
- std::string **filePath**
- float **speedMultiplier**
- [math::Vector2f](#) **scale**
- ParallaxScaleMode **scaleMode**
- [math::Vector2f](#) **sourceSize**
- bool **repeat**
- int **zIndex**
- [math::Vector2f](#) **currentOffset**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/ParallaxComponent.↵  
hpp

## 4.125 ecs::ParallaxRenderingSystem Class Reference

Inheritance diagram for ecs::ParallaxRenderingSystem:



### Protected Member Functions

- void **update** (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

### Private Member Functions

- [math::Vector2f](#) **calculateScale** (const [ParallaxLayer](#) &layer, float screenWidth, float screenHeight)
- void **renderLayer** (const [ParallaxLayer](#) &layer, std::shared\_ptr< [ResourceManager](#) > resourceManager, const [math::Vector2f](#) &basePosition, float screenWidth, float screenHeight)

### Additional Inherited Members

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

- void **updateSystem** (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

## 4.125.1 Member Function Documentation

### 4.125.1.1 update()

```

void ecs::ParallaxRenderingSystem::update (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [protected], [virtual]
  
```

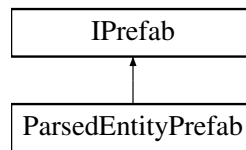
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/ParallaxRendering↵  
System.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/ParallaxRendering↵  
System.cpp

## 4.126 ParsedEntityPrefab Class Reference

Inheritance diagram for ParsedEntityPrefab:



### Public Member Functions

- **ParsedEntityPrefab** (const std::string &name, const std::map< std::type\_index, ComponentAdder > &adders)
- void **addComponent** (std::shared\_ptr< [ecs::IComponent](#) > component, std::type\_index typeIndex)
- const std::vector< std::shared\_ptr< [ecs::IComponent](#) > > & **getComponents** () const
- std::string **getName** () const
- ecs::Entity **instantiate** (const std::shared\_ptr< [ecs::Registry](#) > &registry, const std::shared\_ptr< [ecs::IEntityFactory](#) > &factory, const [ecs::EntityCreationContext](#) &context=ecs::EntityCreationContext::forLocalClient()) override
- ecs::Entity **instantiate** (const std::shared\_ptr< [ecs::Registry](#) > &registry) override

### Private Member Functions

- void **addParsedComponents** (const std::shared\_ptr< [ecs::Registry](#) > &registry, ecs::Entity entity)

### Private Attributes

- std::string **\_name**
- std::vector< std::pair< std::shared\_ptr< [ecs::IComponent](#) >, std::type\_index > > **\_components**
- const std::map< std::type\_index, ComponentAdder > & **\_componentAdders**

## 4.126.1 Member Function Documentation

### 4.126.1.1 instantiate() [1/2]

```
ecs::Entity ParsedEntityPrefab::instantiate (
    const std::shared_ptr< ecs::Registry > & registry) [override], [virtual]
```

Implements [IPrefab](#).

### 4.126.1.2 instantiate() [2/2]

```
ecs::Entity ParsedEntityPrefab::instantiate (
    const std::shared_ptr< ecs::Registry > & registry,
    const std::shared_ptr< ecs::IEntityFactory > & factory,
    const ecs::EntityCreationContext & context = ecs::EntityCreationContext::forLocalClient())
[override], [virtual]
```

Implements [IPrefab](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Prefab/ParsedEntityPrefab.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Prefab/ParsedEntityPrefab.cpp



## 4.127 Parser Class Reference

### Public Member Functions

- **Parser** (std::shared\_ptr< [EntityPrefabManager](#) > prefab, ParsingType type, std::shared\_ptr< [ecs::Registry](#) > registry)
- std::shared\_ptr< [EntityPrefabManager](#) > **getPrefabManager** () const
- void **setPrefabManager** (std::shared\_ptr< [EntityPrefabManager](#) > prefab)
- void **parseAllEntities** (std::string directoryPath)
- void **parseEntity** (std::string entityPath)
- void **instanciateComponentDefinitions** ()
- void **instanciateComponentCreators** ()
- template<typename T>  
void **registerComponent** (const ComponentCreator &creator)
- const std::map< std::type\_index, ComponentAdder > & **getComponentAdders** () const
- ParsingType **getParsingType** () const
- bool **isClientParsing** () const
- bool **isServerParsing** () const
- bool **shouldParseComponent** (std::map< std::string, std::shared\_ptr< [FieldValue](#) > > fields) const
- void **parseMapFromFile** (const std::string &filePath)
- std::shared\_ptr< [MapParser](#) > **getMapParser** () const
- void **setRegistry** (std::shared\_ptr< [ecs::Registry](#) > registry)

### Private Attributes

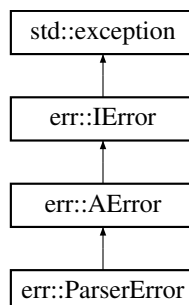
- std::shared\_ptr< [EntityParser](#) > **\_entityParser**
- std::shared\_ptr< [MapParser](#) > **\_mapParser**
- std::shared\_ptr< [EntityPrefabManager](#) > **\_prefabManager**
- std::shared\_ptr< std::map< std::string, std::pair< std::type\_index, std::vector< [Field](#) > > > > **\_↵ componentDefinitions**
- std::map< std::type\_index, ComponentCreator > **\_componentCreators**
- std::map< std::type\_index, ComponentAdder > **\_componentAdders**
- ParsingType **\_parsingType**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/Parser.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/ComponentParserCreators.↵  
cpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/Parser.cpp

## 4.128 err::ParserError Class Reference

Inheritance diagram for err::ParserError:



## Public Types

- enum **ErrorCode** {  
**UNKNOWN** = 1000 , **FILE\_NOT\_FOUND** = 1001 , **INVALID\_FORMAT** = 1002 , **MISSING\_FIELD** = 1003 ,  
**TYPE\_MISMATCH** = 1004 }

## Public Member Functions

- ParserError** (const std::string &message, ErrorCode code=UNKNOWN)
- std::string [getType](#) () const noexcept override

## Public Member Functions inherited from [err::AError](#)

- AError** (const std::string &message, int code=0)
- const char \* [what](#) () const noexcept override
- int [getCode](#) () const noexcept override
- std::string [getDetails](#) () const noexcept override

## Additional Inherited Members

## Protected Attributes inherited from [err::AError](#)

- std::string **m\_message**
- int **m\_code**

## 4.128.1 Member Function Documentation

### 4.128.1.1 [getType\(\)](#)

```
std::string err::ParserError::getType () const [override], [virtual], [noexcept]
```

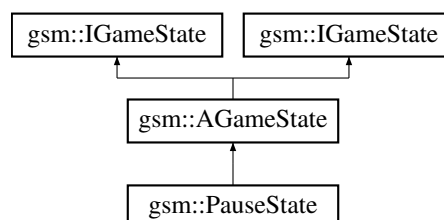
Implements [err::AError](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/ParserError.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/ParserError.cpp

## 4.129 [gsm::PauseState](#) Class Reference

Inheritance diagram for [gsm::PauseState](#):



## Public Member Functions

- **PauseState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void [enter](#) () override
- void [update](#) (float deltaTime) override
- void [exit](#) () override

## Public Member Functions inherited from [gsm::AGameState](#)

- **AGameState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- std::vector< std::shared\_ptr< [ecs::ISystem](#) > > [getSystems](#) () const override
- **AGameState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- std::vector< std::shared\_ptr< [ecs::ISystem](#) > > [getSystems](#) () const override

## Additional Inherited Members

## Protected Member Functions inherited from [gsm::AGameState](#)

- void [addSystem](#) (std::shared\_ptr< [ecs::ISystem](#) > system) override
- void [addSystem](#) (std::shared\_ptr< [ecs::ISystem](#) > system) override

## Protected Attributes inherited from [gsm::AGameState](#)

- std::weak\_ptr< [IGameStateMachine](#) > **\_gsm**
- std::shared\_ptr< [ResourceManager](#) > **\_resourceManager**
- std::vector< std::shared\_ptr< [ecs::ISystem](#) > > **\_systems**

## 4.129.1 Member Function Documentation

### 4.129.1.1 `enter()`

```
void gsm::PauseState::enter () [override], [virtual]
```

Reimplemented from [gsm::AGameState](#).

### 4.129.1.2 `exit()`

```
void gsm::PauseState::exit () [override], [virtual]
```

Reimplemented from [gsm::AGameState](#).

#### 4.129.1.3 update()

```
void gsm::PauseState::update (
    float deltaTime) [override], [virtual]
```

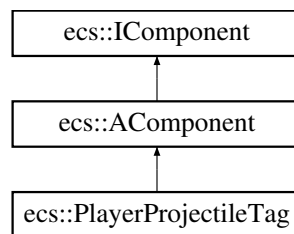
Reimplemented from [gsm::AGameState](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Pause/PauseState.↔  
hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Pause/PauseState.↔  
cpp

### 4.130 ecs::PlayerProjectileTag Class Reference

Inheritance diagram for `ecs::PlayerProjectileTag`:

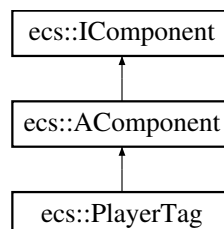


The documentation for this class was generated from the following file:

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/tags/PlayerProjectile↔  
Tag.hpp

### 4.131 ecs::PlayerTag Class Reference

Inheritance diagram for `ecs::PlayerTag`:

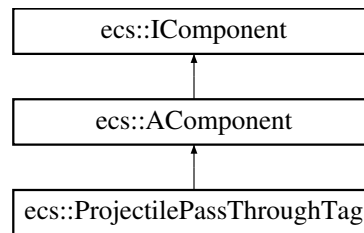


The documentation for this class was generated from the following file:

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/tags/PlayerTag.hpp

## 4.132 ecs::ProjectilePassThroughTag Class Reference

Inheritance diagram for ecs::ProjectilePassThroughTag:

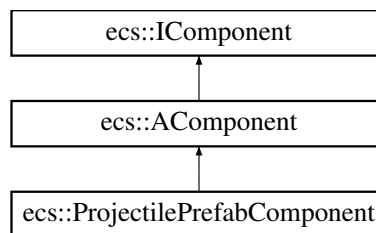


The documentation for this class was generated from the following file:

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/tags/ProjectilePassThroughTag.hpp

## 4.133 ecs::ProjectilePrefabComponent Class Reference

Inheritance diagram for ecs::ProjectilePrefabComponent:



### Public Member Functions

- **ProjectilePrefabComponent** (const std::string &prefabName="")
- std::string **getPrefabName** () const
- void **setPrefabName** (const std::string &prefabName)

### Private Attributes

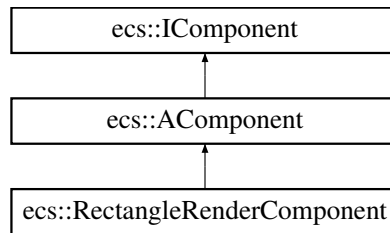
- std::string **\_prefabName**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/ProjectilePrefabComponent.hpp

## 4.134 ecs::RectangleRenderComponent Class Reference

Inheritance diagram for ecs::RectangleRenderComponent:



### Public Member Functions

- **RectangleRenderComponent** (gfx::color\_t color, float width, float height)
- const gfx::color\_t & **getColor** () const
- void **setColor** (const gfx::color\_t &color)
- float **getWidth** () const
- float **getHeight** () const
- void **setSize** (float width, float height)

### Private Attributes

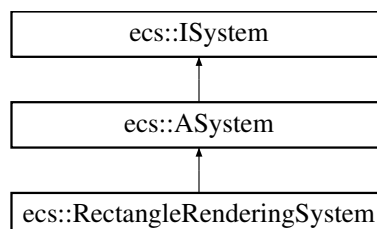
- gfx::color\_t **\_color**
- std::pair< float, float > **\_size**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/RectangleRenderComponent.hpp

## 4.135 ecs::RectangleRenderingSystem Class Reference

Inheritance diagram for ecs::RectangleRenderingSystem:



### Protected Member Functions

- void **update** (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

## Additional Inherited Members

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

- void [updateSystem](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

### 4.135.1 Member Function Documentation

#### 4.135.1.1 update()

```
void ecs::RectangleRenderingSystem::update (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [protected], [virtual]
```

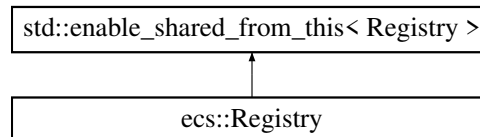
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/RectangleRendering↵  
System.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/RectangleRendering↵  
System.cpp

## 4.136 ecs::Registry Class Reference

Inheritance diagram for [ecs::Registry](#):



## Public Member Functions

- template<typename T>  
void **registerComponent** ()
- template<typename T>  
void **addComponent** (Entity entityId, std::shared\_ptr< T > component)
- template<typename T>  
std::shared\_ptr< T > **getComponent** (Entity entityId) const
- template<typename T>  
std::vector< std::shared\_ptr< T > > **getComponents** (Entity entityId) const
- template<typename T>  
void **removeAllComponents** (Entity entityId)
- template<typename T>  
void **removeOneComponent** (Entity entityId)
- template<typename T>  
bool **hasComponent** (Entity entityId) const
- template<typename... Components>  
[View](#)< Components... > **view** ()
- Entity **getMaxEntityId** () const
- Entity **createEntity** ()
- void **destroyEntity** (Entity entityId)
- Entity **getEntityByNetworkId** (size\_t networkId)
- void **setOnEntityDestroyed** (std::function< void(Entity)> callback)

**Private Attributes**

- Entity **\_nextEntityId**
- std::unordered\_map< std::string, std::shared\_ptr< [IComponentArray](#) > > **\_components**
- std::function< void(Entity)> **\_onEntityDestroyed**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/entity/registry/Registry.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/entity/registry/Registry.cpp

## 4.137 ecs::RemappableKeyBinding Struct Reference

**Public Member Functions**

- **RemappableKeyBinding** (gfx::EventType p, gfx::EventType s)

**Public Attributes**

- gfx::EventType **primary**
- gfx::EventType **secondary**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/InputMapping/InputMapping.hpp

## 4.138 ResourceManager Class Reference

**Public Member Functions**

- template<typename T>  
void **add** (std::shared\_ptr< T > resource)
- template<typename T>  
std::shared\_ptr< T > **get** ()
- template<typename T>  
bool **has** ()
- void **clear** ()
- template<typename T>  
void **remove** ()

**Private Attributes**

- std::unordered\_map< size\_t, std::shared\_ptr< void > > **resources**

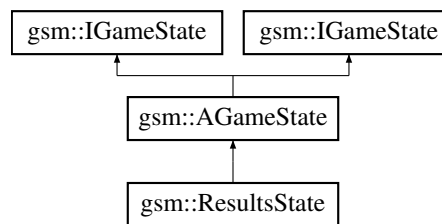
The documentation for this class was generated from the following file:

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/resourceManager/ResourceManager.↵.hpp



## 4.139 gsm::ResultsState Class Reference

Inheritance diagram for gsm::ResultsState:



### Public Member Functions

- **ResultsState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void [enter](#) () override
- void [update](#) (float deltaTime) override
- void [exit](#) () override

### Public Member Functions inherited from [gsm::AGameState](#)

- **AGameState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- std::vector< std::shared\_ptr< [ecs::ISystem](#) > > [getSystems](#) () const override
- **AGameState** (std::shared\_ptr< [IGameStateMachine](#) > gsm, std::shared\_ptr< [ResourceManager](#) > resourceManager)
- std::vector< std::shared\_ptr< [ecs::ISystem](#) > > [getSystems](#) () const override

### Additional Inherited Members

### Protected Member Functions inherited from [gsm::AGameState](#)

- void [addSystem](#) (std::shared\_ptr< [ecs::ISystem](#) > system) override
- void [addSystem](#) (std::shared\_ptr< [ecs::ISystem](#) > system) override

### Protected Attributes inherited from [gsm::AGameState](#)

- std::weak\_ptr< [IGameStateMachine](#) > [\\_gsm](#)
- std::shared\_ptr< [ResourceManager](#) > [\\_resourceManager](#)
- std::vector< std::shared\_ptr< [ecs::ISystem](#) > > [\\_systems](#)

## 4.139.1 Member Function Documentation

### 4.139.1.1 [enter\(\)](#)

```
void gsm::ResultsState::enter () [override], [virtual]
```

Reimplemented from [gsm::AGameState](#).

#### 4.139.1.2 exit()

```
void gsm::ResultsState::exit () [override], [virtual]
```

Reimplemented from [gsm::AGameState](#).

#### 4.139.1.3 update()

```
void gsm::ResultsState::update (
    float deltaTime) [override], [virtual]
```

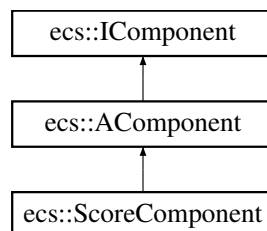
Reimplemented from [gsm::AGameState](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Results/Results↔State.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Results/Results↔State.cpp

## 4.140 ecs::ScoreComponent Class Reference

Inheritance diagram for ecs::ScoreComponent:



### Public Member Functions

- **ScoreComponent** (int score=0)
- int **getScore** () const
- void **setScore** (int score)
- void **addScore** (int amount)

### Private Attributes

- int **\_score**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/Score↔Component.hpp

## 4.141 `gsm::ScoreFeedback` Struct Reference

### Public Attributes

- `std::string` **text**
- `float` **lifetime**
- `float` **maxLifetime**

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

- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Dev/DevState.hpp`
- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/InGame/InGame↵  
State.hpp`

## 4.142 `ScoreIntentComponent` Class Reference

### Public Member Functions

- **`ScoreIntentComponent`** (`int` score=0)
- `int` **`getScore`** () const
- `void` **`setScore`** (`int` newScore)

### Private Attributes

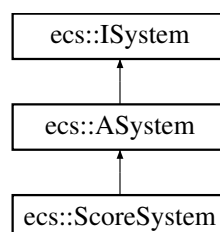
- `int` **`_score`**

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

- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/temporary/ScoreIntent↵  
Component.hpp`

## 4.143 `ecs::ScoreSystem` Class Reference

Inheritance diagram for `ecs::ScoreSystem`:



### Public Member Functions

- `void` **`update`** (`std::shared_ptr`< [ResourceManager](#) > resourceManager, `std::shared_ptr`< [Registry](#) > registry, `float` deltaTime) override

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

- void [updateSystem](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

### 4.143.1 Member Function Documentation

#### 4.143.1.1 update()

```
void ecs::ScoreSystem::update (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [virtual]
```

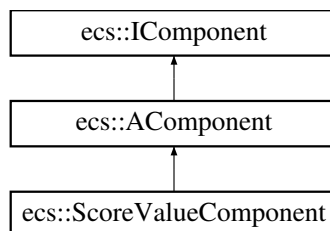
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/score/ScoreSystem.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/score/ScoreSystem.cpp

## 4.144 [ecs::ScoreValueComponent](#) Class Reference

Inheritance diagram for [ecs::ScoreValueComponent](#):



### Public Member Functions

- **ScoreValueComponent** (int scoreValue=0)
- int **getScoreValue** () const
- void **setScoreValue** (int scoreValue)

### Private Attributes

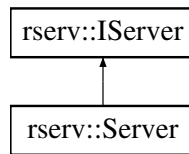
- int **\_scoreValue**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/ScoreValueComponent.hpp

## 4.145 rserv::Server Class Reference

Inheritance diagram for rserv::Server:



### Public Member Functions

- **Server** (std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void [init](#) () override
- void [start](#) () override
- void [stop](#) () override
- void [setConfig](#) (std::shared\_ptr< [ServerConfig](#) > config) override
- std::shared\_ptr< [ServerConfig](#) > [getConfig](#) () const override
- uint16\_t [getPort](#) () const override
- void [setPort](#) (uint16\_t port) override
- int [getState](#) () const override
- void [setState](#) (int state) override
- [operator int](#) () const noexcept override
- std::shared\_ptr< net::INetwork > [getNetwork](#) () const override
- void [setNetwork](#) (std::shared\_ptr< net::INetwork > network) override
- void [onClientConnected](#) (uint8\_t idClient) override
- void [onClientDisconnected](#) (uint8\_t idClient) override
- void [onPacketReceived](#) (uint8\_t idClient, const pm::IPacketManager &packet) override
- std::vector< uint8\_t > [getConnectedClients](#) () const override
- std::vector< asio::ip::udp::endpoint > [getConnectedClientEndpoints](#) () const override
- size\_t [getClientCount](#) () const override
- std::shared\_ptr< std::queue< std::tuple< uint8\_t, constants::EventType, double > > > [getEventQueue](#) () override
- bool [hasEvents](#) () const override
- void [processIncomingPackets](#) () override
- bool [processConnections](#) (std::pair< asio::ip::udp::endpoint, std::vector< uint8\_t > > client) override
- bool [processDisconnections](#) (uint8\_t idClient) override
- bool [processEvents](#) (uint8\_t idClient) override
- bool [processEndOfGame](#) (uint8\_t idClient) override
- bool [processWhoAmI](#) (uint8\_t idClient)
- bool [connectionPacket](#) (asio::ip::udp::endpoint endpoint)
- bool [gameStatePacket](#) ()
- bool [canStartPacket](#) ()
- std::vector< uint64\_t > [spawnPacket](#) (size\_t entity, const std::string prefabName)
- std::vector< uint64\_t > [deathPacket](#) (size\_t entity)
- void [setCurrentMap](#) (const std::vector< uint64\_t > &map)
- std::vector< uint64\_t > [getCurrentMap](#) () const
- bool [isGameStarted](#) () const
- bool [allClientsReady](#) () const
- uint32\_t [getSequenceNumber](#) () const
- std::shared\_ptr< pm::IPacketManager > [getPacketManager](#) () const
- void [incrementSequenceNumber](#) ()
- void [setResourceManager](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void [clearEntityDeltaCache](#) (uint8\_t clientId, uint32\_t entityId)

### Private Member Functions

- void **loadNetworkLibrary** ()
- void **loadBufferLibrary** ()
- void **loadPacketLibrary** ()
- std::vector< uint64\_t > **convertTagComponent** (std::shared\_ptr< [ecs::Registry](#) > registry, ecs::Entity i)
- std::vector< uint64\_t > **convertTransformComponent** (std::shared\_ptr< [ecs::Registry](#) > registry, ecs::Entity i)
- std::vector< uint64\_t > **convertSpeedComponent** (std::shared\_ptr< [ecs::Registry](#) > registry, ecs::Entity i)
- std::vector< uint64\_t > **convertHealthComponent** (std::shared\_ptr< [ecs::Registry](#) > registry, ecs::Entity i)
- std::vector< uint64\_t > **convertColliderComponent** (std::shared\_ptr< [ecs::Registry](#) > registry, ecs::Entity i)
- std::vector< uint64\_t > **convertShootStatComponent** (std::shared\_ptr< [ecs::Registry](#) > registry, ecs::Entity i)
- std::vector< uint64\_t > **convertScoreComponent** (std::shared\_ptr< [ecs::Registry](#) > registry, ecs::Entity i)
- std::vector< uint64\_t > **convertAIMovementPatternComponent** (std::shared\_ptr< [ecs::Registry](#) > registry, ecs::Entity i)
- std::vector< uint64\_t > **convertDamageComponent** (std::shared\_ptr< [ecs::Registry](#) > registry, ecs::Entity i)
- std::vector< uint64\_t > **convertLifetimeComponent** (std::shared\_ptr< [ecs::Registry](#) > registry, ecs::Entity i)
- std::vector< uint64\_t > **convertVelocityComponent** (std::shared\_ptr< [ecs::Registry](#) > registry, ecs::Entity i)
- std::vector< uint64\_t > **convertAIMoverTagComponent** (std::shared\_ptr< [ecs::Registry](#) > registry, ecs::Entity i)
- std::vector< uint64\_t > **convertAIShooterTagComponent** (std::shared\_ptr< [ecs::Registry](#) > registry, ecs::Entity i)
- std::vector< uint64\_t > **convertControllableTagComponent** (std::shared\_ptr< [ecs::Registry](#) > registry, ecs::Entity i)
- std::vector< uint64\_t > **convertEnemyProjectileTagComponent** (std::shared\_ptr< [ecs::Registry](#) > registry, ecs::Entity i)
- std::vector< uint64\_t > **convertGameZoneColliderTagComponent** (std::shared\_ptr< [ecs::Registry](#) > registry, ecs::Entity i)
- std::vector< uint64\_t > **convertMobTagComponent** (std::shared\_ptr< [ecs::Registry](#) > registry, ecs::Entity i)
- std::vector< uint64\_t > **convertObstacleTagComponent** (std::shared\_ptr< [ecs::Registry](#) > registry, ecs::Entity i)
- std::vector< uint64\_t > **convertPlayerProjectileTagComponent** (std::shared\_ptr< [ecs::Registry](#) > registry, ecs::Entity i)
- std::vector< uint64\_t > **convertShooterTagComponent** (std::shared\_ptr< [ecs::Registry](#) > registry, ecs::Entity i)
- std::vector< uint64\_t > **convertProjectilePassThroughTagComponent** (std::shared\_ptr< [ecs::Registry](#) > registry, ecs::Entity i)
- std::vector< uint64\_t > **convertProjectilePrefabComponent** (std::shared\_ptr< [ecs::Registry](#) > registry, ecs::Entity i)
- std::vector< uint64\_t > **convertNetworkIdComponent** (std::shared\_ptr< [ecs::Registry](#) > registry, ecs::Entity i)
- std::vector< uint64\_t > **convertGameZoneComponent** (std::shared\_ptr< [ecs::Registry](#) > registry, ecs::Entity i)

### Private Attributes

- [DLLoader](#)< createNetworkLib\_t > **\_networloader**
- [DLLoader](#)< createBuffer\_t > **\_bufferloader**
- [DLLoader](#)< createPacket\_t > **\_packetloader**

- uint8\_t **\_nextClientId**
- uint32\_t **\_sequenceNumber**
- std::vector< std::tuple< uint8\_t, asio::ip::udp::endpoint, std::string > > **\_clients**
- std::map< uint8\_t, bool > **\_clientsReady**
- std::shared\_ptr< [ServerConfig](#) > **\_config**
- std::shared\_ptr< net::INetwork > **\_network**
- std::shared\_ptr< IBuffer > **\_buffer**
- std::shared\_ptr< pm::IPacketManager > **\_packet**
- std::shared\_ptr< std::queue< std::tuple< uint8\_t, constants::EventType, double > > > **\_eventQueue**
- bool **\_gameStarted**
- std::shared\_ptr< [ResourceManager](#) > **\_resourceManager**
- std::chrono::steady\_clock::time\_point **\_lastGameStateTime**
- [ComponentDeltaTracker](#) **\_deltaTracker**
- std::vector< std::function< std::vector< uint64\_t >(std::shared\_ptr< [ecs::Registry](#) >, ecs::Entity)> > > **\_convertFunctions**

## 4.145.1 Member Function Documentation

### 4.145.1.1 getClientCount()

```
size_t rserv::Server::getClientCount () const [override], [virtual]
```

Implements [rserv::IServer](#).

### 4.145.1.2 getConfig()

```
std::shared_ptr< rserv::ServerConfig > rserv::Server::getConfig () const [override], [virtual]
```

Implements [rserv::IServer](#).

### 4.145.1.3 getConnectedClientEndpoints()

```
std::vector< asio::ip::udp::endpoint > rserv::Server::getConnectedClientEndpoints () const [override], [virtual]
```

Implements [rserv::IServer](#).

### 4.145.1.4 getConnectedClients()

```
std::vector< uint8_t > rserv::Server::getConnectedClients () const [override], [virtual]
```

Implements [rserv::IServer](#).

### 4.145.1.5 getEventQueue()

```
std::shared_ptr< std::queue< std::tuple< uint8_t, constants::EventType, double > > > rserv::Server::getEventQueue () [override], [virtual]
```

Implements [rserv::IServer](#).

#### 4.145.1.6 getNetwork()

```
std::shared_ptr< net::INetwork > rserv::Server::getNetwork () const [override], [virtual]
```

Implements [rserv::IServer](#).

#### 4.145.1.7 getPort()

```
uint16_t rserv::Server::getPort () const [override], [virtual]
```

Implements [rserv::IServer](#).

#### 4.145.1.8 getState()

```
int rserv::Server::getState () const [override], [virtual]
```

Implements [rserv::IServer](#).

#### 4.145.1.9 hasEvents()

```
bool rserv::Server::hasEvents () const [override], [virtual]
```

Implements [rserv::IServer](#).

#### 4.145.1.10 init()

```
void rserv::Server::init () [override], [virtual]
```

Implements [rserv::IServer](#).

#### 4.145.1.11 onClientConnected()

```
void rserv::Server::onClientConnected (  
    uint8_t idClient) [override], [virtual]
```

Implements [rserv::IServer](#).

#### 4.145.1.12 onClientDisconnected()

```
void rserv::Server::onClientDisconnected (  
    uint8_t idClient) [override], [virtual]
```

Implements [rserv::IServer](#).



#### 4.145.1.13 onPacketReceived()

```
void rserv::Server::onPacketReceived (
    uint8_t idClient,
    const pm::IPacketManager & packet) [override], [virtual]
```

Implements [rserv::IServer](#).

#### 4.145.1.14 operator int()

```
rserv::Server::operator int () const [override], [virtual], [noexcept]
```

Implements [rserv::IServer](#).

#### 4.145.1.15 processConnections()

```
bool rserv::Server::processConnections (
    std::pair< asio::ip::udp::endpoint, std::vector< uint8_t > > client) [override],
[virtual]
```

Implements [rserv::IServer](#).

#### 4.145.1.16 processDisconnections()

```
bool rserv::Server::processDisconnections (
    uint8_t idClient) [override], [virtual]
```

Implements [rserv::IServer](#).

#### 4.145.1.17 processEndOfGame()

```
bool rserv::Server::processEndOfGame (
    uint8_t idClient) [override], [virtual]
```

Implements [rserv::IServer](#).

#### 4.145.1.18 processEvents()

```
bool rserv::Server::processEvents (
    uint8_t idClient) [override], [virtual]
```

Implements [rserv::IServer](#).

#### 4.145.1.19 processIncomingPackets()

```
void rserv::Server::processIncomingPackets () [override], [virtual]
```

Implements [rserv::IServer](#).

#### 4.145.1.20 setConfig()

```
void rserv::Server::setConfig (
    std::shared_ptr< ServerConfig > config) [override], [virtual]
```

Implements [rserv::IServer](#).

#### 4.145.1.21 setNetwork()

```
void rserv::Server::setNetwork (
    std::shared_ptr< net::INetwork > network) [override], [virtual]
```

Implements [rserv::IServer](#).

#### 4.145.1.22 setPort()

```
void rserv::Server::setPort (
    uint16_t port) [override], [virtual]
```

Implements [rserv::IServer](#).

#### 4.145.1.23 setState()

```
void rserv::Server::setState (
    int state) [override], [virtual]
```

Implements [rserv::IServer](#).

#### 4.145.1.24 start()

```
void rserv::Server::start () [override], [virtual]
```

Implements [rserv::IServer](#).

#### 4.145.1.25 stop()

```
void rserv::Server::stop () [override], [virtual]
```

Implements [rserv::IServer](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/Server.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/ECSConversions.cpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/Server.cpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/ServerLibsLoading.cpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/ServerReceivePacket.cpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/ServerSentPacket.cpp

## 4.146 rserv::ServerConfig Class Reference

### Public Member Functions

- int **getState** () const
- void **setPort** (uint16\_t port)
- uint16\_t **getPort** () const
- void **setState** (int state)
- void **setNbClients** (int nbClients)
- int **getNbClients** () const
- std::string **getIp** () const
- void **setIp** (std::string ip)
- void **setIsDebug** (bool isDebug)
- bool **getIsDebug** () const

### Private Attributes

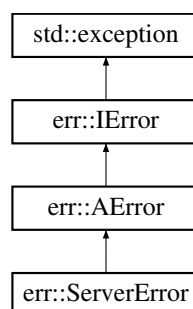
- int **\_state**
- uint16\_t **\_port**
- int **\_nbClients**
- std::string **\_ip**
- bool **\_isDebug**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/ServerConfig.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/ServerConfig.cpp

## 4.147 err::ServerError Class Reference

Inheritance diagram for err::ServerError:



### Public Types

- enum **ErrorCode** {  
**UNKNOWN** = 1000 , **CONNECTION\_FAILED** = 1001 , **TIMEOUT** = 1002 , **INVALID\_REQUEST** = 1003 ,  
**INTERNAL\_ERROR** = 1004 , **LIBRARY\_LOAD\_FAILED** = 1005 , **CONFIG\_ERROR** = 1006 }

## Public Member Functions

- **ServerError** (const std::string &message, ErrorCode code=UNKNOWN)
- std::string [getType](#) () const noexcept override

## Public Member Functions inherited from [err::AError](#)

- **AError** (const std::string &message, int code=0)
- const char \* [what](#) () const noexcept override
- int [getCode](#) () const noexcept override
- std::string [getDetails](#) () const noexcept override

## Additional Inherited Members

## Protected Attributes inherited from [err::AError](#)

- std::string **m\_message**
- int **m\_code**

## 4.147.1 Member Function Documentation

### 4.147.1.1 [getType\(\)](#)

```
std::string err::ServerError::getType () const [override], [virtual], [noexcept]
```

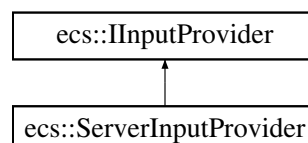
Implements [err::AError](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/ServerError.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/ServerError.cpp

## 4.148 [ecs::ServerInputProvider](#) Class Reference

Inheritance diagram for [ecs::ServerInputProvider](#):



## Public Member Functions

- float [getAxisValue](#) (event\_t axis, size\_t clientID=0) override
- bool [isActionPressed](#) (InputAction action, size\_t clientID=0) override
- float [getActionAxis](#) (InputAction action, size\_t clientID=0) override
- [InputMapping](#) [getInputMapping](#) (size\_t clientID=0) const override
- void [setAxisValue](#) (ecs::InputAction action, float value, size\_t clientID=0)
- void [addClientInputMapping](#) (size\_t clientID, size\_t identity, const [InputMapping](#) &mapping)
- void [updateInputFromEvent](#) (size\_t clientID, constants::EventType eventType, float value)
- std::vector< size\_t > [getConnectedClients](#) () const

## Private Types

- using [InputHandler](#) = void (ServerInputProvider::\*)(size\_t, float)

## Private Member Functions

- void [handleUp](#) (size\_t clientID, float value)
- void [handleDown](#) (size\_t clientID, float value)
- void [handleLeft](#) (size\_t clientID, float value)
- void [handleRight](#) (size\_t clientID, float value)
- void [handleStop](#) (size\_t clientID, float value)
- void [handleShoot](#) (size\_t clientID, float value)

## Private Attributes

- std::vector< std::tuple< size\_t, size\_t, [InputMapping](#) > > [\\_inputMapping](#)
- std::map< size\_t, std::map< ecs::InputAction, float > > [\\_clientAxisValues](#)
- std::vector< InputHandler > [\\_inputHandlers](#)

## Additional Inherited Members

## Public Types inherited from [ecs::IInputProvider](#)

- using [event\\_t](#) = gfx::EventType

### 4.148.1 Member Function Documentation

#### 4.148.1.1 [getActionAxis\(\)](#)

```
float ecs::ServerInputProvider::getActionAxis (
    InputAction action,
    size_t clientID = 0) [override], [virtual]
```

Implements [ecs::IInputProvider](#).

#### 4.148.1.2 `getAxisValue()`

```
float ecs::ServerInputProvider::getAxisValue (
    event_t axis,
    size_t clientID = 0) [override], [virtual]
```

Implements [ecs::IInputProvider](#).

#### 4.148.1.3 `getInputMapping()`

```
InputMapping ecs::ServerInputProvider::getInputMapping (
    size_t clientID = 0) const [override], [virtual]
```

Implements [ecs::IInputProvider](#).

#### 4.148.1.4 `isActionPressed()`

```
bool ecs::ServerInputProvider::isActionPressed (
    InputAction action,
    size_t clientID = 0) [override], [virtual]
```

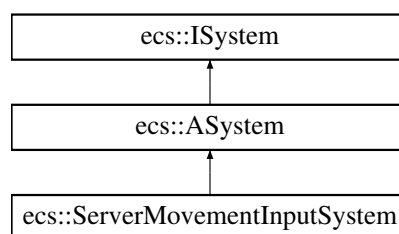
Implements [ecs::IInputProvider](#).

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

- [/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/initResourcesManager/ServerInputProvider.hpp](#)
- [/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/initResourcesManager/ServerInputProvider.cpp](#)

## 4.149 `ecs::ServerMovementInputSystem` Class Reference

Inheritance diagram for `ecs::ServerMovementInputSystem`:



### Public Member Functions

- void [update](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

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

- void [updateSystem](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

## Private Member Functions

- [math::Vector2f](#) [getMovementDirection](#) (std::shared\_ptr< [IInputProvider](#) > inputProvider, size\_t clientID) const
- void [updateInputIntent](#) (std::shared\_ptr< [Registry](#) > registry, Entity entityId, const [math::Vector2f](#) &direction)
- [math::Vector2f](#) [getAnalogStickInput](#) (std::shared\_ptr< [IInputProvider](#) > inputProvider, size\_t clientID) const
- [math::Vector2f](#) [normalizeDirection](#) (const [math::Vector2f](#) &direction) const

## 4.149.1 Member Function Documentation

### 4.149.1.1 update()

```
void ecs::ServerMovementInputSystem::update (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [virtual]
```

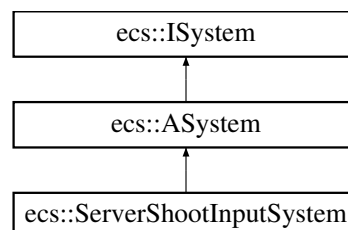
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/systems/input/ServerMovementInput↵  
System.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/systems/input/ServerMovementInput↵  
System.cpp

## 4.150 ecs::ServerShootInputSystem Class Reference

Inheritance diagram for [ecs::ServerShootInputSystem](#):



## Public Member Functions

- void [update](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

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

- void [updateSystem](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

## Private Member Functions

- void [updateShootIntent](#) (std::shared\_ptr< [Registry](#) > registry, Entity entityId)

## 4.150.1 Member Function Documentation

### 4.150.1.1 update()

```
void ecs::ServerShootInputSystem::update (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [virtual]
```

Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/systems/input/ServerShootInput↵  
System.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/systems/input/ServerShootInput↵  
System.cpp

## 4.151 SettingsConfig Class Reference

### Public Types

- enum class **ScreenResolution** {  
    **RES\_800x600** = 0 , **RES\_1024x768** = 1 , **RES\_1280x720** = 2 , **RES\_1920x1080** = 3 ,  
    **FULLSCREEN** = 4 }

### Public Member Functions

- int **getColorBlindnessState** () const
- void **setColorBlindnessState** (int state)
- float **getBrightnessValue** () const
- void **setBrightnessValue** (float value)
- bool **isHighContrastEnabled** () const
- void **setHighContrastEnabled** (bool enabled)
- ui::UIScale **getUIScale** () const
- void **setUIScale** (ui::UIScale scale)
- float **getMusicVolume** () const
- void **setMusicVolume** (float volume)
- float **getSoundVolume** () const
- void **setSoundVolume** (float volume)



- ScreenResolution **getScreenResolution** () const
- void **setScreenResolution** (ScreenResolution resolution)
- int **getTargetFPS** () const
- void **setTargetFPS** (int fps)
- float **getRenderQuality** () const
- void **setRenderQuality** (float quality)
- std::string **getScreenResolutionName** (ScreenResolution resolution) const
- std::pair< int, int > **getScreenResolutionSize** (ScreenResolution resolution) const
- bool **isFullscreen** (ScreenResolution resolution) const
- void **saveAccessibility** (const std::string &filepath=constants::ACCESSIBILITY\_FILE\_PATH)
- void **loadAccessibility** (const std::string &filepath=constants::ACCESSIBILITY\_FILE\_PATH)
- void **saveSettings** (const std::string &filepath=constants::SETTINGS\_FILE\_PATH)
- void **loadSettings** (const std::string &filepath=constants::SETTINGS\_FILE\_PATH)

### Private Attributes

- int **\_colorBlindnessState** = 0
- float **\_brightnessValue** = 1.0f
- bool **\_highContrastEnabled** = false
- ui::UIScale **\_uiScale** = ui::UIScale::Normal
- float **\_musicVolume** = 100.0f
- float **\_soundVolume** = 100.0f
- ScreenResolution **\_screenResolution** = ScreenResolution::RES\_1920x1080
- int **\_targetFPS** = 60
- float **\_renderQuality** = 1.0f

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/SettingsConfig.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/SettingsConfig.cpp

## 4.152 SettingsManager Class Reference

### Public Member Functions

- **SettingsManager** (std::shared\_ptr< [ecs::InputMappingManager](#) > mappingManager, std::shared\_ptr< [ecs::IInputProvider](#) > inputProvider, std::shared\_ptr< [SettingsConfig](#) > settingsConfig)
- void **loadAll** ()
- void **saveAll** ()
- void **saveKeybinds** ()
- void **loadKeybinds** ()
- void **saveAccessibility** ()
- void **loadAccessibility** ()
- void **saveSettings** ()
- void **loadSettings** ()
- void **applyAccessibilityToWindow** (std::shared\_ptr< gfx::IWindow > window)

### Private Attributes

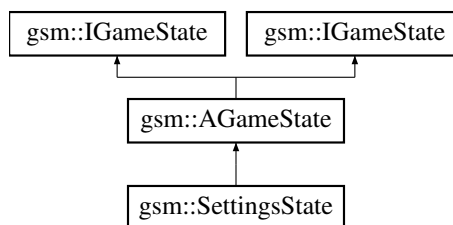
- `std::shared_ptr< ecs::InputMappingManager > _mappingManager`
- `std::shared_ptr< ecs::InputProvider > _inputProvider`
- `std::shared_ptr< SettingsConfig > _settingsConfig`

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

- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/SettingsManager.hpp`
- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/SettingsManager.cpp`

## 4.153 `gsm::SettingsState` Class Reference

Inheritance diagram for `gsm::SettingsState`:



### Public Member Functions

- **SettingsState** (`std::shared_ptr< IGameStateMachine > gsm`, `std::shared_ptr< ResourceManager > resourceManager`)
- `void enter ()` override
- `void update (float deltaTime)` override
- `void exit ()` override

### Public Member Functions inherited from `gsm::AGameState`

- **AGameState** (`std::shared_ptr< IGameStateMachine > gsm`, `std::shared_ptr< ResourceManager > resourceManager`)
- `std::vector< std::shared_ptr< ecs::ISystem > > getSystems ()` const override
- **AGameState** (`std::shared_ptr< IGameStateMachine > gsm`, `std::shared_ptr< ResourceManager > resourceManager`)
- `std::vector< std::shared_ptr< ecs::ISystem > > getSystems ()` const override

### Private Member Functions

- void **renderUI** ()
- void **cycleColorBlindnessFilter** ()
- void **toggleHighContrastFilter** ()
- void **updateBrightnessFilter** (float value)
- void **applyColorBlindnessFilter** (int state)
- void **applyHighContrastFilter** (bool enabled)
- void **cycleUIScale** ()
- void **updateMusicVolume** (float value)
- void **updateSoundVolume** (float value)
- void **updateToggleValue** (bool value)
- void **cycleScreenResolution** ()
- void **updateTargetFPS** (int fps)
- void **updateRenderQuality** (float quality)
- void **setScreenResolution** (SettingsConfig::ScreenResolution resolution)
- void **updateResolutionButtonColors** (SettingsConfig::ScreenResolution current)
- void **startKeyRebind** (ecs::RemappableAction action, bool rebindPrimary, std::shared\_ptr< ui::Button > button)
- void **handleKeyRebind** (gfx::EventType newKey)
- void **updateKeyBindingButtonText** (std::shared\_ptr< ui::Button > button, ecs::RemappableAction action, bool isPrimary)
- std::string **getRemappableActionName** (ecs::RemappableAction action) const
- std::string **getScreenResolutionText** (SettingsConfig::ScreenResolution resolution)
- std::string **getColorBlindnessText** (int state)
- std::string **getUIScaleText** (ui::UIScale scale)

### Private Attributes

- std::unique\_ptr< MouseInputHandler > **\_mouseHandler**
- std::shared\_ptr< ui::Button > **\_backButton**
- std::shared\_ptr< ui::Button > **\_highContrastButton**
- std::shared\_ptr< ui::Button > **\_colorBlindnessButton**
- std::shared\_ptr< ui::Slider > **\_brightnessSlider**
- std::shared\_ptr< ui::Slider > **\_musicVolumeSlider**
- std::shared\_ptr< ui::Slider > **\_soundVolumeSlider**
- std::shared\_ptr< ui::ToggleSwitch > **\_toggleSwitch**
- std::shared\_ptr< ui::Text > **\_toggleLabel**
- std::shared\_ptr< ui::UILayout > **\_toggleLayout**
- std::vector< std::shared\_ptr< ui::Button > > **\_resolutionButtons**
- std::shared\_ptr< ui::Slider > **\_fpsSlider**
- std::shared\_ptr< ui::Slider > **\_renderQualitySlider**
- std::shared\_ptr< ui::Button > **\_scaleButton**
- std::unique\_ptr< ui::UIManager > **\_uiManager**
- std::shared\_ptr< ui::UILayout > **\_settingsLayout**
- std::shared\_ptr< ui::UILayout > **\_leftColumnLayout**
- std::shared\_ptr< ui::UILayout > **\_rightColumnLayout**
- std::shared\_ptr< ui::UILayout > **\_centerColumnLayout**
- std::shared\_ptr< ui::UILayout > **\_titleLabel**
- std::shared\_ptr< ui::Background > **\_background**
- **math::Vector2f** **\_savedViewCenter**
- std::shared\_ptr< SettingsManager > **\_settingsManager**
- std::shared\_ptr< ui::UILayout > **\_moveUpLayout**
- std::shared\_ptr< ui::Text > **\_moveUpLabel**

- `std::shared_ptr< ui::Button > _moveUpPrimaryButton`
- `std::shared_ptr< ui::Button > _moveUpSecondaryButton`
- `std::shared_ptr< ui::UILayout > _moveDownLayout`
- `std::shared_ptr< ui::Text > _moveDownLabel`
- `std::shared_ptr< ui::Button > _moveDownPrimaryButton`
- `std::shared_ptr< ui::Button > _moveDownSecondaryButton`
- `std::shared_ptr< ui::UILayout > _moveLeftLayout`
- `std::shared_ptr< ui::Text > _moveLeftLabel`
- `std::shared_ptr< ui::Button > _moveLeftPrimaryButton`
- `std::shared_ptr< ui::Button > _moveLeftSecondaryButton`
- `std::shared_ptr< ui::UILayout > _moveRightLayout`
- `std::shared_ptr< ui::Text > _moveRightLabel`
- `std::shared_ptr< ui::Button > _moveRightPrimaryButton`
- `std::shared_ptr< ui::Button > _moveRightSecondaryButton`
- `std::shared_ptr< ui::UILayout > _shootLayout`
- `std::shared_ptr< ui::Text > _shootLabel`
- `std::shared_ptr< ui::Button > _shootPrimaryButton`
- `std::shared_ptr< ui::Button > _shootSecondaryButton`
- `bool _isWaitingForKey = false`
- `std::optional< ecs::RemappableAction > _actionToRebind`
- `bool _rebindingPrimary = true`
- `std::string _rebindLabel`
- `std::shared_ptr< ui::Button > _buttonToUpdate`
- `gfx::EventType _originalKey = gfx::EventType::NOTHING`

#### Additional Inherited Members

#### Protected Member Functions inherited from [gsm::AGameState](#)

- `void addSystem (std::shared_ptr< ecs::ISystem > system) override`
- `void addSystem (std::shared_ptr< ecs::ISystem > system) override`

#### Protected Attributes inherited from [gsm::AGameState](#)

- `std::weak_ptr< IGameStateMachine > _gsm`
- `std::shared_ptr< ResourceManager > _resourceManager`
- `std::vector< std::shared_ptr< ecs::ISystem > > _systems`

### 4.153.1 Member Function Documentation

#### 4.153.1.1 enter()

```
void gsm::SettingsState::enter () [override], [virtual]
```

Reimplemented from [gsm::AGameState](#).

#### 4.153.1.2 exit()

```
void gsm::SettingsState::exit () [override], [virtual]
```

Reimplemented from [gsm::AGameState](#).

#### 4.153.1.3 update()

```
void gsm::SettingsState::update (
    float deltaTime) [override], [virtual]
```

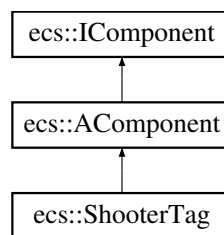
Reimplemented from [gsm::AGameState](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Settings/Settings↔State.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Settings/Settings↔State.cpp

## 4.154 ecs::ShooterTag Class Reference

Inheritance diagram for ecs::ShooterTag:

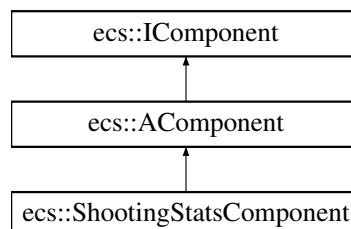


The documentation for this class was generated from the following file:

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/tags/ShooterTag.hpp

## 4.155 ecs::ShootingStatsComponent Class Reference

Inheritance diagram for ecs::ShootingStatsComponent:



### Public Member Functions

- **ShootingStatsComponent** (float fireRate=1.0f, const [MultiShotPattern](#) &pattern=[MultiShotPattern](#)())
- float **getFireRate** () const
- void **setFireRate** (float fireRate)
- [MultiShotPattern](#) **getMultiShotPattern** () const
- void **setMultiShotPattern** (const [MultiShotPattern](#) &pattern)
- float **getCooldownTimer** () const
- void **setCooldownTimer** (float timer)
- bool **canShoot** () const
- void **updateCooldown** (float deltaTime)
- void **resetCooldown** ()

### Private Attributes

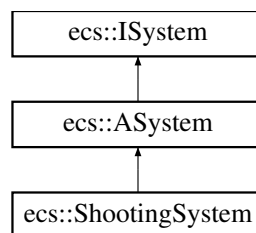
- float **\_fireRate**
- [MultiShotPattern](#) **\_multiShotPattern**
- float **\_cooldownTimer**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/ShootingStatsComponent.hpp

## 4.156 ecs::ShootingSystem Class Reference

Inheritance diagram for ecs::ShootingSystem:



### Public Member Functions

- void **update** (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

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

- void **updateSystem** (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

### Private Member Functions

- void **spawnProjectile** (std::shared\_ptr< [Registry](#) > registry, std::shared\_ptr< [ResourceManager](#) > resourceManager, const std::string &prefabName, const [math::Vector2f](#) &position, float angle, ecs::Entity shooterEntity)
- [math::Vector2f](#) **calculateProjectileVelocity** (float angle, float speed)

## 4.156.1 Member Function Documentation

### 4.156.1.1 update()

```
void ecs::ShootingSystem::update (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [virtual]
```

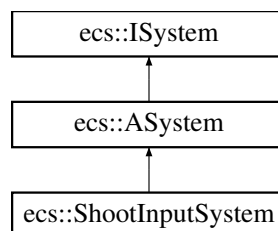
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/shooting/ShootingSystem.↔  
hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/shooting/ShootingSystem.↔  
cpp

## 4.157 ecs::ShootInputSystem Class Reference

Inheritance diagram for ecs::ShootInputSystem:



### Public Member Functions

- void **update** (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

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

- void **updateSystem** (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

## 4.157.1 Member Function Documentation

### 4.157.1.1 update()

```
void ecs::ShootInputSystem::update (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [virtual]
```

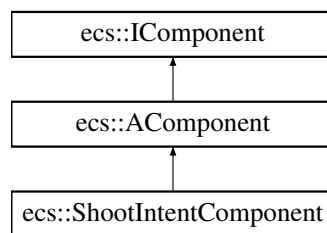
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/input/ShootInputSystem.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/input/ShootInputSystem.cpp

## 4.158 ecs::ShootIntentComponent Class Reference

Inheritance diagram for ecs::ShootIntentComponent:



### Public Member Functions

- **ShootIntentComponent** (float angle=0.0f)
- void **setAngle** (float angle)
- float **getAngle** () const

### Private Attributes

- float **\_angle**
- [math::Vector2f](#) **\_position**

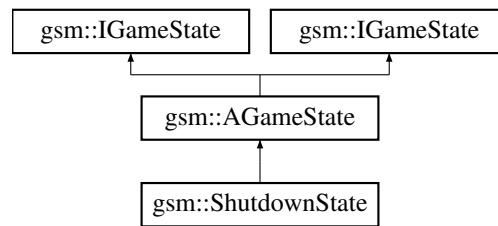
The documentation for this class was generated from the following file:

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/temporary/ShootIntentComponent.hpp



## 4.159 `gsm::ShutdownState` Class Reference

Inheritance diagram for `gsm::ShutdownState`:



### Public Member Functions

- **ShutdownState** (`std::shared_ptr< IGameStateMachine > gsm`, `std::shared_ptr< ResourceManager > resourceManager`)
- void [enter](#) () override

### Public Member Functions inherited from `gsm::AGameState`

- **AGameState** (`std::shared_ptr< IGameStateMachine > gsm`, `std::shared_ptr< ResourceManager > resourceManager`)
- void [update](#) (float deltaTime) override
- void [exit](#) () override
- `std::vector< std::shared_ptr< ecs::ISystem > > getSystems ()` const override
- **AGameState** (`std::shared_ptr< IGameStateMachine > gsm`, `std::shared_ptr< ResourceManager > resourceManager`)
- void [update](#) (float deltaTime) override
- void [exit](#) () override
- `std::vector< std::shared_ptr< ecs::ISystem > > getSystems ()` const override

### Additional Inherited Members

### Protected Member Functions inherited from `gsm::AGameState`

- void [addSystem](#) (`std::shared_ptr< ecs::ISystem > system`) override
- void [addSystem](#) (`std::shared_ptr< ecs::ISystem > system`) override

### Protected Attributes inherited from `gsm::AGameState`

- `std::weak_ptr< IGameStateMachine > \_gsm`
- `std::shared_ptr< ResourceManager > \_resourceManager`
- `std::vector< std::shared_ptr< ecs::ISystem > > \_systems`

### 4.159.1 Member Function Documentation

#### 4.159.1.1 enter()

```
void gsm::ShutdownState::enter () [override], [virtual]
```

Reimplemented from [gsm::AGameState](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/scenes/Shutdown/Shutdown↵  
State.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/scenes/Shutdown/Shutdown↵  
State.cpp

## 4.160 Signal Class Reference

### Static Public Member Functions

- static void **signalHandler** (int signum)
- static void **setupSignalHandlers** ()

### Static Public Attributes

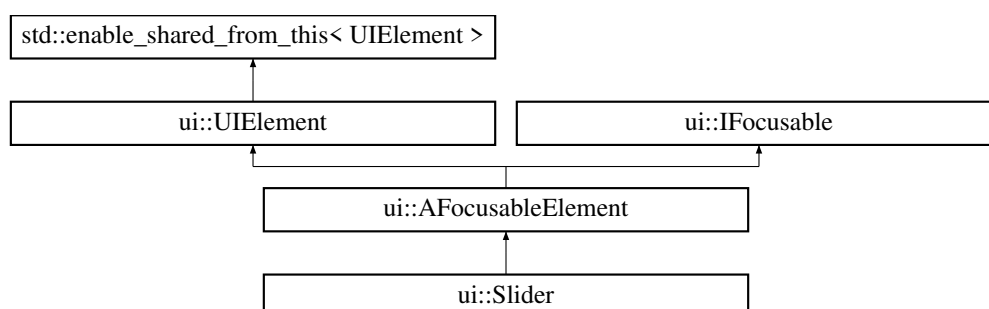
- static volatile sig\_atomic\_t **stopFlag** = 0

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Signal/Signal.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Signal/Signal.cpp

## 4.161 ui::Slider Class Reference

Inheritance diagram for ui::Slider:



## Public Member Functions

- **Slider** (std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void **setMinValue** (float minValue)
- void **setMaxValue** (float maxValue)
- void **setValue** (float value)
- float **getValue** () const
- float **getMinValue** () const
- float **getMaxValue** () const
- void **setStep** (float step)
- float **getStep** () const
- void **setLabel** (const std::string &label)
- const std::string & **getLabel** () const
- void **setLabelColor** (const gfx::color\_t &color)
- void **setFontPath** (const std::string &fontPath)
- void **setBaseFontSize** (size\_t fontSize)
- size\_t **getBaseFontSize** () const
- void **setShowPercentage** (bool show)
- void **setTrackColor** (const gfx::color\_t &color)
- void **setFillColor** (const gfx::color\_t &color)
- void **setHandleColor** (const gfx::color\_t &color)
- void **setHandleHoveredColor** (const gfx::color\_t &color)
- void **setHandleFocusedColor** (const gfx::color\_t &color)
- void **setOnValueChanged** (std::function< void(float)> callback)
- virtual void **render** () override
- virtual void **handleInput** (const [math::Vector2f](#) &mousePos, bool mousePressed) override
- virtual void **onActivated** () override
- virtual bool **onNavigateLeft** () override
- virtual bool **onNavigateRight** () override
- void **incrementValue** ()
- void **decrementValue** ()

## Public Member Functions inherited from [ui::AFocusableElement](#)

- **AFocusableElement** (std::shared\_ptr< [ResourceManager](#) > resourceManager)
- virtual void **setFocused** (bool focused) override
- virtual bool **isFocused** () const override
- virtual bool **canBeFocused** () const override
- virtual void **onFocusGained** () override
- virtual void **onFocusLost** () override
- void **setOnFocusGained** (std::function< void()> callback)
- void **setOnFocusLost** (std::function< void()> callback)
- void **setOnActivated** (std::function< void()> callback)

## Public Member Functions inherited from [ui::UIElement](#)

- **UIElement** (std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void **setPosition** (const [math::Vector2f](#) &position)
- void **setSize** (const [math::Vector2f](#) &size)
- [math::Vector2f](#) **getPosition** () const
- [math::Vector2f](#) **getSize** () const
- [math::Vector2f](#) **getAbsolutePosition** () const
- [math::Vector2f](#) **getAbsoluteSize** () const
- void **setVisible** (bool visible)
- bool **isVisible** () const
- void **setState** (UIState state)
- UIState **getState** () const
- virtual void **setScale** (UIScale scale)
- UIScale **getScale** () const
- void **setParent** (std::weak\_ptr< [UIElement](#) > parent)
- std::shared\_ptr< [UIElement](#) > **getParent** () const
- void **addChild** (std::shared\_ptr< [UIElement](#) > child)
- void **removeChild** (std::shared\_ptr< [UIElement](#) > child)
- const std::vector< std::shared\_ptr< [UIElement](#) > > & **getChildren** () const
- virtual bool **containsPoint** (const [math::Vector2f](#) &point) const
- void **setOnClick** (std::function< void()> callback)
- void **setOnHover** (std::function< void()> callback)
- void **setOnRelease** (std::function< void()> callback)
- virtual void **update** (float deltaTime)

## Private Member Functions

- float **getNormalizedValue** () const
- void **setNormalizedValue** (float normalized)
- [gfx::color\\_t](#) **getCurrentHandleColor** () const
- [size\\_t](#) **getFontSize** () const
- float **getHandleRadius** () const
- float **getTrackHeight** () const
- float **getLabelHeight** () const

## Private Attributes

- float **\_minValue** = 0.0f
- float **\_maxValue** = 1.0f
- float **\_value** = 0.5f
- float **\_step** = 0.1f
- float **\_visualNormalizedValue** = 0.5f
- std::string **\_label**
- [gfx::color\\_t](#) **\_labelColor** = colors::SLIDER\_LABEL
- std::string **\_fontPath** = "assets/fonts/arial.ttf"
- [size\\_t](#) **\_baseFontSize** = constants::BUTTON\_FONT\_SIZE\_BASE
- float **\_outlineThickness** = 2.0f
- bool **\_showPercentage** = true
- [gfx::color\\_t](#) **\_trackColor** = colors::SLIDER\_TRACK
- [gfx::color\\_t](#) **\_fillColor** = colors::SLIDER\_FILL
- [gfx::color\\_t](#) **\_handleColor** = colors::SLIDER\_HANDLE
- [gfx::color\\_t](#) **\_handleHoveredColor** = colors::SLIDER\_HANDLE\_HOVER
- [gfx::color\\_t](#) **\_handleFocusedColor** = colors::SLIDER\_HANDLE\_FOCUSED
- std::function< void(float)> **\_onValueChanged**
- bool **\_isDragging** = false
- bool **\_wasMousePressed** = false

## Additional Inherited Members

### Protected Member Functions inherited from [ui::AFocusableElement](#)

- virtual void **onFocusStateChanged** (bool focused)

### Protected Member Functions inherited from [ui::UIElement](#)

- std::pair< int, int > **getWindowSize** () const
- std::pair< int, int > **getLogicalSize** () const
- float **getScaleFactor** () const

### Protected Attributes inherited from [ui::AFocusableElement](#)

- bool **\_focused** = false
- bool **\_pressedInside** = false
- bool **\_wasPressed** = false
- std::function< void()> **\_onFocusGained**
- std::function< void()> **\_onFocusLost**
- std::function< void()> **\_onActivated**

### Protected Attributes inherited from [ui::UIElement](#)

- std::weak\_ptr< [ResourceManager](#) > **\_resourceManager**
- [math::Vector2f](#) **\_position**
- [math::Vector2f](#) **\_size**
- bool **\_visible** = true
- UIState **\_state** = UIState::Normal
- UIScale **\_scale** = UIScale::Normal
- std::weak\_ptr< [UIElement](#) > **\_parent**
- std::vector< std::shared\_ptr< [UIElement](#) > > **\_children**
- bool **\_pressedInside** = false
- bool **\_wasPressed** = false
- std::function< void()> **\_onClick**
- std::function< void()> **\_onHover**
- std::function< void()> **\_onRelease**

## 4.161.1 Member Function Documentation

### 4.161.1.1 handleInput()

```
void ui::Slider::handleInput (
    const math::Vector2f & mousePos,
    bool mousePressed) [override], [virtual]
```

Reimplemented from [ui::AFocusableElement](#).

#### 4.161.1.2 onActivated()

```
void ui::Slider::onActivated () [override], [virtual]
```

Reimplemented from [ui::AFocusableElement](#).

#### 4.161.1.3 onNavigateLeft()

```
bool ui::Slider::onNavigateLeft () [override], [virtual]
```

Reimplemented from [ui::IFocusable](#).

#### 4.161.1.4 onNavigateRight()

```
bool ui::Slider::onNavigateRight () [override], [virtual]
```

Reimplemented from [ui::IFocusable](#).

#### 4.161.1.5 render()

```
void ui::Slider::render () [override], [virtual]
```

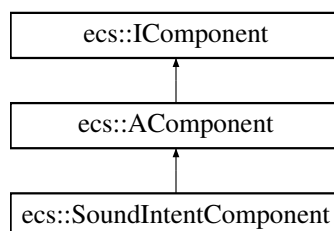
Reimplemented from [ui::UIElement](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/focusable/Slider.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/focusable/Slider.cpp

## 4.162 ecs::SoundIntentComponent Class Reference

Inheritance diagram for `ecs::SoundIntentComponent`:



### Public Member Functions

- **SoundIntentComponent** (const std::string &soundPath="", float volume=100.0f)
- std::string **getSoundPath** () const
- void **setSoundPath** (const std::string &soundPath)
- float **getVolume** () const
- void **setVolume** (float volume)

**Private Attributes**

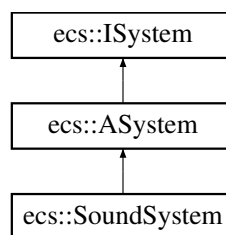
- std::string **\_soundPath**
- float **\_volume**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/temporary/SoundIntentComponent.hpp

**4.163 ecs::SoundSystem Class Reference**

Inheritance diagram for ecs::SoundSystem:

**Protected Member Functions**

- void **update** (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

**Additional Inherited Members****Public Member Functions inherited from [ecs::ASystem](#)**

- void **updateSystem** (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

**4.163.1 Member Function Documentation****4.163.1.1 update()**

```

void ecs::SoundSystem::update (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [protected], [virtual]
  
```

Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/audio/SoundSystem.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/audio/SoundSystem.cpp

## 4.164 ecs::SpatialGrid Class Reference

### Public Member Functions

- **SpatialGrid** (float worldWidth=constants::MAX\_WIDTH, float worldHeight=constants::MAX\_HEIGHT, float cellSize=constants::SPATIAL\_GRID\_CELL\_SIZE, float padding=constants::SPATIAL\_GRID\_PADDING)
- void **clear** ()
- void **insert** (Entity entityId, const [math::FRect](#) &bounds)
- std::vector< Entity > **query** (const [math::FRect](#) &bounds) const
- std::vector< std::pair< Entity, Entity > > **getPotentialPairs** () const
- void **setCellSize** (float cellSize)
- void **setOffset** (float offsetX, float offsetY)
- float **getCellSize** () const
- size\_t **getNumCols** () const
- size\_t **getNumRows** () const
- float **getOffsetX** () const
- float **getOffsetY** () const

### Private Member Functions

- size\_t **getCellIndex** (float x, float y) const
- std::vector< size\_t > **getCellIndices** (const [math::FRect](#) &bounds) const

### Private Attributes

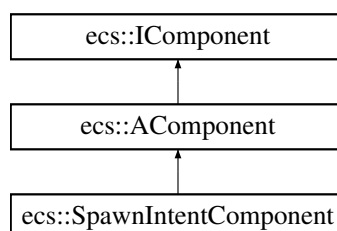
- float **\_worldWidth**
- float **\_worldHeight**
- float **\_cellSize**
- float **\_padding**
- float **\_offsetX**
- float **\_offsetY**
- size\_t **\_numCols**
- size\_t **\_numRows**
- std::vector< std::vector< Entity > > **\_cells**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/SpatialGrid/SpatialGrid.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/SpatialGrid/SpatialGrid.cpp

## 4.165 ecs::SpawnIntentComponent Class Reference

Inheritance diagram for ecs::SpawnIntentComponent:





**Public Member Functions**

- **SpawnIntentComponent** (const std::string &prefabName, const [math::Vector2f](#) &position, float gameViewXTrigger=0.0f)
- **SpawnIntentComponent** (const std::string &prefabName, const [math::Vector2f](#) &position, const [EntityCreationContext](#) &context, float gameViewXTrigger=0.0f)
- void **setPrefabName** (const std::string &prefabName)
- std::string **getPrefabName** () const
- void **setPosition** (const [math::Vector2f](#) &position)
- [math::Vector2f](#) **getPosition** () const
- void **setCreationContext** (const [EntityCreationContext](#) &context)
- [EntityCreationContext](#) **getCreationContext** () const
- void **setGameViewXTrigger** (const float &gameViewXTrigger)
- float **getGameViewXTrigger** () const

**Private Attributes**

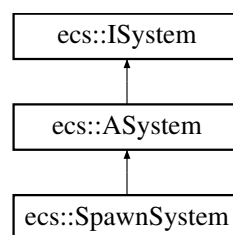
- std::string **\_prefabName**
- [math::Vector2f](#) **\_position**
- [EntityCreationContext](#) **\_creationContext**
- float **\_gameViewXTrigger**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/temporary/SpawnIntentComponent.hpp

## 4.166 ecs::SpawnSystem Class Reference

Inheritance diagram for ecs::SpawnSystem:

**Public Member Functions**

- void **update** (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

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

- void **updateSystem** (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

## 4.166.1 Member Function Documentation

### 4.166.1.1 update()

```
void ecs::SpawnSystem::update (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [virtual]
```

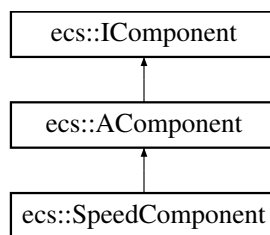
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/spawn/SpawnSystem.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/spawn/SpawnSystem.cpp

## 4.167 ecs::SpeedComponent Class Reference

Inheritance diagram for `ecs::SpeedComponent`:



### Public Member Functions

- **SpeedComponent** (float speed=constants::BASE\_SPEED)
- float **getSpeed** () const
- void **setSpeed** (float speed)

### Private Attributes

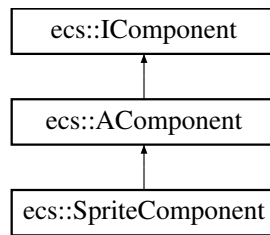
- float **\_speed**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/SpeedComponent.hpp

## 4.168 ecs::SpriteComponent Class Reference

Inheritance diagram for ecs::SpriteComponent:



### Public Member Functions

- **SpriteComponent** (const std::string &texturePath)
- const std::string & **getTexturePath** () const
- void **setTexturePath** (const std::string &path)
- bool **isValid** () const

### Private Attributes

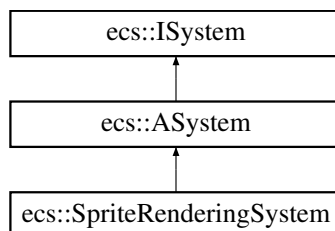
- std::string **\_texturePath**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/SpriteComponent.↔  
hpp

## 4.169 ecs::SpriteRenderingSystem Class Reference

Inheritance diagram for ecs::SpriteRenderingSystem:



### Protected Member Functions

- void **update** (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

### Additional Inherited Members

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

- void [updateSystem](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

## 4.169.1 Member Function Documentation

### 4.169.1.1 update()

```
void ecs::SpriteRenderingSystem::update (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [protected], [virtual]
```

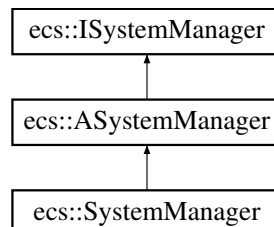
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/SpriteRendering↔  
System.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/SpriteRendering↔  
System.cpp

## 4.170 ecs::SystemManager Class Reference

Inheritance diagram for `ecs::SystemManager`:



### Additional Inherited Members

### Public Member Functions inherited from [ecs::ASystemManager](#)

- void [updateAllSystems](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override
- void [addSystem](#) (std::shared\_ptr< [ISystem](#) > system) override
- void [removeSystem](#) (std::shared\_ptr< [ISystem](#) > system) override

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/systemManager/System↔  
Manager.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/systemManager/System↔  
Manager.cpp

## 4.171 TagRegistry Class Reference

### Public Member Functions

- `template<typename T>`  
void **registerTag** (const std::string &tagName)
- bool **hasTag** (std::shared\_ptr< [ecs::Registry](#) > registry, ecs::Entity entity, const std::string &tagName) const
- std::vector< std::string > **getTags** (std::shared\_ptr< [ecs::Registry](#) > registry, ecs::Entity entity) const

### Static Public Member Functions

- static const [TagRegistry](#) & **getInstance** ()

### Private Member Functions

- **TagRegistry** (const [TagRegistry](#) &)=delete
- [TagRegistry](#) & **operator=** (const [TagRegistry](#) &)=delete
- void **initializeTags** ()

### Private Attributes

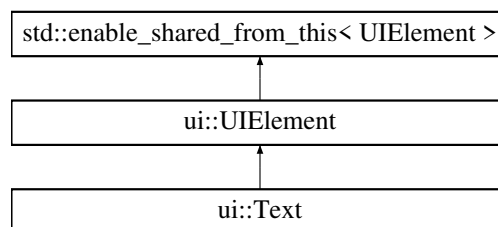
- std::unordered\_map< std::string, std::function< bool(std::shared\_ptr< [ecs::Registry](#) >, ecs::Entity)> > > **tagCheckers**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/interactions/TagRegistry.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/interactions/TagRegistry.cpp

## 4.172 ui::Text Class Reference

Inheritance diagram for ui::Text:



## Public Member Functions

- **Text** (std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void **render** () override
- void **update** (float deltaTime) override
- void **setScale** (UIScale scale) override
- void **setText** (const std::string &text)
- std::string **getText** () const
- void **setTextColor** (const gfx::color\_t &color)
- void **setFontSize** (unsigned int size)
- void **setFontPath** (const std::string &path)
- void **setOutlineColor** (const gfx::color\_t &color)
- void **setOutlineThickness** (float thickness)

## Public Member Functions inherited from [ui::UIElement](#)

- **UIElement** (std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void **setPosition** (const [math::Vector2f](#) &position)
- void **setSize** (const [math::Vector2f](#) &size)
- [math::Vector2f](#) **getPosition** () const
- [math::Vector2f](#) **getSize** () const
- [math::Vector2f](#) **getAbsolutePosition** () const
- [math::Vector2f](#) **getAbsoluteSize** () const
- void **setVisible** (bool visible)
- bool **isVisible** () const
- void **setState** (UIState state)
- UIState **getState** () const
- UIScale **getScale** () const
- void **setParent** (std::weak\_ptr< [UIElement](#) > parent)
- std::shared\_ptr< [UIElement](#) > **getParent** () const
- void **addChild** (std::shared\_ptr< [UIElement](#) > child)
- void **removeChild** (std::shared\_ptr< [UIElement](#) > child)
- const std::vector< std::shared\_ptr< [UIElement](#) > > & **getChildren** () const
- virtual void **handleInput** (const [math::Vector2f](#) &mousePos, bool mousePressed)
- virtual bool **containsPoint** (const [math::Vector2f](#) &point) const
- void **setOnClick** (std::function< void()> callback)
- void **setOnHover** (std::function< void()> callback)
- void **setOnRelease** (std::function< void()> callback)

## Private Attributes

- std::string **\_text**
- gfx::color\_t **\_textColor**
- unsigned int **\_fontSize**
- unsigned int **\_baseFontSize**
- std::string **\_fontPath**
- gfx::color\_t **\_outlineColor**
- float **\_outlineThickness**

## Additional Inherited Members

### Protected Member Functions inherited from [ui::UIElement](#)

- `std::pair< int, int > getWindowSize () const`
- `std::pair< int, int > getLogicalSize () const`
- `float getScaleFactor () const`

### Protected Attributes inherited from [ui::UIElement](#)

- `std::weak_ptr< ResourceManager > _resourceManager`
- `math::Vector2f _position`
- `math::Vector2f _size`
- `bool _visible = true`
- `UIState _state = UIState::Normal`
- `UIScale _scale = UIScale::Normal`
- `std::weak_ptr< UIElement > _parent`
- `std::vector< std::shared_ptr< UIElement > > _children`
- `bool _pressedInside = false`
- `bool _wasPressed = false`
- `std::function< void()> _onClick`
- `std::function< void()> _onHover`
- `std::function< void()> _onRelease`

## 4.172.1 Member Function Documentation

### 4.172.1.1 render()

```
void ui::Text::render () [override], [virtual]
```

Reimplemented from [ui::UIElement](#).

### 4.172.1.2 setScale()

```
void ui::Text::setScale (
    UIScale scale) [override], [virtual]
```

Reimplemented from [ui::UIElement](#).

### 4.172.1.3 update()

```
void ui::Text::update (
    float deltaTime) [override], [virtual]
```

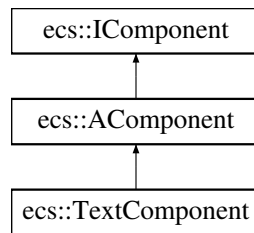
Reimplemented from [ui::UIElement](#).

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

- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/Text.hpp`
- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/Text.cpp`

## 4.173 ecs::TextComponent Class Reference

Inheritance diagram for ecs::TextComponent:



### Public Member Functions

- **TextComponent** (const std::string &text, const std::string &fontPath, gfx::color\_t color=gfx::color\_t{255, 255, 255})
- const std::string & **getText** () const
- const std::string & **getFontPath** () const
- const gfx::color\_t & **getColor** () const
- void **setText** (const std::string &text)
- void **setFontPath** (const std::string &fontPath)
- void **setColor** (const gfx::color\_t &color)

### Private Attributes

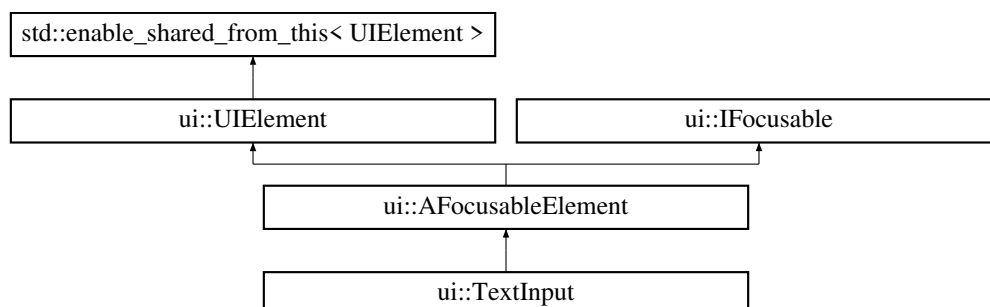
- std::string **\_text**
- std::string **\_fontPath**
- gfx::color\_t **\_color**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/TextComponent.↔hpp

## 4.174 ui::TextInput Class Reference

Inheritance diagram for ui::TextInput:





**Public Member Functions**

- **TextInput** (std::shared\_ptr< [ResourceManager](#) > resourceManager)
- virtual void **render** () override
- void **setText** (const std::string &text)
- const std::string & **getText** () const
- void **setPlaceholder** (const std::string &placeholder)
- const std::string & **getPlaceholder** () const
- void **setTextColor** (const gfx::color\_t &color)
- void **setPlaceholderColor** (const gfx::color\_t &color)
- void **setFontPath** (const std::string &fontPath)
- void **setBaseFontSize** (size\_t fontSize)
- size\_t **getBaseFontSize** () const
- void **setOnTextChanged** (std::function< void(const std::string &)> callback)
- void **setOnSubmit** (std::function< void(const std::string &)> callback)
- virtual void **handleInput** (const [math::Vector2f](#) &mousePos, bool mousePressed) override
- void **handleKeyboardInput** (gfx::EventType event)
- void **handleTextInput** (const std::string &text)
- virtual void **update** (float deltaTime) override

**Public Member Functions inherited from [ui::AFocusableElement](#)**

- **AFocusableElement** (std::shared\_ptr< [ResourceManager](#) > resourceManager)
- virtual void **setFocused** (bool focused) override
- virtual bool **isFocused** () const override
- virtual bool **canBeFocused** () const override
- virtual void **onFocusGained** () override
- virtual void **onFocusLost** () override
- virtual void **onActivated** () override
- void **setOnFocusGained** (std::function< void()> callback)
- void **setOnFocusLost** (std::function< void()> callback)
- void **setOnActivated** (std::function< void()> callback)

**Public Member Functions inherited from [ui::UIElement](#)**

- **UIElement** (std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void **setPosition** (const [math::Vector2f](#) &position)
- void **setSize** (const [math::Vector2f](#) &size)
- [math::Vector2f](#) **getPosition** () const
- [math::Vector2f](#) **getSize** () const
- [math::Vector2f](#) **getAbsolutePosition** () const
- [math::Vector2f](#) **getAbsoluteSize** () const
- void **setVisible** (bool visible)
- bool **isVisible** () const
- void **setState** (UIState state)
- UIState **getState** () const
- virtual void **setScale** (UIScale scale)
- UIScale **getScale** () const
- void **setParent** (std::weak\_ptr< [UIElement](#) > parent)
- std::shared\_ptr< [UIElement](#) > **getParent** () const
- void **addChild** (std::shared\_ptr< [UIElement](#) > child)
- void **removeChild** (std::shared\_ptr< [UIElement](#) > child)
- const std::vector< std::shared\_ptr< [UIElement](#) > > & **getChildren** () const
- virtual bool **containsPoint** (const [math::Vector2f](#) &point) const
- void **setOnClick** (std::function< void()> callback)
- void **setOnHover** (std::function< void()> callback)
- void **setOnRelease** (std::function< void()> callback)

## Public Member Functions inherited from [ui::IFocusable](#)

- virtual bool **onNavigateLeft** ()
- virtual bool **onNavigateRight** ()

## Private Member Functions

- void **insertChar** (char c)
- void **deleteChar** ()
- void **moveCursorLeft** ()
- void **moveCursorRight** ()
- size\_t **getFontSize** () const
- void **updateCursorBlink** (float deltaTime)
- gfx::color\_t **getCurrentColor** () const

## Private Attributes

- std::string **\_text**
- std::string **\_placeholder**
- size\_t **\_cursorPosition** = 0
- float **\_cursorBlinkTimer** = 0.0f
- bool **\_showCursor** = true
- gfx::color\_t **\_textColor** = {0, 0, 0}
- gfx::color\_t **\_placeholderColor** = {128, 128, 128}
- std::string **\_fontPath** = "assets/fonts/arial.ttf"
- size\_t **\_baseFontSize** = 24
- std::function< void(const std::string &)> **\_onTextChanged**
- std::function< void(const std::string &)> **\_onSubmit**
- gfx::color\_t **\_normalColor** = colors::WHITE
- gfx::color\_t **\_hoveredColor** = colors::LIGHT\_GRAY
- gfx::color\_t **\_pressedColor** = colors::DARK\_GRAY
- gfx::color\_t **\_disabledColor** = colors::UI\_DISABLED
- gfx::color\_t **\_focusedColor** = colors::UI\_FOCUSED

## Additional Inherited Members

## Protected Member Functions inherited from [ui::AFocusableElement](#)

- virtual void **onFocusStateChanged** (bool focused)

## Protected Member Functions inherited from [ui::UIElement](#)

- std::pair< int, int > **getWindowSize** () const
- std::pair< int, int > **getLogicalSize** () const
- float **getScaleFactor** () const

## Protected Attributes inherited from [ui::AFocusableElement](#)

- bool **\_focused** = false
- bool **\_pressedInside** = false
- bool **\_wasPressed** = false
- std::function< void()> **\_onFocusGained**
- std::function< void()> **\_onFocusLost**
- std::function< void()> **\_onActivated**

## Protected Attributes inherited from [ui::UIElement](#)

- std::weak\_ptr< [ResourceManager](#) > **\_resourceManager**
- [math::Vector2f](#) **\_position**
- [math::Vector2f](#) **\_size**
- bool **\_visible** = true
- UIState **\_state** = UIState::Normal
- UIScale **\_scale** = UIScale::Normal
- std::weak\_ptr< [UIElement](#) > **\_parent**
- std::vector< std::shared\_ptr< [UIElement](#) > > **\_children**
- bool **\_pressedInside** = false
- bool **\_wasPressed** = false
- std::function< void()> **\_onClick**
- std::function< void()> **\_onHover**
- std::function< void()> **\_onRelease**

### 4.174.1 Member Function Documentation

#### 4.174.1.1 `handleInput()`

```
void ui::TextInput::handleInput (
    const math::Vector2f & mousePos,
    bool mousePressed) [override], [virtual]
```

Reimplemented from [ui::AFocusableElement](#).

#### 4.174.1.2 `render()`

```
void ui::TextInput::render () [override], [virtual]
```

Reimplemented from [ui::UIElement](#).

#### 4.174.1.3 `update()`

```
void ui::TextInput::update (
    float deltaTime) [override], [virtual]
```

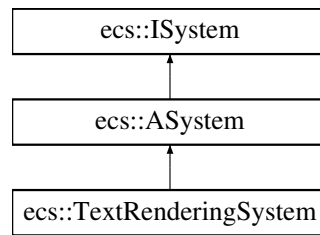
Reimplemented from [ui::UIElement](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/focusable/TextInput.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/focusable/TextInput.cpp

## 4.175 ecs::TextRenderingSystem Class Reference

Inheritance diagram for ecs::TextRenderingSystem:



### Protected Member Functions

- void [update](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

### Additional Inherited Members

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

- void [updateSystem](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

## 4.175.1 Member Function Documentation

### 4.175.1.1 update()

```
void ecs::TextRenderingSystem::update (  
    std::shared_ptr< ResourceManager > resourceManager,  
    std::shared_ptr< Registry > registry,  
    float deltaTime) [override], [protected], [virtual]
```

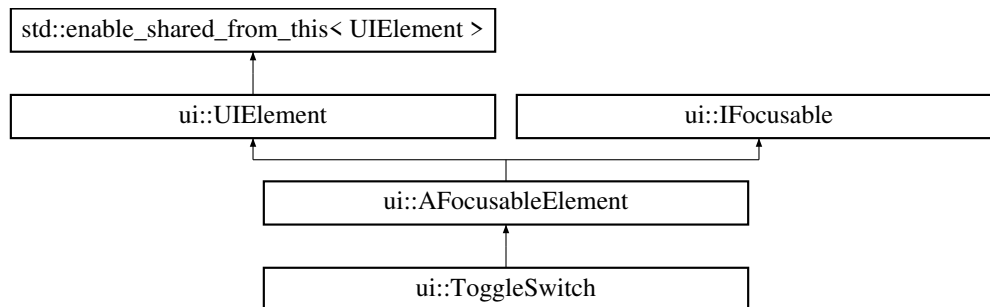
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/TextRendering↵  
System.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/TextRendering↵  
System.cpp

## 4.176 ui::ToggleSwitch Class Reference

Inheritance diagram for ui::ToggleSwitch:



### Public Member Functions

- **ToggleSwitch** (std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void **setValue** (bool value)
- bool **getValue** () const
- void **setFontPath** (const std::string &fontPath)
- void **setBaseFontSize** (size\_t fontSize)
- size\_t **getBaseFontSize** () const
- void **setOnText** (const std::string &text)
- void **setOffText** (const std::string &text)
- void **setTrackColor** (const gfx::color\_t &color)
- void **setHandleColor** (const gfx::color\_t &color)
- void **setHandleHoveredColor** (const gfx::color\_t &color)
- void **setHandleFocusedColor** (const gfx::color\_t &color)
- void **setOnColor** (const gfx::color\_t &color)
- void **setOffColor** (const gfx::color\_t &color)
- void **setOnValueChanged** (std::function< void(bool)> callback)
- virtual void **render** () override
- virtual void **handleInput** (const [math::Vector2f](#) &mousePos, bool mousePressed) override
- virtual bool **containsPoint** (const [math::Vector2f](#) &point) const override

### Public Member Functions inherited from [ui::AFocusableElement](#)

- **AFocusableElement** (std::shared\_ptr< [ResourceManager](#) > resourceManager)
- virtual void **setFocused** (bool focused) override
- virtual bool **isFocused** () const override
- virtual bool **canBeFocused** () const override
- virtual void **onFocusGained** () override
- virtual void **onFocusLost** () override
- virtual void **onActivated** () override
- void **setOnFocusGained** (std::function< void()> callback)
- void **setOnFocusLost** (std::function< void()> callback)
- void **setOnActivated** (std::function< void()> callback)

## Public Member Functions inherited from [ui::UIElement](#)

- **UIElement** (std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void **setPosition** (const [math::Vector2f](#) &position)
- void **setSize** (const [math::Vector2f](#) &size)
- [math::Vector2f](#) **getPosition** () const
- [math::Vector2f](#) **getSize** () const
- [math::Vector2f](#) **getAbsolutePosition** () const
- [math::Vector2f](#) **getAbsoluteSize** () const
- void **setVisible** (bool visible)
- bool **isVisible** () const
- void **setState** (UIState state)
- UIState **getState** () const
- virtual void **setScale** (UIScale scale)
- UIScale **getScale** () const
- void **setParent** (std::weak\_ptr< [UIElement](#) > parent)
- std::shared\_ptr< [UIElement](#) > **getParent** () const
- void **addChild** (std::shared\_ptr< [UIElement](#) > child)
- void **removeChild** (std::shared\_ptr< [UIElement](#) > child)
- const std::vector< std::shared\_ptr< [UIElement](#) > > & **getChildren** () const
- void **setOnClick** (std::function< void()> callback)
- void **setOnHover** (std::function< void()> callback)
- void **setOnRelease** (std::function< void()> callback)
- virtual void **update** (float deltaTime)

## Public Member Functions inherited from [ui::IFocusable](#)

- virtual bool **onNavigateLeft** ()
- virtual bool **onNavigateRight** ()

## Private Attributes

- bool **\_value** = false
- std::string **\_fontPath** = "assets/fonts/arial.ttf"
- size\_t **\_baseFontSize** = constants::BUTTON\_FONT\_SIZE\_BASE
- std::string **\_onText** = "ON"
- std::string **\_offText** = "OFF"
- gfx::color\_t **\_trackColor** = colors::TOGGLE\_TRACK
- gfx::color\_t **\_handleColor** = colors::TOGGLE\_HANDLE
- gfx::color\_t **\_handleHoveredColor** = colors::TOGGLE\_HANDLE\_HOVER
- gfx::color\_t **\_handleFocusedColor** = colors::TOGGLE\_HANDLE\_FOCUSED
- gfx::color\_t **\_onColor** = colors::TOGGLE\_ON
- gfx::color\_t **\_offColor** = colors::TOGGLE\_OFF
- std::function< void(bool)> **\_onValueChanged**
- bool **\_isHovered** = false

## Additional Inherited Members

## Protected Member Functions inherited from [ui::AFocusableElement](#)

- virtual void **onFocusStateChanged** (bool focused)

## Protected Member Functions inherited from [ui::UIElement](#)

- `std::pair< int, int > getWindowSize () const`
- `std::pair< int, int > getLogicalSize () const`
- `float getScaleFactor () const`

## Protected Attributes inherited from [ui::AFocusableElement](#)

- `bool _focused = false`
- `bool _pressedInside = false`
- `bool _wasPressed = false`
- `std::function< void()> _onFocusGained`
- `std::function< void()> _onFocusLost`
- `std::function< void()> _onActivated`

## Protected Attributes inherited from [ui::UIElement](#)

- `std::weak_ptr< ResourceManager > _resourceManager`
- `math::Vector2f _position`
- `math::Vector2f _size`
- `bool _visible = true`
- `UIState _state = UIState::Normal`
- `UIScale _scale = UIScale::Normal`
- `std::weak_ptr< UIElement > _parent`
- `std::vector< std::shared_ptr< UIElement > > _children`
- `bool _pressedInside = false`
- `bool _wasPressed = false`
- `std::function< void()> _onClick`
- `std::function< void()> _onHover`
- `std::function< void()> _onRelease`

## 4.176.1 Member Function Documentation

### 4.176.1.1 containsPoint()

```
bool ui::ToggleSwitch::containsPoint (
    const math::Vector2f & point) const [override], [virtual]
```

Reimplemented from [ui::UIElement](#).

### 4.176.1.2 handleInput()

```
void ui::ToggleSwitch::handleInput (
    const math::Vector2f & mousePos,
    bool mousePressed) [override], [virtual]
```

Reimplemented from [ui::AFocusableElement](#).

### 4.176.1.3 render()

```
void ui::ToggleSwitch::render () [override], [virtual]
```

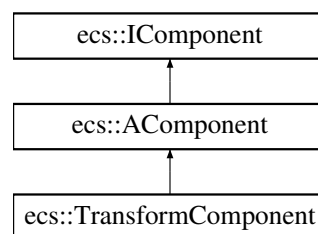
Reimplemented from [ui::UIElement](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/focusable/ToggleSwitch.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/focusable/ToggleSwitch.cpp

## 4.177 ecs::TransformComponent Class Reference

Inheritance diagram for `ecs::TransformComponent`:



### Public Member Functions

- **TransformComponent** ([math::Vector2f](#) position=[math::Vector2f](#)(0.0f, 0.0f), float rotation=0.0f, [math::Vector2f](#) scale=[math::Vector2f](#)(1.0f, 1.0f))
- [math::Vector2f](#) **getPosition** () const
- void **setPosition** ([math::Vector2f](#) position)
- float **getRotation** () const
- void **setRotation** (float rotation)
- [math::Vector2f](#) **getScale** () const
- void **setScale** ([math::Vector2f](#) scale)

### Private Attributes

- [math::Vector2f](#) **\_position**
- float **\_rotation**
- [math::Vector2f](#) **\_scale**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/TransformComponent.hpp



## 4.178 ecs::Transition Struct Reference

### Public Attributes

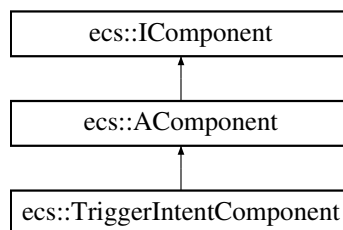
- std::string **from**
- std::string **to**
- std::vector< [AnimationCondition](#) > **conditions**
- bool **playRewind** = false

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

- [/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/AnimationComponent.hpp](#)

## 4.179 ecs::TriggerIntentComponent Class Reference

Inheritance diagram for ecs::TriggerIntentComponent:



### Public Member Functions

- **TriggerIntentComponent** (Entity self=0, Entity other=0)
- Entity **getSelf** () const
- void **setSelf** (Entity self)
- Entity **getOther** () const
- void **setOther** (Entity other)

### Private Attributes

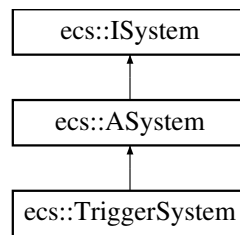
- Entity **\_self**
- Entity **\_other**

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

- [/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/temporary/TriggerIntentComponent.hpp](#)

## 4.180 ecs::TriggerSystem Class Reference

Inheritance diagram for ecs::TriggerSystem:



### Public Member Functions

- void [update](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

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

- void [updateSystem](#) (std::shared\_ptr< [ResourceManager](#) > resourceManager, std::shared\_ptr< [Registry](#) > registry, float deltaTime) override

### Private Member Functions

- void [buildSpatialGrid](#) (std::shared\_ptr< [Registry](#) > registry)
- bool [checkCollision](#) (const [TransformComponent](#) &transformA, const [ColliderComponent](#) &colliderA, const [TransformComponent](#) &transformB, const [ColliderComponent](#) &colliderB)
- bool [shouldCollide](#) (std::shared\_ptr< [Registry](#) > registry, size\_t entityA, const [ColliderComponent](#) &colliderA, size\_t entityB)

### Private Attributes

- [SpatialGrid](#) \_spatialGrid

## 4.180.1 Member Function Documentation

### 4.180.1.1 update()

```

void ecs::TriggerSystem::update (
    std::shared_ptr< ResourceManager > resourceManager,
    std::shared_ptr< Registry > registry,
    float deltaTime) [override], [virtual]
  
```

Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/interactions/TriggerSystem.↵  
hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/interactions/TriggerSystem.↵  
cpp

## 4.181 ui::UIElement Class Reference

Inheritance diagram for ui::UIElement:



### Public Member Functions

- **UIElement** (std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void **setPosition** (const [math::Vector2f](#) &position)
- void **setSize** (const [math::Vector2f](#) &size)
- [math::Vector2f](#) **getPosition** () const
- [math::Vector2f](#) **getSize** () const
- [math::Vector2f](#) **getAbsolutePosition** () const
- [math::Vector2f](#) **getAbsoluteSize** () const
- void **setVisible** (bool visible)
- bool **isVisible** () const
- void **setState** (UIState state)
- UIState **getState** () const
- virtual void **setScale** (UIScale scale)
- UIScale **getScale** () const
- void **setParent** (std::weak\_ptr< [UIElement](#) > parent)
- std::shared\_ptr< [UIElement](#) > **getParent** () const
- void **addChild** (std::shared\_ptr< [UIElement](#) > child)
- void **removeChild** (std::shared\_ptr< [UIElement](#) > child)
- const std::vector< std::shared\_ptr< [UIElement](#) > > & **getChildren** () const
- virtual void **handleInput** (const [math::Vector2f](#) &mousePos, bool mousePressed)
- virtual bool **containsPoint** (const [math::Vector2f](#) &point) const
- void **setOnClick** (std::function< void()> callback)
- void **setOnHover** (std::function< void()> callback)
- void **setOnRelease** (std::function< void()> callback)
- virtual void **render** ()
- virtual void **update** (float deltaTime)

### Protected Member Functions

- std::pair< int, int > **getWindowSize** () const
- std::pair< int, int > **getLogicalSize** () const
- float **getScaleFactor** () const

### Protected Attributes

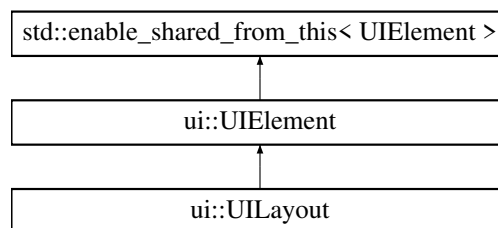
- `std::weak_ptr< ResourceManager > _resourceManager`
- `math::Vector2f _position`
- `math::Vector2f _size`
- `bool _visible = true`
- `UIState _state = UIState::Normal`
- `UIScale _scale = UIScale::Normal`
- `std::weak_ptr< UIElement > _parent`
- `std::vector< std::shared_ptr< UIElement > > _children`
- `bool _pressedInside = false`
- `bool _wasPressed = false`
- `std::function< void()> _onClick`
- `std::function< void()> _onHover`
- `std::function< void()> _onRelease`

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

- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/base/UIElement.hpp`
- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/base/UIElement.cpp`

## 4.182 ui::UILayout Class Reference

Inheritance diagram for `ui::UILayout`:



### Public Member Functions

- **UILayout** (`std::shared_ptr< ResourceManager > resourceManager`, `const LayoutConfig &config=LayoutConfig()`)
- void **addElement** (`std::shared_ptr< UIElement > element`)
- void **removeElement** (`std::shared_ptr< UIElement > element`)
- void **clearElements** ()
- void **setDirection** (`LayoutDirection direction`)
- void **setAlignment** (`LayoutAlignment alignment`)
- void **setSpacing** (`float spacing`)
- void **setPadding** (`const math::Vector2f &padding`)
- void **setOffset** (`const math::Vector2f &offset`)
- void **setAutoResize** (`bool autoResize`)
- void **setAnchor** (`AnchorX anchorX`, `AnchorY anchorY`)
- `LayoutDirection` **getDirection** () const
- `LayoutAlignment` **getAlignment** () const
- `float` **getSpacing** () const
- `math::Vector2f` **getPadding** () const
- `bool` **isAutoResize** () const
- void **updateLayout** ()
- void **setScale** (`UIScale scale`) override
- void **render** () override
- void **update** (`float deltaTime`) override
- `float` **getScaledSpacing** () const
- void **applyAnchor** ()

## Public Member Functions inherited from [ui::UIElement](#)

- **UIElement** (std::shared\_ptr< [ResourceManager](#) > resourceManager)
- void **setPosition** (const [math::Vector2f](#) &position)
- void **setSize** (const [math::Vector2f](#) &size)
- [math::Vector2f](#) **getPosition** () const
- [math::Vector2f](#) **getSize** () const
- [math::Vector2f](#) **getAbsolutePosition** () const
- [math::Vector2f](#) **getAbsoluteSize** () const
- void **setVisible** (bool visible)
- bool **isVisible** () const
- void **setState** (UIState state)
- UIState **getState** () const
- UIScale **getScale** () const
- void **setParent** (std::weak\_ptr< [UIElement](#) > parent)
- std::shared\_ptr< [UIElement](#) > **getParent** () const
- void **addChild** (std::shared\_ptr< [UIElement](#) > child)
- void **removeChild** (std::shared\_ptr< [UIElement](#) > child)
- const std::vector< std::shared\_ptr< [UIElement](#) > > & **getChildren** () const
- virtual void **handleInput** (const [math::Vector2f](#) &mousePos, bool mousePressed)
- virtual bool **containsPoint** (const [math::Vector2f](#) &point) const
- void **setOnClick** (std::function< void()> callback)
- void **setOnHover** (std::function< void()> callback)
- void **setOnRelease** (std::function< void()> callback)

## Private Member Functions

- void **calculatePositions** ()
- float **getTotalSize** () const
- [math::Vector2f](#) **calculateElementPosition** (size\_t index, float totalSize) const

## Private Attributes

- [LayoutConfig](#) **\_config**
- std::vector< std::shared\_ptr< [UIElement](#) > > **\_layoutElements**

## Additional Inherited Members

## Protected Member Functions inherited from [ui::UIElement](#)

- std::pair< int, int > **getWindowSize** () const
- std::pair< int, int > **getLogicalSize** () const
- float **getScaleFactor** () const

## Protected Attributes inherited from [ui::UIElement](#)

- `std::weak_ptr< ResourceManager > _resourceManager`
- `math::Vector2f _position`
- `math::Vector2f _size`
- `bool _visible = true`
- `UIState _state = UIState::Normal`
- `UIScale _scale = UIScale::Normal`
- `std::weak_ptr< UIElement > _parent`
- `std::vector< std::shared_ptr< UIElement > > _children`
- `bool _pressedInside = false`
- `bool _wasPressed = false`
- `std::function< void()> _onClick`
- `std::function< void()> _onHover`
- `std::function< void()> _onRelease`

### 4.182.1 Member Function Documentation

#### 4.182.1.1 `render()`

```
void ui::UILayout::render () [override], [virtual]
```

Reimplemented from [ui::UIElement](#).

#### 4.182.1.2 `setScale()`

```
void ui::UILayout::setScale (
    UIScale scale) [override], [virtual]
```

Reimplemented from [ui::UIElement](#).

#### 4.182.1.3 `update()`

```
void ui::UILayout::update (
    float deltaTime) [override], [virtual]
```

Reimplemented from [ui::UIElement](#).

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

- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/core/UILayout.hpp`
- `/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/core/UILayout.cpp`

## 4.183 ui::UIManager Class Reference

### Public Member Functions

- void **addElement** (std::shared\_ptr< [UIElement](#) > element)
- void **removeElement** (std::shared\_ptr< [UIElement](#) > element)
- void **clearElements** ()
- void **update** (float deltaTime)
- void **render** ()
- void **handleMouseDown** (const [math::Vector2f](#) &mousePos, bool mousePressed)
- bool **handleNavigationInput** (ecs::InputAction action)
- bool **handleNavigationInputs** (std::shared\_ptr< [ecs::IInputProvider](#) > inputProvider, float deltaTime)
- void **handleKeyboardInput** (gfx::EventType event)
- void **handleTextInput** (const std::string &text)
- std::shared\_ptr< [UINavigationManager](#) > **getNavigationManager** ()
- void **setNavigationEnabled** (bool enabled)
- bool **isNavigationEnabled** () const
- bool **focusFirstElement** ()
- void **clearFocus** ()
- std::shared\_ptr< [IFocusable](#) > **getFocusedElement** () const
- void **setGlobalScale** (UIScale scale)
- void **cycleGlobalScale** ()
- UIScale **getGlobalScale** () const
- void **setOnBack** (std::function< void()> callback)
- bool **isMouseHoveringAnyElement** (const [math::Vector2f](#) &mousePos) const

### Private Member Functions

- bool **hasMouseMoved** (const [math::Vector2f](#) &mousePos)
- void **refreshNavigationElements** ()

### Private Attributes

- std::vector< std::shared\_ptr< [UIElement](#) > > **\_elements**
- std::shared\_ptr< [UINavigationManager](#) > **\_navigationManager**
- [math::Vector2f](#) **\_lastMousePos**
- bool **\_mouseMovementDetected**
- float **\_navigationCooldown** = 0.0f
- UIScale **\_globalScale** = UIScale::Normal
- std::function< void()> **\_onBack**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/manager/UIManager.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/manager/UIManager.cpp

## 4.184 ui::UINavigationManager Class Reference

### Public Member Functions

- void **addFocusableElement** (std::shared\_ptr< [IFocusable](#) > element)
- void **removeFocusableElement** (std::shared\_ptr< [IFocusable](#) > element)
- void **clearFocusableElements** ()
- bool **handleNavigationInput** (ecs::InputAction action)
- bool **setFocus** (std::shared\_ptr< [IFocusable](#) > element)
- std::shared\_ptr< [IFocusable](#) > **getFocusedElement** () const
- void **clearFocus** ()
- bool **focusFirstElement** ()
- bool **focusNextElement** ()
- bool **focusPreviousElement** ()
- void **setNavigationEnabled** (bool enabled)
- bool **isNavigationEnabled** () const
- void **setOnFocusChanged** (std::function< void(std::shared\_ptr< [IFocusable](#) >)> callback)
- void **onMouseMovement** ()
- void **enableFocus** ()
- bool **isFocusDisabled** () const

### Private Member Functions

- void **cleanupExpiredElements** ()
- int **getCurrentFocusedIndex** () const
- bool **navigateInDirection** (NavigationDirection direction)

### Private Attributes

- std::vector< std::weak\_ptr< [IFocusable](#) > > **\_focusableElements**
- std::weak\_ptr< [IFocusable](#) > **\_currentFocused**
- bool **\_navigationEnabled**
- bool **\_focusDisabled**
- std::function< void(std::shared\_ptr< [IFocusable](#) >)> **\_onFocusChanged**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/navigation/UINavigationManager.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/navigation/UINavigationManager.cpp

## 4.185 Utils Class Reference

### Public Member Functions

- void **helper** ()
- void **parseCli** (int ac, char \*\*av, std::shared\_ptr< [ClientNetwork](#) > clientNetwork)
- void **helper** ()
- void **parsCli** (int ac, char \*\*av, std::shared\_ptr< [rserv::ServerConfig](#) > config)

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/Utils.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/Utils.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/Utils.cpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/Utils.cpp



## 4.186 math::Vector2f Class Reference

### Public Member Functions

- **Vector2f** (float x=0.0f, float y=0.0f)
- **Vector2f** ([Vector2f](#) const &other)
- float **getX** () const
- void **setX** (float x)
- float **getY** () const
- void **setY** (float y)
- [Vector2f](#) **getVector** () const
- [Vector2f](#) **operator\*** (float scalar) const
- [Vector2f](#) **operator-** ([Vector2f](#) const &other) const
- [Vector2f](#) **operator+** ([Vector2f](#) const &other) const
- void **operator=** ([Vector2f](#) const &other)
- void **operator+=** ([Vector2f](#) const &other)
- void **operator-=** ([Vector2f](#) const &other)
- void **operator\*=** (float scalar)
- void **operator/=** (float scalar)

### Private Attributes

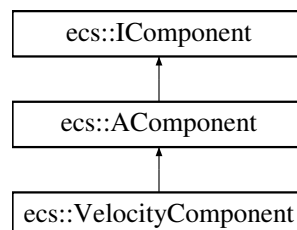
- float **\_x**
- float **\_y**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/types/Vector2f.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/types/Vector2f.cpp

## 4.187 ecs::VelocityComponent Class Reference

Inheritance diagram for ecs::VelocityComponent:



### Public Member Functions

- **VelocityComponent** ([math::Vector2f](#) velocity=[math::Vector2f](#)(0.0f, 0.0f))
- [math::Vector2f](#) **getVelocity** () const
- void **setVelocity** ([math::Vector2f](#) velocity)

**Private Attributes**

- [math::Vector2f](#) **\_velocity**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/Velocity↵  
Component.hpp

## 4.188 **ecs::View**< **Components** > Class Template Reference

**Classes**

- class [Iterator](#)

**Public Member Functions**

- **View** (std::shared\_ptr< [Registry](#) > registry)
- [Iterator](#) **begin** ()
- [Iterator](#) **end** ()

**Private Attributes**

- std::shared\_ptr< [Registry](#) > **\_registry**

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

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/entity/registry/Registry.hpp
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/view/View.hpp

# Chapter 5

## File Documentation

### 5.1 ClientNetwork.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ClientNetwork
00006 */
00007
00008
00009 #ifndef CLIENTNETWORK_HPP_
00010 #define CLIENTNETWORK_HPP_
00011
00012 #include <memory>
00013 #include <thread>
00014 #include <queue>
00015 #include <mutex>
00016 #include <condition_variable>
00017 #include <atomic>
00018 #include <map>
00019
00020 #include "../common/DLLoader/DLLoader.hpp"
00021 #include "../common/DLLoader/LoaderType.hpp"
00022 #include "../libs/Network/INetwork.hpp"
00023 #include "../common/constants.hpp"
00024 #include "../common/resourceManager/ResourceManager.hpp"
00025 #include "../common/gsm/IGameStateMachine.hpp"
00026
00027 namespace ecs {
00028     class Registry;
00029     using Entity = std::size_t;
00030 }
00031
00032 struct NetworkEvent {
00033     constants::EventType eventType;
00034     double depth;
00035 };
00036
00037 class ClientNetwork {
00038     public:
00039         ClientNetwork();
00040         ~ClientNetwork();
00041
00042         void init();
00043         void start();
00044         void stop();
00045         void connect();
00046
00047         uint16_t getPort() const;
00048         void setPort(int port);
00049
00050         std::string getIp() const;
00051         void setIp(const std::string &ip);
00052         std::shared_ptr<net::INetwork> getNetwork() const;
00053
00054         void setDebugMode(bool isDebug);
00055         bool isDebugMode() const;
00056
00057         void loadNetworkLibrary();
```

```

00058     void loadBufferLibrary();
00059     void loadPacketLibrary();
00060
00061     void sendConnectionData(std::vector<uint8_t> packet);
00062
00063     std::string getName() const;
00064     void setName(const std::string &name);
00065
00066     uint8_t getIdClient() const;
00067     void setIdClient(uint8_t idClient);
00068
00069     net::ConnectionState getConnectionState() const;
00070
00071     /* Packet Handling */
00072     void eventPacket(const constants::EventType &eventType, double depth);
00073     void disconnectionPacket();
00074     void connectionPacket();
00075     void sendReady();
00076     void sendWhoAmI();
00077
00078     void addToEventQueue(const NetworkEvent &event);
00079
00080     bool isConnected() const;
00081     bool isReady() const;
00082     std::atomic<bool> _isConnected;
00083     std::atomic<bool> _ready;
00084
00085     void setResourceManager(std::shared_ptr<ResourceManager> resourceManager);
00086     void setGameStateMachine(std::shared_ptr<gsm::IGameStateMachine> gsm);
00087     std::shared_ptr<gsm::IGameStateMachine> getGameStateMachine() const;
00088
00089     void redoServerEndpoint();
00090
00091     protected:
00092         std::pair<int, std::chrono::steady_clock::time_point> tryConnection(const int maxRetries, int
retryCount, std::chrono::steady_clock::time_point lastRetryTime);
00093         void handlePacketType(uint8_t type);
00094
00095     private:
00096         typedef void (ClientNetwork::*PacketHandler)();
00097         PacketHandler _packetHandlers[constants::MAX_INDEX_PACKET_TYPE];
00098
00099         void handleNoOp();
00100         void handleConnectionAcceptation();
00101         void handleGameState();
00102         void handleEndMap();
00103         void handleEndGame();
00104         void handleCanStart();
00105         void handleEntitySpawn();
00106         void handleEntityDeath();
00107         void handleWhoAmI();
00108
00109         typedef size_t (ClientNetwork::*ComponentParser)(const std::vector<uint64_t> &, size_t,
ecs::Entity);
00110         std::map<uint64_t, ComponentParser> _componentParsers;
00111
00112         ecs::Entity findOrCreateNetworkEntity(std::shared_ptr<ecs::Registry> registry, size_t
networkId);
00113
00114         size_t parsePlayerTagComponent(const std::vector<uint64_t> &payload, size_t index, ecs::Entity
entityId);
00115         size_t parseTransformComponent(const std::vector<uint64_t> &payload, size_t index, ecs::Entity
entityId);
00116         size_t parseSpeedComponent(const std::vector<uint64_t> &payload, size_t index, ecs::Entity
entityId);
00117         size_t parseHealthComponent(const std::vector<uint64_t> &payload, size_t index, ecs::Entity
entityId);
00118         size_t parseColliderComponent(const std::vector<uint64_t> &payload, size_t index, ecs::Entity
entityId);
00119         size_t parseShootingStatsComponent(const std::vector<uint64_t> &payload, size_t index,
ecs::Entity entityId);
00120         size_t parseScoreComponent(const std::vector<uint64_t> &payload, size_t index, ecs::Entity
entityId);
00121         size_t parseAIMovementPatternComponent(const std::vector<uint64_t> &payload, size_t index,
ecs::Entity entityId);
00122         size_t parseDamageComponent(const std::vector<uint64_t> &payload, size_t index, ecs::Entity
entityId);
00123         size_t parseLifetimeComponent(const std::vector<uint64_t> &payload, size_t index, ecs::Entity
entityId);
00124         size_t parseVelocityComponent(const std::vector<uint64_t> &payload, size_t index, ecs::Entity
entityId);
00125         size_t parseAIMoverTagComponent(const std::vector<uint64_t> &payload, size_t index,
ecs::Entity entityId);
00126         size_t parseAIShooterTagComponent(const std::vector<uint64_t> &payload, size_t index,
ecs::Entity entityId);
00127         size_t parseControllableTagComponent(const std::vector<uint64_t> &payload, size_t index,
ecs::Entity entityId);

```

```

00128         size_t parseEnemyProjectileTagComponent(const std::vector<uint64_t> &payload, size_t index,
00129         ecs::Entity entityId);
00129         size_t parseGameZoneColliderTagComponent(const std::vector<uint64_t> &payload, size_t index,
00130         ecs::Entity entityId);
00130         size_t parseMobTagComponent(const std::vector<uint64_t> &payload, size_t index, ecs::Entity
00131         entityId);
00131         size_t parseObstacleTagComponent(const std::vector<uint64_t> &payload, size_t index,
00132         ecs::Entity entityId);
00132         size_t parsePlayerProjectileTagComponent(const std::vector<uint64_t> &payload, size_t index,
00133         ecs::Entity entityId);
00133         size_t parseScoreTagComponent(const std::vector<uint64_t> &payload, size_t index, ecs::Entity
00134         entityId);
00134         size_t parseShooterTagComponent(const std::vector<uint64_t> &payload, size_t index,
00135         ecs::Entity entityId);
00135         size_t parseProjectilePassThroughTagComponent(const std::vector<uint64_t> &payload, size_t
00136         index, ecs::Entity entityId);
00136         size_t parseProjectilePrefabComponent(const std::vector<uint64_t> &payload, size_t index,
00137         ecs::Entity entityId);
00137         size_t parseNetworkIdComponent(const std::vector<uint64_t> &payload, size_t index, ecs::Entity
00138         entityId);
00138         size_t parseGameZoneComponent(const std::vector<uint64_t> &payload, size_t index, ecs::Entity
00139         entityId);
00139
00140         DLLoader<createNetworkLib_t> _networkloader;
00141         DLLoader<createBuffer_t> _bufferloader;
00142         DLLoader<createPacket_t> _packetloader;
00143
00144         std::shared_ptr<net::INetwork> _network;
00145         std::shared_ptr<IBuffer> _receptionBuffer;
00146         std::shared_ptr<IBuffer> _sendBuffer;
00147         std::shared_ptr<pm::IPacketManager> _packet;
00148
00149         std::shared_ptr<ResourceManager> _resourceManager;
00150         std::shared_ptr<gsm::IGameStateMachine> _gsm;
00151
00152         uint32_t _sequenceNumber;
00153         uint16_t _port;
00154         std::string _ip;
00155         std::string _name;
00156         std::vector<std::string> _clientNames;
00157         bool _isDebug;
00158
00159
00160         uint8_t _idClient;
00161         asio::ip::udp::endpoint _serverEndpoint;
00162
00163         std::queue<NetworkEvent> _eventQueue;
00164         std::mutex _queueMutex;
00165         std::condition_variable _queueCond;
00166
00167         bool _shouldConnect;
00168     };
00169
00170 #endif /* !CLIENTNETWORK_HPP_ */

```

## 5.2 colors.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** Colors
00006  */
00007
00008 #ifndef COLORS_HPP_
00009 #define COLORS_HPP_
00010
00011 #include "../libs/Multimedia/IWindow.hpp"
00012
00013 namespace colors {
00014     const gfx::color_t BLACK = {0, 0, 0};
00015     const gfx::color_t WHITE = {255, 255, 255};
00016     const gfx::color_t RED = {255, 0, 0};
00017     const gfx::color_t GREEN = {0, 255, 0};
00018     const gfx::color_t BLUE = {0, 0, 255};
00019     const gfx::color_t YELLOW = {255, 255, 0};
00020     const gfx::color_t CYAN = {0, 255, 255};
00021     const gfx::color_t MAGENTA = {255, 0, 255};
00022     const gfx::color_t GRAY = {128, 128, 128};
00023     const gfx::color_t LIGHT_GRAY = {192, 192, 192};
00024     const gfx::color_t DARK_GRAY = {64, 64, 64};
00025     const gfx::color_t ORANGE = {255, 165, 0};
00026     const gfx::color_t PURPLE = {128, 0, 128};

```

```

00027     const gfx::color_t PINK = {255, 192, 203};
00028     const gfx::color_t BROWN = {165, 42, 42};
00029
00030     // UI Colors
00031     const gfx::color_t UI_BACKGROUND = {10, 10, 30};           // Dark space blue
00032     const gfx::color_t UI_ACCENT = {0, 255, 255};             // Cyan accent
00033     const gfx::color_t UI_TEXT = {255, 255, 255};             // White text
00034     const gfx::color_t UI_TEXT_SECONDARY = {192, 192, 192};    // Light gray
00035     const gfx::color_t UI_OUTLINE = {0, 0, 0};                 // Black outline
00036
00037     // Button Colors
00038     const gfx::color_t BUTTON_PRIMARY = {0, 100, 200};         // Deep blue
00039     const gfx::color_t BUTTON_PRIMARY_HOVER = {0, 150, 255};   // Bright blue
00040     const gfx::color_t BUTTON_PRIMARY_PRESSED = {0, 50, 150};  // Dark blue
00041     const gfx::color_t BUTTON_SECONDARY = {100, 0, 150};        // Purple
00042     const gfx::color_t BUTTON_SECONDARY_HOVER = {150, 0, 200};  // Bright purple
00043     const gfx::color_t BUTTON_SECONDARY_PRESSED = {75, 0, 112}; // Dark purple
00044
00045     // Panel/Frame Colors
00046     const gfx::color_t PANEL_BACKGROUND = {20, 20, 50};         // Slightly lighter
00047     than UI background
00048     const gfx::color_t PANEL_BORDER = {0, 200, 255};           // Cyan border
00049
00050     // Slider Colors
00051     const gfx::color_t SLIDER_TRACK = {50, 50, 100};           // Dark blue track
00052     const gfx::color_t SLIDER_FILL = {0, 150, 255};            // Bright blue fill
00053     const gfx::color_t SLIDER_HANDLE = {0, 100, 200};          // Blue handle
00054     const gfx::color_t SLIDER_HANDLE_HOVER = {0, 150, 255};    // Bright blue hover
00055     const gfx::color_t SLIDER_HANDLE_FOCUSED = {0, 200, 255};  // Cyan focused
00056     const gfx::color_t SLIDER_LABEL = {255, 255, 255};         // White label
00057
00058     // Toggle Switch Colors
00059     const gfx::color_t TOGGLE_TRACK = {100, 50, 150};          // Purple track
00060     const gfx::color_t TOGGLE_HANDLE = {150, 100, 200};        // Light purple handle
00061     const gfx::color_t TOGGLE_HANDLE_HOVER = {200, 150, 255};  // Bright purple hover
00062     const gfx::color_t TOGGLE_HANDLE_FOCUSED = {255, 200, 255}; // Pinkish purple
00063     focused
00064     const gfx::color_t TOGGLE_ON = {150, 0, 200};              // Purple on
00065     const gfx::color_t TOGGLE_OFF = {50, 0, 100};              // Dark purple off
00066
00067     // General UI States
00068     const gfx::color_t UI_HOVER = {0, 150, 255};               // Bright blue hover
00069     const gfx::color_t UI_FOCUSED = {0, 200, 255};              // Cyan focused
00070     const gfx::color_t UI_DISABLED = {100, 100, 100};          // Gray disabled
00071 }
00072 #endif // COLORS_HPP_

```

## 5.3 AnimationComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** AnimationComponent
00006 */
00007
00008 #ifndef ANIMATIONCOMPONENT_HPP_
00009 #define ANIMATIONCOMPONENT_HPP_
00010
00011 #include <unordered_map>
00012 #include <vector>
00013 #include <functional>
00014 #include <string>
00015 #include "../common/components/base/AComponent.hpp"
00016 #include "../common/types/FRect.hpp"
00017 #include "../common/ECS/entity/Entity.hpp"
00018
00019 namespace ecs {
00020
00021     class Registry;
00022
00023     struct AnimationClip {
00024         std::string texturePath;
00025         float frameWidth;
00026         float frameHeight;
00027         int frameCount;
00028         float startWidth;
00029         float startHeight;
00030         float speed;
00031         bool loop;
00032     };
00033 }

```

```

00034 struct AnimationCondition {
00035     std::string param;
00036     bool equals;
00037 };
00038
00039 struct Transition {
00040     std::string from;
00041     std::string to;
00042     std::vector<AnimationCondition> conditions;
00043     bool playRewind = false;
00044 };
00045
00046 class AnimationComponent : public AComponent {
00047 public:
00048     AnimationComponent()
00049         : _currentState(""), _timer(0.f), _isPlaying(false), _currentFrame(0),
00050         _rewindStartFrame(-1) {
00051         this->_states = {};
00052         this->_playRewind = false;
00053         this->_transitions = {};
00054         this->_frameRect = math::FRect();
00055     }
00056
00057     void addState(const std::string& name, std::shared_ptr<AnimationClip> clip) {
00058         _states[name] = clip;
00059     }
00060
00061     void addTransition(const std::string& from, const std::string& to,
00062         const std::vector<AnimationCondition>& conditions,
00063         bool playRewind = false) {
00064         _transitions.push_back({from, to, conditions, playRewind});
00065     }
00066
00067     void setCurrentState(const std::string& state) {
00068         if (_states.find(state) != _states.end()) {
00069             _currentState = state;
00070             _timer = 0.f;
00071             _isPlaying = true;
00072             _currentFrame = 0;
00073             _playRewind = false;
00074             _stateJustChanged = true;
00075
00076             auto clip = _states[state];
00077             _minAnimationTime = clip->speed;
00078             _frameRect = math::FRect(clip->startWidth, clip->startHeight, clip->frameWidth,
00079                 clip->frameHeight);
00080         }
00081     }
00082
00083     const std::string& getCurrentState() const { return _currentState; }
00084     float getTimer() const { return _timer; }
00085     void setTimer(float timer) { _timer = timer; }
00086     bool isPlaying() const { return _isPlaying; }
00087     void setPlaying(bool playing) { _isPlaying = playing; }
00088     bool isPlayingRewind() const { return _playRewind; }
00089     void setPlayingRewind(bool rewind) { _playRewind = rewind; }
00090
00091     int getRewindStartFrame() const { return _rewindStartFrame; }
00092     void setRewindStartFrame(int frame) { _rewindStartFrame = frame; }
00093
00094     std::shared_ptr<const AnimationClip> getCurrentClip() const {
00095         auto it = _states.find(_currentState);
00096         return it != _states.end() ? it->second : nullptr;
00097     }
00098
00099     const std::vector<Transition>& getTransitions() const { return _transitions; }
00100
00101     int getCurrentFrame() const { return _currentFrame; }
00102     void setCurrentFrame(int frame) { _currentFrame = frame; }
00103
00104     const math::FRect& getFrameRect() const { return _frameRect; }
00105     void setFrameRect(const math::FRect& rect) { _frameRect = rect; }
00106
00107     bool isValid() const { return !_states.empty(); }
00108
00109     bool isAnimationFinished() const {
00110         auto clip = getCurrentClip();
00111         if (!clip) return true;
00112         if (clip->loop) return false;
00113         int currentFrame = static_cast<int>(_timer / clip->speed);
00114         if (_playRewind) {
00115             return currentFrame >= clip->frameCount;
00116         } else {
00117             return currentFrame >= clip->frameCount - 1;
00118         }
00119     }
00120
00121 }

```

```

00119         void setStateJustChanged(bool changed) { _stateJustChanged = changed; }
00120         bool getStateJustChanged() const { return _stateJustChanged; }
00121
00122         void setMinAnimationTime(float time) { _minAnimationTime = time; }
00123         float getMinAnimationTime() const { return _minAnimationTime; }
00124
00125     private:
00126         std::unordered_map<std::string, std::shared_ptr<AnimationClip> _states;
00127         std::vector<Transition> _transitions;
00128         std::string _currentState;
00129         float _timer;
00130         bool _isPlaying;
00131         bool _playRewind;
00132         int _currentFrame;
00133         int _rewindStartFrame;
00134         math::FRect _frameRect;
00135         bool _stateJustChanged = false;
00136         float _minAnimationTime = 0.0f;
00137 };
00138
00139 } // namespace ecs
00140
00141 #endif /* !ANIMATIONCOMPONENT_HPP_ */

```

## 5.4 HealthBarComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** HealthBarComponent
00006 */
00007
00008 #ifndef HEALTHBARCOMPONENT_HPP_
00009 #define HEALTHBARCOMPONENT_HPP_
00010
00011 #include "../common/components/base/AComponent.hpp"
00012
00013 namespace ecs {
00014
00015     class HealthBarComponent : public AComponent {
00016     public:
00017         HealthBarComponent() = default;
00018         ~HealthBarComponent() = default;
00019     };
00020
00021 } // namespace ecs
00022
00023 #endif /* !HEALTHBARCOMPONENT_HPP_ */

```

## 5.5 HitboxRenderComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** HitboxRenderComponent
00006 */
00007
00008 #ifndef HITBOXRENDERCOMPONENT_HPP_
00009 #define HITBOXRENDERCOMPONENT_HPP_
00010
00011 #include "../common/components/base/AComponent.hpp"
00012 #include "../libs/Multimedia/IWindow.hpp"
00013
00014 namespace ecs {
00015
00016     class HitboxRenderComponent : public AComponent {
00017     public:
00018         HitboxRenderComponent() : _color{255, 255, 255}, _outlineThickness(1.0f) {}
00019         HitboxRenderComponent(gfx::color_t color, float outlineThickness = 1.0f)
00020             : _color(color), _outlineThickness(outlineThickness) {}
00021
00022         ~HitboxRenderComponent() = default;
00023
00024         const gfx::color_t& getColor() const { return _color; }
00025         void setColor(const gfx::color_t& color) { _color = color; }
00026

```



```

00027         float getOutlineThickness() const { return _outlineThickness; }
00028         void setOutlineThickness(float thickness) { _outlineThickness = thickness; }
00029
00030     private:
00031         gfx::color_t _color;
00032         float _outlineThickness;
00033 };
00034
00035 } // namespace ecs
00036
00037 #endif /* !HITBOXRENDERCOMPONENT_HPP_ */

```

## 5.6 MusicComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** MusicComponent
00006 */
00007
00008 #ifndef MUSICCOMPONENT_HPP_
00009 #define MUSICCOMPONENT_HPP_
00010
00011 #include "../common/components/base/AComponent.hpp"
00012 #include <string>
00013
00014 namespace ecs {
00015
00016     typedef enum MusicState {
00017         PLAYING = 0,
00018         PAUSED = 1,
00019         CHANGING = 2,
00020         STOPPED = 3
00021     } MusicState;
00022
00023     class MusicComponent : public AComponent {
00024     public:
00025         MusicComponent(std::string musicFile = "", MusicState initialState = STOPPED, float volume =
00026             100.0f, bool loop = false)
00027             : _currentMusic(musicFile), _state(initialState), _volume(volume), _loop(loop) {};
00028         ~MusicComponent() = default;
00029
00030         void playMusic() { _state = PLAYING; };
00031         void pauseMusic() { _state = PAUSED; };
00032         void stopMusic() { _state = STOPPED; };
00033         bool isPlaying() const { return _state == PLAYING; };
00034         MusicState getState() const { return _state; };
00035         void playNewMusic(const std::string& musicFile) {
00036             _currentMusic = musicFile;
00037             _state = CHANGING;
00038         };
00039
00040         std::string getCurrentMusic() const { return _currentMusic; };
00041         void setCurrentMusic(const std::string& musicFile) { _currentMusic = musicFile; };
00042
00043         float getVolume() const { return _volume; };
00044         void setVolume(float volume) { _volume = volume; };
00045
00046         bool isLooping() const { return _loop; };
00047         void setLoop(bool loop) { _loop = loop; };
00048
00049     protected:
00050     private:
00051         std::string _currentMusic;
00052         MusicState _state;
00053         float _volume;
00054         bool _loop;
00055     };
00056
00057 } // namespace ecs
00058
00059 #endif /* !MUSICCOMPONENT_HPP_ */

```

## 5.7 ParallaxComponent.hpp

```

00001 /*

```

```

00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ParallaxComponent
00006 */
00007
00008 #ifndef PARALLAXCOMPONENT_HPP_
00009 #define PARALLAXCOMPONENT_HPP_
00010
00011 #include "../common/components/base/AComponent.hpp"
00012 #include "../common/types/Vector2f.hpp"
00013 #include "../common/constants.hpp"
00014 #include <string>
00015 #include <vector>
00016
00017 namespace ecs {
00018
00019 enum class ParallaxScaleMode {
00020     FIT_SCREEN = 0,
00021     STRETCH = 1,
00022     MANUAL = 2,
00023 };
00024
00025 struct ParallaxLayer {
00026     std::string name;
00027     std::string filePath;
00028     float speedMultiplier;
00029     math::Vector2f scale;
00030     ParallaxScaleMode scaleMode;
00031     math::Vector2f sourceSize;
00032     bool repeat;
00033     int zIndex;
00034     math::Vector2f currentOffset;
00035
00036     ParallaxLayer()
00037         : name(""),
00038         filePath(""),
00039         speedMultiplier(1.0f),
00040         scale(1.0f, 1.0f),
00041         scaleMode(ParallaxScaleMode::FIT_SCREEN),
00042         sourceSize(constants::DEFAULT_TEXTURE_WIDTH, constants::DEFAULT_TEXTURE_HEIGHT),
00043         repeat(true),
00044         zIndex(0),
00045         currentOffset(0.0f, 0.0f) {}
00046 };
00047
00048 class ParallaxComponent : public AComponent {
00049     public:
00050         ParallaxComponent()
00051             : _baseScrollSpeed(1.0f),
00052             _direction(constants::BACKGROUND_PARALLAX_DIRECTION_X,
00053                 constants::BACKGROUND_PARALLAX_DIRECTION_Y),
00054             _layers() {}
00055
00056         ~ParallaxComponent() = default;
00057
00058         float getBaseScrollSpeed() const { return _baseScrollSpeed; }
00059         const math::Vector2f& getDirection() const { return _direction; }
00060         const std::vector<ParallaxLayer>& getLayers() const { return _layers; }
00061
00062         void setBaseScrollSpeed(float speed) { _baseScrollSpeed = speed; }
00063         void setDirection(const math::Vector2f& direction) { _direction = direction; }
00064         void addLayer(const ParallaxLayer& layer) { _layers.push_back(layer); }
00065         void clearLayers() { _layers.clear(); }
00066
00067         void updateLayerOffsets(const math::Vector2f& direction, float baseSpeed, float deltaTime) {
00068             for (auto& layer : _layers) {
00069                 float speed = baseSpeed * layer.speedMultiplier;
00070                 math::Vector2f movement(
00071                     direction.getX() * speed * deltaTime,
00072                     direction.getY() * speed * deltaTime
00073                 );
00074                 layer.currentOffset = math::Vector2f(
00075                     layer.currentOffset.getX() + movement.getX(),
00076                     layer.currentOffset.getY() + movement.getY()
00077                 );
00078             }
00079
00080             size_t getLayerCount() const { return _layers.size(); }
00081
00082             void sortLayersByZIndex() {
00083                 std::sort(_layers.begin(), _layers.end(),
00084                     [](const ParallaxLayer& a, const ParallaxLayer& b) {
00085                         return a.zIndex < b.zIndex;
00086                     });
00087             }

```

```

00088
00089     private:
00090         float _baseScrollSpeed;
00091         math::Vector2f _direction;
00092         std::vector<ParallaxLayer> _layers;
00093     };
00094
00095 } // namespace ecs
00096
00097 #endif /* !PARALLAXCOMPONENT_HPP_ */

```

## 5.8 RectangleRenderComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** RectangleRenderComponent
00006 */
00007
00008 #ifndef RECTANGLERENDERCOMPONENT_HPP_
00009 #define RECTANGLERENDERCOMPONENT_HPP_
00010
00011 #include "../common/components/base/AComponent.hpp"
00012 #include "../common/libraries/Multimedia/IWindow.hpp"
00013
00014 namespace ecs {
00015
00016 class RectangleRenderComponent : public AComponent {
00017     public:
00018         RectangleRenderComponent() : _color{255, 255, 255}, _size{10.0f, 10.0f} {}
00019         RectangleRenderComponent(gfx::color_t color, float width, float height)
00020             : _color(color), _size{width, height} {}
00021
00022         ~RectangleRenderComponent() = default;
00023
00024         const gfx::color_t& getColor() const { return _color; }
00025         void setColor(const gfx::color_t& color) { _color = color; }
00026
00027         float getWidth() const { return _size.first; }
00028         float getHeight() const { return _size.second; }
00029         void setSize(float width, float height) { _size = {width, height}; }
00030
00031     private:
00032         gfx::color_t _color;
00033         std::pair<float, float> _size;
00034 };
00035
00036 } // namespace ecs
00037
00038 #endif /* !RECTANGLERENDERCOMPONENT_HPP_ */

```

## 5.9 SpriteComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** SpriteComponent
00006 */
00007
00008 #ifndef SPRITECOMPONENT_HPP_
00009 #define SPRITECOMPONENT_HPP_
00010
00011 #include "../common/components/base/AComponent.hpp"
00012 #include "../common/types/FRect.hpp"
00013 #include <string>
00014
00015 namespace ecs {
00016
00017 class SpriteComponent : public AComponent {
00018     public:
00019         SpriteComponent() : _texturePath("") {}
00020         SpriteComponent(const std::string& texturePath)
00021             : _texturePath(texturePath) {}
00022
00023         ~SpriteComponent() = default;
00024         const std::string& getTexturePath() const { return _texturePath; }

```

```

00025         void setTexturePath(const std::string& path) { _texturePath = path; }
00026         bool isValid() const { return !_texturePath.empty(); }
00027
00028     private:
00029         std::string _texturePath;
00030 };
00031
00032 } // namespace ecs
00033
00034 #endif /* !SPRITECOMPONENT_HPP_ */

```

## 5.10 TextComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** TextComponent
00006 */
00007
00008 #ifndef TEXTCOMPONENT_HPP_
00009 #define TEXTCOMPONENT_HPP_
00010
00011 #include <string>
00012 #include "../common/components/base/AComponent.hpp"
00013 #include "../libs/Multimedia/IWindow.hpp"
00014
00015 namespace ecs {
00016
00017     class TextComponent : public AComponent {
00018     public:
00019         TextComponent(const std::string& text, const std::string& fontPath,
00020             gfx::color_t color = gfx::color_t{255, 255, 255})
00021             : _text(text), _fontPath(fontPath), _color(color) {}
00022         ~TextComponent() {}
00023
00024         const std::string& getText() const { return _text; }
00025         const std::string& getFontPath() const { return _fontPath; }
00026         const gfx::color_t& getColor() const { return _color; }
00027
00028         void setText(const std::string& text) { _text = text; }
00029         void setFontPath(const std::string& fontPath) { _fontPath = fontPath; }
00030         void setColor(const gfx::color_t& color) { _color = color; }
00031
00032     protected:
00033     private:
00034         std::string _text;
00035         std::string _fontPath;
00036         gfx::color_t _color;
00037 };
00038
00039 } // namespace ecs
00040
00041 #endif /* !TEXTCOMPONENT_HPP_ */

```

## 5.11 BackgroundMusicTag.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** BackgroundMusicTag
00006 */
00007
00008 #ifndef BACKGROUNDMUSICTAG_HPP_
00009 #define BACKGROUNDMUSICTAG_HPP_
00010
00011 #include "../common/components/base/AComponent.hpp"
00012
00013 namespace ecs {
00014
00015     class BackgroundMusicTag : public AComponent {
00016     public:
00017         BackgroundMusicTag() = default;
00018         ~BackgroundMusicTag() = default;
00019 };
00020
00021 } // namespace ecs
00022
00023 #endif /* !BACKGROUNDMUSICTAG_HPP_ */

```

## 5.12 MusicIntentComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** MusicIntentComponent
00006 */
00007
00008 #ifndef MUSICINTENTCOMPONENT_HPP_
00009 #define MUSICINTENTCOMPONENT_HPP_
00010
00011 #include "../common/components/base/AComponent.hpp"
00012 #include <string>
00013
00014 namespace ecs {
00015
00016 typedef enum MusicAction {
00017     PLAY = 0,
00018     PAUSE = 1,
00019     CHANGE = 2
00020 } MusicAction;
00021
00022 class MusicIntentComponent : public AComponent {
00023     public:
00024         MusicIntentComponent(MusicAction action = PLAY, const std::string &musicPath = "", float
volume = 100.0f)
00025             : _action(action), _musicPath(musicPath), _volume(volume) {
00026         };
00027         ~MusicIntentComponent() = default;
00028
00029         MusicAction getAction() const { return _action; };
00030         void setAction(MusicAction action) { _action = action; };
00031
00032         std::string getMusicPath() const { return _musicPath; };
00033         void setMusicPath(const std::string &musicPath) { _musicPath = musicPath; };
00034
00035         float getVolume() const { return _volume; };
00036         void setVolume(float volume) { _volume = volume; };
00037
00038     private:
00039         MusicAction _action;
00040         std::string _musicPath;
00041         float _volume;
00042 };
00043
00044 } // namespace ecs
00045
00046 #endif /* !MUSICINTENTCOMPONENT_HPP_ */

```

## 5.13 SoundIntentComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** SoundIntentComponent
00006 */
00007
00008 #ifndef SOUNDINTENTCOMPONENT_HPP_
00009 #define SOUNDINTENTCOMPONENT_HPP_
00010
00011 #include "../common/components/base/AComponent.hpp"
00012 #include <string>
00013
00014 namespace ecs {
00015
00016 class SoundIntentComponent : public AComponent {
00017     public:
00018         SoundIntentComponent(const std::string &soundPath = "", float volume = 100.0f)
00019             : _soundPath(soundPath), _volume(volume) {
00020         };
00021         ~SoundIntentComponent() = default;
00022
00023         std::string getSoundPath() const { return _soundPath; };
00024         void setSoundPath(const std::string &soundPath) { _soundPath = soundPath; };
00025
00026         float getVolume() const { return _volume; };
00027         void setVolume(float volume) { _volume = volume; };
00028
00029     private:
00030         std::string _soundPath;

```

```

00031         float _volume;
00032     };
00033
00034 } // namespace ecs
00035
00036 #endif /* !SOUNDINTENTCOMPONENT_HPP_ */

```

## 5.14 constants.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** Client Constants
00006 */
00007
00008 #ifndef CLIENT_CONSTANTS_HPP_
00009 #define CLIENT_CONSTANTS_HPP_
00010
00011 #include "../common/constants.hpp"
00012
00013 namespace constants {
00014     /* Timeout */
00015     constexpr int NETWORK_TIMEOUT = 5;
00016
00017     /* Window */
00018     constexpr int WINDOW_WIDTH = 1920;
00019     constexpr int WINDOW_HEIGHT = 1080;
00020
00021     /* UI */
00022     constexpr float INVALID_MOUSE_POSITION = -1.0f;
00023     constexpr float NAVIGATION_COOLDOWN_TIME = 0.1f;
00024     constexpr float UI_SCALE_SMALL = 0.75f;
00025     constexpr float UI_SCALE_NORMAL = 1.0f;
00026     constexpr float UI_SCALE_LARGE = 1.25f;
00027     constexpr size_t BUTTON_FONT_SIZE_BASE = 24;
00028     constexpr double SMALL_FONT_SIZE_MULTIPLIER = 0.5;
00029
00030     /* View */
00031     constexpr float VIEW_SMOOTHING_SPEED = 15.0f;
00032
00033     enum MouseButton {
00034         LEFT = 0,
00035         RIGHT = 1,
00036         MIDDLE = 2,
00037     };
00038
00039     /* Accessibility Filters */
00040     const std::string FILTER_HIGH_CONTRAST_SHADER_PATH = "assets/shaders/highcontrast.frag";
00041     const std::string FILTER_PROTANOPIA_SHADER_PATH = "assets/shaders/protanopia.frag";
00042     const std::string FILTER_DEUTERANOPIA_SHADER_PATH = "assets/shaders/deuteranopia.frag";
00043     const std::string FILTER_TRITANOPIA_SHADER_PATH = "assets/shaders/tritanopia.frag";
00044     const std::string FILTER_BRIGHTNESS_SHADER_PATH = "assets/shaders/brightness.frag";
00045     const std::string FILTER_BRIGHTNESS_UNIFORM_NAME = "brightness";
00046
00047     /* Health Bar */
00048     constexpr float HEALTH_BAR_OFFSET_Y = -10.0f;
00049     constexpr float HEALTH_BAR_HEIGHT = 5.0f;
00050     constexpr float HEALTH_BAR_OUTLINE_THICKNESS = 2.0f;
00051
00052     /* Settings Parsing Constants */
00053     const std::string ACCESSIBILITY_COLOR_BLINDNESS_STATE = "colorBlindnessState";
00054     const std::string ACCESSIBILITY_BRIGHTNESS_VALUE = "brightnessValue";
00055     const std::string ACCESSIBILITY_HIGH_CONTRAST_ENABLED = "highContrastEnabled";
00056     const std::string SETTINGS_UI_SCALE = "uiScale";
00057     const std::string SETTINGS_MUSIC_VOLUME = "musicVolume";
00058     const std::string SETTINGS_SOUND_VOLUME = "soundVolume";
00059     const std::string SETTINGS_SCREEN_RESOLUTION = "screenResolution";
00060     const std::string SETTINGS_TARGET_FPS = "targetFPS";
00061     const std::string SETTINGS_RENDER_QUALITY = "renderQuality";
00062     const std::string KEYBIND_PRIMARY = "primary";
00063     const std::string KEYBIND_SECONDARY = "secondary";
00064     const std::string KEYBIND_TOGGLE_MODE = "toggle_mode";
00065
00066     /* Paths */
00067     const std::string SAVES_DIRECTORY = "saves";
00068     const std::string KEYBINDS_FILE_PATH = "saves/keybinds.json";
00069     const std::string ACCESSIBILITY_FILE_PATH = "saves/accessibility.json";
00070     const std::string SETTINGS_FILE_PATH = "saves/settings.json";
00071     const std::string UI_BACKGROUND_EARTH_PATH = "assets/ui/earth.jpg";
00072
00073     /* Home page input place holders */
00074     const std::string IP_PLACEHOLDER = "Enter an IP address";

```

```

00075     const std::string PORT_PLACEHOLDER = "Enter a port";
00076 }
00077
00078 #endif /* !CLIENT_CONSTANTS_HPP_ */

```

## 5.15 constants.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** Constants
00006 */
00007
00008 #ifndef CONSTANTS_HPP_
00009 #define CONSTANTS_HPP_
00010
00011 #include <cstdint>
00012 #include <string>
00013 #include "types/Vector2f.hpp"
00014
00015 namespace constants {
00016     /* Network Defaults */
00017     constexpr int DEFAULT_SERVER_PORT = 4242;
00018     const std::string DEFAULT_SERVER_IP = "127.0.0.1"; // 127.0.0.1
00019
00020     /* Game Defaults */
00021     constexpr float BASE_SPEED = 100.0f;
00022     constexpr float EPS = 1e-6f;
00023     constexpr float PLAYER_BASE_SPEED = 300.0f;
00024     constexpr float GAMEPAD_DEADZONE = 0.15f;
00025     constexpr float AXIS_MAX_VALUE = 100.0f;
00026     constexpr int SMOOTH_MOVEMENT_ITERATIONS = 4;
00027
00028     /* Texture Defaults */
00029     constexpr float DEFAULT_TEXTURE_WIDTH = 1920.0f;
00030     constexpr float DEFAULT_TEXTURE_HEIGHT = 1080.0f;
00031
00032     enum class EventType {
00033         UP = 0,
00034         DOWN = 1,
00035         LEFT = 2,
00036         RIGHT = 3,
00037         SHOOT = 4,
00038         STOP = 5,
00039     };
00040     constexpr int MAX_RETRY_CONNECTIONS = 3;
00041     /* Paths */
00042     const std::string CONFIG_PATH = "configs/entities/";
00043
00044     /* Collision Rules JSON Keys */
00045     const std::string COLLISION_SOLID_KEY = "solid";
00046     const std::string COLLISION_TRIGGER_KEY = "trigger";
00047     const std::string COLLISION_PUSH_KEY = "push";
00048     const std::string COLLISION_ALLOW_KEY = "allow";
00049
00050     /* Parsing constants */
00051     const std::string SERVER_VALUE = "server";
00052     const std::string CLIENT_VALUE = "client";
00053     const std::string BOTH_VALUE = "both";
00054
00055     const std::string TRANSFORMCOMPONENT = "TransformComponent";
00056     const std::string VELOCITYCOMPONENT = "VelocityComponent";
00057     const std::string SPEEDCOMPONENT = "SpeedComponent";
00058     const std::string SPRITECOMPONENT = "SpriteComponent";
00059     const std::string ANIMATIONCOMPONENT = "AnimationComponent";
00060     const std::string SHOOTINGSTATSCOMPONENT = "ShootingStatsComponent";
00061     const std::string RECTANGLERENDERCOMPONENT = "RectangleRenderComponent";
00062     const std::string PROJECTILEPREFABCOMPONENT = "ProjectilePrefabComponent";
00063     const std::string TEXTCOMPONENT = "TextComponent";
00064     const std::string HEALTHBARCOMPONENT = "HealthBarComponent";
00065
00066     const std::string SCORECOMPONENT = "ScoreComponent";
00067     const std::string SCOREVALUECOMPONENT = "ScoreValueComponent";
00068     const std::string DAMAGECOMPONENT = "DamageComponent";
00069     const std::string HEALTHCOMPONENT = "HealthComponent";
00070     const std::string INTERACTIONCONFIGCOMPONENT = "InteractionConfigComponent";
00071     const std::string AIMOVEMENTPATTERNCOMPONENT = "AIMovementPatternComponent";
00072     const std::string SCORE_FIELD = "score";
00073     const std::string SCOREVALUE_FIELD = "scoreValue";
00074     const std::string DAMAGE_FIELD = "damage";
00075     const std::string HEALTH_FIELD = "health";
00076     const std::string TARGET_FIELD = "target";

```

```

00077     const std::string POSITION_FIELD = "position";
00078     const std::string OFFSET_FIELD = "offset";
00079     const std::string SCALE_FIELD = "scale";
00080     const std::string ROTATION_FIELD = "rotation";
00081     const std::string SPEED_FIELD = "speed";
00082     const std::string FILEPATH_FIELD = "filePath";
00083     const std::string ANIMATIONPATH_FIELD = "animationPath";
00084     const std::string FRAMEWIDTH_FIELD = "frameWidth";
00085     const std::string FRAMEHEIGHT_FIELD = "frameHeight";
00086     const std::string FRAMECOUNT_FIELD = "frameCount";
00087     const std::string STARTWIDTH_FIELD = "startWidth";
00088     const std::string STARTHEIGHT_FIELD = "startHeight";
00089     const std::string SIZE_FIELD = "size";
00090     const std::string FIRERATE_FIELD = "fireRate";
00091     const std::string SHOTCOUNT_FIELD = "shotCount";
00092     const std::string ANGLEOFFSET_FIELD = "angleOffset";
00093     const std::string SPREADANGLE_FIELD = "spreadAngle";
00094     const std::string DEFAULTBEHAVIOR_FIELD = "defaultBehavior";
00095     const std::string ZIGZAGAMPLITUDE_FIELD = "zigzagAmplitude";
00096     const std::string ZIGZAGFREQUENCY_FIELD = "zigzagFrequency";
00097     const std::string DETECTIONRANGE_FIELD = "detectionRange";
00098     const std::string VERTICALDEADZONE_FIELD = "verticalDeadzone";
00099     const std::string STRAIGHT_LINE_VALUE = "StraightLine";
00100     const std::string ZIGZAG_VALUE = "Zigzag";
00101     const std::string VERTICAL_MIRROR_VALUE = "VerticalMirror";
00102     const std::string FOLLOW_RIGHT_VALUE = "FollowRight";
00103     const std::string WIDTH_FIELD = "width";
00104     const std::string HEIGHT_FIELD = "height";
00105     const std::string COLOR_FIELD = "color";
00106     const std::string R_FIELD = "r";
00107     const std::string G_FIELD = "g";
00108     const std::string B_FIELD = "b";
00109
00110     const std::string STATES_FIELD = "states";
00111     const std::string INITIALSTATE_FIELD = "initialState";
00112     const std::string TRANSITIONS_FIELD = "transitions";
00113     const std::string CONDITIONS_FIELD = "conditions";
00114     const std::string CONDITION_FIELD = "condition";
00115     const std::string PARAM_FIELD = "param";
00116     const std::string EQUALS_FIELD = "equals";
00117     const std::string FROM_FIELD = "from";
00118     const std::string BASESCROLLSPEED_FIELD = "baseScrollSpeed";
00119     const std::string DIRECTION_FIELD = "direction";
00120     const std::string LAYERS_FIELD = "layers";
00121     const std::string ZONERECT_FIELD = "zoneRect";
00122     const std::string NAME_FIELD = "name";
00123     const std::string SPEEDMULTIPLIER_FIELD = "speedMultiplier";
00124     const std::string SCALEMODE_FIELD = "scaleMode";
00125     const std::string SOURCESIZE_FIELD = "sourceSize";
00126     const std::string REPEAT_FIELD = "repeat";
00127     const std::string ZINDEX_FIELD = "zIndex";
00128     const std::string TO_FIELD = "to";
00129     const std::string REWIND_FIELD = "rewind";
00130     const std::string TEXTUREPATH_FIELD = "texturePath";
00131     const std::string LOOP_FIELD = "loop";
00132     const std::string SCALEMODE_FITSCREEN = "FIT_SCREEN";
00133     const std::string SCALEMODE_STRETCH = "STRETCH";
00134     const std::string SCALEMODE_MANUAL = "MANUAL";
00135
00136     const std::string COLLISION_TYPE_SOLID = "Solid";
00137     const std::string COLLISION_TYPE_TRIGGER = "Trigger";
00138     const std::string COLLISION_TYPE_PUSH = "Push";
00139     const std::string COLLISION_TYPE_NONE = "None";
00140
00141     const std::string X_FIELD = "x";
00142     const std::string Y_FIELD = "y";
00143
00144     const std::string PREFABNAME_FIELD = "prefabName";
00145     const std::string LIFETIMECOMPONENT = "LifetimeComponent";
00146     const std::string LIFETIME_FIELD = "lifetime";
00147     const std::string LIFESPANCOMPONENT = "LifeSpanComponent";
00148     const std::string LIFESPAN_FIELD = "lifespan";
00149     const std::string BACKGROUNDMUSICTAG = "BackGroundMusicTag";
00150     const std::string TEXT_FIELD = "text";
00151     const std::string FONTPATH_FIELD = "fontPath";
00152     const std::string SOUNDINTENTCOMPONENT = "SoundIntentComponent";
00153     const std::string SOUND_FILE_FIELD = "soundFile";
00154
00155     const std::string MAPPINGS_FIELD = "mappings";
00156     const std::string TAGS_FIELD = "tags";
00157     const std::string TOENTITY_FIELD = "toEntity";
00158     const std::string TOSELF_FIELD = "toSelf";
00159
00160     const std::string MUSICCOMPONENT = "MusicComponent";
00161     const std::string MUSICFILE_FIELD = "musicFile";
00162     const std::string VOLUME_FIELD = "volume";
00163     const std::string INITIALSTATEMUSIC_FIELD = "initialState";

```



```

00164     const std::string PLAYING_FIELD = "PLAYING";
00165     const std::string PAUSED_FIELD = "PAUSED";
00166     const std::string STOPPED_FIELD = "STOPPED";
00167     const std::string CHANGING_FIELD = "CHANGING";
00168
00169     const float MAX_HEIGHT = 1080.0f;
00170     const float MAX_WIDTH = 1920.0f;
00171     const float GAME_ZONE_BOUNDARY_THICKNESS = 100.0f;
00172
00173     const float SPATIAL_GRID_CELL_SIZE = 128.0f;
00174     const float SPATIAL_GRID_PADDING = 200.0f;
00175     const float OUT_OF_BOUNDS_MARGIN = 200.0f;
00176
00177     /* Map parsing constants */
00178     const std::string BACKGROUND_FIELD = "background";
00179     const std::string BACKGROUND_SCROLL_SPEED_FIELD = "scrollSpeed";
00180     const std::string MUSIC_FIELD = "music";
00181     const std::string POWERUPS_FIELD = "powerUps";
00182     const std::string OBSTACLES_FIELD = "obstacles";
00183     const std::string WAVES_FIELD = "waves";
00184     const std::string POSITIONS_FIELD = "positions";
00185     const std::string TYPE_FIELD = "type";
00186     const std::string FROMX_FIELD = "fromX";
00187     const std::string FROMY_FIELD = "fromY";
00188     const std::string POSX_FIELD = "posX";
00189     const std::string POSY_FIELD = "posY";
00190     const std::string COUNT_FIELD = "count";
00191     const std::string GAMEXTRIGGER_FIELD = "gameXTrigger";
00192     const std::string DISTRIBUTIONX_FIELD = "distributionX";
00193     const std::string DISTRIBUTIONY_FIELD = "distributionY";
00194     const std::string ENEMIES_FIELD = "enemies";
00195     const std::string MIN_FIELD = "min";
00196     const std::string MAX_FIELD = "max";
00197     const math::Vector2f BACKGROUND_POSITION = math::Vector2f(0.0f, 0.0f);
00198     const float BACKGROUND_PARALLAX_DIRECTION_X = -1.0f;
00199     const float BACKGROUND_PARALLAX_DIRECTION_Y = 0.0f;
00200     const std::string EMPTY_PREFAB = "empty";
00201
00202     const std::string HORIZONTAL_LINE_TYPE = "horizontalLine";
00203     const std::string VERTICAL_LINE_TYPE = "verticalLine";
00204     const std::string UNIQUE_TYPE = "unique";
00205     const std::string RANDOM_TYPE = "random";
00206     const std::string UNIFORM_TYPE = "uniform";
00207
00208     /* Animation conditions */
00209     const std::string VELOCITY_UP_CONDITION = "isVelocityUp";
00210     const std::string VELOCITY_DOWN_CONDITION = "isVelocityDown";
00211
00212     /* Tags */
00213     const std::string CONTROLLABLETAG = "ControllableTag";
00214     const std::string PLAYERTAG = "PlayerTag";
00215     const std::string COLLIDERCOMPONENT = "ColliderComponent";
00216     const std::string MOBTAG = "MobTag";
00217     const std::string SHOOTERTAG = "ShooterTag";
00218     const std::string PLAYERPROJECTILETAG = "PlayerProjectileTag";
00219     const std::string ENEMYPROJECTILETAG = "EnemyProjectileTag";
00220     const std::string PROJECTILEPASSTHROUGHTAG = "ProjectilePassThroughTag";
00221     const std::string PARALLAXCOMPONENT = "ParallaxComponent";
00222     const std::string GAMEZONECOMPONENT = "GameZoneComponent";
00223     const std::string GAMEZONECOLLIDERTAG = "GameZoneColliderTag";
00224     const std::string OBSTACLETAG = "ObstacleTag";
00225     const std::string AIMOVERTAG = "AIMoverTag";
00226     const std::string AISHOOTERTAG = "AIShooterTag";
00227
00228     /* Action constants */
00229     const std::string DEALDEATH_ACTION = "DealDeath";
00230     const std::string TAKEDEATH_ACTION = "TakeDeath";
00231     const std::string DEALDAMAGE_ACTION = "DealDamage";
00232     const std::string TAKEDAMAGE_ACTION = "TakeDamage";
00233
00234     /* Prefabs */
00235     const std::string GAME_ZONE_PREFAB = "gamezone";
00236     const std::string SMALL_EXPLOSION = "small_explosion";
00237     const std::string BIG_EXPLOSION = "big_explosion";
00238
00239     /* AI Movement Pattern defaults */
00240     constexpr float DEFAULT_ZIGZAG_AMPLITUDE = 80.0f;
00241     constexpr float DEFAULT_ZIGZAG_FREQUENCY = 2.0f;
00242     constexpr float DEFAULT_DETECTION_RANGE = 800.0f;
00243     constexpr float DEFAULT_VERTICAL_DEADZONE = 10.0f;
00244     constexpr float DEFAULT_TIMER = 0.0f;
00245
00246     /* Packet constants */
00247     constexpr std::uint8_t PACKET_NO_OP = 0x00;
00248     constexpr std::uint8_t PACKET_CONNECTION = 0x01;
00249     constexpr std::uint8_t PACKET_ACCEPT = 0x02;
00250     constexpr std::uint8_t PACKET_DISC = 0x03;

```

```

00251     constexpr std::uint8_t PACKET_EVENT = 0x04;
00252     constexpr std::uint8_t PACKET_GAME_STATE = 0x05;
00253     constexpr std::uint8_t PACKET_MAP = 0x06;
00254     constexpr std::uint8_t PACKET_END_MAP = 0x07;
00255     constexpr std::uint8_t PACKET_END_GAME = 0x08;
00256     constexpr std::uint8_t PACKET_CAN_START = 0x09;
00257     constexpr std::uint8_t PACKET_CLIENT_READY = 0x0A;
00258     constexpr std::uint8_t PACKET_SPAWN = 0x0B;
00259     constexpr std::uint8_t PACKET_DEATH = 0x0C;
00260     constexpr std::uint8_t PACKET_WHOAMI = 0x0D;
00261
00262     const int MAX_INDEX_PACKET_TYPE = 14;
00263 }
00264
00265 #endif /* !CONSTANTS_HPP_ */

```

## 5.16 Core.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** Core.hpp
00006 */
00007
00008 #ifndef CORE_HPP_
00009 #define CORE_HPP_
00010
00011 #include <memory>
00012 #include <thread>
00013 #include "../common/resourceManager/ResourceManager.hpp"
00014 #include "ClientNetwork.hpp"
00015 #include "../libs/Multimedia/IWindow.hpp"
00016 #include "../libs/Multimedia/IEvent.hpp"
00017 #include "../libs/Multimedia/IAudio.hpp"
00018 #include "gsm/machine/GameStateMachine.hpp"
00019 #include "../common/DLLoader/DLLoader.hpp"
00020 #include "../common/Parser/Parser.hpp"
00021
00022 class Core
00023 {
00024     public:
00025         Core();
00026         ~Core();
00027
00028         void initFirstScene();
00029         void run();
00030         void startNetwork();
00031
00032         std::shared_ptr<ClientNetwork> getNetwork();
00033
00034     private:
00035         std::shared_ptr<DLLoader<gfx::createWindow_t> _windowLoader;
00036         std::shared_ptr<DLLoader<gfx::createEvent_t> _eventLoader;
00037         std::shared_ptr<DLLoader<gfx::createAudio_t> _audioLoader;
00038
00039         std::shared_ptr<ResourceManager> _resourceManager;
00040         std::shared_ptr<gsm::GameStateMachine> _gsm;
00041         std::shared_ptr<ecs::Registry> _registry;
00042         std::shared_ptr<ClientNetwork> _clientNetwork;
00043         std::shared_ptr<Parser> _parser;
00044         std::thread _networkThread;
00045
00046         void initNetwork();
00047         void initLibraries();
00048         void networkLoop();
00049 };
00050
00051 #endif /* !CORE_HPP_ */

```

## 5.17 Core.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** Core
00006 */

```

```

00007
00008 #ifndef CORE_HPP_
00009 #define CORE_HPP_
00010
00011 #include <memory>
00012 #include <stack>
00013 #include "Server.hpp"
00014 #include "Utils.hpp"
00015 #include "ServerConfig.hpp"
00016 #include "../common/resourceManager/ResourceManager.hpp"
00017 #include <thread>
00018 #include "../common/ECS/entity/registry/Registry.hpp"
00019 #include "../common/Parser/Parser.hpp"
00020 #include "../common/systems/systemManager/ISystemManager.hpp"
00021 #include "gsm/machine/GameStateMachine.hpp"
00022 #include "initResourcesManager/ServerInputProvider.hpp"
00023
00024 class Core {
00025     public:
00026         Core();
00027         ~Core();
00028
00029         void init();
00030         void loop();
00031
00032         std::shared_ptr<rserv::Server> getServer() const;
00033         std::shared_ptr<rserv::ServerConfig> getConfig() const;
00034         std::shared_ptr<ResourceManager> getResourceManager() const;
00035         std::shared_ptr<ecs::Registry> getRegistry() const;
00036         std::shared_ptr<Parser> getParser() const;
00037         std::shared_ptr<ecs::ISystemManager> getSystemsManager() const;
00038         std::shared_ptr<gsm::GameStateMachine> getGameStateMachine() const;
00039         void processServerEvents();
00040
00041     protected:
00042     private:
00043         std::shared_ptr<Utils> _utils;
00044         std::shared_ptr<rserv::Server> _server;
00045         std::thread _serverThread;
00046
00047         std::shared_ptr<ResourceManager> _resourceManager;
00048         std::shared_ptr<ecs::Registry> _registry;
00049         std::shared_ptr<Parser> _parser;
00050         std::shared_ptr<ecs::ISystemManager> _systemsManager;
00051         std::shared_ptr<gsm::GameStateMachine> _gsm;
00052         std::shared_ptr<ecs::ServerInputProvider> _inputProvider;
00053 };
00054
00055 #endif /* !CORE_HPP_ */

```

## 5.18 AGameStateMachine.hpp

```

00001 #pragma once
00002
00003 #include "../common/gsm/IGameStateMachine.hpp"
00004 #include "../common/gsm/IGameState.hpp"
00005
00006 namespace gsm {
00007
00008 class AGameStateMachine : public IGameStateMachine {
00009     public:
00010         AGameStateMachine();
00011         ~AGameStateMachine() override = default;
00012
00013         void changeState(std::shared_ptr<IGameState> newState) override;
00014         void pushState(std::shared_ptr<IGameState> newState) override;
00015         void popState() override;
00016         void requestStateChange(std::shared_ptr<IGameState> newState) override;
00017         void requestStatePush(std::shared_ptr<IGameState> newState) override;
00018         void requestStatePop() override;
00019
00020         void update(float deltaTime) override;
00021
00022     protected:
00023         std::stack<std::shared_ptr<IGameState>> _states;
00024         std::shared_ptr<IGameState> _pendingChangeState;
00025         std::shared_ptr<IGameState> _pendingPushState;
00026         bool _pendingPopState = false;
00027 };
00028
00029 } // namespace gsm

```

## 5.19 AGameStateMachine.hpp

```

00001 #pragma once
00002
00003 #include "../../common/gsm/IGameStateMachine.hpp"
00004 #include "../../common/gsm/IGameState.hpp"
00005
00006 namespace gsm {
00007
00008 class AGameStateMachine : public IGameStateMachine {
00009 public:
00010     AGameStateMachine();
00011     ~AGameStateMachine() override;
00012
00013     void changeState(std::shared_ptr<IGameState> newState) override;
00014     void pushState(std::shared_ptr<IGameState> newState) override;
00015     void popState() override;
00016
00017     void update(float deltaTime) override;
00018
00019 protected:
00020     std::stack<std::shared_ptr<IGameState>> _states;
00021 };
00022
00023 } // namespace gsm

```

## 5.20 GameStateMachine.hpp

```

00001 #pragma once
00002
00003 #include "AGameStateMachine.hpp"
00004
00005 namespace gsm {
00006
00007 class GameStateMachine : public AGameStateMachine {
00008 public:
00009     GameStateMachine();
00010     ~GameStateMachine() override = default;
00011 };
00012
00013 } // namespace gsm

```

## 5.21 GameStateMachine.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** GameStateMachine
00006 */
00007
00008 #pragma once
00009
00010 #include "AGameStateMachine.hpp"
00011
00012 namespace gsm {
00013
00014 class GameStateMachine : public AGameStateMachine {
00015 public:
00016     GameStateMachine();
00017     ~GameStateMachine() override = default;
00018
00019     void requestStateChange(std::shared_ptr<IGameState> newState) override;
00020     void requestStatePush(std::shared_ptr<IGameState> newState) override;
00021     void requestStatePop() override;
00022 };
00023
00024 } // namespace gsm

```

## 5.22 AGameState.hpp

```

00001 #pragma once
00002

```

```

00003 #include "../common/gsm/IGameState.hpp"
00004 #include "../common/resourceManager/ResourceManager.hpp"
00005
00006 namespace gsm {
00007
00008 class AGameState : public IGameState {
00009     public:
00010         AGameState(std::shared_ptr<IGameStateMachine> gsm, std::shared_ptr<ResourceManager>
resourceManager);
00011         ~AGameState() override = default;
00012
00013         void enter() override;
00014         void update(float deltaTime) override;
00015         void exit() override;
00016         std::vector<std::shared_ptr<ecs::ISystem>> getSystems() const override;
00017
00018     protected:
00019         void addSystem(std::shared_ptr<ecs::ISystem> system) override;
00020         std::weak_ptr<IGameStateMachine> _gsm;
00021         std::shared_ptr<ResourceManager> _resourceManager;
00022         std::vector<std::shared_ptr<ecs::ISystem>> _systems;
00023 };
00024
00025 } // namespace gsm

```

## 5.23 AGameState.hpp

```

00001 #pragma once
00002
00003 #include "../common/gsm/IGameState.hpp"
00004 #include "resourceManager/ResourceManager.hpp"
00005
00006 namespace gsm {
00007
00008 class AGameState : public IGameState {
00009     public:
00010         AGameState(std::shared_ptr<IGameStateMachine> gsm, std::shared_ptr<ResourceManager>
resourceManager);
00011         ~AGameState() override;
00012
00013         void enter() override;
00014         void update(float deltaTime) override;
00015         void exit() override;
00016         std::vector<std::shared_ptr<ecs::ISystem>> getSystems() const override;
00017
00018     protected:
00019         void addSystem(std::shared_ptr<ecs::ISystem> system) override;
00020         std::weak_ptr<IGameStateMachine> _gsm;
00021         std::shared_ptr<ResourceManager> _resourceManager;
00022         std::vector<std::shared_ptr<ecs::ISystem>> _systems;
00023 };
00024
00025 } // namespace gsm

```

## 5.24 BootState.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** BootState
00006 */
00007
00008 #ifndef BOOTSTATE_HPP_
00009 #define BOOTSTATE_HPP_
00010
00011 #include "../base/AGameState.hpp"
00012 #include "resourceManager/ResourceManager.hpp"
00013
00014 namespace gsm {
00015
00016 class BootState : public AGameState {
00017     public:
00018         BootState(std::shared_ptr<IGameStateMachine> gsm, std::shared_ptr<ResourceManager>
resourceManager);
00019         ~BootState() override = default;
00020
00021         void enter() override;

```

```

00022     void update(float deltaTime) override;
00023     void exit() override;
00024 };
00025
00026 } // namespace gsm
00027
00028 #endif // BOOTSTATE_HPP_

```

## 5.25 BootState.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** BootState
00006  */
00007
00008 #ifndef SERVER_BOOTSTATE_HPP_
00009 #define SERVER_BOOTSTATE_HPP_
00010
00011 #include "../AGameState.hpp"
00012 #include "resourceManager/ResourceManager.hpp"
00013
00014 namespace gsm {
00015
00016 class BootState : public AGameState {
00017 public:
00018     BootState(std::shared_ptr<IGameStateMachine> gsm, std::shared_ptr<ResourceManager>
00019 resourceManager);
00020     ~BootState() override = default;
00021     void enter() override;
00022 };
00023
00024 } // namespace gsm
00025
00026 #endif // SERVER_BOOTSTATE_HPP_

```

## 5.26 DevState.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** DevState
00006  */
00007
00008 #ifndef DEVSTATE_HPP_
00009 #define DEVSTATE_HPP_
00010
00011 #include "../base/AGameState.hpp"
00012 #include <vector>
00013 #include <string>
00014 #include <memory>
00015 #include "resourceManager/ResourceManager.hpp"
00016 #include "../common/ECS/entity/registry/Registry.hpp"
00017 #include "../common/Prefab/entityPrefabManager/EntityPrefabManager.hpp"
00018 #include "../common/Parser/Parser.hpp"
00019 #include "../libs/Multimedia/IWindow.hpp"
00020
00021 namespace gsm {
00022
00023 struct ScoreFeedback {
00024     std::string text;
00025     float lifetime;
00026     float maxLifetime;
00027 };
00028
00029 class DevState : public AGameState {
00030 public:
00031     DevState(std::shared_ptr<IGameStateMachine> gsm, std::shared_ptr<ResourceManager>
00032 resourceManager);
00033     ~DevState() override = default;
00034     void enter() override;
00035     void update(float deltaTime) override;
00036     void exit() override;
00037

```

```

00038     private:
00039         void renderHUD();
00040         void drawHealthHUD(std::shared_ptr<gfx::IWindow> window, float health, float maxHealth);
00041         void drawScoreHUD(std::shared_ptr<gfx::IWindow> window, int score);
00042
00043     private:
00044         std::shared_ptr<ecs::Registry> _registry;
00045         std::shared_ptr<EntityPrefabManager> _prefabManager;
00046         std::shared_ptr<Parser> _parser;
00047         int _previousScore;
00048         int _previousHealth;
00049         std::vector<ScoreFeedback> _scoreFeedbacks;
00050         std::vector<ScoreFeedback> _healthFeedbacks;
00051 };
00052
00053 } // namespace gsm
00054
00055 #endif // DEVSTATE_HPP_
00056

```

## 5.27 InGameState.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** InGameState
00006  */
00007
00008 #ifndef INGAMESTATE_HPP_
00009 #define INGAMESTATE_HPP_
00010
00011 #include "../base/AGameState.hpp"
00012 #include "resourceManager/ResourceManager.hpp"
00013 #include "../common/Prefab/entityPrefabManager/EntityPrefabManager.hpp"
00014 #include "../common/Parser/Parser.hpp"
00015 #include "../libs/Multimedia/IWindow.hpp"
00016 #include <vector>
00017 #include <string>
00018 #include <memory>
00019
00020 namespace gsm {
00021
00022     struct ScoreFeedback {
00023         std::string text;
00024         float lifetime;
00025         float maxLifetime;
00026     };
00027
00028     class InGameState : public AGameState {
00029     public:
00030         InGameState(std::shared_ptr<IGameStateMachine> gsm, std::shared_ptr<ResourceManager>
resourceManager);
00031         ~InGameState() override = default;
00032
00033         void enter() override;
00034         void update(float deltaTime) override;
00035         void exit() override;
00036
00037     private:
00038         void renderHUD();
00039         void drawHealthHUD(std::shared_ptr<gfx::IWindow> window, float health, float maxHealth);
00040         void drawScoreHUD(std::shared_ptr<gfx::IWindow> window, int score);
00041
00042     private:
00043         std::shared_ptr<ecs::Registry> _registry;
00044         std::shared_ptr<EntityPrefabManager> _prefabManager;
00045         std::shared_ptr<Parser> _parser;
00046         int _previousScore;
00047         int _previousHealth;
00048         std::vector<ScoreFeedback> _scoreFeedbacks;
00049         std::vector<ScoreFeedback> _healthFeedbacks;
00050     };
00051
00052 } // namespace gsm
00053
00054 #endif // INGAMESTATE_HPP_

```

## 5.28 InGameState.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** InGameState
00006 */
00007
00008 #ifndef SERVER_INGAMESTATE_HPP_
00009 #define SERVER_INGAMESTATE_HPP_
00010
00011 #include "../AGameState.hpp"
00012 #include "resourceManager/ResourceManager.hpp"
00013 #include "systems/base/ISystem.hpp"
00014
00015 namespace gsm {
00016
00017 class InGameState : public AGameState {
00018 public:
00019     InGameState(std::shared_ptr<IGameStateMachine> gsm, std::shared_ptr<ResourceManager>
00020 resourceManager);
00021     ~InGameState() override = default;
00022     void enter() override;
00023
00024 private:
00025 };
00026
00027 };
00028 } // namespace gsm
00029
00030 #endif // SERVER_INGAMESTATE_HPP_

```

## 5.29 LoadingState.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** LoadingState
00006 */
00007
00008 #ifndef LOADINGSTATE_HPP_
00009 #define LOADINGSTATE_HPP_
00010
00011 #include "../base/AGGameState.hpp"
00012 #include "resourceManager/ResourceManager.hpp"
00013
00014 namespace gsm {
00015
00016 class LoadingState : public AGameState {
00017 public:
00018     LoadingState(std::shared_ptr<IGameStateMachine> gsm, std::shared_ptr<ResourceManager>
00019 resourceManager);
00020     ~LoadingState() override = default;
00021     void enter() override;
00022     void update(float deltaTime) override;
00023     void exit() override;
00024 };
00025
00026 } // namespace gsm
00027
00028 #endif // LOADINGSTATE_HPP_

```

## 5.30 LoadingState.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** LoadingState
00006 */
00007
00008 #ifndef SERVER_LOADINGSTATE_HPP_

```



```

00009 #define SERVER_LOADINGSTATE_HPP_
00010
00011 #include "../AGameState.hpp"
00012 #include "resourceManager/ResourceManager.hpp"
00013
00014 namespace gsm {
00015
00016 class LoadingState : public AGameState {
00017 public:
00018     LoadingState(std::shared_ptr<IGameStateMachine> gsm, std::shared_ptr<ResourceManager>
resourceManager);
00019     ~LoadingState() override = default;
00020
00021     void enter() override;
00022 };
00023
00024 } // namespace gsm
00025
00026 #endif // SERVER_LOADINGSTATE_HPP_

```

## 5.31 LobbyState.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** LobbyState
00006 */
00007
00008 #ifndef LOBBYSTATE_HPP_
00009 #define LOBBYSTATE_HPP_
00010
00011 #include "../base/AGameState.hpp"
00012 #include "resourceManager/ResourceManager.hpp"
00013
00014 namespace gsm {
00015
00016 class LobbyState : public AGameState {
00017 public:
00018     LobbyState(std::shared_ptr<IGameStateMachine> gsm, std::shared_ptr<ResourceManager>
resourceManager);
00019     ~LobbyState() override = default;
00020
00021     void enter() override;
00022     void update(float deltaTime) override;
00023     void exit() override;
00024 };
00025
00026 } // namespace gsm
00027
00028 #endif // LOBBYSTATE_HPP_

```

## 5.32 LobbyState.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** LobbyState
00006 */
00007
00008 #ifndef SERVER_LOBBYSTATE_HPP_
00009 #define SERVER_LOBBYSTATE_HPP_
00010
00011 #include "../AGameState.hpp"
00012 #include "resourceManager/ResourceManager.hpp"
00013
00014 namespace gsm {
00015
00016 class LobbyState : public AGameState {
00017 public:
00018     LobbyState(std::shared_ptr<IGameStateMachine> gsm, std::shared_ptr<ResourceManager>
resourceManager);
00019     ~LobbyState() override = default;
00020
00021     void enter() override;
00022     void update(float deltaTime) override;
00023 };

```

```

00024
00025 } // namespace gsm
00026
00027 #endif // SERVER_LOBBYSTATE_HPP_

```

### 5.33 LobbyBrowserState.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** LobbyBrowserState
00006 */
00007
00008 #ifndef LOBBYBROWSERSTATE_HPP_
00009 #define LOBBYBROWSERSTATE_HPP_
00010
00011 #include "../base/AGameState.hpp"
00012 #include "resourceManager/ResourceManager.hpp"
00013
00014 namespace gsm {
00015
00016 class LobbyBrowserState : public AGameState {
00017 public:
00018     LobbyBrowserState(std::shared_ptr<IGameStateMachine> gsm, std::shared_ptr<ResourceManager>
00019         resourceManager);
00020     ~LobbyBrowserState() override = default;
00021     void enter() override;
00022     void update(float deltaTime) override;
00023     void exit() override;
00024 };
00025
00026 } // namespace gsm
00027
00028 #endif // LOBBYBROWSERSTATE_HPP_

```

### 5.34 MainMenuState.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** MainMenuState
00006 */
00007
00008 #ifndef MAINMENUSTATE_HPP_
00009 #define MAINMENUSTATE_HPP_
00010
00011 #include "../base/AGameState.hpp"
00012 #include "resourceManager/ResourceManager.hpp"
00013 #include "../input/MouseInputHandler.hpp"
00014 #include "../ui/elements/focusable/Button.hpp"
00015 #include "../ui/manager/UIManager.hpp"
00016 #include "../ui/core/UILayout.hpp"
00017 #include "../ui/elements/Background.hpp"
00018 #include "../ui/elements/focusable/TextInput.hpp"
00019
00020
00021 namespace gsm {
00022
00023 class MainMenuState : public AGameState {
00024 public:
00025     MainMenuState(std::shared_ptr<IGameStateMachine> gsm, std::shared_ptr<ResourceManager>
00026         resourceManager);
00027     ~MainMenuState() override = default;
00028     void enter() override;
00029     void update(float deltaTime) override;
00030     void exit() override;
00031
00032 private:
00033     void renderUI();
00034     void updatePlayButtonText();
00035
00036 private:
00037     std::unique_ptr<MouseInputHandler> _mouseHandler;
00038     std::shared_ptr<ui::Button> _playButton;

```

```

00039     std::shared_ptr<ui::Button> _settingsButton;
00040     std::shared_ptr<ui::Button> _quitButton;
00041     std::shared_ptr<ui::Button> _connectButton;
00042     std::unique_ptr<ui::UIManager> _uiManager;
00043     std::shared_ptr<ui::UILayout> _leftLayout;
00044     std::shared_ptr<ui::UILayout> _mainMenuLayout;
00045
00046     std::shared_ptr<ui::UILayout> _rightLayout;
00047     std::shared_ptr<ui::Button> _devButton;
00048
00049     std::shared_ptr<ui::TextInput> _ipInput;
00050     std::shared_ptr<ui::TextInput> _portInput;
00051
00052     std::shared_ptr<ui::Background> _background;
00053 };
00054
00055 } // namespace gsm
00056
00057 #endif // MAINMENUSTATE_HPP_

```

## 5.35 PauseState.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** PauseState
00006  */
00007
00008 #ifndef PAUSESTATE_HPP_
00009 #define PAUSESTATE_HPP_
00010
00011 #include "../base/AGameState.hpp"
00012 #include "resourceManager/ResourceManager.hpp"
00013
00014 namespace gsm {
00015
00016 class PauseState : public AGameState {
00017 public:
00018     PauseState(std::shared_ptr<IGameStateMachine> gsm, std::shared_ptr<ResourceManager>
00019 resourceManager);
00019     ~PauseState() override = default;
00020
00021     void enter() override;
00022     void update(float deltaTime) override;
00023     void exit() override;
00024 };
00025
00026 } // namespace gsm
00027
00028 #endif // PAUSESTATE_HPP_

```

## 5.36 ResultsState.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** ResultsState
00006  */
00007
00008 #ifndef RESULTSSSTATE_HPP_
00009 #define RESULTSSSTATE_HPP_
00010
00011 #include "../base/AGameState.hpp"
00012 #include "resourceManager/ResourceManager.hpp"
00013
00014 namespace gsm {
00015
00016 class ResultsState : public AGameState {
00017 public:
00018     ResultsState(std::shared_ptr<IGameStateMachine> gsm, std::shared_ptr<ResourceManager>
00019 resourceManager);
00019     ~ResultsState() override = default;
00020
00021     void enter() override;
00022     void update(float deltaTime) override;
00023     void exit() override;

```

```

00024 };
00025
00026 } // namespace gsm
00027
00028 #endif // RESULTSSTATE_HPP_

```

## 5.37 SettingsState.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** SettingsState
00006 */
00007
00008 #ifndef SETTINGSSTATE_HPP_
00009 #define SETTINGSSTATE_HPP_
00010
00011 #include "../base/AGameState.hpp"
00012 #include "resourceManager/ResourceManager.hpp"
00013 #include "../input/MouseInputHandler.hpp"
00014 #include "../ui/elements/focusable/Button.hpp"
00015 #include "../ui/elements/focusable/Slider.hpp"
00016 #include "../ui/elements/focusable/ToggleSwitch.hpp"
00017 #include "../ui/elements/Text.hpp"
00018 #include "../ui/elements/Background.hpp"
00019 #include "../ui/manager/UISManager.hpp"
00020 #include "../ui/core/UILayout.hpp"
00021 #include "../common/types/Vector2f.hpp"
00022 #include "../common/InputMapping/InputAction.hpp"
00023 #include "../libs/Multimedia/EventTypes.hpp"
00024 #include <optional>
00025 #include "../SettingsManager.hpp"
00026
00027 namespace gsm {
00028
00029 class SettingsState : public AGameState {
00030 public:
00031     SettingsState(std::shared_ptr<IGameStateMachine> gsm, std::shared_ptr<ResourceManager>
resourceManager);
00032     ~SettingsState() override = default;
00033
00034     void enter() override;
00035     void update(float deltaTime) override;
00036     void exit() override;
00037
00038 private:
00039     void renderUI();
00040     void cycleColorBlindnessFilter();
00041     void toggleHighContrastFilter();
00042     void updateBrightnessFilter(float value);
00043     void applyColorBlindnessFilter(int state);
00044     void applyHighContrastFilter(bool enabled);
00045     void cycleUIScale();
00046     void updateMusicVolume(float value);
00047     void updateSoundVolume(float value);
00048     void updateToggleValue(bool value);
00049     void cycleScreenResolution();
00050     void updateTargetFPS(int fps);
00051     void updateRenderQuality(float quality);
00052     void setScreenResolution(SettingsConfig::ScreenResolution resolution);
00053     void updateResolutionButtonColors(SettingsConfig::ScreenResolution current);
00054
00055     void startKeyRebind(ecs::RemappableAction action, bool rebindPrimary, std::shared_ptr<ui::Button>
button);
00056     void handleKeyRebind(gfx::EventType newKey);
00057     void updateKeyBindingButtonText(std::shared_ptr<ui::Button> button, ecs::RemappableAction action,
bool isPrimary);
00058     std::string getRemappableActionName(ecs::RemappableAction action) const;
00059
00060     std::string getScreenResolutionText(SettingsConfig::ScreenResolution resolution);
00061
00062 private:
00063     std::unique_ptr<MouseInputHandler> _mouseHandler;
00064     std::shared_ptr<ui::Button> _backButton;
00065     std::shared_ptr<ui::Button> _highContrastButton;
00066     std::shared_ptr<ui::Button> _colorBlindnessButton;
00067     std::shared_ptr<ui::Slider> _brightnessSlider;
00068     std::shared_ptr<ui::Slider> _musicVolumeSlider;
00069     std::shared_ptr<ui::Slider> _soundVolumeSlider;
00070     std::shared_ptr<ui::ToggleSwitch> _toggleSwitch;
00071     std::shared_ptr<ui::Text> _toggleLabel;
00072     std::shared_ptr<ui::UILayout> _toggleLayout;

```

```

00073     std::vector<std::shared_ptr<ui::Button> _resolutionButtons;
00074     std::shared_ptr<ui::Slider> _fpsSlider;
00075     std::shared_ptr<ui::Slider> _renderQualitySlider;
00076     std::shared_ptr<ui::Button> _scaleButton;
00077     std::unique_ptr<ui::UIManager> _uiManager;
00078     std::shared_ptr<ui::UILayout> _settingsLayout;
00079     std::shared_ptr<ui::UILayout> _leftColumnLayout;
00080     std::shared_ptr<ui::UILayout> _rightColumnLayout;
00081     std::shared_ptr<ui::UILayout> _centerColumnLayout;
00082     std::shared_ptr<ui::UILayout> _titleLabel;
00083     std::shared_ptr<ui::Background> _background;
00084     math::Vector2f _savedViewCenter;
00085
00086     std::shared_ptr<SettingsManager> _settingsManager;
00087
00088     std::shared_ptr<ui::UILayout> _moveUpLayout;
00089     std::shared_ptr<ui::Text> _moveUpLabel;
00090     std::shared_ptr<ui::Button> _moveUpPrimaryButton;
00091     std::shared_ptr<ui::Button> _moveUpSecondaryButton;
00092
00093     std::shared_ptr<ui::UILayout> _moveDownLayout;
00094     std::shared_ptr<ui::Text> _moveDownLabel;
00095     std::shared_ptr<ui::Button> _moveDownPrimaryButton;
00096     std::shared_ptr<ui::Button> _moveDownSecondaryButton;
00097
00098     std::shared_ptr<ui::UILayout> _moveLeftLayout;
00099     std::shared_ptr<ui::Text> _moveLeftLabel;
00100     std::shared_ptr<ui::Button> _moveLeftPrimaryButton;
00101     std::shared_ptr<ui::Button> _moveLeftSecondaryButton;
00102
00103     std::shared_ptr<ui::UILayout> _moveRightLayout;
00104     std::shared_ptr<ui::Text> _moveRightLabel;
00105     std::shared_ptr<ui::Button> _moveRightPrimaryButton;
00106     std::shared_ptr<ui::Button> _moveRightSecondaryButton;
00107
00108     std::shared_ptr<ui::UILayout> _shootLayout;
00109     std::shared_ptr<ui::Text> _shootLabel;
00110     std::shared_ptr<ui::Button> _shootPrimaryButton;
00111     std::shared_ptr<ui::Button> _shootSecondaryButton;
00112
00113     bool _isWaitingForKey = false;
00114     std::optional<ecs::RemappableAction> _actionToRebind;
00115     bool _rebindingPrimary = true;
00116     std::string _rebindLabel;
00117     std::shared_ptr<ui::Button> _buttonToUpdate;
00118     gfx::EventType _originalKey = gfx::EventType::NOTHING;
00119
00120     std::string getColorBlindnessText(int state);
00121     std::string getUIScaleText(ui::UIScale scale);
00122 };
00123
00124 } // namespace gsm
00125
00126 #endif // SETTINGSSTATE_HPP_

```

## 5.38 GraphicalInputProvider.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** GraphicalInputProvider
00006  */
00007
00008 #include "../common/InputMapping/IInputProvider.hpp"
00009 #include "../libs/Multimedia/IEvent.hpp"
00010 #include "../common/InputMapping/InputMappingManager.hpp"
00011 #include <memory>
00012
00013 namespace ecs {
00014
00015     class GraphicalInputProvider : public IInputProvider {
00016     public:
00017         GraphicalInputProvider(std::shared_ptr<gfx::IEvent> eventSystem,
00018                               std::shared_ptr<InputMappingManager> mappingManager);
00019         ~GraphicalInputProvider() override = default;
00020
00021         float getAxisValue(event_t axis, size_t clientID = 0) override;
00022
00023         bool isActionPressed(InputAction action, size_t clientID = 0) override;
00024         float getActionAxis(InputAction action, size_t clientID = 0) override;
00025         InputMapping getInputMapping(size_t clientID = 0) const override;
00026

```

```

00027         void setToggleMode(bool enabled);
00028         bool isToggleMode() const;
00029
00030     private:
00031         std::shared_ptr<gfx::IEvent> _eventSystem;
00032         std::shared_ptr<InputMappingManager> _mappingManager;
00033         bool _toggleMode;
00034         std::map<InputAction, bool> _toggledStates;
00035         std::map<InputAction, bool> _lastKeyState;
00036         std::map<std::pair<InputAction, gfx::EventType>, bool> _keyPressedState;
00037         std::map<std::pair<InputAction, gfx::EventType>, bool> _toggledKeyStates;
00038         std::map<std::pair<InputAction, gfx::EventType>, int> _lastToggleFrame;
00039         int _currentFrame;
00040     };
00041
00042 } // namespace ecs

```

## 5.39 initResourcesManager.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** initResourcesManager
00006 */
00007
00008 #ifndef initResourcesManager_HPP_
00009 #define initResourcesManager_HPP_
00010
00011 #include "../common/resourceManager/ResourceManager.hpp"
00012 #include <memory>
00013 #include "../common/Parser/Parser.hpp"
00014 #include "../common/DLLoader/DLLoader.hpp"
00015 #include "../libs/Multimedia/IWindow.hpp"
00016 #include "../libs/Multimedia/IEvent.hpp"
00017 #include "../libs/Multimedia/IAudio.hpp"
00018
00019 std::shared_ptr<ResourceManager> initResourcesManager(
00020     std::shared_ptr<DLLoader<gfx::createWindow_t>>,
00021     std::shared_ptr<DLLoader<gfx::createEvent_t>>,
00022     std::shared_ptr<DLLoader<gfx::createAudio_t>>,
00023     std::shared_ptr<ClientNetwork>,
00024     std::shared_ptr<Parser> parser
00025 );
00026
00027 #endif /* !initResourcesManager_HPP_ */

```

## 5.40 initResourcesManager.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** initResourcesManager
00006 */
00007
00008 #ifndef initResourcesManager_HPP_
00009 #define initResourcesManager_HPP_
00010
00011 #include "../common/resourceManager/ResourceManager.hpp"
00012 #include "../Server.hpp"
00013 #include "../common/ECS/entity/registry/Registry.hpp"
00014 #include "../common/Parser/Parser.hpp"
00015 #include "../common/systems/systemManager/ISystemManager.hpp"
00016 #include "../gsm/machine/GameStateMachine.hpp"
00017 #include <memory>
00018
00019 std::shared_ptr<ResourceManager> initResourcesManager(
00020     std::shared_ptr<rserve::Server> server,
00021     std::shared_ptr<ecs::Registry> registry,
00022     std::shared_ptr<Parser> parser,
00023     std::shared_ptr<ecs::ISystemManager> systemsManager,
00024     std::shared_ptr<gsm::GameStateMachine> gameStateMachine,
00025     std::shared_ptr<ecs::IInputProvider> inputProvider
00026 );
00027
00028 #endif /* !initResourcesManager_HPP_ */

```

## 5.41 MouseInputHandler.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** MouseInputHandler
00006  */
00007
00008  #ifndef MOUSEINPUTHANDLER_HPP_
00009  #define MOUSEINPUTHANDLER_HPP_
00010
00011  #include <memory>
00012  #include <optional>
00013  #include "../common/resourceManager/ResourceManager.hpp"
00014  #include "../common/types/Vector2f.hpp"
00015  #include "../constants.hpp"
00016
00017  struct MouseClickInfo {
00018      math::Vector2f position;
00019      constants::MouseButton button;
00020  };
00021
00022  class MouseInputHandler {
00023  public:
00024      MouseInputHandler(std::shared_ptr<ResourceManager> resourceManager);
00025      ~MouseInputHandler() = default;
00026
00027      std::optional<MouseClickInfo> pollMouseClicked();
00028      math::Vector2f getMousePosition() const;
00029      math::Vector2f getWorldMousePosition() const;
00030      math::Vector2f getNormalizedMousePosition() const;
00031      bool isMouseButtonPressed(int button) const;
00032
00033  private:
00034      std::weak_ptr<ResourceManager> _resourceManager;
00035  };
00036
00037  #endif /* !MOUSEINPUTHANDLER_HPP_ */

```

## 5.42 NetworkStateComponent.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** NetworkStateComponent
00006  */
00007
00008  #ifndef NETWORKSTATECOMPONENT_HPP_
00009  #define NETWORKSTATECOMPONENT_HPP_
00010
00011  #include "../common/components/base/AComponent.hpp"
00012  #include "../common/types/Vector2f.hpp"
00013  #include <chrono>
00014
00015  namespace ecs {
00016
00017      struct NetworkTransformState {
00018          math::Vector2f position;
00019          float rotation;
00020          math::Vector2f scale;
00021          std::chrono::steady_clock::time_point timestamp;
00022
00023          NetworkTransformState()
00024              : position(0.0f, 0.0f)
00025              , rotation(0.0f)
00026              , scale(1.0f, 1.0f)
00027              , timestamp(std::chrono::steady_clock::now()) {}
00028      };
00029
00030      struct NetworkHealthState {
00031          uint32_t health;
00032          uint32_t baseHealth;
00033          std::chrono::steady_clock::time_point timestamp;
00034
00035          NetworkHealthState()
00036              : health(0)
00037              , baseHealth(0)
00038              , timestamp(std::chrono::steady_clock::now()) {}
00039      };
00040

```

```

00041 class NetworkStateComponent : public AComponent {
00042     public:
00043         NetworkStateComponent()
00044             : _hasTransform(false)
00045             , _hasHealth(false)
00046             , _interpolationTime(0.1f) {}
00047
00048         ~NetworkStateComponent() = default;
00049
00050         void setCurrentTransform(const math::Vector2f& pos, float rot, const math::Vector2f& scale) {
00051             if (_hasTransform) {
00052                 _previousTransform = _currentTransform;
00053             }
00054             _currentTransform.position = pos;
00055             _currentTransform.rotation = rot;
00056             _currentTransform.scale = scale;
00057             _currentTransform.timestamp = std::chrono::steady_clock::now();
00058             _hasTransform = true;
00059         }
00060
00061         bool hasTransform() const { return _hasTransform; }
00062         const NetworkTransformState& getPreviousTransform() const { return _previousTransform; }
00063         const NetworkTransformState& getCurrentTransform() const { return _currentTransform; }
00064
00065         void setCurrentHealth(uint32_t health, uint32_t baseHealth) {
00066             if (_hasHealth) {
00067                 _previousHealth = _currentHealth;
00068             }
00069             _currentHealth.health = health;
00070             _currentHealth.baseHealth = baseHealth;
00071             _currentHealth.timestamp = std::chrono::steady_clock::now();
00072             _hasHealth = true;
00073         }
00074
00075         bool hasHealth() const { return _hasHealth; }
00076         const NetworkHealthState& getPreviousHealth() const { return _previousHealth; }
00077         const NetworkHealthState& getCurrentHealth() const { return _currentHealth; }
00078
00079         void setInterpolationTime(float time) { _interpolationTime = time; }
00080         float getInterpolationTime() const { return _interpolationTime; }
00081
00082         float getTransformInterpolationFactor() const {
00083             if (!_hasTransform) return 1.0f;
00084             auto now = std::chrono::steady_clock::now();
00085             auto elapsed = std::chrono::duration<float>(now - _currentTransform.timestamp).count();
00086             if (elapsed >= _interpolationTime) return 1.0f;
00087             return elapsed / _interpolationTime;
00088         }
00089
00090     private:
00091         NetworkTransformState _previousTransform;
00092         NetworkTransformState _currentTransform;
00093         bool _hasTransform;
00094         NetworkHealthState _previousHealth;
00095         NetworkHealthState _currentHealth;
00096         bool _hasHealth;
00097         float _interpolationTime;
00098 };
00099
00100 } // namespace ecs
00101
00102 #endif /* !NETWORKSTATECOMPONENT_HPP_ */

```

## 5.43 DefaultPacketHandlers.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** Default packet handlers registration (client-side)
00006 */
00007
00008 #ifndef CLIENT_DEFAULT_PACKET_HANDLERS_HPP_
00009 #define CLIENT_DEFAULT_PACKET_HANDLERS_HPP_
00010
00011 #include <memory>
00012 #include "../libs/Packet/IPacketManager.hpp"
00013
00014 namespace rcli::packet {
00015     bool registerDefaultPacketHandlers(std::shared_ptr<pm::IPacketManager> packet);
00016 }
00017
00018 #endif // CLIENT_DEFAULT_PACKET_HANDLERS_HPP_

```



## 5.44 DefaultPacketHandlers.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** Default packet handlers registration (common)
00006 */
00007
00008 #ifndef COMMON_DEFAULT_PACKET_HANDLERS_HPP_
00009 #define COMMON_DEFAULT_PACKET_HANDLERS_HPP_
00010
00011 #include <memory>
00012 #include "../libs/Package/IPacketManager.hpp"
00013
00014 namespace common::packet {
00015     bool registerDefaultPacketHandlers(std::shared_ptr<pm::IPacketManager> packet);
00016 }
00017
00018 #endif // COMMON_DEFAULT_PACKET_HANDLERS_HPP_

```

## 5.45 DefaultPacketHandlers.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** Default packet handlers registration (server-side)
00006 */
00007
00008 #ifndef DEFAULT_PACKET_HANDLERS_HPP_
00009 #define DEFAULT_PACKET_HANDLERS_HPP_
00010
00011 #include <memory>
00012 #include "../libs/Package/IPacketManager.hpp"
00013
00014 namespace rserv::packet {
00015     bool registerDefaultPacketHandlers(std::shared_ptr<pm::IPacketManager> packet);
00016 }
00017
00018 #endif // DEFAULT_PACKET_HANDLERS_HPP_

```

## 5.46 SettingsConfig.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** SettingsConfig
00006 */
00007
00008 #ifndef SETTINGSCONFIG_HPP_
00009 #define SETTINGSCONFIG_HPP_
00010
00011 #include "ui/elements/base/UIElement.hpp"
00012 #include "constants.hpp"
00013
00014 class SettingsConfig {
00015 public:
00016     SettingsConfig() = default;
00017     ~SettingsConfig() = default;
00018
00019     int getColorBlindnessState() const { return _colorBlindnessState; }
00020     void setColorBlindnessState(int state) { _colorBlindnessState = state; }
00021
00022     float getBrightnessValue() const { return _brightnessValue; }
00023     void setBrightnessValue(float value) { _brightnessValue = value; }
00024
00025     bool isHighContrastEnabled() const { return _highContrastEnabled; }
00026     void setHighContrastEnabled(bool enabled) { _highContrastEnabled = enabled; }
00027
00028     ui::UIScale getUIScale() const { return _uiScale; }
00029     void setUIScale(ui::UIScale scale) { _uiScale = scale; }
00030
00031     float getMusicVolume() const { return _musicVolume; }
00032     void setMusicVolume(float volume) { _musicVolume = volume; }
00033

```

```

00034     float getSoundVolume() const { return _soundVolume; }
00035     void setSoundVolume(float volume) { _soundVolume = volume; }
00036
00037     enum class ScreenResolution {
00038         RES_800x600 = 0,
00039         RES_1024x768 = 1,
00040         RES_1280x720 = 2,
00041         RES_1920x1080 = 3,
00042         FULLSCREEN = 4
00043     };
00044
00045     ScreenResolution getScreenResolution() const { return _screenResolution; }
00046     void setScreenResolution(ScreenResolution resolution) { _screenResolution = resolution; }
00047
00048     int getTargetFPS() const { return _targetFPS; }
00049     void setTargetFPS(int fps) { _targetFPS = fps; }
00050
00051     float getRenderQuality() const { return _renderQuality; }
00052     void setRenderQuality(float quality) { _renderQuality = quality; }
00053
00054     std::string getScreenResolutionName(ScreenResolution resolution) const;
00055     std::pair<int, int> getScreenResolutionSize(ScreenResolution resolution) const;
00056     bool isFullscreen(ScreenResolution resolution) const;
00057
00058     void saveAccessibility(const std::string& filepath = constants::ACCESSIBILITY_FILE_PATH);
00059     void loadAccessibility(const std::string& filepath = constants::ACCESSIBILITY_FILE_PATH);
00060
00061     void saveSettings(const std::string& filepath = constants::SETTINGS_FILE_PATH);
00062     void loadSettings(const std::string& filepath = constants::SETTINGS_FILE_PATH);
00063
00064 private:
00065     int _colorBlindnessState = 0;
00066     float _brightnessValue = 1.0f;
00067     bool _highContrastEnabled = false;
00068     ui::UIScale _uiScale = ui::UIScale::Normal;
00069     float _musicVolume = 100.0f;
00070     float _soundVolume = 100.0f;
00071     ScreenResolution _screenResolution = ScreenResolution::RES_1920x1080;
00072     int _targetFPS = 60;
00073     float _renderQuality = 1.0f;
00074 };
00075
00076 #endif // SETTINGSCONFIG_HPP_

```

## 5.47 SettingsManager.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** SettingsManager
00006 */
00007
00008 #ifndef SETTINGSMANAGER_HPP_
00009 #define SETTINGSMANAGER_HPP_
00010
00011 #include <memory>
00012 #include "../common/InputMapping/InputMappingManager.hpp"
00013 #include "../common/InputMapping/IInputProvider.hpp"
00014 #include "SettingsConfig.hpp"
00015 #include "../libs/Multimedia/IWindow.hpp"
00016
00017 class SettingsManager {
00018 public:
00019     SettingsManager(std::shared_ptr<ecs::InputMappingManager> mappingManager,
00020                     std::shared_ptr<ecs::IInputProvider> inputProvider,
00021                     std::shared_ptr<SettingsConfig> settingsConfig);
00022     ~SettingsManager() = default;
00023
00024     void loadAll();
00025     void saveAll();
00026
00027     void saveKeybinds();
00028     void loadKeybinds();
00029
00030     void saveAccessibility();
00031     void loadAccessibility();
00032
00033     void saveSettings();
00034     void loadSettings();
00035
00036     void applyAccessibilityToWindow(std::shared_ptr<gfx::IWindow> window);
00037

```

```

00038 private:
00039     std::shared_ptr<ecs::InputMappingManager> _mappingManager;
00040     std::shared_ptr<ecs::IInputProvider> _inputProvider;
00041     std::shared_ptr<SettingsConfig> _settingsConfig;
00042 };
00043
00044 #endif /* !SETTINGSMANAGER_HPP_ */

```

## 5.48 MusicSystem.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** MusicSystem
00006 */
00007
00008 #ifndef MUSICSYSTEM_HPP_
00009 #define MUSICSYSTEM_HPP_
00010
00011 #include "../common/systems/base/ASystem.hpp"
00012 #include <memory>
00013
00014 namespace ecs {
00015
00016 class MusicSystem : public ASystem {
00017     public:
00018         MusicSystem();
00019         ~MusicSystem() override = default;
00020
00021     protected:
00022         void update(
00023             std::shared_ptr<ResourceManager> resourceManager,
00024             std::shared_ptr<Registry> registry,
00025             float deltaTime
00026         ) override;
00027 };
00028
00029 } // namespace ecs
00030
00031 #endif /* !MUSICSYSTEM_HPP_ */

```

## 5.49 SoundSystem.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** SoundSystem
00006 */
00007
00008 #ifndef SOUNDSYSTEM_HPP_
00009 #define SOUNDSYSTEM_HPP_
00010
00011 #include "../common/systems/base/ASystem.hpp"
00012 #include <memory>
00013
00014 namespace ecs {
00015
00016 class SoundSystem : public ASystem {
00017     public:
00018         SoundSystem();
00019         ~SoundSystem() override = default;
00020
00021     protected:
00022         void update(
00023             std::shared_ptr<ResourceManager> resourceManager,
00024             std::shared_ptr<Registry> registry,
00025             float deltaTime
00026         ) override;
00027 };
00028
00029 } // namespace ecs
00030
00031 #endif /* !SOUNDSYSTEM_HPP_ */

```

## 5.50 MovementInputSystem.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** MovementInputSystem
00006 */
00007
00008 #ifndef MOVEMENTINPUTSYSTEM_HPP_
00009 #define MOVEMENTINPUTSYSTEM_HPP_
00010
00011 #include "../common/systems/base/ASystem.hpp"
00012 #include "../common/components/temporary/InputIntentComponent.hpp"
00013 #include "../common/InputMapping/IInputProvider.hpp"
00014 #include "../common/InputMapping/InputAction.hpp"
00015 #include <memory>
00016
00017 namespace gfx {
00018     class IEvent;
00019 }
00020
00021 namespace ecs {
00022
00023     class MovementInputSystem : public ASystem {
00024     public:
00025         MovementInputSystem();
00026         ~MovementInputSystem() = default;
00027
00028         void update(std::shared_ptr<ResourceManager> resourceManager, std::shared_ptr<Registry>
registry, float deltaTime) override;
00029
00030     private:
00031         math::Vector2f getMovementDirection(std::shared_ptr<ResourceManager> resourceManager) const;
00032         void updateInputIntent(std::shared_ptr<Registry> registry, Entity entityId, const
math::Vector2f &direction);
00033         math::Vector2f getAnalogStickInput(std::shared_ptr<IInputProvider> inputProvider) const;
00034         void sendAxisEvents(std::shared_ptr<ResourceManager> resourceManager, const math::Vector2f
&direction);
00035         bool _wasMovingLastFrame = false;
00036     };
00037
00038 } // namespace ecs
00039
00040 #endif /* !MOVEMENTINPUTSYSTEM_HPP_ */

```

## 5.51 ShootInputSystem.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ShootInputSystem
00006 */
00007
00008 #ifndef SHOOTINPUTSYSTEM_HPP_
00009 #define SHOOTINPUTSYSTEM_HPP_
00010
00011 #include <memory>
00012 #include "../common/systems/base/ASystem.hpp"
00013
00014 namespace ecs {
00015
00016     class ShootInputSystem : public ASystem {
00017     public:
00018         ShootInputSystem();
00019         ~ShootInputSystem() = default;
00020
00021         void update(
00022             std::shared_ptr<ResourceManager> resourceManager,
00023             std::shared_ptr<Registry> registry,
00024             float deltaTime
00025         ) override;
00026
00027     private:
00028     };
00029
00030 }
00031
00032 #endif /* !SHOOTINPUTSYSTEM_HPP_ */

```

## 5.52 NetworkInterpolationSystem.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** NetworkInterpolationSystem
00006 */
00007
00008 #ifndef NETWORKINTERPOLATIONSYSTEM_HPP_
00009 #define NETWORKINTERPOLATIONSYSTEM_HPP_
00010
00011 #include "../common/systems/base/ASystem.hpp"
00012 #include "../common/ECS/entity/registry/Registry.hpp"
00013 #include "../interpolation/NetworkStateComponent.hpp"
00014 #include "../common/components/permanent/TransformComponent.hpp"
00015 #include "../common/components/permanent/HealthComponent.hpp"
00016
00017 namespace ecs {
00018
00019 class NetworkInterpolationSystem : public ASystem {
00020     public:
00021         NetworkInterpolationSystem() = default;
00022         ~NetworkInterpolationSystem() override = default;
00023
00024         void update(std::shared_ptr<ResourceManager> resourceManager,
00025                     std::shared_ptr<Registry> registry,
00026                     float deltaTime) override;
00027
00028     private:
00029         void interpolateTransform(std::shared_ptr<NetworkStateComponent> networkState,
00030                                   std::shared_ptr<TransformComponent> transform);
00031 };
00032
00033 } // namespace ecs
00034
00035 #endif /* !NETWORKINTERPOLATIONSYSTEM_HPP_ */

```

## 5.53 AnimationRenderingSystem.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** AnimationRenderingSystem
00006 */
00007
00008 #ifndef ANIMATIONRENDERINGSYSTEM_HPP_
00009 #define ANIMATIONRENDERINGSYSTEM_HPP_
00010
00011
00012 #include "../common/systems/base/ASystem.hpp"
00013 #include <memory>
00014 #include "../components/rendering/AnimationComponent.hpp"
00015 #include "../common/ECS/entity/Entity.hpp"
00016 #include "../common/ECS/entity/registry/Registry.hpp"
00017 #include "../common/Parser/Animation/AnimationConditionFactory.hpp"
00018 namespace ecs {
00019
00020 class AnimationRenderingSystem : public ASystem {
00021     public:
00022         AnimationRenderingSystem();
00023         ~AnimationRenderingSystem() override = default;
00024
00025     protected:
00026         void update(std::shared_ptr<ResourceManager> resourceManager,
00027                     std::shared_ptr<Registry> registry, float deltaTime) override;
00028 };
00029
00030 } // namespace ecs
00031
00032 #endif /* !ANIMATIONRENDERINGSYSTEM_HPP_ */

```

## 5.54 GameZoneRenderingSystem.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025

```

```

00003 ** ryanR-type
00004 ** File description:
00005 ** GameZoneRenderingSystem
00006 */
00007
00008 #ifndef GAMEZONERENDERINGSYSTEM_HPP_
00009 #define GAMEZONERENDERINGSYSTEM_HPP_
00010
00011 #include "../common/systems/base/ASystem.hpp"
00012 #include "../common/components/permanent/GameZoneComponent.hpp"
00013 #include <memory>
00014
00015 namespace ecs {
00016
00017 class GameZoneRenderingSystem : public ASystem {
00018     public:
00019         GameZoneRenderingSystem();
00020         ~GameZoneRenderingSystem() override = default;
00021
00022     protected:
00023         void update(std::shared_ptr<ResourceManager> resourceManager,
00024                     std::shared_ptr<Registry> registry, float deltaTime) override;
00025 };
00026
00027 } // namespace ecs
00028
00029 #endif /* !GAMEZONERENDERINGSYSTEM_HPP_ */

```

## 5.55 GameZoneViewSystem.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** GameZoneViewSystem
00006 */
00007
00008 #ifndef GAMEZONEVIEWSYSTEM_HPP_
00009 #define GAMEZONEVIEWSYSTEM_HPP_
00010
00011 #include "../common/systems/base/ASystem.hpp"
00012 #include "../common/components/permanent/GameZoneComponent.hpp"
00013 #include <memory>
00014
00015 namespace ecs {
00016
00017 class GameZoneViewSystem : public ASystem {
00018     public:
00019         GameZoneViewSystem();
00020         ~GameZoneViewSystem() override = default;
00021
00022     protected:
00023         void update(std::shared_ptr<ResourceManager> resourceManager,
00024                     std::shared_ptr<Registry> registry, float deltaTime) override;
00025 };
00026
00027 } // namespace ecs
00028
00029 #endif /* !GAMEZONEVIEWSYSTEM_HPP_ */

```

## 5.56 HealthBarRenderingSystem.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** HealthBarRenderingSystem
00006 */
00007
00008 #ifndef HEALTHBARRENDERINGSYSTEM_HPP_
00009 #define HEALTHBARRENDERINGSYSTEM_HPP_
00010
00011 #include "../common/systems/base/ASystem.hpp"
00012 #include <memory>
00013
00014 namespace ecs {
00015
00016 class HealthBarRenderingSystem : public ASystem {

```

```

00017     public:
00018         HealthBarRenderingSystem();
00019         ~HealthBarRenderingSystem() override = default;
00020
00021     protected:
00022         void update(std::shared_ptr<ResourceManager> resourceManager,
00023             std::shared_ptr<Registry> registry, float deltaTime) override;
00024 };
00025
00026 } // namespace ecs
00027
00028 #endif /* !HEALTHBARRENDERINGSYSTEM_HPP_ */

```

## 5.57 HitboxRenderingSystem.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** HitboxRenderingSystem
00006 */
00007
00008 #ifndef HITBOXRENDERINGSYSTEM_HPP_
00009 #define HITBOXRENDERINGSYSTEM_HPP_
00010
00011 #include "../common/systems/base/ASystem.hpp"
00012 #include <memory>
00013
00014 namespace ecs {
00015
00016     class HitboxRenderingSystem : public ASystem {
00017     public:
00018         HitboxRenderingSystem();
00019         ~HitboxRenderingSystem() override = default;
00020
00021     protected:
00022         void update(std::shared_ptr<ResourceManager> resourceManager,
00023             std::shared_ptr<Registry> registry, float deltaTime) override;
00024 };
00025
00026 } // namespace ecs
00027
00028 #endif /* !HITBOXRENDERINGSYSTEM_HPP_ */

```

## 5.58 ParallaxRenderingSystem.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ParallaxRenderingSystem
00006 */
00007
00008 #ifndef PARALLAXRENDERINGSYSTEM_HPP_
00009 #define PARALLAXRENDERINGSYSTEM_HPP_
00010
00011 #include "../common/systems/base/ASystem.hpp"
00012 #include "../components/rendering/ParallaxComponent.hpp"
00013 #include "../common/types/Vector2f.hpp"
00014 #include <memory>
00015
00016 namespace ecs {
00017
00018     class ParallaxRenderingSystem : public ASystem {
00019     public:
00020         ParallaxRenderingSystem();
00021         ~ParallaxRenderingSystem() override = default;
00022
00023     protected:
00024         void update(std::shared_ptr<ResourceManager> resourceManager,
00025             std::shared_ptr<Registry> registry, float deltaTime) override;
00026
00027     private:
00028         math::Vector2f calculateScale(const ParallaxLayer& layer,
00029             float screenWidth, float screenHeight);
00030
00031         void renderLayer(const ParallaxLayer& layer,
00032             std::shared_ptr<ResourceManager> resourceManager,

```

```

00033         const math::Vector2f& basePosition,
00034         float screenWidth, float screenHeight);
00035     };
00036
00037 } // namespace ecs
00038
00039 #endif /* !PARALLAXRENDERINGSYSTEM_HPP_ */

```

## 5.59 RectangleRenderingSystem.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** RectangleRenderingSystem
00006 */
00007
00008 #ifndef RECTANGLERENDERINGSYSTEM_HPP_
00009 #define RECTANGLERENDERINGSYSTEM_HPP_
00010
00011 #include "../common/systems/base/ASystem.hpp"
00012 #include <memory>
00013
00014 namespace ecs {
00015
00016 class RectangleRenderingSystem : public ASystem {
00017     public:
00018         RectangleRenderingSystem();
00019         ~RectangleRenderingSystem() override = default;
00020
00021     protected:
00022         void update(std::shared_ptr<ResourceManager> resourceManager,
00023                     std::shared_ptr<Registry> registry, float deltaTime) override;
00024 };
00025
00026 } // namespace ecs
00027
00028 #endif /* !RECTANGLERENDERINGSYSTEM_HPP_ */

```

## 5.60 SpriteRenderingSystem.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** SpriteRenderingSystem
00006 */
00007
00008 #ifndef SPRITERENDERINGSYSTEM_HPP_
00009 #define SPRITERENDERINGSYSTEM_HPP_
00010
00011 #include "../common/systems/base/ASystem.hpp"
00012 #include <memory>
00013
00014 namespace ecs {
00015
00016 class SpriteRenderingSystem : public ASystem {
00017     public:
00018         SpriteRenderingSystem();
00019         ~SpriteRenderingSystem() override = default;
00020
00021     protected:
00022         void update(std::shared_ptr<ResourceManager> resourceManager,
00023                     std::shared_ptr<Registry> registry, float deltaTime) override;
00024 };
00025
00026 } // namespace ecs
00027
00028 #endif /* !SPRITERENDERINGSYSTEM_HPP_ */

```

## 5.61 TextRenderingSystem.hpp

```

00001 /*

```



```

00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** TextRenderingSystem
00006 */
00007
00008 #ifndef TEXTRENDERINGSYSTEM_HPP_
00009 #define TEXTRENDERINGSYSTEM_HPP_
00010
00011 #include "../common/systems/base/ASystem.hpp"
00012 #include <memory>
00013
00014 namespace ecs {
00015
00016 class TextRenderingSystem : public ASystem {
00017     public:
00018         TextRenderingSystem() ;
00019         ~TextRenderingSystem() override = default;
00020
00021     protected:
00022         void update(std::shared_ptr<ResourceManager> resourceManager,
00023             std::shared_ptr<Registry> registry, float deltaTime) override;
00024 };
00025
00026 } // namespace ecs
00027
00028 #endif /* !TEXTRENDERINGSYSTEM_HPP_ */

```

## 5.62 AFocusableElement.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** AFocusableElement
00006 */
00007
00008 #ifndef AFOCUSABLEELEMENT_HPP_
00009 #define AFOCUSABLEELEMENT_HPP_
00010
00011 #include "../elements/base/UIElement.hpp"
00012 #include "IFocusable.hpp"
00013 #include <functional>
00014
00015 namespace ui {
00016
00017 class AFocusableElement : public UIElement, public IFocusable {
00018     public:
00019         explicit AFocusableElement(std::shared_ptr<ResourceManager> resourceManager);
00020         virtual ~AFocusableElement() = default;
00021
00022         virtual void setFocused(bool focused) override;
00023         virtual bool isFocused() const override;
00024         virtual bool canBeFocused() const override;
00025         virtual void onFocusGained() override;
00026         virtual void onFocusLost() override;
00027         virtual void onActivated() override;
00028
00029         void setOnFocusGained(std::function<void()> callback);
00030         void setOnFocusLost(std::function<void()> callback);
00031         void setOnActivated(std::function<void()> callback);
00032
00033         virtual void handleInput(const math::Vector2f& mousePos, bool mousePressed) override;
00034
00035     protected:
00036         bool _focused = false;
00037         bool _pressedInside = false;
00038         bool _wasPressed = false;
00039         std::function<void()> _onFocusGained;
00040         std::function<void()> _onFocusLost;
00041         std::function<void()> _onActivated;
00042
00043         virtual void onFocusStateChanged(bool focused);
00044 };
00045
00046 } // namespace ui
00047
00048 #endif /* !AFOCUSABLEELEMENT_HPP_ */

```

## 5.63 IFocusable.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** IFocusable
00006 */
00007
00008 #ifndef IFOCUSABLE_HPP_
00009 #define IFOCUSABLE_HPP_
00010
00011 #include <memory>
00012
00013 namespace ui {
00014
00015 class IFocusable {
00016     public:
00017         virtual ~IFocusable() = default;
00018
00019         virtual void setFocused(bool focused) = 0;
00020         virtual bool isFocused() const = 0;
00021         virtual bool canBeFocused() const = 0;
00022
00023         virtual void onFocusGained() = 0;
00024         virtual void onFocusLost() = 0;
00025         virtual void onActivated() = 0;
00026
00027         virtual bool onNavigateLeft() { return false; }
00028         virtual bool onNavigateRight() { return false; }
00029 };
00030
00031 } // namespace ui
00032
00033 #endif /* !IFOCUSABLE_HPP_ */

```

## 5.64 UILayout.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** UILayout
00006 */
00007
00008 #ifndef UILAYOUT_HPP_
00009 #define UILAYOUT_HPP_
00010
00011 #include <memory>
00012 #include <vector>
00013 #include "../elements/base/UIElement.hpp"
00014 #include "../common/types/Vector2f.hpp"
00015
00016 namespace ui {
00017
00018 enum class LayoutDirection {
00019     Horizontal,
00020     Vertical
00021 };
00022
00023 enum class LayoutAlignment {
00024     Start,
00025     Center,
00026     End
00027 };
00028
00029 enum class AnchorX {
00030     None,
00031     Left,
00032     Center,
00033     Right
00034 };
00035
00036 enum class AnchorY {
00037     None,
00038     Top,
00039     Center,
00040     Bottom
00041 };
00042
00043 struct LayoutConfig {
00044     LayoutDirection direction = LayoutDirection::Vertical;

```

```

00045     LayoutAlignment alignment = LayoutAlignment::Start;
00046     float spacing = 0.0f;
00047     math::Vector2f padding = math::Vector2f(0.0f, 0.0f);
00048     math::Vector2f offset = math::Vector2f(0.0f, 0.0f);
00049     bool autoResize = false;
00050     AnchorX anchorX = AnchorX::None;
00051     AnchorY anchorY = AnchorY::None;
00052 };
00053
00054 class UILayout : public UIElement {
00055     public:
00056         UILayout(std::shared_ptr<ResourceManager> resourceManager, const LayoutConfig& config =
LayoutConfig());
00057         ~UILayout() override = default;
00058
00059         void addElement(std::shared_ptr<UIElement> element);
00060         void removeElement(std::shared_ptr<UIElement> element);
00061         void clearElements();
00062
00063         void setDirection(LayoutDirection direction);
00064         void setAlignment(LayoutAlignment alignment);
00065         void setSpacing(float spacing);
00066         void setPadding(const math::Vector2f& padding);
00067         void setOffset(const math::Vector2f& offset);
00068         void setAutoResize(bool autoResize);
00069         void setAnchor(AnchorX anchorX, AnchorY anchorY);
00070
00071         LayoutDirection getDirection() const;
00072         LayoutAlignment getAlignment() const;
00073         float getSpacing() const;
00074         math::Vector2f getPadding() const;
00075         bool isAutoResize() const;
00076
00077         void updateLayout();
00078
00079         void setScale(UIScale scale) override;
00080
00081         void render() override;
00082         void update(float deltaTime) override;
00083
00084         float getScaledSpacing() const;
00085         void applyAnchor(); private:
00086         LayoutConfig _config;
00087         std::vector<std::shared_ptr<UIElement>> _layoutElements;
00088
00089         void calculatePositions();
00090         float getTotalSize() const;
00091         math::Vector2f calculateElementPosition(size_t index, float totalSize) const;
00092 };
00093
00094 } // namespace ui
00095
00096 #endif /* !UILAYOUT_HPP */

```

## 5.65 Background.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** Background
00006 */
00007
00008 #ifndef BACKGROUND_HPP_
00009 #define BACKGROUND_HPP_
00010
00011 #include "base/UIElement.hpp"
00012 #include "../common/constants.hpp"
00013 #include <string>
00014 #include <vector>
00015
00016 namespace ui {
00017
00018 class Background : public UIElement {
00019     public:
00020         Background(std::shared_ptr<ResourceManager> resourceManager);
00021         ~Background() override = default;
00022
00023         void render() override;
00024         void update(float deltaTime) override;
00025
00026         void addLayer(const std::string& texturePath, float speedX, float speedY = 0.0f,

```

```

00027         const math::Vector2f& sourceSize = math::Vector2f(constants::MAX_WIDTH,
00028         constants::MAX_HEIGHT));
00028
00029 private:
00030     struct Layer {
00031         std::string texturePath;
00032         float speedX;
00033         float speedY;
00034         math::Vector2f sourceSize;
00035         float offsetX = 0.0f;
00036         float offsetY = 0.0f;
00037     };
00038
00039     float calculateScale(const Layer& layer, float screenWidth);
00040
00041     std::vector<Layer> _layers;
00042 };
00043
00044 } // namespace ui
00045
00046 #endif // BACKGROUND_HPP_

```

## 5.66 UIElement.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** UIElement
00006 */
00007
00008 #ifndef UIELEMENT_HPP_
00009 #define UIELEMENT_HPP_
00010
00011 #include <memory>
00012 #include <vector>
00013 #include <functional>
00014 #include "../common/types/Vector2f.hpp"
00015 #include "../common/resourceManager/ResourceManager.hpp"
00016
00017 namespace ui {
00018
00019 enum class UIState {
00020     Normal,
00021     Hovered,
00022     Pressed,
00023     Disabled,
00024     Focused
00025 };
00026
00027 enum class UIScale {
00028     Small,
00029     Normal,
00030     Large
00031 };
00032
00033 class UIElement : public std::enable_shared_from_this<UIElement> {
00034 public:
00035     UIElement(std::shared_ptr<ResourceManager> resourceManager);
00036     virtual ~UIElement() = default;
00037
00038     void setPosition(const math::Vector2f& position);
00039     void setSize(const math::Vector2f& size);
00040     math::Vector2f getPosition() const;
00041     math::Vector2f getSize() const;
00042
00043     math::Vector2f getAbsolutePosition() const;
00044     math::Vector2f getAbsoluteSize() const;
00045
00046     void setVisible(bool visible);
00047     bool isVisible() const;
00048
00049     void setState(UIState state);
00050     UIState getState() const;
00051
00052     virtual void setScale(UIScale scale);
00053     UIScale getScale() const;
00054
00055     void setParent(std::weak_ptr<UIElement> parent);
00056     std::shared_ptr<UIElement> getParent() const;
00057     void addChild(std::shared_ptr<UIElement> child);
00058     void removeChild(std::shared_ptr<UIElement> child);
00059     const std::vector<std::shared_ptr<UIElement>>& getChildren() const;

```

```

00060
00061     virtual void handleInput(const math::Vector2f& mousePos, bool mousePressed);
00062     virtual bool containsPoint(const math::Vector2f& point) const;
00063
00064     void setOnClick(std::function<void()> callback);
00065     void setOnHover(std::function<void()> callback);
00066     void setOnRelease(std::function<void()> callback);
00067
00068     virtual void render();
00069
00070     virtual void update(float deltaTime);
00071
00072     protected:
00073         std::weak_ptr<ResourceManager> _resourceManager;
00074         math::Vector2f _position;
00075         math::Vector2f _size;
00076         bool _visible = true;
00077         UIState _state = UIState::Normal;
00078         UIScale _scale = UIScale::Normal;
00079         std::weak_ptr<UIElement> _parent;
00080         std::vector<std::shared_ptr<UIElement>> _children;
00081
00082         bool _pressedInside = false;
00083         bool _wasPressed = false;
00084
00085         std::function<void()> _onClick;
00086         std::function<void()> _onHover;
00087         std::function<void()> _onRelease;
00088
00089         std::pair<int, int> getWindowSize() const;
00090         std::pair<int, int> getLogicalSize() const;
00091
00092         float getScaleFactor() const;
00093 };
00094
00095 } // namespace ui
00096
00097 #endif /* !UIELEMENT_HPP_ */

```

## 5.67 Button.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** Button
00006 */
00007
00008 #ifndef BUTTON_HPP_
00009 #define BUTTON_HPP_
00010
00011 #include "../core/AFocusableElement.hpp"
00012 #include <string>
00013 #include "../libs/Multimedia/IWindow.hpp"
00014 #include "../constants.hpp"
00015 #include "../colors.hpp"
00016
00017 namespace ui {
00018
00019     class Button : public AFocusableElement {
00020     public:
00021         explicit Button(std::shared_ptr<ResourceManager> resourceManager);
00022         virtual ~Button() = default;
00023
00024         void setText(const std::string& text);
00025         const std::string& getText() const;
00026         void setTextColor(const gfx::color_t& color);
00027         void setFontPath(const std::string& fontPath);
00028
00029         void setNormalColor(const gfx::color_t& color);
00030         void setHoveredColor(const gfx::color_t& color);
00031         void setPressedColor(const gfx::color_t& color);
00032         void setDisabledColor(const gfx::color_t& color);
00033         void setFocusedColor(const gfx::color_t& color);
00034         void setBaseFontSize(size_t fontSize);
00035         size_t getBaseFontSize() const;
00036
00037         virtual void render() override;
00038
00039     private:
00040         std::string _text;
00041         gfx::color_t _textColor = colors::UI_TEXT;
00042         std::string _fontPath = "assets/fonts/arial.ttf";

```

```

00043
00044     gfx::color_t _normalColor = colors::BUTTON_PRIMARY;
00045     gfx::color_t _hoveredColor = colors::BUTTON_PRIMARY_HOVER;
00046     gfx::color_t _pressedColor = colors::BUTTON_PRIMARY_PRESSED;
00047     gfx::color_t _disabledColor = colors::UI_DISABLED;
00048     gfx::color_t _focusedColor = colors::UI_FOCUSED;
00049     size_t _baseFontSize = constants::BUTTON_FONT_SIZE_BASE;
00050
00051     gfx::color_t getCurrentColor() const;
00052     size_t getFontSize() const;
00053 };
00054
00055 } // namespace ui
00056
00057 #endif /* !BUTTON_HPP_ */

```

## 5.68 Slider.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** Slider
00006 */
00007
00008 #ifndef SLIDER_HPP_
00009 #define SLIDER_HPP_
00010
00011 #include "../core/AFocusableElement.hpp"
00012 #include <string>
00013 #include <functional>
00014 #include "../libs/Multimedia/IWindow.hpp"
00015 #include "../constants.hpp"
00016 #include "../colors.hpp"
00017
00018 namespace ui {
00019
00020 class Slider : public AFocusableElement {
00021     public:
00022         explicit Slider(std::shared_ptr<ResourceManager> resourceManager);
00023         virtual ~Slider() = default;
00024
00025         void setMinValue(float minValue);
00026         void setMaxValue(float maxValue);
00027         void setValue(float value);
00028         float getValue() const;
00029         float getMinValue() const;
00030         float getMaxValue() const;
00031
00032         void setStep(float step);
00033         float getStep() const;
00034
00035         void setLabel(const std::string& label);
00036         const std::string& getLabel() const;
00037         void setLabelColor(const gfx::color_t& color);
00038         void setFontPath(const std::string& fontPath);
00039         void setBaseFontSize(size_t fontSize);
00040         size_t getBaseFontSize() const;
00041         void setShowPercentage(bool show);
00042
00043         void setTrackColor(const gfx::color_t& color);
00044         void setFillColor(const gfx::color_t& color);
00045         void setHandleColor(const gfx::color_t& color);
00046         void setHandleHoveredColor(const gfx::color_t& color);
00047         void setHandleFocusedColor(const gfx::color_t& color);
00048
00049         void setOnValueChanged(std::function<void(float)> callback);
00050
00051         virtual void render() override;
00052         virtual void handleInput(const math::Vector2f& mousePos, bool mousePressed) override;
00053         virtual void onActivated() override;
00054         virtual bool onNavigateLeft() override;
00055         virtual bool onNavigateRight() override;
00056
00057         void incrementValue();
00058         void decrementValue();
00059
00060     private:
00061         float _minValue = 0.0f;
00062         float _maxValue = 1.0f;
00063         float _value = 0.5f;
00064         float _step = 0.1f;
00065         float _visualNormalizedValue = 0.5f;

```

```

00066
00067     std::string _label;
00068     gfx::color_t _labelColor = colors::SLIDER_LABEL;
00069     std::string _fontPath = "assets/fonts/arial.ttf";
00070     size_t _baseFontSize = constants::BUTTON_FONT_SIZE_BASE;
00071     float _outlineThickness = 2.0f;
00072     bool _showPercentage = true;
00073
00074     gfx::color_t _trackColor = colors::SLIDER_TRACK;
00075     gfx::color_t _fillColor = colors::SLIDER_FILL;
00076     gfx::color_t _handleColor = colors::SLIDER_HANDLE;
00077     gfx::color_t _handleHoveredColor = colors::SLIDER_HANDLE_HOVER;
00078     gfx::color_t _handleFocusedColor = colors::SLIDER_HANDLE_FOCUSED;
00079
00080     std::function<void(float)> _onValueChanged;
00081
00082     bool _isDragging = false;
00083     bool _wasMousePressed = false;
00084
00085     float getNormalizedValue() const;
00086     void setNormalizedValue(float normalized);
00087     gfx::color_t getCurrentHandleColor() const;
00088     size_t getFontSize() const;
00089     float getHandleRadius() const;
00090     float getTrackHeight() const;
00091     float getLabelHeight() const;
00092 };
00093
00094 } // namespace ui
00095
00096 #endif /* !SLIDER_HPP_ */

```

## 5.69 TextInput.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** TextInput
00006  */
00007
00008 #ifndef TEXTINPUT_HPP_
00009 #define TEXTINPUT_HPP_
00010
00011 #include "../core/AFocusableElement.hpp"
00012 #include <string>
00013 #include <functional>
00014 #include "../libs/Multimedia/EventTypes.hpp"
00015 #include "../libs/Multimedia/IWindow.hpp"
00016 #include "../libs/colors.hpp"
00017
00018 namespace ui {
00019
00020 class TextInput : public AFocusableElement {
00021     public:
00022         TextInput(std::shared_ptr<ResourceManager> resourceManager);
00023         ~TextInput();
00024
00025         virtual void render() override;
00026
00027         void setText(const std::string& text);
00028         const std::string& getText() const;
00029         void setPlaceholder(const std::string& placeholder);
00030         const std::string& getPlaceholder() const;
00031         void setTextColor(const gfx::color_t& color);
00032         void setPlaceholderColor(const gfx::color_t& color);
00033         void setFontPath(const std::string& fontPath);
00034         void setBaseFontSize(size_t fontSize);
00035         size_t getBaseFontSize() const;
00036
00037         void setOnTextChanged(std::function<void(const std::string&)> callback);
00038         void setOnSubmit(std::function<void(const std::string&)> callback);
00039
00040         virtual void handleInput(const math::Vector2f& mousePos, bool mousePressed) override;
00041         void handleKeyboardInput(gfx::EventType event);
00042         void handleTextInput(const std::string& text);
00043
00044         virtual void update(float deltaTime) override;
00045
00046     private:
00047         std::string _text;
00048         std::string _placeholder;
00049         size_t _cursorPosition = 0;

```

```

00050         float _cursorBlinkTimer = 0.0f;
00051         bool _showCursor = true;
00052
00053         gfx::color_t _textColor = {0, 0, 0};
00054         gfx::color_t _placeholderColor = {128, 128, 128};
00055         std::string _fontPath = "assets/fonts/arial.ttf";
00056         size_t _baseFontSize = 24;
00057
00058         std::function<void(const std::string&)> _onTextChanged;
00059         std::function<void(const std::string&)> _onSubmit;
00060
00061         void insertChar(char c);
00062         void deleteChar();
00063         void moveCursorLeft();
00064         void moveCursorRight();
00065         size_t getFontSize() const;
00066         void updateCursorBlink(float deltaTime);
00067
00068         gfx::color_t _normalColor = colors::WHITE;
00069         gfx::color_t _hoveredColor = colors::LIGHT_GRAY;
00070         gfx::color_t _pressedColor = colors::DARK_GRAY;
00071         gfx::color_t _disabledColor = colors::UI_DISABLED;
00072         gfx::color_t _focusedColor = colors::UI_FOCUSED;
00073         gfx::color_t _getCurrentColor() const;
00074     };
00075
00076 }
00077
00078 #endif /* !TEXTINPUT_HPP_ */

```

## 5.70 ToggleSwitch.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ToggleSwitch
00006 */
00007
00008 #ifndef TOGGLESWITCH_HPP_
00009 #define TOGGLESWITCH_HPP_
00010
00011 #include "../core/AFocusableElement.hpp"
00012 #include <string>
00013 #include <functional>
00014 #include "../libs/Multimedia/IWindow.hpp"
00015 #include "../constants.hpp"
00016 #include "../colors.hpp"
00017
00018 namespace ui {
00019
00020 class ToggleSwitch : public AFocusableElement {
00021     public:
00022         explicit ToggleSwitch(std::shared_ptr<ResourceManager> resourceManager);
00023         virtual ~ToggleSwitch() = default;
00024
00025         void setValue(bool value);
00026         bool getValue() const;
00027
00028         void setFontPath(const std::string& fontPath);
00029         void setBaseFontSize(size_t fontSize);
00030         size_t getBaseFontSize() const;
00031
00032         void setOnText(const std::string& text);
00033         void setOffText(const std::string& text);
00034
00035         void setTrackColor(const gfx::color_t& color);
00036         void setHandleColor(const gfx::color_t& color);
00037         void setHandleHoveredColor(const gfx::color_t& color);
00038         void setHandleFocusedColor(const gfx::color_t& color);
00039         void setOnColor(const gfx::color_t& color);
00040         void setOffColor(const gfx::color_t& color);
00041
00042         void setOnValueChanged(std::function<void(bool)> callback);
00043
00044         virtual void render() override;
00045         virtual void handleInput(const math::Vector2f& mousePos, bool mousePressed) override;
00046         virtual bool containsPoint(const math::Vector2f& point) const override;
00047
00048     private:
00049         bool _value = false;
00050         std::string _fontPath = "assets/fonts/arial.ttf";
00051         size_t _baseFontSize = constants::BUTTON_FONT_SIZE_BASE;

```



```

00052         std::string _onText = "ON";
00053         std::string _offText = "OFF";
00054
00055         gfx::color_t _trackColor = colors::TOGGLE_TRACK;
00056         gfx::color_t _handleColor = colors::TOGGLE_HANDLE;
00057         gfx::color_t _handleHoveredColor = colors::TOGGLE_HANDLE_HOVER;
00058         gfx::color_t _handleFocusedColor = colors::TOGGLE_HANDLE_FOCUSED;
00059         gfx::color_t _onColor = colors::TOGGLE_ON;
00060         gfx::color_t _offColor = colors::TOGGLE_OFF;
00061
00062         std::function<void(bool)> _onValueChanged;
00063
00064         bool _isHovered = false;
00065     };
00066
00067 } // namespace ui
00068
00069 #endif // TOGGLESWITCH_HPP_

```

## 5.71 Text.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** Text
00006 */
00007
00008 #ifndef TEXT_HPP_
00009 #define TEXT_HPP_
00010
00011 #include "base/UIElement.hpp"
00012 #include "../common/types/Vector2f.hpp"
00013 #include "../libs/Multimedia/IWindow.hpp"
00014 #include "resourceManager/ResourceManager.hpp"
00015 #include <memory>
00016 #include <string>
00017 #include "../colors.hpp"
00018
00019 namespace ui {
00020
00021 class Text : public UIElement {
00022 public:
00023     Text(std::shared_ptr<ResourceManager> resourceManager);
00024     ~Text() override = default;
00025
00026     void render() override;
00027     void update(float deltaTime) override;
00028     void setScale(UIScale scale) override;
00029
00030     void setText(const std::string& text);
00031     std::string getText() const;
00032
00033     void setTextColor(const gfx::color_t& color);
00034     void setFontSize(unsigned int size);
00035     void setFontPath(const std::string& path);
00036     void setOutlineColor(const gfx::color_t& color);
00037     void setOutlineThickness(float thickness);
00038
00039 private:
00040     std::string _text;
00041     gfx::color_t _textColor;
00042     unsigned int _fontSize;
00043     unsigned int _baseFontSize;
00044     std::string _fontPath;
00045     gfx::color_t _outlineColor;
00046     float _outlineThickness;
00047 };
00048
00049 } // namespace ui
00050
00051 #endif // TEXT_HPP_

```

## 5.72 UIManager.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type

```

```

00004 ** File description:
00005 ** UIManager
00006 */
00007
00008 #ifndef UIMANAGER_HPP_
00009 #define UIMANAGER_HPP_
00010
00011 #include <memory>
00012 #include <vector>
00013 #include "../elements/base/UIElement.hpp"
00014 #include "../navigation/UINavigationManager.hpp"
00015 #include "../../common/InputMapping/InputAction.hpp"
00016 #include "../../common/InputMapping/IInputProvider.hpp"
00017 #include "../../common/types/Vector2f.hpp"
00018 #include "../../client/constants.hpp"
00019
00020 namespace ui {
00021
00022 class UIManager {
00023     public:
00024         UIManager();
00025         ~UIManager() = default;
00026
00027         void addElement(std::shared_ptr<UIElement> element);
00028         void removeElement(std::shared_ptr<UIElement> element);
00029         void clearElements();
00030
00031         void update(float deltaTime);
00032
00033         void render();
00034
00035         void handleMouseInput(const math::Vector2f& mousePos, bool mousePressed);
00036         bool handleNavigationInput(ecs::InputAction action);
00037         bool handleNavigationInputs(std::shared_ptr<ecs::IInputProvider> inputProvider, float
deltaTime);
00038         void handleKeyboardInput(gfx::EventType event);
00039         void handleTextInput(const std::string& text);
00040
00041         std::shared_ptr<UINavigationManager> getNavigationManager();
00042
00043         void setNavigationEnabled(bool enabled);
00044         bool isNavigationEnabled() const;
00045
00046         bool focusFirstElement();
00047         void clearFocus();
00048
00049         std::shared_ptr<IFocusable> getFocusedElement() const;
00050
00051         void setGlobalScale(UIScale scale);
00052         void cycleGlobalScale();
00053         UIScale getGlobalScale() const;
00054
00055         void setOnBack(std::function<void()> callback);
00056
00057         bool isMouseHoveringAnyElement(const math::Vector2f& mousePos) const;
00058
00059     private:
00060         std::vector<std::shared_ptr<UIElement>> _elements;
00061         std::shared_ptr<UINavigationManager> _navigationManager;
00062         math::Vector2f _lastMousePos;
00063         bool _mouseMovementDetected;
00064
00065         float _navigationCooldown = 0.0f;
00066         UIScale _globalScale = UIScale::Normal;
00067         std::function<void()> _onBack;
00068
00069         bool hasMouseMoved(const math::Vector2f& mousePos);
00070
00071         void refreshNavigationElements();
00072 };
00073
00074 } // namespace ui
00075
00076 #endif /* !UIMANAGER_HPP_ */

```

## 5.73 UINavigationManager.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** UINavigationManager
00006 */

```

```

00007
00008 #ifndef UINAVIGATIONMANAGER_HPP_
00009 #define UINAVIGATIONMANAGER_HPP_
00010
00011 #include <vector>
00012 #include <memory>
00013 #include <functional>
00014 #include "../core/IFocusable.hpp"
00015 #include "../../common/InputMapping/InputAction.hpp"
00016 #include "../../common/types/Vector2f.hpp"
00017
00018 namespace ui {
00019
00020 enum class NavigationDirection {
00021     Up,
00022     Down,
00023     Left,
00024     Right
00025 };
00026
00027 class UINavigationManager {
00028     public:
00029         UINavigationManager();
00030         ~UINavigationManager() = default;
00031
00032         void addFocusableElement(std::shared_ptr<IFocusable> element);
00033         void removeFocusableElement(std::shared_ptr<IFocusable> element);
00034         void clearFocusableElements();
00035
00036         bool handleNavigationInput(ecs::InputAction action);
00037
00038         bool setFocus(std::shared_ptr<IFocusable> element);
00039         std::shared_ptr<IFocusable> getFocusedElement() const;
00040         void clearFocus();
00041
00042         bool focusFirstElement();
00043         bool focusNextElement();
00044         bool focusPreviousElement();
00045
00046         void setNavigationEnabled(bool enabled);
00047         bool isNavigationEnabled() const;
00048
00049         void setOnFocusChanged(std::function<void(std::shared_ptr<IFocusable>)> callback);
00050
00051         void onMouseMovement();
00052
00053         void enableFocus();
00054
00055         bool isFocusDisabled() const;
00056
00057     private:
00058         std::vector<std::weak_ptr<IFocusable>> _focusableElements;
00059         std::weak_ptr<IFocusable> _currentFocused;
00060         bool _navigationEnabled;
00061         bool _focusDisabled;
00062         std::function<void(std::shared_ptr<IFocusable>)> _onFocusChanged;
00063
00064         void cleanupExpiredElements();
00065         int getCurrentFocusedIndex() const;
00066         bool navigateInDirection(NavigationDirection direction);
00067 };
00068
00069 } // namespace ui
00070
00071 #endif /* !UINAVIGATIONMANAGER_HPP_ */

```

## 5.74 Utils.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** Utils
00006 */
00007 #include "ClientNetwork.hpp"
00008
00009 #ifndef UTILS_HPP_
00010 #define UTILS_HPP_
00011
00012 class Utils {
00013     public:
00014         Utils();
00015         ~Utils();

```

```

00016
00017         void helper();
00018         void parseCli(int ac, char **av, std::shared_ptr<ClientNetwork> clientNetwork);
00019
00020     protected:
00021     private:
00022 };
00023
00024 #endif /* !UTILS_HPP_ */

```

## 5.75 Utils.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** Utils
00006 */
00007
00008 #include <vector>
00009 #include <memory>
00010
00011 #include "ServerConfig.hpp"
00012
00013 #ifndef UTILS_HPP_
00014 #define UTILS_HPP_
00015
00016 class Utils {
00017     public:
00018         Utils();
00019         ~Utils();
00020
00021         void helper();
00022         void parsCli(int ac, char **av, std::shared_ptr<rserv::ServerConfig> config);
00023     protected:
00024     private:
00025 };
00026
00027 #endif /* !UTILS_HPP_ */

```

## 5.76 CollisionRules.hpp

```

00001 #ifndef COLLISIONRULES_HPP_
00002 #define COLLISIONRULES_HPP_
00003
00004 #include <vector>
00005 #include <string>
00006 #include "CollisionRulesData.hpp"
00007 #include "../components/permanent/ColliderComponent.hpp"
00008
00009 namespace ecs {
00010
00011 class CollisionRules {
00012     public:
00013         static const CollisionRules& getInstance();
00014
00015         static void initWithData(const CollisionRulesData& data);
00016
00017         bool canCollide(
00018             CollisionType type,
00019             const std::vector<std::string>& tagsA,
00020             const std::vector<std::string>& tagsB
00021         ) const;
00022
00023     private:
00024         CollisionRules();
00025         ~CollisionRules() = default;
00026         CollisionRules(const CollisionRules&) = delete;
00027         CollisionRules& operator=(const CollisionRules&) = delete;
00028
00029         const std::vector<CollisionRule>& getAllowRules(CollisionType type) const;
00030
00031         std::shared_ptr<std::vector<CollisionRule>> _solidAllowRules;
00032         std::shared_ptr<std::vector<CollisionRule>> _triggerAllowRules;
00033         std::shared_ptr<std::vector<CollisionRule>> _pushAllowRules;
00034
00035         bool entityMatchesGroup(
00036             const std::vector<std::string>& entityTags,

```

```

00037         const std::vector<std::string>& group
00038     ) const;
00039     bool ruleMatches(
00040         const CollisionRule& rule,
00041         const std::vector<std::string>& tagsA,
00042         const std::vector<std::string>& tagsB
00043     ) const;
00044 };
00045
00046 } // namespace ecs
00047
00048 #endif // COLLISIONRULES_HPP_

```

## 5.77 CollisionRulesData.hpp

```

00001 #ifndef COLLISION_RULES_DATA_HPP_
00002 #define COLLISION_RULES_DATA_HPP_
00003
00004 #include <vector>
00005 #include <string>
00006 #include <memory>
00007
00008 namespace ecs {
00009
00010 struct CollisionRule {
00011     std::vector<std::string> groupA;
00012     std::vector<std::string> groupB;
00013 };
00014
00015 struct CollisionRulesData {
00016     std::shared_ptr<std::vector<CollisionRule>> solidAllowRules;
00017     std::shared_ptr<std::vector<CollisionRule>> triggerAllowRules;
00018     std::shared_ptr<std::vector<CollisionRule>> pushAllowRules;
00019 };
00020
00021 } // namespace ecs
00022
00023 #endif // COLLISION_RULES_DATA_HPP_

```

## 5.78 AComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** AComponent
00006 */
00007
00008 #ifndef ACOMPONENT_HPP_
00009 #define ACOMPONENT_HPP_
00010
00011 #include "IComponent.hpp"
00012
00013 namespace ecs {
00014
00015 class AComponent : public IComponent {
00016     public:
00017         AComponent();
00018         ~AComponent();
00019
00020     protected:
00021     private:
00022 };
00023
00024 } // namespace ecs
00025
00026 #endif /* !ACOMPONENT_HPP_ */

```

## 5.79 IComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type

```

```

00004 ** File description:
00005 ** IComponent
00006 */
00007
00008 #ifndef ICOMPONENT_HPP_
00009 #define ICOMPONENT_HPP_
00010
00011 namespace ecs {
00012
00013 class IComponent {
00014     public:
00015         IComponent() = default;
00016         virtual ~IComponent() = default;
00017
00018     protected:
00019     private:
00020 };
00021
00022 } // namespace ecs
00023
00024 #endif /* !ICOMPONENT_HPP_ */

```

## 5.80 AIMovementPatternComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** AIMovementPatternComponent
00006 */
00007
00008 #ifndef AIMOVEMENTPATTERNCOMPONENT_HPP
00009 #define AIMOVEMENTPATTERNCOMPONENT_HPP
00010
00011 #include "../base/AComponent.hpp"
00012 #include "../constants.hpp"
00013
00014 namespace ecs {
00015
00016 enum class AIMovementPattern {
00017     STRAIGHT_LINE = 0,
00018     ZIGZAG = 1,
00019     VERTICAL_MIRROR = 2,
00020     FOLLOW_RIGHT = 3
00021 };
00022
00023 class AIMovementPatternComponent : public AComponent {
00024     public:
00025         AIMovementPatternComponent(
00026             AIMovementPattern p = AIMovementPattern::STRAIGHT_LINE
00027         ) : pattern(p) {}
00028         ~AIMovementPatternComponent() = default;
00029
00030         AIMovementPattern getPattern() const { return pattern; }
00031         float getZigzagAmplitude() const { return zigzagAmplitude; }
00032         float getZigzagFrequency() const { return zigzagFrequency; }
00033         float getDetectionRange() const { return detectionRange; }
00034         float getVerticalDeadzone() const { return verticalDeadzone; }
00035         float getTimer() const { return timer; }
00036
00037         void setPattern(AIMovementPattern p) { pattern = p; }
00038         void setZigzagAmplitude(float amplitude) { zigzagAmplitude = amplitude; }
00039         void setZigzagFrequency(float frequency) { zigzagFrequency = frequency; }
00040         void setDetectionRange(float range) { detectionRange = range; }
00041         void setVerticalDeadzone(float deadzone) { verticalDeadzone = deadzone; }
00042         void setTimer(float t) { timer = t; }
00043
00044     public:
00045         AIMovementPattern pattern;
00046         float zigzagAmplitude = constants::DEFAULT_ZIGZAG_AMPLITUDE;
00047         float zigzagFrequency = constants::DEFAULT_ZIGZAG_FREQUENCY;
00048         float detectionRange = constants::DEFAULT_DETECTION_RANGE;
00049         float verticalDeadzone = constants::DEFAULT_VERTICAL_DEADZONE;
00050         float timer = constants::DEFAULT_TIMER;
00051 };
00052
00053 } // namespace ecs
00054
00055 #endif /* !AIMOVEMENTPATTERNCOMPONENT_HPP */

```

## 5.81 ColliderComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ColliderComponent
00006 */
00007
00008 #ifndef COLLIDERCOMPONENT_HPP_
00009 #define COLLIDERCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include "../types/FRect.hpp"
00013 #include "../types/Vector2f.hpp"
00014
00015 namespace ecs {
00016
00017 enum class CollisionType {
00018     None = 0,
00019     Solid = 1,
00020     Trigger = 2,
00021     Push = 3
00022 };
00023
00024 class ColliderComponent : public AComponent {
00025 public:
00026     ColliderComponent(math::Vector2f offset = math::Vector2f(0.0f, 0.0f), math::Vector2f size =
00027         math::Vector2f(0.0f, 0.0f), CollisionType type = CollisionType::Solid)
00028         : _offset(offset), _size(size), _type(type) {};
00029     ~ColliderComponent() = default;
00030
00031     math::Vector2f getOffset() const { return _offset; };
00032     void setOffset(math::Vector2f offset) { _offset = offset; };
00033
00034     math::Vector2f getSize() const { return _size; };
00035     void setSize(math::Vector2f size) { _size = size; };
00036
00037     CollisionType getType() const { return _type; };
00038     void setType(CollisionType type) { _type = type; };
00039
00040     math::FRect getHitbox(math::Vector2f entityPosition, math::Vector2f scale =
00041         math::Vector2f(1.0f, 1.0f)) const {
00042         return math::FRect(entityPosition.getX() + _offset.getX(), entityPosition.getY() +
00043             _offset.getY(), _size.getX() * scale.getX(), _size.getY() * scale.getY());
00044     };
00045     math::FRect getScaledHitbox(math::Vector2f entityPosition, math::Vector2f scale) const {
00046         return getHitbox(entityPosition, scale);
00047     };
00048 private:
00049     math::Vector2f _offset;
00050     math::Vector2f _size;
00051     CollisionType _type;
00052 };
00053 } // namespace ecs
00054
00055 #endif /* !COLLIDERCOMPONENT_HPP_ */

```

## 5.82 DamageComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** DamageComponent
00006 */
00007
00008 #ifndef DAMAGECOMPONENT_HPP_
00009 #define DAMAGECOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012
00013 namespace ecs {
00014
00015 class DamageComponent : public AComponent {
00016 public:
00017     DamageComponent(float damage = 0.0f) : _damage(damage) {};
00018     ~DamageComponent() override = default;
00019

```

```

00020         float getDamage() const { return _damage; }
00021         void setDamage(float damage) { _damage = damage; }
00022
00023     private:
00024         float _damage;
00025 };
00026
00027 } // namespace ecs
00028
00029 #endif /* !DAMAGECOMPONENT_HPP_ */

```

## 5.83 GameZoneComponent.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** GameZoneComponent
00006  */
00007
00008 #ifndef GAMEZONECOMPONENT_HPP_
00009 #define GAMEZONECOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include "../types/FRect.hpp"
00013 #include "../constants.hpp"
00014
00015 namespace ecs {
00016
00017     class GameZoneComponent : public AComponent {
00018     public:
00019         GameZoneComponent(math::FRect zone = math::FRect(0.0f, 0.0f, constants::MAX_WIDTH,
00020             constants::MAX_HEIGHT))
00021             : _zone(zone) {};
00022         ~GameZoneComponent() = default;
00023
00024         math::FRect getZone() const { return _zone; };
00025         void setZone(math::FRect zone) { _zone = zone; };
00026
00027     private:
00028         math::FRect _zone;
00029 };
00030 } // namespace ecs
00031
00032 #endif /* !GAMEZONECOMPONENT_HPP_ */

```

## 5.84 HealthComponent.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** HealthComponent
00006  */
00007
00008 #ifndef HEALTHCOMPONENT_HPP_
00009 #define HEALTHCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include "../ECS/entity/Entity.hpp"
00013
00014 namespace ecs {
00015
00016     class HealthComponent : public AComponent {
00017     public:
00018         HealthComponent(float health = 100) : _health(health), _baseHealth(health),
00019             _lastDamageSource(0) {};
00020         ~HealthComponent() override = default;
00021
00022         float getHealth() const { return _health; };
00023         void setHealth(float health) { _health = health; };
00024
00025         void decreaseHealth(float quantity) { _health -= quantity; };
00026
00027         float getBaseHealth() const { return _baseHealth; };
00028         void setBaseHealth(float health) { _baseHealth = health; };
00029
00030     private:
00031         int _lastDamageSource;
00032     };
00033 }

```



```

00029         ecs::Entity getLastDamageSource() const { return _lastDamageSource; }
00030         void setLastDamageSource(ecs::Entity source) { _lastDamageSource = source; }
00031
00032     private:
00033         float _health;
00034         float _baseHealth;
00035         ecs::Entity _lastDamageSource;
00036 };
00037
00038 }
00039
00040 #endif /* !HEALTHCOMPONENT_HPP_ */

```

## 5.85 InteractionConfigComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** InteractionConfigComponent
00006 */
00007
00008 #ifndef INTERACTIONCONFIGCOMPONENT_HPP_
00009 #define INTERACTIONCONFIGCOMPONENT_HPP_
00010
00011 #include <vector>
00012 #include <string>
00013 #include "../base/AComponent.hpp"
00014
00015 namespace ecs {
00016
00017     struct InteractionMapping {
00018         std::vector<std::string> targetTags;
00019         std::vector<std::string> actionsToOther;
00020         std::vector<std::string> actionsToSelf;
00021     };
00022
00023     class InteractionConfigComponent : public AComponent {
00024     public:
00025         InteractionConfigComponent() {
00026             _mappings = std::vector<InteractionMapping>();
00027         };
00028         InteractionConfigComponent(const std::vector<InteractionMapping>& mappings)
00029             : _mappings(mappings) {}
00030         ~InteractionConfigComponent() = default;
00031
00032         const std::vector<InteractionMapping>& getMappings() const { return _mappings; }
00033         void setMappings(const std::vector<InteractionMapping>& mappings) { _mappings = mappings; }
00034         void addMapping(const InteractionMapping& mapping) { _mappings.push_back(mapping); }
00035
00036     private:
00037         std::vector<InteractionMapping> _mappings;
00038     };
00039
00040 } // namespace ecs
00041
00042 #endif /* !INTERACTIONCONFIGCOMPONENT_HPP_ */

```

## 5.86 LifetimeComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** LifetimeComponent
00006 */
00007
00008 #ifndef LIFETIMECOMPONENT_HPP_
00009 #define LIFETIMECOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012
00013 namespace ecs {
00014
00015     class LifetimeComponent : public AComponent {
00016     public:
00017         LifetimeComponent(float lifetime = 0.0f) : _lifetime(lifetime) {};
00018         ~LifetimeComponent() = default;

```

```

00019
00020         float getLifetime() const { return _lifetime; };
00021         void setLifetime(float lifetime) { _lifetime = lifetime; };
00022
00023     private:
00024         float _lifetime;
00025 };
00026
00027 } // ecs
00028
00029
00030 #endif /* !LIFETIMECOMPONENT_HPP_ */

```

## 5.87 NetworkIdComponent.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** NetworkIdComponent
00006  */
00007
00008 #ifndef NETWORKIDCOMPONENT_HPP_
00009 #define NETWORKIDCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include <cstdint>
00013
00014 namespace ecs {
00015
00016     class NetworkIdComponent : public AComponent {
00017     public:
00018         explicit NetworkIdComponent(size_t networkId = 0)
00019             : _networkId(networkId) {}
00020
00021         ~NetworkIdComponent() = default;
00022
00023         size_t getNetworkId() const { return _networkId; }
00024         void setNetworkId(size_t networkId) { _networkId = networkId; }
00025
00026     private:
00027         size_t _networkId;
00028     };
00029
00030 } // namespace ecs
00031
00032 #endif /* !NETWORKIDCOMPONENT_HPP_ */

```

## 5.88 OwnerComponent.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** OwnerComponent
00006  */
00007
00008 #ifndef OWNERCOMPONENT_HPP_
00009 #define OWNERCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012
00013 namespace ecs {
00014
00015     class OwnerComponent : public AComponent {
00016     public:
00017         OwnerComponent(ecs::Entity owner = 0) : _owner(owner) {};
00018         ~OwnerComponent() override = default;
00019
00020         ecs::Entity getOwner() const { return _owner; }
00021         void setOwner(ecs::Entity owner) { _owner = owner; }
00022
00023     private:
00024         ecs::Entity _owner;
00025     };
00026
00027 } // namespace ecs
00028
00029 #endif /* !OWNERCOMPONENT_HPP_ */

```

## 5.89 ProjectilePrefabComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ProjectilePrefabComponent
00006 */
00007
00008 #ifndef PROJECTILEPREFABCOMPONENT_HPP_
00009 #define PROJECTILEPREFABCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include <string>
00013
00014 namespace ecs {
00015
00016 class ProjectilePrefabComponent : public AComponent {
00017     public:
00018         ProjectilePrefabComponent(const std::string &prefabName = "")
00019             : _prefabName(prefabName) {};
00020         ~ProjectilePrefabComponent() = default;
00021
00022         std::string getPrefabName() const { return _prefabName; };
00023         void setPrefabName(const std::string &prefabName) { _prefabName = prefabName; };
00024     private:
00025         std::string _prefabName;
00026 };
00027
00028 }
00029
00030 #endif /* !PROJECTILEPREFABCOMPONENT_HPP_ */

```

## 5.90 ScoreComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ScoreComponent
00006 */
00007
00008 #ifndef SCORECOMPONENT_HPP_
00009 #define SCORECOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012
00013 namespace ecs {
00014
00015 class ScoreComponent : public AComponent {
00016     public:
00017         ScoreComponent(int score = 0) : _score(score) {};
00018         ~ScoreComponent() {};
00019
00020         int getScore() const { return _score; }
00021         void setScore(int score) { _score = score; }
00022         void addScore(int amount) { _score += amount; }
00023
00024     private:
00025         int _score;
00026 };
00027
00028 } // namespace ecs
00029
00030 #endif /* !SCORECOMPONENT_HPP_ */

```

## 5.91 ScoreValueComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ScoreValueComponent
00006 */
00007
00008 #ifndef SCOREVALUECOMPONENT_HPP_
00009 #define SCOREVALUECOMPONENT_HPP_

```

```

00010
00011 #include "../base/AComponent.hpp"
00012
00013 namespace ecs {
00014
00015 class ScoreValueComponent : public AComponent {
00016     public:
00017         ScoreValueComponent(int scoreValue = 0) : _scoreValue(scoreValue) {};
00018         ~ScoreValueComponent() override = default;
00019
00020         int getScoreValue() const { return _scoreValue; }
00021         void setScoreValue(int scoreValue) { _scoreValue = scoreValue; }
00022
00023     private:
00024         int _scoreValue;
00025 };
00026
00027 } // namespace ecs
00028
00029 #endif /* !SCOREVALUECOMPONENT_HPP_ */

```

## 5.92 ShootingStatsComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ShootingStatsComponent
00006 */
00007
00008 #ifndef SHOOTINGSTATSCOMPONENT_HPP_
00009 #define SHOOTINGSTATSCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include "../Prefab/IPrefab.hpp"
00013 #include <memory>
00014 #include <vector>
00015
00016 namespace ecs {
00017
00018 struct MultiShotPattern {
00019     int shotCount = 1;
00020     float angleSpread = 0.0f;
00021     float offsetDistance = 0.0f;
00022
00023     MultiShotPattern() = default;
00024     MultiShotPattern(int count, float spread, float offset)
00025         : shotCount(count), angleSpread(spread), offsetDistance(offset) {}
00026 };
00027
00028 class ShootingStatsComponent : public AComponent {
00029     public:
00030         ShootingStatsComponent(
00031             float fireRate = 1.0f,
00032             const MultiShotPattern &pattern = MultiShotPattern()
00033         ) : _fireRate(fireRate),
00034             _multiShotPattern(pattern),
00035             _cooldownTimer(0.0f) {};
00036         ~ShootingStatsComponent() = default;
00037
00038         float getFireRate() const { return _fireRate; };
00039         void setFireRate(float fireRate) { _fireRate = fireRate; };
00040
00041         MultiShotPattern getMultiShotPattern() const { return _multiShotPattern; };
00042         void setMultiShotPattern(const MultiShotPattern &pattern) { _multiShotPattern = pattern; };
00043
00044         float getCooldownTimer() const { return _cooldownTimer; };
00045         void setCooldownTimer(float timer) { _cooldownTimer = timer; };
00046
00047         bool canShoot() const { return _cooldownTimer <= 0.0f; };
00048
00049         void updateCooldown(float deltaTime) {
00050             if (_cooldownTimer > 0.0f) {
00051                 _cooldownTimer -= deltaTime;
00052             }
00053         };
00054
00055         void resetCooldown() {
00056             _cooldownTimer = 1.0f / _fireRate;
00057         };
00058
00059     private:
00060         float _fireRate;

```

```

00061         MultiShotPattern _multiShotPattern;
00062         float _cooldownTimer;
00063     };
00064
00065 } // namespace ecs
00066
00067 #endif /* !SHOOTINGSTATSCOMPONENT_HPP_ */

```

## 5.93 SpeedComponent.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** SpeedComponent
00006  */
00007
00008 #ifndef SPEEDCOMPONENT_HPP_
00009 #define SPEEDCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include "constants.hpp"
00013
00014 namespace ecs {
00015
00016     class SpeedComponent : public AComponent {
00017     public:
00018         SpeedComponent(float speed = constants::BASE_SPEED) : _speed(speed) {};
00019         ~SpeedComponent() = default;
00020
00021         float getSpeed() const { return _speed; };
00022         void setSpeed(float speed) { _speed = speed; };
00023     private:
00024         float _speed;
00025     };
00026
00027 } // namespace ecs
00028
00029 #endif /* !SPEEDCOMPONENT_HPP_ */

```

## 5.94 TransformComponent.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** TransformComponent
00006  */
00007
00008 #ifndef TRANSFORMCOMPONENT_HPP_
00009 #define TRANSFORMCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include "../types/Vector2f.hpp"
00013
00014 namespace ecs {
00015
00016     class TransformComponent : public AComponent {
00017     public:
00018         TransformComponent(math::Vector2f position = math::Vector2f(0.0f, 0.0f), float rotation =
00019             0.0f, math::Vector2f scale = math::Vector2f(1.0f, 1.0f))
00019             : _position(position), _rotation(rotation), _scale(scale) {};
00020         ~TransformComponent() = default;
00021
00022         math::Vector2f getPosition() const { return _position; };
00023         void setPosition(math::Vector2f position) { _position = position; };
00024
00025         float getRotation() const { return _rotation; };
00026         void setRotation(float rotation) { _rotation = rotation; };
00027
00028         math::Vector2f getScale() const { return _scale; };
00029         void setScale(math::Vector2f scale) { _scale = scale; };
00030
00031     private:
00032         math::Vector2f _position;
00033         float _rotation;
00034         math::Vector2f _scale;
00035     };

```

```

00036
00037 } // namespace ecs
00038
00039 #endif /* !TRANSFORMCOMPONENT_HPP_ */

```

## 5.95 VelocityComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** VelocityComponent
00006 */
00007
00008 #ifndef VELOCITYCOMPONENT_HPP_
00009 #define VELOCITYCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include "../types/Vector2f.hpp"
00013
00014 namespace ecs {
00015
00016 class VelocityComponent : public AComponent {
00017     public:
00018         VelocityComponent(math::Vector2f velocity = math::Vector2f(0.0f, 0.0f)) : _velocity(velocity)
00019         {};
00020         ~VelocityComponent() = default;
00021
00022         math::Vector2f getVelocity() const { return _velocity; };
00023         void setVelocity(math::Vector2f velocity) { _velocity = velocity; };
00024     private:
00025         math::Vector2f _velocity;
00026 };
00027 } // namespace ecs
00028
00029 #endif /* !VELOCITYCOMPONENT_HPP_ */

```

## 5.96 AIMoverTag.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** AIMoverTag
00006 */
00007
00008 #ifndef AIMOVERTAG_HPP
00009 #define AIMOVERTAG_HPP
00010
00011 #include "../base/AComponent.hpp"
00012
00013 namespace ecs {
00014
00015 struct AIMoverTag : public AComponent {
00016     AIMoverTag() = default;
00017 };
00018
00019 } // namespace ecs
00020
00021 #endif /* !AIMOVERTAG_HPP */

```

## 5.97 AIShooterTag.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** AIShooterTag
00006 */
00007
00008 #ifndef AISHOOTERTAG_HPP
00009 #define AISHOOTERTAG_HPP

```

```

00010
00011 #include "../base/AComponent.hpp"
00012
00013 namespace ecs {
00014
00015 struct AIShooterTag : public AComponent {
00016     AIShooterTag() = default;
00017 };
00018
00019 } // namespace ecs
00020
00021 #endif /* !AISHOOTERTAG_HPP */

```

## 5.98 ControllableTag.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ControllableTag
00006 */
00007
00008 #ifndef CONTROLLABLETAG_HPP_
00009 #define CONTROLLABLETAG_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012
00013 namespace ecs {
00014
00015 class ControllableTag : public AComponent {
00016     public:
00017         ControllableTag() = default;
00018         ~ControllableTag() = default;
00019 };
00020
00021 } // namespace ecs
00022
00023 #endif /* !CONTROLLABLETAG_HPP_ */

```

## 5.99 EnnemyProjectileTag.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** EnnemyProjectileTag
00006 */
00007
00008 #ifndef ENNEMYPROJECTILETAG_HPP_
00009 #define ENNEMYPROJECTILETAG_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012
00013 namespace ecs {
00014
00015 class EnnemyProjectileTag : public AComponent {
00016     public:
00017         EnnemyProjectileTag() = default;
00018         ~EnnemyProjectileTag() = default;
00019 };
00020
00021 } // namespace ecs
00022
00023 #endif /* !ENNEMYPROJECTILETAG_HPP_ */

```

## 5.100 GameZoneColliderTag.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** GameZoneColliderTag
00006 */

```

```

00007
00008 #ifndef GAMEZONECOLLIDERTAG_HPP_
00009 #define GAMEZONECOLLIDERTAG_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012
00013 namespace ecs {
00014
00015 class GameZoneColliderTag : public AComponent {
00016     public:
00017         GameZoneColliderTag() = default;
00018         ~GameZoneColliderTag() = default;
00019 };
00020
00021 } // namespace ecs
00022
00023 #endif /* !GAMEZONECOLLIDERTAG_HPP_ */

```

## 5.101 LocalPlayerTag.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** LocalPlayerTag
00006 */
00007
00008 #ifndef LOCALPLAYERTAG_HPP_
00009 #define LOCALPLAYERTAG_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012
00013 namespace ecs {
00014
00015 class LocalPlayerTag : public AComponent {
00016     public:
00017         LocalPlayerTag() = default;
00018         ~LocalPlayerTag() = default;
00019 };
00020
00021 } // namespace ecs
00022
00023 #endif /* !LOCALPLAYERTAG_HPP_ */

```

## 5.102 MobTag.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** MobTag
00006 */
00007
00008 #ifndef MOBTAG_HPP_
00009 #define MOBTAG_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012
00013 namespace ecs {
00014
00015 class MobTag : public AComponent {
00016     public:
00017         MobTag() = default;
00018         ~MobTag() = default;
00019 };
00020
00021 } // namespace ecs
00022
00023 #endif /* !MOBTAG_HPP_ */

```

## 5.103 ObstacleTag.hpp

```

00001 /*

```



```

00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** ObstacleTag
00006  */
00007
00008  #ifndef OBSTACLETAG_HPP_
00009  #define OBSTACLETAG_HPP_
00010
00011  #include "../base/AComponent.hpp"
00012
00013  namespace ecs {
00014
00015  class ObstacleTag : public AComponent {
00016      public:
00017          ObstacleTag() = default;
00018          ~ObstacleTag() = default;
00019  };
00020
00021  } // namespace ecs
00022
00023  #endif /* !OBSTACLETAG_HPP_ */

```

## 5.104 PlayerProjectileTag.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** PlayerProjectileTag
00006  */
00007
00008  #ifndef PLAYERPROJECTILETAG_HPP_
00009  #define PLAYERPROJECTILETAG_HPP_
00010
00011  #include "../base/AComponent.hpp"
00012
00013  namespace ecs {
00014
00015  class PlayerProjectileTag : public AComponent {
00016      public:
00017          PlayerProjectileTag() = default;
00018          ~PlayerProjectileTag() = default;
00019  };
00020
00021  } // namespace ecs
00022
00023  #endif /* !PLAYERPROJECTILETAG_HPP_ */

```

## 5.105 PlayerTag.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** PlayerTag
00006  */
00007
00008  #ifndef PLAYERTAG_HPP_
00009  #define PLAYERTAG_HPP_
00010
00011  #include "../base/AComponent.hpp"
00012
00013  namespace ecs {
00014
00015  class PlayerTag : public AComponent {
00016      public:
00017          PlayerTag() = default;
00018          ~PlayerTag() = default;
00019  };
00020
00021  } // namespace ecs
00022
00023  #endif /* !PLAYERTAG_HPP_ */

```

## 5.106 ProjectilePassThroughTag.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ProjectilePassThroughTag
00006 */
00007
00008 #ifndef PROJECTILEPASSTHROUGHTAG_HPP_
00009 #define PROJECTILEPASSTHROUGHTAG_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012
00013 namespace ecs {
00014
00015 class ProjectilePassThroughTag : public AComponent {
00016     public:
00017         ProjectilePassThroughTag() = default;
00018         ~ProjectilePassThroughTag() = default;
00019 };
00020
00021 } // namespace ecs
00022
00023 #endif /* !PROJECTILEPASSTHROUGHTAG_HPP_ */

```

## 5.107 ShooterTag.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ShooterTag
00006 */
00007
00008 #ifndef SHOOTERTAG_HPP_
00009 #define SHOOTERTAG_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012
00013 namespace ecs {
00014
00015 class ShooterTag : public AComponent {
00016     public:
00017         ShooterTag() = default;
00018         ~ShooterTag() = default;
00019 };
00020
00021 }
00022
00023 #endif /* !SHOOTERTAG_HPP_ */

```

## 5.108 DamageIntentComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** DamageIntentComponent
00006 */
00007
00008 #ifndef DAMAGEINTENTCOMPONENT_HPP_
00009 #define DAMAGEINTENTCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include "../ECS/entity/Entity.hpp"
00013
00014 namespace ecs {
00015
00016 class DamageIntentComponent : public AComponent {
00017     public:
00018         DamageIntentComponent(float damages = 0.0f, ecs::Entity source = 0) : _damages(damages),
00019         _source(source) {};
00019         ~DamageIntentComponent() = default;
00020
00021         float getDamages() { return _damages; };
00022         void setDamages(float damages) { _damages = damages; };

```

```

00023
00024     ecs::Entity getSource() const { return _source; };
00025     void setSource(ecs::Entity source) { _source = source; };
00026
00027     private:
00028         float _damages;
00029         ecs::Entity _source;
00030 };
00031
00032 } // namespace ecs
00033
00034 #endif /* !DAMAGEINTENTCOMPONENT_HPP_ */

```

## 5.109 DeathIntentComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** DeathIntentComponent
00006 */
00007
00008 #ifndef DEATHINTENTCOMPONENT_HPP_
00009 #define DEATHINTENTCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include "../../ECS/entity/Entity.hpp"
00013
00014 namespace ecs {
00015
00016 class DeathIntentComponent : public AComponent {
00017     public:
00018         DeathIntentComponent(ecs::Entity source = 0) : _source(source) {};
00019         ~DeathIntentComponent() = default;
00020
00021         ecs::Entity getSource() const { return _source; }
00022         void setSource(ecs::Entity source) { _source = source; }
00023
00024     private:
00025         ecs::Entity _source;
00026 };
00027
00028 }
00029
00030 #endif /* !DEATHINTENTCOMPONENT_HPP_ */

```

## 5.110 InputIntentComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** InputIntentComponent
00006 */
00007
00008 #ifndef INPUTINTENTCOMPONENT_HPP_
00009 #define INPUTINTENTCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include "../../types/Vector2f.hpp"
00013
00014 namespace ecs {
00015
00016 class InputIntentComponent : public AComponent {
00017     public:
00018         InputIntentComponent(const math::Vector2f &direction = math::Vector2f(0.0f, 0.0f))
00019             : _direction(direction) {
00020         };
00021         ~InputIntentComponent() {
00022             _direction = math::Vector2f(0.0f, 0.0f);
00023         };
00024
00025         math::Vector2f getDirection() const { return _direction; };
00026         void setDirection(const math::Vector2f &direction) { _direction = direction; };
00027
00028     private:
00029         math::Vector2f _direction;
00030 };

```

```

00031
00032 } // namespace ecs
00033
00034 #endif /* !INPUTINTENTCOMPONENT_HPP_ */

```

## 5.111 MovementIntentComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** MovementIntentComponent
00006 */
00007
00008 #ifndef MOVEMENTINTENTCOMPONENT_HPP_
00009 #define MOVEMENTINTENTCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include "../../types/Vector2f.hpp"
00013
00014 namespace ecs {
00015
00016 class MovementIntentComponent : public AComponent {
00017     public:
00018         MovementIntentComponent(const math::Vector2f &direction = math::Vector2f(0.0f, 0.0f), bool
active = false)
00019             : _direction(direction), _active(active) {
00020         };
00021         ~MovementIntentComponent() = default;
00022
00023         math::Vector2f getDirection() const { return _direction; };
00024         void setDirection(const math::Vector2f &direction) { _direction = direction; };
00025
00026         bool isActive() const { return _active; };
00027         void setActive(bool active) { _active = active; };
00028
00029     private:
00030         math::Vector2f _direction;
00031         bool _active;
00032 };
00033
00034 } // namespace ecs
00035
00036 #endif /* !MOVEMENTINTENTCOMPONENT_HPP_ */

```

## 5.112 ScoreIntentComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ScoreIntentComponent
00006 */
00007
00008 #ifndef SCOREINTENTCOMPONENT_HPP_
00009 #define SCOREINTENTCOMPONENT_HPP_
00010
00011 class ScoreIntentComponent {
00012     public:
00013         ScoreIntentComponent(int score = 0) : _score(score) {};
00014         ~ScoreIntentComponent() {};
00015
00016         int getScore() const { return _score; };
00017         void setScore(int newScore) { _score = newScore; };
00018     protected:
00019     private:
00020         int _score;
00021 };
00022
00023 #endif /* !SCOREINTENTCOMPONENT_HPP_ */

```

## 5.113 ShootIntentComponent.hpp

```

00001 /*

```

```

00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ShootIntentComponent
00006 */
00007
00008 #ifndef SHOOTINTENTCOMPONENT_HPP_
00009 #define SHOOTINTENTCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include "../types/Vector2f.hpp"
00013
00014 namespace ecs {
00015
00016 class ShootIntentComponent : public AComponent {
00017     public:
00018         ShootIntentComponent(float angle = 0.0f) : _angle(angle) {
00019             _position = math::Vector2f(0.0f, 0.0f);
00020         }
00021         ~ShootIntentComponent() = default;
00022
00023         void setAngle(float angle) { _angle = angle; }
00024         float getAngle() const { return _angle; }
00025     private:
00026         float _angle;
00027         math::Vector2f _position;
00028 };
00029
00030 } // namespace ecs
00031
00032 #endif /* !SHOOTINTENTCOMPONENT_HPP_ */

```

## 5.114 SpawnIntentComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** SpawnIntentComponent
00006 */
00007
00008 #ifndef SPAWNINTENTCOMPONENT_HPP_
00009 #define SPAWNINTENTCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include <string>
00013 #include <optional>
00014 #include "../types/Vector2f.hpp"
00015 #include "../ECS/entity/EntityCreationContext.hpp"
00016
00017 namespace ecs {
00018
00019 class SpawnIntentComponent : public AComponent {
00020     public:
00021         SpawnIntentComponent(
00022             const std::string &prefabName,
00023             const math::Vector2f &position,
00024             float gameViewXTrigger = 0.0f
00025         ) : _prefabName(prefabName),
00026             _position(position),
00027             _creationContext(EntityCreationContext::forLocalClient()),
00028             _gameViewXTrigger(gameViewXTrigger) {}
00029
00030         SpawnIntentComponent(
00031             const std::string &prefabName,
00032             const math::Vector2f &position,
00033             const EntityCreationContext &context,
00034             float gameViewXTrigger = 0.0f
00035         ) : _prefabName(prefabName),
00036             _position(position),
00037             _creationContext(context),
00038             _gameViewXTrigger(gameViewXTrigger) {}
00039
00040         ~SpawnIntentComponent() = default;
00041
00042         void setPrefabName(const std::string &prefabName) { _prefabName = prefabName; }
00043         std::string getPrefabName() const { return _prefabName; }
00044
00045         void setPosition(const math::Vector2f &position) { _position = position; }
00046         math::Vector2f getPosition() const { return _position; }
00047
00048         void setCreationContext(const EntityCreationContext &context) {
00049             _creationContext = context;
00050         }
00051     private:
00052         std::string _prefabName;
00053         math::Vector2f _position;
00054         EntityCreationContext _creationContext;
00055         float _gameViewXTrigger;
00056 };
00057
00058 } // namespace ecs
00059
00060 #endif /* !SPAWNINTENTCOMPONENT_HPP_ */

```

```

00050     }
00051     EntityCreationContext getCreationContext() const { return _creationContext; }
00052
00053     void setGameViewXTrigger(const float &gameViewXTrigger) { _gameViewXTrigger =
gameViewXTrigger; };
00054     float getGameViewXTrigger() const { return _gameViewXTrigger; };
00055
00056     private:
00057         std::string _prefabName;
00058         math::Vector2f _position;
00059         EntityCreationContext _creationContext;
00060         float _gameViewXTrigger;
00061 };
00062
00063 }
00064
00065 #endif /* !SPAWNINTENTCOMPONENT_HPP_ */

```

## 5.115 TriggerIntentComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** TriggerIntentComponent
00006 */
00007
00008 #ifndef TRIGGERINTENTCOMPONENT_HPP_
00009 #define TRIGGERINTENTCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include "../../ECS/entity/Entity.hpp"
00013
00014 namespace ecs {
00015
00016 class TriggerIntentComponent : public AComponent {
00017     public:
00018         TriggerIntentComponent(Entity self = 0, Entity other = 0) : _self(self), _other(other) {};
00019         ~TriggerIntentComponent() override = default;
00020
00021         Entity getSelf() const { return _self; }
00022         void setSelf(Entity self) { _self = self; }
00023
00024         Entity getOther() const { return _other; }
00025         void setOther(Entity other) { _other = other; }
00026
00027     private:
00028         Entity _self;
00029         Entity _other;
00030 };
00031
00032 } // namespace ecs
00033
00034 #endif /* !TRIGGERINTENTCOMPONENT_HPP_ */

```

## 5.116 debug.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** debug
00006 */
00007
00008 #ifndef DEBUG_HPP_
00009 #define DEBUG_HPP_
00010
00011 #ifdef _WIN32
00012     #ifndef WIN32_LEAN_AND_MEAN
00013         #define WIN32_LEAN_AND_MEAN
00014     #endif
00015     #include <windows.h>
00016     #ifndef ERROR
00017         #undef ERROR
00018     #endif
00019     #ifndef INFO
00020         #undef INFO
00021     #endif

```

```

00022     #ifdef WARNING
00023     #undef WARNING
00024     #endif
00025 #endif
00026
00027
00028 #include <string>
00029
00030 namespace debug {
00031
00032     enum debugType {
00033         NETWORK = 0,
00034         ECS = 1,
00035         CORE = 2
00036     };
00037
00038     enum debugLevel {
00039         INFO = 0,
00040         WARNING = 1,
00041         ERROR = 2
00042     };
00043
00044     class Debug {
00045     public:
00046         ~Debug() = default;
00047         static void printDebug(const bool isDebug, const std::string &message, debugType type,
                                debugLevel level);
00048     };
00049
00050 } // namespace debug
00051
00052 #endif /* !DEBUG_HPP_ */

```

## 5.117 DLLoader.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** r-type
00004 ** File description:
00005 ** DLLoader
00006 */
00007
00008 #ifndef DLLOADER_HPP_
00009 #define DLLOADER_HPP_
00010
00011 #ifdef _WIN32
00012     #include <windows.h>
00013     #define RTLD_LAZY 0
00014 #else
00015     #include <dlfcn.h>
00016 #endif
00017
00018 #include <iostream>
00019 #include <ostream>
00020 #include <memory>
00021 #include "ILoader.hpp"
00022
00023 template <typename T>
00024
00025 class DLLoader : public ILoader {
00026     private:
00027     #ifdef _WIN32
00028         HMODULE _handler = nullptr;
00029         mutable std::string _lastError;
00030     #else
00031         void *_handler = nullptr;
00032     #endif
00033
00034     public:
00035         ~DLLoader() override {
00036             if (_handler != nullptr) {
00037                 Close();
00038             }
00039         }
00040
00041         void *getHandler() const override {
00042             return _handler;
00043         };
00044
00045         void *Open(const char *path, int flag = RTLD_LAZY) override {
00046     #ifdef _WIN32
00047             (void)flag;
00048             _handler = LoadLibraryA(path);

```

```

00049         if (!_handler) {
00050             _lastError = "Failed to load library: " + std::string(path);
00051         }
00052     #else
00053         _handler = dlopen(path, flag);
00054     #endif
00055     return _handler;
00056 };
00057
00058 void *Symbol(const char *symbolName) override {
00059     #ifdef _WIN32
00060         void *symbol = (void*)GetProcAddress(_handler, symbolName);
00061         if (!symbol) {
00062             _lastError = "Failed to get symbol: " + std::string(symbolName);
00063             std::cerr << "GetProcAddress error: " << _lastError << std::endl;
00064             return nullptr;
00065         }
00066         return symbol;
00067     #else
00068         void *symbol = dlsym(_handler, symbolName);
00069         const char *error = dlerror();
00070         if (error) {
00071             std::cerr << "dlerror: " << error << std::endl;
00072             return nullptr;
00073         }
00074         return symbol;
00075     #endif
00076 };
00077
00078 T getSymbol(const char *symbolName) {
00079     #ifdef _WIN32
00080         return reinterpret_cast<T>(GetProcAddress(_handler, symbolName));
00081     #else
00082         return reinterpret_cast<T>(dlsym(_handler, symbolName));
00083     #endif
00084 };
00085
00086 int Close() override{
00087     if (_handler == nullptr)
00088         return -1;
00089     #ifdef _WIN32
00090     int result = FreeLibrary(_handler) ? 0 : -1;
00091     #else
00092     int result = dlclose(_handler);
00093     #endif
00094     _handler = nullptr;
00095     return result;
00096 };
00097
00098 const char *Error() override {
00099     #ifdef _WIN32
00100         return _lastError.c_str();
00101     #else
00102         return dlerror();
00103     #endif
00104 };
00105 };
00106
00107 #endif /* !DLLOADER_HPP_ */

```

## 5.118 ILoader.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** r-type
00004 ** File description:
00005 ** ILoader
00006 */
00007
00008 #ifndef ILoader_HPP_
00009 #define ILoader_HPP_
00010
00011
00012 class ILoader {
00013     public:
00014         virtual ~ILoader() = default;
00015
00016         virtual void *Open(const char *path, int flag) = 0;
00017         virtual void *Symbol(const char *symbolName) = 0;
00018         virtual int Close() = 0;
00019         virtual const char *Error() = 0;
00020         virtual void *getHandler() const = 0;
00021

```



```

00022     protected:
00023     private:
00024 };
00025
00026 #endif /* !Loader_HPP_ */

```

## 5.119 LoaderType.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** r-type
00004 ** File description:
00005 ** LoaderType
00006 */
00007
00008 #ifndef LOADERTYPE_HPP_
00009 #define LOADERTYPE_HPP_
00010
00011 enum ModuleType_t{
00012     MULTIMEDIA_MODULE = 0,
00013     NETWORK_SERVER_MODULE = 1,
00014     NETWORK_CLIENT_MODULE = 2,
00015     PACKET_MODULE = 3,
00016     BUFFER_MODULE = 4,
00017     UNKNOWN_MODULE
00018 };
00019
00020 typedef ModuleType_t (*getTypeFunc_t)();
00021
00022 typedef void (*createNetworkLib_t)();
00023 typedef void (*createBuffer_t)();
00024 typedef void (*createPacket_t)();
00025
00026 #define pathLoad "./libraries"
00027
00028 #ifdef _WIN32
00029     #define multimediaLib "Multimedia"
00030     #define networkServerLib "NetworkServer"
00031     #define networkClientLib "NetworkClient"
00032     #define bufferLib "Buffer"
00033     #define packetLib "Packet"
00034     #define sharedLibExt ".dll"
00035 #elif __APPLE__
00036     #define multimediaLib "libMultimedia"
00037     #define networkServerLib "libNetworkServer"
00038     #define networkClientLib "libNetworkClient"
00039     #define bufferLib "libBuffer"
00040     #define packetLib "libPacket"
00041     #define sharedLibExt ".dylib"
00042 #else
00043     #define multimediaLib "libMultimedia"
00044     #define networkServerLib "libNetworkServer"
00045     #define networkClientLib "libNetworkClient"
00046     #define bufferLib "libBuffer"
00047     #define packetLib "libPacket"
00048     #define sharedLibExt ".so"
00049 #endif
00050
00051 #endif /* !LOADERTYPE_HPP_ */

```

## 5.120 AComponentArray.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** AComposantType
00006 */
00007
00008 #ifndef AComposantType_HPP_
00009 #define AComposantType_HPP_
00010
00011 #include "IComponentArray.hpp"
00012 #include "../components/base/IComponent.hpp"
00013 #include <vector>
00014 #include <memory>
00015
00016 namespace ecs {

```

```

00017
00018 template <typename T>
00019 class AComponentArray : public IComponentArray {
00020     public:
00021         AComponentArray();
00022         ~AComponentArray() override;
00023
00024         void add(Entity entityId, std::shared_ptr<T> component);
00025         std::shared_ptr<T> get(Entity entityId) const;
00026         std::vector<std::shared_ptr<T>> getAll(Entity entityId) const;
00027         void removeComponents(Entity entityId) override;
00028         void removeOneComponent(Entity entityId) override;
00029         bool has(Entity entityId) const;
00030
00031         Entity getMaxEntityId() const override;
00032
00033     private:
00034         std::vector<std::vector<std::shared_ptr<T>>> _components;
00035 };
00036
00037 } // namespace ecs
00038
00039 #include "AComponentArray.hpp"
00040
00041 #endif /* !AComposantType_HPP_ */

```

## 5.121 IComponentArray.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** IComponentArray
00006 */
00007
00008 #ifndef ICOMPONENTARRAY_HPP_
00009 #define ICOMPONENTARRAY_HPP_
00010
00011 #include "../.../components/base/IComponent.hpp"
00012 #include "../Entity.hpp"
00013
00014 namespace ecs {
00015
00016     class IComponentArray {
00017     public:
00018         virtual ~IComponentArray() = default;
00019         virtual Entity getMaxEntityId() const = 0;
00020         virtual void removeComponents(Entity entityId) = 0;
00021         virtual void removeOneComponent(Entity entityId) = 0;
00022     };
00023
00024 } // namespace ecs
00025
00026 #endif /* !ICOMPONENTARRAY_HPP_ */

```

## 5.122 Entity.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** Entity
00006 */
00007
00008 #ifndef ENTITY_HPP_
00009 #define ENTITY_HPP_
00010
00011 #include <cstdint>
00012
00013 namespace ecs {
00014
00015     using Entity = size_t;
00016
00017 } // namespace ecs
00018
00019 #endif /* !ENTITY_HPP_ */

```

## 5.123 EntityCreationContext.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** EntityCreationContext
00006 */
00007
00008 #ifndef ENTITYCREATIONCONTEXT_HPP_
00009 #define ENTITYCREATIONCONTEXT_HPP_
00010
00011 #include <cstddef>
00012 #include <optional>
00013
00014 namespace ecs {
00015
00016 enum class EntityCreationOrigin {
00017     SERVER,
00018     CLIENT_FROM_NETWORK,
00019     CLIENT_LOCAL
00020 };
00021
00022 struct EntityCreationContext {
00023     EntityCreationOrigin origin = EntityCreationOrigin::CLIENT_LOCAL;
00024     std::optional<size_t> networkId = std::nullopt;
00025
00026     static EntityCreationContext forServer(std::optional<size_t> existingId = std::nullopt) {
00027         return {EntityCreationOrigin::SERVER, existingId};
00028     }
00029
00030     static EntityCreationContext forNetworkSync(size_t serverNetworkId) {
00031         return {EntityCreationOrigin::CLIENT_FROM_NETWORK, serverNetworkId};
00032     }
00033
00034     static EntityCreationContext forLocalClient() {
00035         return {EntityCreationOrigin::CLIENT_LOCAL, std::nullopt};
00036     }
00037
00038     bool shouldHaveNetworkId() const {
00039         return origin == EntityCreationOrigin::SERVER ||
00040             origin == EntityCreationOrigin::CLIENT_FROM_NETWORK;
00041     }
00042 };
00043
00044 } // namespace ecs
00045
00046 #endif /* !ENTITYCREATIONCONTEXT_HPP_ */

```

## 5.124 EntityFactory.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** EntityFactory
00006 */
00007
00008 #ifndef ENTITYFACTORY_HPP_
00009 #define ENTITYFACTORY_HPP_
00010
00011 #include "IEntityFactory.hpp"
00012 #include <atomic>
00013
00014 namespace ecs {
00015
00016 class EntityFactory : public IEntityFactory {
00017 public:
00018     explicit EntityFactory(size_t startingNetworkId = 1);
00019     ~EntityFactory() override;
00020
00021     Entity createEntity(
00022         const std::shared_ptr<Registry>& registry,
00023         const EntityCreationContext& context = EntityCreationContext::forLocalClient()
00024     ) override;
00025
00026     size_t getNextNetworkId() const override;
00027     void setNextNetworkId(size_t nextId) override;
00028
00029 private:
00030     size_t resolveNetworkId(const EntityCreationContext& context);
00031     std::atomic<size_t> _nextNetworkId;

```

```

00032         std::atomic<size_t> _nextLocalId;
00033     };
00034
00035 } // namespace ecs
00036
00037 #endif /* !ENTITYFACTORY_HPP_ */

```

## 5.125 IEntityFactory.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** IEntityFactory
00006 */
00007
00008 #ifndef ENTITYFACTORY_HPP_
00009 #define ENTITYFACTORY_HPP_
00010
00011 #include <memory>
00012 #include <string>
00013 #include "../Entity.hpp"
00014 #include "../EntityCreationContext.hpp"
00015 #include "../registry/Registry.hpp"
00016
00017 namespace ecs {
00018
00019 class IEntityFactory {
00020     public:
00021         virtual ~IEntityFactory() = default;
00022
00023         virtual Entity createEntity(
00024             const std::shared_ptr<Registry>& registry,
00025             const EntityCreationContext& context = EntityCreationContext::forLocalClient()
00026         ) = 0;
00027
00028         virtual size_t getNextNetworkId() const = 0;
00029         virtual void setNextNetworkId(size_t nextId) = 0;
00030 };
00031
00032 } // namespace ecs
00033
00034 #endif /* !ENTITYFACTORY_HPP_ */

```

## 5.126 Registry.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** Registry
00006 */
00007
00008 #ifndef REGISTRY_HPP_
00009 #define REGISTRY_HPP_
00010
00011 #include "../.../components/base/IComponent.hpp"
00012 #include "../componentArray/IComponentArray.hpp"
00013 #include "../componentArray/AComponentArray.hpp"
00014 #include <memory>
00015 #include <unordered_map>
00016 #include <string>
00017 #include <functional>
00018
00019 namespace ecs {
00020
00021 template <typename... Components> class View;
00022 template <typename... Components> class Group;
00023
00024 class Registry : public std::enable_shared_from_this<Registry> {
00025     public:
00026         Registry();
00027         ~Registry();
00028
00029         template <typename T>
00030         void registerComponent();
00031
00032         template <typename T>

```

```

00033     void addComponent(Entity entityId, std::shared_ptr<T> component);
00034
00035     template <typename T>
00036     std::shared_ptr<T> getComponent(Entity entityId) const;
00037
00038     template <typename T>
00039     std::vector<std::shared_ptr<T>> getComponents(Entity entityId) const;
00040
00041     template <typename T>
00042     void removeAllComponents(Entity entityId);
00043
00044     template <typename T>
00045     void removeOneComponent(Entity entityId);
00046
00047     template <typename T>
00048     bool hasComponent(Entity entityId) const;
00049
00050     template <typename... Components>
00051     View<Components...> view();
00052
00053     Entity getMaxEntityId() const;
00054
00055     Entity createEntity();
00056     void destroyEntity(Entity entityId);
00057
00058     Entity getEntityByNetworkId(size_t networkId);
00059
00060     void setOnEntityDestroyed(std::function<void(Entity)> callback);
00061 protected:
00062 private:
00063     Entity _nextEntityId;
00064     std::unordered_map<std::string, std::shared_ptr<IComponentArray>> _components;
00065     std::function<void(Entity)> _onEntityDestroyed;
00066 };
00067
00068 } // namespace ecs
00069
00070 #include "Registry.hpp"
00071
00072 #endif /* !REGISTRY_HPP_ */

```

## 5.127 View.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** View
00006  */
00007
00008 #ifndef VIEW_HPP_
00009 #define VIEW_HPP_
00010
00011 #include <vector>
00012 #include <memory>
00013 #include <type_traits>
00014
00015 namespace ecs {
00016
00017     template <typename... Components>
00018     class View {
00019     public:
00020         View(std::shared_ptr<Registry> registry);
00021
00022         class Iterator;
00023
00024         Iterator begin();
00025         Iterator end();
00026
00027         class Iterator {
00028         public:
00029             Iterator(std::shared_ptr<Registry> registry, size_t entityId, size_t maxEntityId);
00030             bool operator!=(const Iterator& other) const;
00031             Iterator& operator++();
00032             size_t operator*() const;
00033
00034         private:
00035             bool hasAllComponents() const;
00036             std::shared_ptr<Registry> _registry;
00037             size_t _entityId;
00038             size_t _maxEntityId;
00039         };
00040

```

```

00041     private:
00042         std::shared_ptr<Registry> _registry;
00043     };
00044
00045
00046 } // namespace ecs
00047
00048 #include "View.hpp"
00049
00050 #endif /* !VIEW_HPP_ */

```

## 5.128 AError.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** AError
00006 */
00007
00008 #ifndef AERROR_HPP_
00009 #define AERROR_HPP_
00010
00011 #include <string>
00012 #include "IError.hpp"
00013
00014 namespace err {
00015
00016 class AError : public IError {
00017     public:
00018         AError(const std::string &message, int code = 0);
00019
00020         virtual ~AError() noexcept override = default;
00021         const char *what() const noexcept override;
00022         int getCode() const noexcept override;
00023         std::string getDetails() const noexcept override;
00024
00025         virtual std::string getType() const noexcept override = 0;
00026
00027     protected:
00028         std::string m_message;
00029         int m_code;
00030 };
00031
00032 }
00033
00034 #endif /* !AERROR_HPP_ */

```

## 5.129 ClientError.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ClientError
00006 */
00007
00008 #ifndef CLIENTERROR_HPP_
00009 #define CLIENTERROR_HPP_
00010
00011 #include "AError.hpp"
00012
00013 namespace err {
00014
00015 class ClientError : public AError {
00016     public:
00017         enum ErrorCode {
00018             UNKNOWN = 2000,
00019             CONNECTION_FAILED = 2001,
00020             DISCONNECTED = 2002,
00021             TIMEOUT = 2003,
00022             NOT_INITIALIZED = 2004,
00023             CAN_NOT_OPEN_FILE = 2005
00024         };
00025
00026         ClientError(const std::string &message, ErrorCode code = UNKNOWN);
00027         ~ClientError() override;
00028

```

```

00029         std::string getType() const noexcept override;
00030     protected:
00031     private:
00032 };
00033
00034 } // namespace err
00035
00036 #endif /* !CLIENTERROR_HPP_ */

```

## 5.130 ClientNetworkError.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ClientNetworkError
00006 */
00007
00008 #ifndef CLIENTNETWORKERROR_HPP_
00009 #define CLIENTNETWORKERROR_HPP_
00010
00011 #include "AError.hpp"
00012
00013 namespace err {
00014
00015     class ClientNetworkError : public AError {
00016     public:
00017         enum ErrorCode {
00018             UNKNOWN = 1000,
00019             CONNECTION_FAILED = 1001,
00020             TIMEOUT = 1002,
00021             INVALID_REQUEST = 1003,
00022             INTERNAL_ERROR = 1004,
00023             LIBRARY_LOAD_FAILED = 1005,
00024             CONFIG_ERROR = 1006
00025         };
00026
00027         ClientNetworkError(const std::string &message, ErrorCode code = UNKNOWN);
00028         virtual ~ClientNetworkError() noexcept = default;
00029         std::string getType() const noexcept override;
00030
00031     private:
00032 };
00033
00034 }
00035
00036 #endif /* !CLIENTNETWORKERROR_HPP_ */

```

## 5.131 IError.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** IError
00006 */
00007
00008 #ifndef IERROR_HPP_
00009 #define IERROR_HPP_
00010
00011 #include <string>
00012 #include <exception>
00013
00014 namespace err {
00015
00016     class IError : public std::exception {
00017     public:
00018
00019         virtual ~IError() noexcept = default;
00020         virtual const char *what() const noexcept override = 0;
00021         virtual int getCode() const noexcept = 0;
00022         virtual std::string getType() const noexcept = 0;
00023         virtual std::string getDetails() const noexcept = 0;
00024
00025     protected:
00026     private:
00027 };
00028

```

```

00029 }
00030
00031 #endif /* !ERROR_HPP_ */
00032

```

## 5.132 LibrairiesLoadError.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** LibrairiesLoadError
00006 */
00007
00008 #ifndef LIBRAIRIESLOADERROR_HPP_
00009 #define LIBRAIRIESLOADERROR_HPP_
00010
00011 #include "AError.hpp"
00012
00013 namespace err {
00014
00015 class LibrairiesLoadError : public AError {
00016     public:
00017         enum ErrorCode {
00018             UNKNOWN = 1000,
00019             LIBRARY_NOT_FOUND = 1001,
00020             SYMBOL_NOT_FOUND = 1002
00021         };
00022
00023         LibrairiesLoadError(const std::string &message, ErrorCode code = UNKNOWN);
00024         ~LibrairiesLoadError() override;
00025         std::string getType() const noexcept override;
00026
00027     protected:
00028     private:
00029 };
00030
00031 }
00032
00033 #endif /* !LIBRAIRIESLOADERROR_HPP_ */

```

## 5.133 PacketError.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** R-Type
00004 ** File description:
00005 ** Header
00006 */
00007
00008 #ifndef PACKET_ERROR_HPP
00009 #define PACKET_ERROR_HPP
00010
00011 #include "AError.hpp"
00012
00013 namespace err {
00014
00015 class PacketError : public AError {
00016     public:
00017         enum ErrorCode {
00018             UNKNOWN = 1000,
00019             SERIALIZER_ATTRIBUTION_FAILED = 1001,
00020             STRING_FORMATTING_ERROR = 1002
00021         };
00022
00023         PacketError(const std::string &message, ErrorCode code = UNKNOWN);
00024         ~PacketError() override;
00025         std::string getType() const noexcept override;
00026
00027     protected:
00028     private:
00029 };
00030
00031 }
00032
00033 #endif /* !PACKET_ERROR_HPP */

```



## 5.134 ParserError.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ParserError
00006 */
00007
00008 #ifndef PARSERERROR_HPP_
00009 #define PARSERERROR_HPP_
00010
00011 #include "AError.hpp"
00012 namespace err {
00013
00014 class ParserError : public AError {
00015     public:
00016         enum ErrorCode {
00017             UNKNOWN = 1000,
00018             FILE_NOT_FOUND = 1001,
00019             INVALID_FORMAT = 1002,
00020             MISSING_FIELD = 1003,
00021             TYPE_MISMATCH = 1004
00022         };
00023
00024         ParserError(const std::string &message, ErrorCode code = UNKNOWN);
00025         virtual ~ParserError() noexcept = default;
00026         std::string getType() const noexcept override;
00027 };
00028
00029 } // namespace err
00030
00031 #endif /* !PARSERERROR_HPP_ */

```

## 5.135 ServerError.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ServerError
00006 */
00007
00008 #ifndef SERVER_ERROR_HPP
00009 #define SERVER_ERROR_HPP
00010
00011 #include "AError.hpp"
00012
00013 namespace err {
00014
00015 class ServerError : public AError {
00016     public:
00017         enum ErrorCode {
00018             UNKNOWN = 1000,
00019             CONNECTION_FAILED = 1001,
00020             TIMEOUT = 1002,
00021             INVALID_REQUEST = 1003,
00022             INTERNAL_ERROR = 1004,
00023             LIBRARY_LOAD_FAILED = 1005,
00024             CONFIG_ERROR = 1006
00025         };
00026
00027         ServerError(const std::string &message, ErrorCode code = UNKNOWN);
00028         virtual ~ServerError() noexcept = default;
00029         std::string getType() const noexcept override;
00030
00031     private:
00032 };
00033
00034 }
00035
00036 #endif /* !SERVER_ERROR_HPP */

```

## 5.136 IGameState.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025

```

```

00003  ** ryanR-type
00004  ** File description:
00005  ** IGameState
00006  */
00007
00008  #pragma once
00009
00010  #include <memory>
00011  #include <vector>
00012  #include "../systems/base/ISystem.hpp"
00013
00014  namespace gsm {
00015
00016  class IGameStateMachine;
00017
00018  class IGameState {
00019  public:
00020      virtual ~IGameState() = default;
00021
00022      virtual void enter() = 0;
00023      virtual void update(float deltaTime) = 0;
00024      virtual void exit() = 0;
00025      virtual void addSystem(std::shared_ptr<ecs::ISystem> system) = 0;
00026      virtual std::vector<std::shared_ptr<ecs::ISystem> getSystems() const = 0;
00027  };
00028
00029  } // namespace gsm

```

## 5.137 IGameStateMachine.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** IGameStateMachine
00006  */
00007
00008  #pragma once
00009
00010  #include <memory>
00011  #include <stack>
00012
00013  namespace gsm {
00014
00015  class IGameState;
00016
00017  class IGameStateMachine {
00018  public:
00019      virtual ~IGameStateMachine() = default;
00020
00021      virtual void changeState(std::shared_ptr<IGameState> newState) = 0;
00022      virtual void pushState(std::shared_ptr<IGameState> newState) = 0;
00023      virtual void popState() = 0;
00024      virtual void requestStateChange(std::shared_ptr<IGameState> newState) = 0;
00025      virtual void requestStatePush(std::shared_ptr<IGameState> newState) = 0;
00026      virtual void requestStatePop() = 0;
00027
00028      virtual void update(float deltaTime) = 0;
00029  };
00030
00031  } // namespace gsm

```

## 5.138 IInputProvider.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** IInputProvider
00006  */
00007
00008  #ifndef IINPUTPROVIDER_HPP_
00009  #define IINPUTPROVIDER_HPP_
00010
00011  #include <utility>
00012  #include "../libs/Multimedia/EventTypes.hpp"
00013  #include "InputAction.hpp"
00014  #include "InputMapping.hpp"

```

```

00015
00016 namespace ecs {
00017
00018 class IInputProvider {
00019     public:
00020         using event_t = gfx::EventType;
00021         virtual ~IInputProvider() = default;
00022
00023         virtual float getAxisValue(event_t axis, size_t clientID = 0) = 0;
00024         virtual bool isActionPressed(InputAction action, size_t clientID = 0) = 0;
00025         virtual float getActionAxis(InputAction action, size_t clientID = 0) = 0;
00026         virtual InputMapping getInputMapping(size_t clientID = 0) const = 0;
00027 };
00028
00029 } // namespace ecs
00030
00031 #endif /* !INPUTPROVIDER_HPP_ */

```

## 5.139 InputAction.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** InputAction
00006 */
00007
00008 #ifndef INPUTACTION_HPP_
00009 #define INPUTACTION_HPP_
00010
00011 namespace ecs {
00012
00013 enum class InputAction {
00014     MOVE_X,
00015     MOVE_Y,
00016     SHOOT,
00017     PAUSE,
00018     MENU_UP,
00019     MENU_DOWN,
00020     MENU_LEFT,
00021     MENU_RIGHT,
00022     MENU_SELECT,
00023     MENU_BACK,
00024 };
00025
00026 enum class RemappableAction {
00027     MOVE_LEFT,
00028     MOVE_RIGHT,
00029     MOVE_UP,
00030     MOVE_DOWN,
00031     SHOOT,
00032 };
00033
00034 } // namespace ecs
00035
00036 #endif /* !INPUTACTION_HPP_ */

```

## 5.140 InputMapping.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** InputMapping
00006 */
00007
00008 #ifndef INPUTMAPPING_HPP_
00009 #define INPUTMAPPING_HPP_
00010
00011 #include <map>
00012 #include <vector>
00013 #include "../libs/Multimedia/EventTypes.hpp"
00014 #include "InputAction.hpp"
00015
00016 namespace ecs {
00017
00018 struct RemappableKeyBinding {
00019     gfx::EventType primary;

```

```

00020     gfx::EventType secondary;
00021
00022     RemappableKeyBinding()
00023     : primary(gfx::EventType::NOTHING), secondary(gfx::EventType::NOTHING) {}
00024     RemappableKeyBinding(gfx::EventType p, gfx::EventType s)
00025     : primary(p), secondary(s) {}
00026 };
00027
00028 struct InputMapping {
00029     std::map<RemappableAction, RemappableKeyBinding> remappableKeys;
00030     std::map<InputAction, std::map<gfx::EventType, float> fixedMappings;
00031
00032     std::map<InputAction, std::map<gfx::EventType, float> getAllMappings() const {
00033         std::map<InputAction, std::map<gfx::EventType, float> all = fixedMappings;
00034
00035         for (const auto& [action, binding] : remappableKeys) {
00036             InputAction inputAction;
00037             switch (action) {
00038                 case RemappableAction::MOVE_LEFT: inputAction = InputAction::MOVE_X; break;
00039                 case RemappableAction::MOVE_RIGHT: inputAction = InputAction::MOVE_X; break;
00040                 case RemappableAction::MOVE_UP: inputAction = InputAction::MOVE_Y; break;
00041                 case RemappableAction::MOVE_DOWN: inputAction = InputAction::MOVE_Y; break;
00042                 case RemappableAction::SHOOT: inputAction = InputAction::SHOOT; break;
00043             }
00044
00045             float value = (action == RemappableAction::MOVE_LEFT ||
00046                 action == RemappableAction::MOVE_UP) ? -1.0f : 1.0f;
00047
00048             if (binding.primary != gfx::EventType::NOTHING) {
00049                 all[inputAction][binding.primary] = value;
00050             }
00051             if (binding.secondary != gfx::EventType::NOTHING) {
00052                 all[inputAction][binding.secondary] = value;
00053             }
00054         }
00055
00056         return all;
00057     }
00058 };
00059
00060 } // namespace ecs
00061
00062 #endif /* !INPUTMAPPING_HPP_ */

```

## 5.141 InputMappingManager.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** InputMappingManager
00006  */
00007
00008 #ifndef INPUTMAPPINGMANAGER_HPP_
00009 #define INPUTMAPPINGMANAGER_HPP_
00010
00011 #include <string>
00012 #include <vector>
00013 #include "InputMapping.hpp"
00014
00015 namespace ecs {
00016
00017     class InputMappingManager {
00018     public:
00019         InputMappingManager();
00020         ~InputMappingManager() = default;
00021
00022         void loadDefault();
00023
00024         void setMapping(const InputMapping& mapping);
00025         const InputMapping& getMapping() const;
00026         InputMapping& getMutableMapping();
00027
00028         gfx::EventType getKeyForRemappableAction(RemappableAction action, bool getPrimary = true) const;
00029         void remapKey(RemappableAction action, gfx::EventType newKey, bool setPrimary);
00030
00031         static std::string eventTypeToString(gfx::EventType eventType);
00032         bool isKeyboardKey(gfx::EventType eventType);
00033         static gfx::EventType stringToEventType(const std::string& str);
00034         static std::string remappableActionToString(RemappableAction action);
00035         static RemappableAction stringToRemappableAction(const std::string& str);
00036
00037     private:

```

```

00038     InputMapping _mapping;
00039 };
00040
00041 } // namespace ecs
00042
00043 #endif /* !INPUTMAPPINGMANAGER_HPP_ */

```

## 5.142 GameStateHandlers.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** Game state pack/unpack handlers (common)
00006 */
00007
00008 #ifndef COMMON_GAME_STATE_HANDLERS_HPP_
00009 #define COMMON_GAME_STATE_HANDLERS_HPP_
00010
00011 #include <memory>
00012 #include "DefaultPacketHandlers.hpp"
00013
00014 namespace common::packet {
00015     bool registerGameStateHandlers(std::shared_ptr<pm::IPacketManager> packet);
00016 }
00017
00018 #endif // COMMON_GAME_STATE_HANDLERS_HPP_

```

## 5.143 AnimationConditionFactory.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** AnimationConditionFactory
00006 */
00007
00008 #ifndef ANIMATIONCONDITIONFACTORY_HPP_
00009 #define ANIMATIONCONDITIONFACTORY_HPP_
00010
00011 #include <functional>
00012 #include <string>
00013 #include <unordered_map>
00014 #include <memory>
00015 #include "../ECS/entity/Entity.hpp"
00016 #include "../ECS/entity/registry/Registry.hpp"
00017
00018 namespace ecs {
00019
00020 class AnimationConditionFactory {
00021     public:
00022         using ConditionFunction = std::function<bool(std::shared_ptr<Registry>, Entity)>;
00023
00024         static const AnimationConditionFactory& getInstance();
00025
00026         void registerCondition(const std::string& name, ConditionFunction condition);
00027         bool evaluateCondition(const std::string& name, std::shared_ptr<Registry> registry, Entity
entity) const;
00028         bool hasCondition(const std::string& name) const;
00029         void unregisterCondition(const std::string& name);
00030         void clearConditions();
00031
00032         static bool getConditionValue(const std::string& param, std::shared_ptr<Registry> registry,
Entity entity);
00033
00034     private:
00035         AnimationConditionFactory();
00036         void initializeConditions();
00037
00038         AnimationConditionFactory(const AnimationConditionFactory&) = delete;
00039         AnimationConditionFactory& operator=(const AnimationConditionFactory&) = delete;
00040
00041         std::unordered_map<std::string, ConditionFunction> _conditions;
00042 };
00043
00044 } // namespace ecs
00045
00046 #endif /* !ANIMATIONCONDITIONFACTORY_HPP_ */

```

## 5.144 CollisionRulesParser.hpp

```

00001 #ifndef COLLISION_RULES_PARSER_HPP_
00002 #define COLLISION_RULES_PARSER_HPP_
00003
00004 #include <string>
00005 #include <map>
00006 #include <vector>
00007 #include <nlohmann/json.hpp>
00008 #include "../CollisionRules/CollisionRulesData.hpp"
00009
00010 namespace ecs {
00011
00012 class CollisionRulesParser {
00013     public:
00014         static CollisionRulesData parseFromFile(const std::string& filePath);
00015         static CollisionRulesData parseFromJsonString(const std::string& jsonString);
00016
00017     private:
00018         static void parseRulesForType(
00019             const nlohmann::json& typeJson,
00020             std::shared_ptr<std::vector<CollisionRule>> allowRules
00021         );
00022 };
00023
00024 } // namespace ecs
00025
00026 #endif // COLLISION_RULES_PARSER_HPP_

```

## 5.145 ComposantParser.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ComposantParser
00006 */
00007
00008 #ifndef COMPOSANTPARSER_HPP_
00009 #define COMPOSANTPARSER_HPP_
00010
00011 #include <string>
00012 #include <memory>
00013 #include <map>
00014 #include <typeindex>
00015 #include "../components/base/IComponent.hpp"
00016 #include "../ParserParam.hpp"
00017 #include <nlohmann/json.hpp>
00018 #include <functional>
00019
00020 class ComposantParser {
00021     public:
00022         using ShouldParseComponentCallback = std::function<bool(const std::map<std::string,
00023             std::shared_ptr<FieldValue>&>>>);
00024
00025         ComposantParser(std::shared_ptr<const std::map<std::string, std::pair<std::type_index,
00026             std::vector<Field>>> componentDefinitions,
00027             const std::map<std::type_index, ComponentCreator> &componentCreators,
00028             const ShouldParseComponentCallback &shouldParseCallback = nullptr);
00029         ~ComposantParser();
00030
00031         std::pair<std::shared_ptr<ecs::IComponent>, std::type_index> parseComponent(const std::string
00032             &componentName, const nlohmann::json &componentData);
00033
00034     protected:
00035     private:
00036         std::shared_ptr<FieldValue> parseFieldValue(const nlohmann::json &jsonValue, FieldType type);
00037         std::shared_ptr<const std::map<std::string, std::pair<std::type_index, std::vector<Field>>>
00038             _componentDefinitions;
00039         const std::map<std::type_index, ComponentCreator> &_componentCreators;
00040         ShouldParseComponentCallback _shouldParseCallback;
00041 };
00042
00043 #endif /* !COMPOSANTPARSER_HPP_ */

```

## 5.146 EntityParser.hpp

```

00001 /*

```

```

00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** EntityParser
00006  */
00007
00008  #ifndef ENTITYPARSER_HPP_
00009  #define ENTITYPARSER_HPP_
00010
00011  #include <string>
00012  #include <vector>
00013  #include <memory>
00014  #include <map>
00015  #include "../components/base/IComponent.hpp"
00016  #include "../ParserParam.hpp"
00017  #include "../ComposantParser/ComposantParser.hpp"
00018  #include "../Prefab/IPrefab.hpp"
00019  #include "../Prefab/ParsedEntityPrefab.hpp"
00020  #include <nlohmann/json.hpp>
00021
00022  class EntityParser {
00023  public:
00024      using ShouldParseComponentCallback = ComposantParser::ShouldParseComponentCallback;
00025      EntityParser(
00026          std::shared_ptr<const std::map<std::string, std::pair<std::type_index, std::vector<Field>>>
00027          componentDefinitions,
00028          const std::map<std::type_index, ComponentCreator> &componentCreators,
00029          const std::map<std::type_index, ComponentAdder> &componentAdders,
00030          const ShouldParseComponentCallback &shouldParseCallback = nullptr
00031      );
00032      ~EntityParser();
00033      std::shared_ptr<IPrefab> parseEntity(const std::string &filePath);
00034
00035  protected:
00036  private:
00037      ComposantParser _composantParser;
00038      const std::map<std::type_index, ComponentAdder> &_componentAdders;
00039      ShouldParseComponentCallback _shouldParseCallback;
00040  };
00041
00042  #endif /* !ENTITYPARSER_HPP_ */

```

## 5.147 MapParser.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** MapParser
00006  */
00007
00008  #ifndef MAPPARSER_HPP_
00009  #define MAPPARSER_HPP_
00010
00011  #include <string>
00012  #include <memory>
00013  #include <vector>
00014  #include <nlohmann/json.hpp>
00015  #include "../ECS/entity/registry/Registry.hpp"
00016  #include "../ECS/entity/EntityCreationContext.hpp"
00017  #include "../Prefab/entityPrefabManager/EntityPrefabManager.hpp"
00018  #include "../constants.hpp"
00019  #include "../types/Vector2f.hpp"
00020
00021  class MapParser {
00022  public:
00023      MapParser(
00024          std::shared_ptr<EntityPrefabManager> prefabManager,
00025          std::shared_ptr<ecs::Registry> registry
00026      );
00027      ~MapParser();
00028
00029      void parseMapFromFile(const std::string& filePath);
00030      void parseMap(const nlohmann::json& mapJson);
00031
00032      void generateMapEntities();
00033
00034      nlohmann::json getMapJson() const;
00035      void setMapJson(const nlohmann::json& mapJson);
00036
00037      void setCreationContext(const ecs::EntityCreationContext& context);
00038      ecs::EntityCreationContext getCreationContext() const;

```

```

00039
00040     private:
00041         std::shared_ptr<EntityPrefabManager> _prefabManager;
00042         std::shared_ptr<ecs::Registry> _registry;
00043         ecs::EntityCreationContext _creationContext;
00044         nlohmann::json _mapJson;
00045
00046         void createBackgroundEntity(const std::string &entityName);
00047         void createMusicEntity(const std::string &prefabName);
00048         void createGameZoneEntity(float scrollSpeed);
00049
00050         void parsePowerUps(const nlohmann::json &powerUps);
00051         void parseObstacles(const nlohmann::json &obstacles);
00052         void parseWaves(const nlohmann::json &waves);
00053
00054         std::vector<float> getPositionsFromDistrib(
00055             int count,
00056             const nlohmann::json &distribution,
00057             float limit
00058         );
00059         ecs::Entity createEntityFromPrefab(
00060             const std::string &prefabName,
00061             float x, float y
00062         );
00063     };
00064
00065 #endif /* !MAPPARSER_HPP_ */

```

## 5.148 Parser.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** Parser
00006  */
00007
00008 #ifndef PARSER_HPP_
00009 #define PARSER_HPP_
00010
00011 #include "../Prefab/IPrefab.hpp"
00012 #include <memory>
00013 #include "../EntityParser/EntityParser.hpp"
00014 #include "../Prefab/entityPrefabManager/EntityPrefabManager.hpp"
00015 #include "ParserParam.hpp"
00016 #include "../MapParser/MapParser.hpp"
00017 #include "../common/ECS/entity/registry/Registry.hpp"
00018
00019 typedef enum {
00020     CLIENT = 0,
00021     SERVER = 1
00022 } ParsingType;
00023
00024 class Parser {
00025     public:
00026         Parser(std::shared_ptr<EntityPrefabManager> prefab, ParsingType type,
00027             std::shared_ptr<ecs::Registry> registry);
00028         ~Parser();
00029
00030         std::shared_ptr<EntityPrefabManager> getPrefabManager() const;
00031         void setPrefabManager(std::shared_ptr<EntityPrefabManager> prefab);
00032         void parseAllEntities(std::string directoryPath);
00033         void parseEntity(std::string entityPath);
00034
00035         void instantiateComponentDefinitions();
00036         void instantiateComponentCreators();
00037
00038         template<typename T>
00039         void registerComponent(const ComponentCreator& creator);
00040
00041         const std::map<std::type_index, ComponentAdder>& getComponentAdders() const;
00042         ParsingType getParsingType() const;
00043         bool isClientParsing() const;
00044         bool isServerParsing() const;
00045         bool shouldParseComponent(std::map<std::string, std::shared_ptr<FieldValue>> fields) const;
00046
00047         void parseMapFromFile(const std::string& filePath);
00048
00049         std::shared_ptr<MapParser> getMapParser() const;
00050         void setRegistry(std::shared_ptr<ecs::Registry> registry);
00051
00052     private:
00053         std::shared_ptr<EntityParser> _entityParser;

```



```

00054         std::shared_ptr<MapParser> _mapParser;
00055         std::shared_ptr<EntityPrefabManager> _prefabManager;
00056
00057         std::shared_ptr<std::map<std::string, std::pair<std::type_index, std::vector<Field>>>
    _componentDefinitions;
00058         std::map<std::type_index, ComponentCreator> _componentCreators;
00059         std::map<std::type_index, ComponentAdder> _componentAdders;
00060         ParsingType _parsingType;
00061     };
00062
00063 #endif /* !PARSER_HPP_ */

```

## 5.149 ParserParam.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ParserParam
00006 */
00007
00008 #ifndef PARSERPARAM_HPP_
00009 #define PARSERPARAM_HPP_
00010
00011 #include <string>
00012 #include <vector>
00013 #include <map>
00014 #include <variant>
00015 #include <functional>
00016 #include <memory>
00017 #include "../types/Vector2f.hpp"
00018 #include <nlohmann/json.hpp>
00019 #include "../components/base/IComponent.hpp"
00020 #include "../components/permanent/TransformComponent.hpp"
00021 #include "../components/permanent/VelocityComponent.hpp"
00022 #include "../components/permanent/SpeedComponent.hpp"
00023 #include "../client/components/rendering/SpriteComponent.hpp"
00024 #include "../client/components/rendering/AnimationComponent.hpp"
00025 #include "../components/tags/ControllableTag.hpp"
00026 #include "../components/tags/PlayerTag.hpp"
00027 #include "../components/permanent/ColliderComponent.hpp"
00028
00029 enum class ParserParam {
00030     NONE = 0,
00031     NAME = 1,
00032     COMPONENTS = 2,
00033 };
00034
00035 enum class FieldType {
00036     VECTOR2F = 0,
00037     FLOAT = 1,
00038     STRING = 2,
00039     INT = 3,
00040     BOOL = 4,
00041     OBJECT = 5,
00042     JSON = 6,
00043     UNDEFINED = 7
00044 };
00045
00046 using FieldValueMap = std::map<std::string, std::shared_ptr<struct FieldValue>>;
00047 using FieldValueVariant = std::variant<math::Vector2f, float, std::string, int, bool, FieldValueMap,
    nlohmann::json>;
00048
00049 struct FieldValue : FieldValueVariant {
00050     using FieldValueVariant::FieldValueVariant;
00051     using FieldValueVariant::operator=;
00052
00053     template<typename T>
00054     FieldValue(T&& value) : FieldValueVariant(std::forward<T>(value)) {}
00055 };
00056
00057 struct Field {
00058     std::string name = "";
00059     FieldType type;
00060     bool optional = false;
00061     std::shared_ptr<FieldValue> defaultValue = nullptr;
00062
00063     Field(std::string n, FieldType t, bool opt = false, std::shared_ptr<FieldValue> def = nullptr)
00064         : name(std::move(n)), type(t), optional(opt), defaultValue(std::move(def)) {}
00065 };
00066
00067 #include <typeindex>
00068 #include "../ECS/entity/registry/Registry.hpp"

```

```

00069 #include "../ECS/entity/Entity.hpp"
00070
00071 using ComponentCreator = std::function<std::shared_ptr<ecs::IComponent>(const std::map<std::string,
std::shared_ptr<FieldValue>&&>>);
00072 using ComponentAdder = std::function<void(std::shared_ptr<ecs::Registry>, ecs::Entity,
std::shared_ptr<ecs::IComponent>>);
00073
00074 #endif /* !PARSERPARAM_HPP_ */

```

## 5.150 APrefab.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** APrefab
00006 */
00007
00008 #ifndef APREFAB_HPP_
00009 #define APREFAB_HPP_
00010
00011 #include "IPrefab.hpp"
00012 #include "../ECS/entity/registry/Registry.hpp"
00013 #include "../ECS/entity/factory/EntityFactory.hpp"
00014
00015 class APrefab : public IPrefab {
00016     public:
00017         APrefab() = default;
00018         virtual ~APrefab() = default;
00019
00020         ecs::Entity instantiate(
00021             const std::shared_ptr<ecs::Registry>& registry,
00022             const std::shared_ptr<ecs::IEntityFactory>& factory,
00023             const ecs::EntityCreationContext& context = ecs::EntityCreationContext::forLocalClient()
00024         ) override;
00025
00026         ecs::Entity instantiate(const std::shared_ptr<ecs::Registry>& registry) override;
00027 };
00028
00029 #endif /* !APREFAB_HPP_ */

```

## 5.151 EntityPrefabManager.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** EntityPrefabManager
00006 */
00007
00008 #ifndef ENTITYPREFABMANAGER_HPP_
00009 #define ENTITYPREFABMANAGER_HPP_
00010
00011 #include <vector>
00012 #include <string>
00013 #include <map>
00014 #include <memory>
00015 #include <functional>
00016 #include "../ECS/entity/Entity.hpp"
00017 #include "../ECS/entity/EntityCreationContext.hpp"
00018 #include "../ECS/entity/factory/IEntityFactory.hpp"
00019 #include "../ECS/entity/factory/EntityFactory.hpp"
00020 #include "../IPrefab.hpp"
00021
00022 class EntityPrefabManager
00023 {
00024     public:
00025         EntityPrefabManager();
00026         ~EntityPrefabManager();
00027
00028         void registerPrefab(const std::string &name, const std::shared_ptr<IPrefab> &prefab);
00029         std::shared_ptr<IPrefab> getPrefab(const std::string &name) const;
00030
00031         ecs::Entity createEntityFromPrefab(
00032             const std::string &prefabName,
00033             const std::shared_ptr<ecs::Registry> &registry,
00034             const ecs::EntityCreationContext &context
00035         );
00036 };

```

```

00036
00037     ecs::Entity createEntityFromPrefab(
00038         const std::string &prefabName,
00039         const std::shared_ptr<ecs::Registry> &registry
00040     );
00041
00042     bool hasPrefab(const std::string &name) const;
00043     void deletePrefab(const std::string &name);
00044     void clearPrefabs();
00045
00046     std::shared_ptr<ecs::IEntityFactory> getEntityFactory() const;
00047     void setEntityFactory(std::shared_ptr<ecs::IEntityFactory> factory);
00048
00049     void setOnEntityCreated(std::function<void(ecs::Entity, const std::string&)> callback);
00050
00051     private:
00052         std::map<std::string, std::shared_ptr<IPrefab>> _prefabs;
00053         std::shared_ptr<ecs::IEntityFactory> _entityFactory;
00054         std::function<void(ecs::Entity, const std::string&)> _onEntityCreated;
00055 };
00056
00057 #endif /* !ENTITYPREFABMANAGER_HPP_ */

```

## 5.152 IPrefab.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** IPrefab
00006 */
00007
00008 #ifndef IPREFAB_HPP_
00009 #define IPREFAB_HPP_
00010 #include <memory>
00011 #include "../ECS/entity/registry/Registry.hpp"
00012 #include "../ECS/entity/Entity.hpp"
00013 #include "../ECS/entity/EntityCreationContext.hpp"
00014 #include "../ECS/entity/factory/IEntityFactory.hpp"
00015
00016 class IPrefab {
00017     public:
00018         virtual ~IPrefab() = default;
00019
00020         virtual ecs::Entity instantiate(
00021             const std::shared_ptr<ecs::Registry>& registry,
00022             const std::shared_ptr<ecs::IEntityFactory>& factory,
00023             const ecs::EntityCreationContext& context = ecs::EntityCreationContext::forLocalClient()
00024         ) = 0;
00025
00026         virtual ecs::Entity instantiate(const std::shared_ptr<ecs::Registry>& registry) = 0;
00027 };
00028
00029 #endif /* !IPREFAB_HPP_ */

```

## 5.153 ParsedEntityPrefab.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ParsedEntityPrefab
00006 */
00007
00008 #ifndef PARSEENTITYPREFAB_HPP_
00009 #define PARSEENTITYPREFAB_HPP_
00010
00011 #include "IPrefab.hpp"
00012 #include <vector>
00013 #include <memory>
00014 #include <string>
00015 #include <typeindex>
00016 #include <map>
00017 #include <functional>
00018 #include "../components/base/IComponent.hpp"
00019 #include "../ECS/entity/registry/Registry.hpp"
00020 #include "../ECS/entity/EntityCreationContext.hpp"
00021 #include "../ECS/entity/factory/IEntityFactory.hpp"

```

```

00022 #include "../Parser/ParserParam.hpp"
00023
00024 class ParsedEntityPrefab : public IPrefab {
00025     public:
00026         ParsedEntityPrefab(const std::string& name, const std::map<std::type_index, ComponentAdder>&
adders);
00027         ~ParsedEntityPrefab();
00028
00029         void addComponent(std::shared_ptr<ecs::IComponent> component, std::type_index typeIndex);
00030         const std::vector<std::shared_ptr<ecs::IComponent>>& getComponents() const;
00031         std::string getName() const;
00032
00033         ecs::Entity instantiate(
00034             const std::shared_ptr<ecs::Registry>& registry,
00035             const std::shared_ptr<ecs::IEntityFactory>& factory,
00036             const ecs::EntityCreationContext& context = ecs::EntityCreationContext::forLocalClient()
00037         ) override;
00038
00039         ecs::Entity instantiate(const std::shared_ptr<ecs::Registry>& registry) override;
00040
00041     private:
00042         std::string _name;
00043         std::vector<std::pair<std::shared_ptr<ecs::IComponent>, std::type_index>> _components;
00044         const std::map<std::type_index, ComponentAdder>& _componentAdders;
00045
00046         void addParsedComponents(
00047             const std::shared_ptr<ecs::Registry>& registry,
00048             ecs::Entity entity
00049         );
00050 };
00051
00052 #endif /* !PARSEDENTITYPREFAB_HPP_ */

```

## 5.154 ResourceManager.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ResourceManager
00006 */
00007
00008 #ifndef RESOURCEMANAGER_HPP_
00009 #define RESOURCEMANAGER_HPP_
00010
00011 #include <unordered_map>
00012 #include <memory>
00013
00014
00015 class ResourceManager {
00016     public:
00017         template<typename T>
00018         void add(std::shared_ptr<T> resource);
00019
00020         template<typename T>
00021         std::shared_ptr<T> get();
00022
00023         template<typename T>
00024         bool has();
00025
00026         void clear() {
00027             resources.clear();
00028         }
00029
00030         template<typename T>
00031         void remove() {
00032             resources.erase(typeid(T).hash_code());
00033         }
00034     private:
00035         std::unordered_map<size_t, std::shared_ptr<void>> resources;
00036 };
00037
00038
00039 #include "ResourceManager.hpp"
00040
00041 #endif /* !RESOURCEMANAGER_HPP_ */

```

## 5.155 Signal.hpp

```

00001 /*

```

```

00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** Signal
00006  */
00007
00008  #ifndef SIGNAL_HPP_
00009  #define SIGNAL_HPP_
00010
00011  #ifdef _WIN32
00012      #ifndef _WIN32_WINNT
00013          #define _WIN32_WINNT 0x0A00
00014      #endif
00015
00016      #ifndef WIN32_LEAN_AND_MEAN
00017          #define WIN32_LEAN_AND_MEAN
00018      #endif
00019  #endif
00020
00021  #include <csignal>
00022
00023  class Signal {
00024  public:
00025      Signal();
00026      ~Signal();
00027
00028      static volatile sig_atomic_t stopFlag;
00029      static void signalHandler(int signum);
00030      static void setupSignalHandlers();
00031
00032  protected:
00033  private:
00034  };
00035
00036  #endif /* !SIGNAL_HPP_ */

```

## 5.156 SpatialGrid.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** SpatialGrid
00006  */
00007
00008  #ifndef SPATIALGRID_HPP_
00009  #define SPATIALGRID_HPP_
00010
00011  #include <vector>
00012  #include <unordered_set>
00013  #include <memory>
00014  #include <cmath>
00015  #include <cstdint>
00016  #include "../types/FRect.hpp"
00017  #include "../types/Vector2f.hpp"
00018  #include "../constants.hpp"
00019
00020  namespace ecs {
00021
00022      using Entity = size_t;
00023
00024      class SpatialGrid {
00025      public:
00026          SpatialGrid(
00027              float worldWidth = constants::MAX_WIDTH,
00028              float worldHeight = constants::MAX_HEIGHT,
00029              float cellSize = constants::SPATIAL_GRID_CELL_SIZE,
00030              float padding = constants::SPATIAL_GRID_PADDING
00031          );
00032          ~SpatialGrid() = default;
00033
00034          void clear();
00035          void insert(Entity entityId, const math::FRect& bounds);
00036          std::vector<Entity> query(const math::FRect& bounds) const;
00037          std::vector<std::pair<Entity, Entity>> getPotentialPairs() const;
00038          void setCellSize(float cellSize);
00039          void setOffset(float offsetX, float offsetY);
00040
00041          float getCellSize() const { return _cellSize; }
00042          size_t getNumCols() const { return _numCols; }
00043          size_t getNumRows() const { return _numRows; }
00044          float getOffsetX() const { return _offsetX; }
00045          float getOffsetY() const { return _offsetY; }

```

```

00046
00047     private:
00048         size_t getCellIndex(float x, float y) const;
00049         std::vector<size_t> getCellIndices(const math::FRect& bounds) const;
00050
00051         float _worldWidth;
00052         float _worldHeight;
00053         float _cellSize;
00054         float _padding;
00055         float _offsetX;
00056         float _offsetY;
00057         size_t _numCols;
00058         size_t _numRows;
00059         std::vector<std::vector<Entity>> _cells;
00060 };
00061
00062 }
00063
00064 #endif /* !SPATIALGRID_HPP_ */

```

## 5.157 AIMovementSystem.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** AIMovementSystem
00006  */
00007
00008 #ifndef AIMOVEMENTSYSTEM_HPP
00009 #define AIMOVEMENTSYSTEM_HPP
00010
00011 #include "../base/ASystem.hpp"
00012 #include "../components/permanent/AIMovementPatternComponent.hpp"
00013 #include "../components/permanent/TransformComponent.hpp"
00014 #include "../components/permanent/SpeedComponent.hpp"
00015 #include "../components/temporary/InputIntentComponent.hpp"
00016 #include <optional>
00017
00018 namespace ecs {
00019
00020 class AIMovementSystem : public ASystem {
00021 public:
00022     void update(
00023         std::shared_ptr<ResourceManager> resourceManager,
00024         std::shared_ptr<Registry> registry,
00025         float deltaTime) override;
00026
00027 private:
00028     void executeStraightLine(
00029         std::shared_ptr<AIMovementPatternComponent> movement,
00030         std::shared_ptr<TransformComponent> transform,
00031         std::shared_ptr<InputIntentComponent> inputIntent);
00032     void executeZigzag(
00033         std::shared_ptr<AIMovementPatternComponent> movement,
00034         std::shared_ptr<TransformComponent> transform,
00035         std::shared_ptr<InputIntentComponent> inputIntent,
00036         float deltaTime);
00037     void executeVerticalMirror(
00038         std::shared_ptr<Registry> registry,
00039         std::shared_ptr<AIMovementPatternComponent> movement,
00040         std::shared_ptr<TransformComponent> transform,
00041         std::shared_ptr<InputIntentComponent> inputIntent);
00042     void executeFollowRight(
00043         std::shared_ptr<AIMovementPatternComponent> movement,
00044         std::shared_ptr<TransformComponent> transform,
00045         std::shared_ptr<InputIntentComponent> inputIntent,
00046         float deltaTime);
00047     std::optional<size_t> findNearestPlayer(
00048         std::shared_ptr<Registry> registry,
00049         const math::Vector2f& position,
00050         float range);
00051 };
00052
00053 } // namespace ecs
00054
00055 #endif /* !AIMOVEMENTSYSTEM_HPP_ */

```

## 5.158 AIShootingSystem.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** AIShootingSystem
00006 */
00007
00008 #ifndef AISHOOTINGSYSTEM_HPP
00009 #define AISHOOTINGSYSTEM_HPP
00010
00011 #include "../base/ASystem.hpp"
00012 #include "../components/permanent/ShootingStatsComponent.hpp"
00013 #include "../components/permanent/TransformComponent.hpp"
00014
00015 namespace ecs {
00016
00017 class AIShootingSystem : public ASystem {
00018 public:
00019     void update(
00020         std::shared_ptr<ResourceManager> resourceManager,
00021         std::shared_ptr<Registry> registry,
00022         float deltaTime) override;
00023 };
00024
00025 } // namespace ecs
00026
00027 #endif /* !AISHOOTINGSYSTEM_HPP */
00028

```

## 5.159 ASystem.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ASystem
00006 */
00007
00008 #ifndef ASYSTEM_HPP_
00009 #define ASYSTEM_HPP_
00010
00011 #include <memory>
00012
00013 #include "ISystem.hpp"
00014 #include "../resourceManager/ResourceManager.hpp"
00015 #include "../ECS/entity/registry/Registry.hpp"
00016
00017 namespace ecs {
00018
00019 class ASystem : public ISystem {
00020 public:
00021     ASystem();
00022     ~ASystem() = default;
00023     void updateSystem(std::shared_ptr<ResourceManager> resourceManager, std::shared_ptr<Registry>
registry, float deltaTime) override;
00024
00025     protected:
00026         virtual void update(std::shared_ptr<ResourceManager> resourceManager,
std::shared_ptr<Registry> registry, float deltaTime) = 0;
00027
00028     private:
00029 };
00030
00031 } // namespace ecs
00032
00033 #endif /* !ASYSTEM_HPP_ */

```

## 5.160 ISystem.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ISystem
00006 */

```

```

00007
00008 #ifndef ISystem_HPP_
00009 #define ISystem_HPP_
00010
00011 #include "../resourceManager/ResourceManager.hpp"
00012 #include "../ECS/entity/registry/Registry.hpp"
00013 #include <memory>
00014
00015 namespace ecs {
00016
00017 class ISystem {
00018     public:
00019         virtual ~ISystem() = default;
00020         virtual void updateSystem(std::shared_ptr<ResourceManager> resourceManager,
00021                                 std::shared_ptr<Registry> registry, float deltaTime) = 0;
00022 };
00023 } // namespace ecs
00024
00025 #endif /* !ISystem_HPP_ */

```

## 5.161 OutOfBoundsSystem.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** OutOfBoundsSystem
00006 */
00007
00008 #ifndef OUTOFBOUNDSSYSTEM_HPP_
00009 #define OUTOFBOUNDSSYSTEM_HPP_
00010
00011 #include "../base/ASystem.hpp"
00012
00013 namespace ecs {
00014
00015 class OutOfBoundsSystem : public ASystem {
00016     public:
00017         OutOfBoundsSystem();
00018         ~OutOfBoundsSystem() = default;
00019
00020         void update(
00021             std::shared_ptr<ResourceManager> resourceManager,
00022             std::shared_ptr<Registry> registry,
00023             float deltaTime
00024         ) override;
00025
00026     private:
00027         float _margin;
00028 };
00029
00030 }
00031
00032 #endif /* !OUTOFBOUNDSSYSTEM_HPP_ */

```

## 5.162 DeathSystem.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** DeathSystem
00006 */
00007
00008 #ifndef DEATHSYSTEM_HPP_
00009 #define DEATHSYSTEM_HPP_
00010
00011 #include "../base/ASystem.hpp"
00012 #include <memory>
00013 #include "../types/Vector2f.hpp"
00014
00015 namespace ecs {
00016
00017 class DeathSystem : public ASystem {
00018     public:
00019         DeathSystem();
00020         ~DeathSystem() = default;

```



```

00021
00022     void update(
00023         std::shared_ptr<ResourceManager> resourceManager,
00024         std::shared_ptr<Registry> registry,
00025         float deltaTime
00026     ) override;
00027
00028     private:
00029         math::Vector2f getFirstHitboxCenter(
00030             std::shared_ptr<Registry> registry,
00031             ecs::Entity entity
00032         );
00033         void spawnExplosionAtMobCenter(
00034             std::shared_ptr<ResourceManager> resourceManager,
00035             std::shared_ptr<Registry> registry,
00036             ecs::Entity mobEntity,
00037             const std::string& prefabName
00038         );
00039     };
00040
00041 }
00042
00043 #endif /* !DEATHSYSTEM_HPP_ */

```

## 5.163 HealthSystem.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** HealthSystem
00006 */
00007
00008 #ifndef HEALTHSYSTEM_HPP_
00009 #define HEALTHSYSTEM_HPP_
00010
00011 #include "../base/ASystem.hpp"
00012 #include <memory>
00013
00014 namespace ecs {
00015
00016     class HealthSystem : public ASystem {
00017     public:
00018         HealthSystem();
00019         ~HealthSystem() override = default;
00020
00021         void update(
00022             std::shared_ptr<ResourceManager> resourceManager,
00023             std::shared_ptr<Registry> registry,
00024             float deltaTime
00025         ) override;
00026
00027     private:
00028         void _handleDamageUpdates(std::shared_ptr<Registry> registry);
00029         void _handleHealthUpdates(std::shared_ptr<Registry> registry);
00030     };
00031
00032 }
00033
00034 #endif /* !HEALTHSYSTEM_HPP_ */

```

## 5.164 InputNormalizer.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** InputNormalizer
00006 */
00007
00008 #ifndef INPUTNORMALIZER_HPP_
00009 #define INPUTNORMALIZER_HPP_
00010
00011 #include "../types/Vector2f.hpp"
00012
00013 namespace ecs {
00014
00015     class InputNormalizer {

```

```

00016     public:
00017         static math::Vector2f normalizeDirection(const math::Vector2f &direction);
00018         static math::Vector2f normalizeAnalogInput(float rawX, float rawY);
00019 };
00020
00021 } // namespace ecs
00022
00023 #endif /* !INPUTNORMALIZER_HPP_ */

```

## 5.165 ActionFactory.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** ActionFactory
00006  */
00007
00008 #ifndef ACTIONFACTORY_HPP_
00009 #define ACTIONFACTORY_HPP_
00010
00011 #include <functional>
00012 #include <memory>
00013 #include <string>
00014 #include <unordered_map>
00015 #include "../ECS/entity/Entity.hpp"
00016
00017 namespace ecs {
00018     class Registry;
00019 }
00020
00021 class ActionFactory {
00022     public:
00023         static const ActionFactory& getInstance();
00024
00025         using ActionFunction = std::function<void(std::shared_ptr<ecs::Registry>, ecs::Entity,
00026         ecs::Entity)>;
00027
00028         void registerAction(const std::string& actionId, ActionFunction action);
00029
00030         void executeAction(
00031             const std::string& actionId,
00032             std::shared_ptr<ecs::Registry> registry,
00033             ecs::Entity self, ecs::Entity other) const;
00034
00035         bool hasAction(const std::string& actionId) const;
00036     private:
00037         ActionFactory();
00038         ~ActionFactory() = default;
00039         ActionFactory(const ActionFactory&) = delete;
00040         ActionFactory& operator=(const ActionFactory&) = delete;
00041
00042         void initializeConditions();
00043
00044         std::unordered_map<std::string, ActionFunction> _actions;
00045 };
00046
00047 #endif /* !ACTIONFACTORY_HPP_ */

```

## 5.166 InteractionSystem.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** InteractionSystem
00006  */
00007
00008 #ifndef INTERACTIONSYSYSTEM_HPP_
00009 #define INTERACTIONSYSYSTEM_HPP_
00010
00011 #include "../base/ASystem.hpp"
00012 #include <memory>
00013 #include <string>
00014
00015 namespace ecs {
00016     class Registry;

```

```

00017 }
00018
00019 namespace ecs {
00020
00021 class InteractionSystem : public ASystem {
00022     public:
00023         InteractionSystem();
00024         ~InteractionSystem() = default;
00025
00026         void update(
00027             std::shared_ptr<ResourceManager> resourceManager,
00028             std::shared_ptr<Registry> registry,
00029             float deltaTime
00030         ) override;
00031
00032     private:
00033 };
00034
00035 } // namespace ecs
00036
00037
00038 #endif /* !INTERACTIONSYSTEM_HPP_ */

```

## 5.167 TagRegistry.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** TagRegistry
00006  */
00007
00008 #ifndef TAGREGISTRY_HPP_
00009 #define TAGREGISTRY_HPP_
00010
00011 #include <unordered_map>
00012 #include <functional>
00013 #include <memory>
00014 #include <string>
00015 #include "../ECS/entity/Entity.hpp"
00016 #include "../ECS/entity/registry/Registry.hpp"
00017
00018 class TagRegistry {
00019     public:
00020         static const TagRegistry& getInstance();
00021
00022         template<typename T>
00023         void registerTag(const std::string& tagName) {
00024             _tagCheckers[tagName] = [](std::shared_ptr<ecs::Registry> reg, ecs::Entity ent) {
00025                 return reg->hasComponent<T>(ent);
00026             };
00027         }
00028
00029         bool hasTag(std::shared_ptr<ecs::Registry> registry, ecs::Entity entity, const std::string&
tagName) const;
00030         std::vector<std::string> getTags(std::shared_ptr<ecs::Registry> registry, ecs::Entity entity)
const;
00031
00032     private:
00033         TagRegistry();
00034         ~TagRegistry() = default;
00035         TagRegistry(const TagRegistry&) = delete;
00036         TagRegistry& operator=(const TagRegistry&) = delete;
00037
00038         void initializeTags();
00039
00040         std::unordered_map<std::string,
00041             std::function<bool(std::shared_ptr<ecs::Registry>, ecs::Entity)>> _tagCheckers;
00042 };
00043
00044 #endif /* !TAGREGISTRY_HPP_ */

```

## 5.168 TriggerSystem.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:

```

```

00005 ** TriggerSystem
00006 */
00007
00008 #ifndef TRIGGERSYSTEM_HPP_
00009 #define TRIGGERSYSTEM_HPP_
00010
00011 #include "../base/ASystem.hpp"
00012 #include "../components/base/IComponent.hpp"
00013 #include "../components/temporary/TriggerIntentComponent.hpp"
00014 #include "../components/permanent/TransformComponent.hpp"
00015 #include "../components/permanent/ColliderComponent.hpp"
00016 #include "../CollisionRules/CollisionRules.hpp"
00017 #include "../SpatialGrid/SpatialGrid.hpp"
00018 #include "TagRegistry.hpp"
00019
00020 namespace ecs {
00021
00022 class TriggerSystem : public ASystem {
00023     public:
00024         TriggerSystem();
00025         ~TriggerSystem() = default;
00026
00027         void update(
00028             std::shared_ptr<ResourceManager> resourceManager,
00029             std::shared_ptr<Registry> registry,
00030             float deltaTime
00031         ) override;
00032
00033     private:
00034         void buildSpatialGrid(
00035             std::shared_ptr<Registry> registry
00036         );
00037
00038         bool checkCollision(
00039             const TransformComponent& transformA,
00040             const ColliderComponent& colliderA,
00041             const TransformComponent& transformB,
00042             const ColliderComponent& colliderB
00043         );
00044
00045         bool shouldCollide(
00046             std::shared_ptr<Registry> registry,
00047             size_t entityA,
00048             const ColliderComponent& colliderA,
00049             size_t entityB
00050         );
00051
00052         SpatialGrid _spatialGrid;
00053 };
00054
00055 }
00056
00057 #endif /* !TRIGGERSYSTEM_HPP_ */

```

## 5.169 LifetimeSystem.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** LifetimeSystem
00006 */
00007
00008 #ifndef LIFETIMESYSTEM_HPP_
00009 #define LIFETIMESYSTEM_HPP_
00010
00011 #include "../base/ASystem.hpp"
00012 #include <memory>
00013
00014 namespace ecs {
00015
00016 class LifetimeSystem : public ASystem {
00017     public:
00018         LifetimeSystem();
00019         ~LifetimeSystem() = default;
00020
00021         void update(
00022             std::shared_ptr<ResourceManager> resourceManager,
00023             std::shared_ptr<Registry> registry,
00024             float deltaTime
00025         ) override;
00026 };
00027

```

```

00028 }
00029
00030 #endif /* !LIFETIMESYSTEM_HPP_ */

```

## 5.170 InputToVelocitySystem.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** VelocitySystem
00006 */
00007
00008 #ifndef VELOCITYSYSTEM_HPP_
00009 #define VELOCITYSYSTEM_HPP_
00010
00011 #include "../base/ASystem.hpp"
00012
00013 namespace ecs {
00014
00015 class InputToVelocitySystem : public ASystem {
00016     public:
00017         InputToVelocitySystem();
00018         ~InputToVelocitySystem() = default;
00019
00020         void update(std::shared_ptr<ResourceManager> resourceManager, std::shared_ptr<Registry>
00021 registry, float deltaTime) override;
00022 };
00023 } // namespace ecs
00024
00025 #endif /* !VELOCITYSYSTEM_HPP_ */

```

## 5.171 MovementSystem.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** MovementSystem
00006 */
00007
00008 #ifndef MOVEMENTSYSTEM_HPP_
00009 #define MOVEMENTSYSTEM_HPP_
00010
00011 #include "../base/ASystem.hpp"
00012 #include "../components/base/IComponent.hpp"
00013 #include "../components/temporary/MovementIntentComponent.hpp"
00014 #include "../components/permanent/TransformComponent.hpp"
00015 #include "../components/permanent/SpeedComponent.hpp"
00016 #include "../components/permanent/VelocityComponent.hpp"
00017 #include "../components/permanent/ColliderComponent.hpp"
00018 #include "../CollisionRules/CollisionRules.hpp"
00019 #include "../systems/interactions/TagRegistry.hpp"
00020 #include "../SpatialGrid/SpatialGrid.hpp"
00021
00022 namespace ecs {
00023
00024 class MovementSystem : public ASystem {
00025     public:
00026         MovementSystem();
00027         ~MovementSystem() = default;
00028
00029         void update(
00030             std::shared_ptr<ResourceManager> resourceManager,
00031             std::shared_ptr<Registry> registry,
00032             float deltaTime
00033         ) override;
00034
00035     private:
00036         void buildSpatialGrid(std::shared_ptr<Registry> registry);
00037
00038         bool checkCollision(
00039             std::shared_ptr<Registry> registry,
00040             size_t entityId,
00041             math::Vector2f newPos
00042         );
00043         math::Vector2f calculateSmoothMovement(

```

```

00044         std::shared_ptr<Registry> registry,
00045         size_t entityId,
00046         math::Vector2f startPos,
00047         math::Vector2f desiredPos
00048     );
00049     math::Vector2f calculateSlidingMovement(
00050         std::shared_ptr<Registry> registry,
00051         size_t entityId,
00052         math::Vector2f basePos,
00053         math::Vector2f desiredPos
00054     );
00055     math::Vector2f calculateSmoothSlidingPosition(
00056         std::shared_ptr<Registry> registry,
00057         size_t entityId,
00058         math::Vector2f startPos,
00059         math::Vector2f desiredPos
00060     );
00061     void handlePushCollision(
00062         std::shared_ptr<Registry> registry,
00063         size_t entityId,
00064         math::Vector2f finalPos,
00065         float deltaTime
00066     );
00067     bool shouldCollide(
00068         std::shared_ptr<Registry> registry,
00069         size_t entityA,
00070         const ColliderComponent& colliderA,
00071         size_t entityB
00072     );
00073     bool checkCollisionWithBoundaries(
00074         std::shared_ptr<Registry> registry,
00075         size_t entityId,
00076         math::Vector2f newPos
00077     );
00078
00079     SpatialGrid _spatialGrid;
00080     std::vector<Entity> _boundaryEntities;
00081 };
00082
00083 }
00084
00085 #endif /* !MOVEMENTSYSTEM_HPP_ */

```

## 5.172 ScoreSystem.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** ScoreSystem
00006  */
00007
00008 #ifndef SCORESYSTEM_HPP_
00009 #define SCORESYSTEM_HPP_
00010
00011 #include "../base/ASystem.hpp"
00012
00013 namespace ecs {
00014
00015     class ScoreSystem : public ASystem {
00016     public:
00017         ScoreSystem();
00018         ~ScoreSystem();
00019         void update(
00020             std::shared_ptr<ResourceManager> resourceManager,
00021             std::shared_ptr<Registry> registry,
00022             float deltaTime
00023         ) override;
00024     protected:
00025     private:
00026     };
00027
00028 } // namespace ecs
00029
00030 #endif /* !SCORESYSTEM_HPP_ */

```

## 5.173 ShootingSystem.hpp

```

00001 /*

```

```

00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** ShootingSystem
00006  */
00007
00008 #ifndef SHOOTINGSYSTEM_HPP_
00009 #define SHOOTINGSYSTEM_HPP_
00010
00011 #include "../base/ASystem.hpp"
00012 #include "../../components/base/IComponent.hpp"
00013 #include "../../components/temporary/ShootIntentComponent.hpp"
00014 #include "../../components/permanent/ShootingStatsComponent.hpp"
00015 #include "../../components/permanent/TransformComponent.hpp"
00016 #include "../../components/permanent/VelocityEngineComponent.hpp"
00017 #include "../../types/Vector2f.hpp"
00018 #include <cmath>
00019 #include <string>
00020
00021 namespace ecs {
00022
00023 class ShootingSystem : public ASystem {
00024     public:
00025         ShootingSystem();
00026         ~ShootingSystem() = default;
00027
00028         void update(
00029             std::shared_ptr<ResourceManager> resourceManager,
00030             std::shared_ptr<Registry> registry,
00031             float deltaTime
00032         ) override;
00033
00034     private:
00035         void spawnProjectile(
00036             std::shared_ptr<Registry> registry,
00037             std::shared_ptr<ResourceManager> resourceManager,
00038             const std::string& prefabName,
00039             const math::Vector2f &position,
00040             float angle,
00041             ecs::Entity shooterEntity
00042         );
00043
00044         math::Vector2f calculateProjectileVelocity(
00045             float angle,
00046             float speed
00047         );
00048 };
00049
00050 } // namespace ecs
00051
00052 #endif /* !SHOOTINGSYSTEM_HPP_ */

```

## 5.174 SpawnSystem.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** SpawnSystem
00006  */
00007
00008 #ifndef SPAWNSYSTEM_HPP_
00009 #define SPAWNSYSTEM_HPP_
00010
00011 #include "../base/ASystem.hpp"
00012
00013 namespace ecs {
00014
00015 class SpawnSystem : public ASystem {
00016     public:
00017         SpawnSystem();
00018         ~SpawnSystem() = default;
00019
00020         void update(
00021             std::shared_ptr<ResourceManager> resourceManager,
00022             std::shared_ptr<Registry> registry,
00023             float deltaTime
00024         ) override;
00025 };
00026
00027 }
00028
00029 #endif /* !SPAWNSYSTEM_HPP_ */

```

## 5.175 ASystemManager.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ASystemManager
00006 */
00007
00008 #ifndef ASYSTEMMANAGER_HPP_
00009 #define ASYSTEMMANAGER_HPP_
00010
00011 #include <vector>
00012 #include <memory>
00013
00014 #include "ISystemManager.hpp"
00015 #include "../resourceManager/ResourceManager.hpp"
00016 #include "../ECS/entity/registry/Registry.hpp"
00017 #include "../base/ISystem.hpp"
00018
00019 namespace ecs {
00020
00021 class ASystemManager : public ISystemManager {
00022     public:
00023         ASystemManager();
00024         ~ASystemManager();
00025         void updateAllSystems(std::shared_ptr<ResourceManager> resourceManager,
00026                               std::shared_ptr<Registry> registry, float deltaTime) override;
00027         void addSystem(std::shared_ptr<ISystem> system) override;
00028         void removeSystem(std::shared_ptr<ISystem> system) override;
00029     private:
00030         std::vector<std::shared_ptr<ISystem>> _systems;
00031 };
00032
00033 } // namespace ecs
00034
00035 #endif /* !ASYSTEMMANAGER_HPP_ */

```

## 5.176 ISystemManager.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ISystemManager
00006 */
00007
00008 #ifndef ISYSTEMMANAGER_HPP_
00009 #define ISYSTEMMANAGER_HPP_
00010
00011 #include <memory>
00012
00013 #include "../base/ISystem.hpp"
00014 #include "../resourceManager/ResourceManager.hpp"
00015 #include "../ECS/entity/registry/Registry.hpp"
00016
00017 namespace ecs {
00018
00019 class ISystemManager {
00020     public:
00021         virtual ~ISystemManager() = default;
00022         virtual void updateAllSystems(std::shared_ptr<ResourceManager> resourceManager,
00023                                       std::shared_ptr<Registry> registry, float deltaTime) = 0;
00024         virtual void addSystem(std::shared_ptr<ISystem> system) = 0;
00025         virtual void removeSystem(std::shared_ptr<ISystem> system) = 0;
00026 };
00027
00028 } // namespace ecs
00029
00029 #endif /* !ISYSTEMMANAGER_HPP_ */

```

## 5.177 SystemManager.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type

```



```

00004 ** File description:
00005 ** SystemManager
00006 */
00007
00008 #ifndef SYSTEMMANAGER_HPP_
00009 #define SYSTEMMANAGER_HPP_
00010
00011 #include "ASystemManager.hpp"
00012
00013 namespace ecs {
00014
00015 class SystemManager : public ASystemManager {
00016 public:
00017     SystemManager();
00018     ~SystemManager() = default;
00019
00020 protected:
00021 private:
00022 };
00023
00024 } // namespace ecs
00025
00026 #endif /* !SYSTEMMANAGER_HPP_ */

```

## 5.178 translationToECS.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** R-Type
00004 ** File description:
00005 ** Header
00006 */
00007
00008 #ifndef TRASLATION_TO_ECS_HPP_
00009 #define TRASLATION_TO_ECS_HPP_
00010
00011 enum componentType {
00012     PLAYER_TAG = 0x00,
00013     TRANSFORM = 0x01,
00014     SPEED_COMP = 0x02,
00015     HEALTH = 0x03,
00016     COLLIDER = 0x04,
00017     SHOOTING_STATS = 0x05,
00018     SCORE = 0x06,
00019     AI_MOVEMENT_PATTERN = 0x07,
00020     DAMAGE = 0x08,
00021     LIFETIME = 0x09,
00022     VELOCITY = 0x0A,
00023     AI_MOVER_TAG = 0x0B,
00024     AI_SHOOTER_TAG = 0x0C,
00025     CONTROLLABLE_TAG = 0x0D,
00026     ENEMY_PROJECTILE_TAG = 0x0E,
00027     GAME_ZONE_COLLIDER_TAG = 0x0F,
00028     MOB_TAG = 0x10,
00029     OBSTACLE_TAG = 0x11,
00030     PLAYER_PROJECTILE_TAG = 0x12,
00031     SCORE_TAG = 0x13,
00032     SHOOTER_TAG = 0x14,
00033     PROJECTILE_PASS_THROUGH_TAG = 0x15,
00034     PROJECTILE_PREFAB = 0x16,
00035     NETWORK_ID = 0x17,
00036     GAME_ZONE = 0x18
00037 };
00038
00039 #endif /* !TRASLATION_TO_ECS_HPP_ */

```

## 5.179 Chrono.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** Chrono wrapper for std::chrono
00006 */
00007
00008 #ifndef CHRONO_HPP_
00009 #define CHRONO_HPP_
00010

```

```

00011 #include <chrono>
00012
00013 namespace math {
00014
00015 class Chrono {
00016     public:
00017         Chrono();
00018         ~Chrono() = default;
00019
00020         void start();
00021         void stop();
00022         void reset();
00023         float getElapsedSeconds() const;
00024         float getElapsedMilliseconds() const;
00025         bool isRunning() const;
00026
00027     private:
00028         std::chrono::high_resolution_clock::time_point _startTime;
00029         std::chrono::high_resolution_clock::time_point _stopTime;
00030         bool _isRunning;
00031 };
00032
00033 } // namespace math
00034
00035 #endif /* !CHRONO_HPP_ */

```

## 5.180 FRect.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** FRect
00006  */
00007
00008 #ifndef FRECT_HPP_
00009 #define FRECT_HPP_
00010
00011 namespace math {
00012
00013 class FRect {
00014     public:
00015         FRect();
00016         FRect(float left, float top, float width, float height);
00017         FRect(FRect const &other);
00018         ~FRect() = default;
00019
00020         float getLeft() const;
00021         void setLeft(float left);
00022         float getTop() const;
00023         void setTop(float top);
00024         float getWidth() const;
00025         void setWidth(float width);
00026         float getHeight() const;
00027         void setHeight(float height);
00028
00029         bool contains(float x, float y) const;
00030         bool intersects(FRect const &other) const;
00031         bool intersects(FRect const &other, FRect &intersection) const;
00032
00033         FRect &operator=(FRect const &other);
00034         bool operator==(FRect const &other) const;
00035         bool operator!=(FRect const &other) const;
00036
00037     private:
00038         float left;
00039         float top;
00040         float width;
00041         float height;
00042 };
00043
00044 } // namespace math
00045
00046 #endif /* !FRECT_HPP_ */

```

## 5.181 Vector2f.hpp

```

00001 /*

```

```

00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** Vector2f
00006  */
00007
00008  #ifndef VECTOR2F_HPP_
00009  #define VECTOR2F_HPP_
00010
00011  namespace math {
00012
00013  class Vector2f {
00014  public:
00015      Vector2f(float x = 0.0f, float y = 0.0f);
00016      Vector2f(Vector2f const &other);
00017      ~Vector2f() = default;
00018
00019      float getX() const;
00020      void setX(float x);
00021      float getY() const;
00022      void setY(float y);
00023
00024      Vector2f getVector() const;
00025      Vector2f operator*(float scalar) const;
00026      Vector2f operator-(Vector2f const &other) const;
00027      Vector2f operator+(Vector2f const &other) const;
00028      void operator=(Vector2f const &other);
00029      void operator+=(Vector2f const &other);
00030      void operator-=(Vector2f const &other);
00031      void operator*=(float scalar);
00032      void operator/=(float scalar);
00033  private:
00034      float _x;
00035      float _y;
00036  };
00037
00038  } // namespace math
00039
00040  #endif /* !VECTOR2F_HPP_ */

```

## 5.182 Constants.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2025
00003  ** R-Type
00004  ** File description:
00005  ** Header
00006  */
00007
00008  #ifndef SERVER_CONSTANTS_HPP_
00009  #define SERVER_CONSTANTS_HPP_
00010
00011  #include "../common/constants.hpp"
00012
00013  namespace constants {
00014      /* TPS */
00015      constexpr long TPS = 50;
00016      constexpr long CD_TPS = 20;
00017
00018      /* Core */
00019      constexpr long SERVER_THREAD_SLEEP_MS = 10;
00020      constexpr int SERVER_UP = 1;
00021
00022      /* Server */
00023      constexpr uint8_t ID_SERVER = 0;
00024      constexpr uint8_t BITMASK_INT = 32;
00025
00026      /* Packets */
00027      constexpr char END_OFSTRING_ST = '\r';
00028      constexpr char END_OFSTRING_ND = '\n';
00029      constexpr char END_OFSTRING_TRD = '\0';
00030  }
00031
00032  #endif /* !SERVER_CONSTANTS */

```

## 5.183 ComponentDeltaTracker.hpp

```

00001  /*

```

```

00002 ** EPITECH PROJECT, 2025
00003 ** R-Type
00004 ** File description:
00005 ** Component Delta Tracker - Track and send only modified components
00006 */
00007
00008 #ifndef COMPONENT_DELTA_TRACKER_HPP
00009 #define COMPONENT_DELTA_TRACKER_HPP
00010
00011 #include <unordered_map>
00012 #include <vector>
00013 #include <stdint>
00014 #include <map>
00015 #include "../common/translationToECS.hpp"
00016
00017 namespace rserv {
00018
00019 struct EntitySnapshot {
00020     uint32_t entityId;
00021     uint32_t componentMask;
00022     std::map<uint8_t, std::vector<uint64_t> components;
00023
00024     EntitySnapshot() : entityId(0), componentMask(0) {}
00025 };
00026
00027 class ComponentDeltaTracker {
00028 public:
00029     std::vector<uint64_t> createEntityDelta(uint8_t clientId, uint32_t entityId, const
EntitySnapshot& currentSnapshot);
00030     std::vector<uint64_t> createMultiEntityDelta(uint8_t clientId, const
std::vector<EntitySnapshot>& entities);
00031     EntitySnapshot applyDelta(uint8_t clientId, const std::vector<uint64_t>& deltaPayload);
00032     void clearClientCache(uint8_t clientId);
00033     void clearEntityCache(uint8_t clientId, uint32_t entityId);
00034     void clearAllCaches();
00035
00036 private:
00037     std::unordered_map<uint8_t, std::unordered_map<uint32_t, EntitySnapshot> _clientEntityCache;
00038
00039     std::vector<uint64_t> serializeFullSnapshot(uint32_t entityId, const EntitySnapshot&
snapshot);
00040     std::vector<uint64_t> serializeDelta(uint32_t entityId, uint32_t changedMask, const
std::map<uint8_t, std::vector<uint64_t>& changedComponents);
00041 };
00042
00043 } // namespace rserv
00044
00045 #endif // COMPONENT_DELTA_TRACKER_HPP

```

## 5.184 ComponentSerializer.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** R-Type
00004 ** File description:
00005 ** Component Serializer - Helper for converting ECS components to/from packets
00006 */
00007
00008 #ifndef COMPONENT_SERIALIZER_HPP
00009 #define COMPONENT_SERIALIZER_HPP
00010
00011 #include "ComponentDeltaTracker.hpp"
00012 #include <cstring>
00013 #include <string>
00014
00015 namespace rserv {
00016 class ComponentSerializer {
00017 public:
00018     static std::vector<uint64_t> serializePosition(uint32_t x, uint32_t y);
00019     static void deserializePosition(const std::vector<uint64_t>& data, uint32_t& x, uint32_t& y);
00020     static std::vector<uint64_t> serializeVelocity(int32_t vx, int32_t vy);
00021     static void deserializeVelocity(const std::vector<uint64_t>& data, int32_t& vx, int32_t& vy);
00022     static std::vector<uint64_t> serializeHealth(uint32_t current, uint32_t max);
00023     static void deserializeHealth(const std::vector<uint64_t>& data, uint32_t& current, uint32_t&
max);
00024     static std::vector<uint64_t> serializeCollider(uint32_t x, uint32_t y, uint32_t width,
uint32_t height, uint32_t rotation);
00025     static std::vector<uint64_t> serializeShootingStats(uint32_t fireRate, uint32_t damage,
uint32_t lastShot);
00026     static std::vector<uint64_t> serializeScore(uint64_t score);
00027     static std::vector<uint64_t> serializeDamage(uint32_t damage);
00028     static std::vector<uint64_t> serializeLifetime(uint64_t lifetime);
00029     static std::vector<uint64_t> serializeSpeed(uint64_t speed);

```

```

00030         static std::vector<uint64_t> serializeAIMovementPattern(uint32_t patternId);
00031         static std::vector<uint64_t> serializeNetworkId(uint32_t networkId);
00032         static std::vector<uint64_t> serializeGameZone(uint32_t x, uint32_t y, uint32_t width,
uint32_t height);
00033         static EntitySnapshot createSnapshotFromComponents(uint32_t entityId, const
std::vector<uint64_t>& componentData);
00034         static std::vector<uint64_t> snapshotToComponentData(const EntitySnapshot& snapshot);
00035         static bool isTagComponent(uint8_t component);
00036         static bool isOneParamComponent(uint8_t component);
00037     };
00038 } // namespace rserv
00039
00040 #endif // COMPONENT_SERIALIZER_HPP

```

## 5.185 GameEndState.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** GameEndState
00006 */
00007
00008 #ifndef SERVER_GAMEENDSTATE_HPP_
00009 #define SERVER_GAMEENDSTATE_HPP_
00010
00011 #include "../AGameState.hpp"
00012 #include "resourceManager/ResourceManager.hpp"
00013
00014 namespace gsm {
00015
00016 class GameEndState : public AGameState {
00017 public:
00018     GameEndState(std::shared_ptr<IGameStateMachine> gsm, std::shared_ptr<ResourceManager>
resourceManager);
00019     ~GameEndState() override = default;
00020
00021     void enter() override;
00022 };
00023
00024 } // namespace gsm
00025
00026 #endif // SERVER_GAMEENDSTATE_HPP_

```

## 5.186 ShutdownState.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ShutdownState
00006 */
00007
00008 #ifndef SERVER_SHUTDOWNSTATE_HPP_
00009 #define SERVER_SHUTDOWNSTATE_HPP_
00010
00011 #include "../AGameState.hpp"
00012 #include "resourceManager/ResourceManager.hpp"
00013
00014 namespace gsm {
00015
00016 class ShutdownState : public AGameState {
00017 public:
00018     ShutdownState(std::shared_ptr<IGameStateMachine> gsm, std::shared_ptr<ResourceManager>
resourceManager);
00019     ~ShutdownState() override = default;
00020
00021     void enter() override;
00022 };
00023
00024 } // namespace gsm
00025
00026 #endif // SERVER_SHUTDOWNSTATE_HPP_

```

## 5.187 ServerInputProvider.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ServerInputProvider
00006 */
00007
00008 #ifndef SERVERINPUTPROVIDER_HPP_
00009 #define SERVERINPUTPROVIDER_HPP_
00010
00011 #include "../common/InputMapping/IInputProvider.hpp"
00012 #include "../common/InputMapping/InputMapping.hpp"
00013 #include "../common/constants.hpp"
00014 #include <iostream>
00015 #include <vector>
00016 #include <algorithm>
00017
00018 namespace ecs {
00019
00020 class ServerInputProvider : public IInputProvider {
00021     public:
00022         ServerInputProvider();
00023         ~ServerInputProvider() override = default;
00024
00025         float getAxisValue(event_t axis, size_t clientID = 0) override;
00026
00027         bool isActionPressed(InputAction action, size_t clientID = 0) override;
00028         float getActionAxis(InputAction action, size_t clientID = 0) override;
00029         InputMapping getInputMapping(size_t clientID = 0) const override;
00030
00031         void setAxisValue(ecs::InputAction action, float value, size_t clientID = 0);
00032
00033         void addClientInputMapping(size_t clientID, size_t identity, const InputMapping& mapping);
00034         void updateInputFromEvent(size_t clientID, constants::EventType eventType, float value);
00035         std::vector<size_t> getConnectedClients() const;
00036
00037     private:
00038         std::vector<std::tuple<size_t, size_t, InputMapping>> _inputMapping;
00039         std::map<size_t, std::map<ecs::InputAction, float>> _clientAxisValues;
00040
00041         using InputHandler = void (ServerInputProvider::*)(size_t, float);
00042         std::vector<InputHandler> _inputHandlers;
00043
00044         void handleUp(size_t clientID, float value);
00045         void handleDown(size_t clientID, float value);
00046         void handleLeft(size_t clientID, float value);
00047         void handleRight(size_t clientID, float value);
00048         void handleStop(size_t clientID, float value);
00049         void handleShoot(size_t clientID, float value);
00050
00051 };
00052
00053 } // namespace ecs
00054
00055 #endif /* !SERVERINPUTPROVIDER_HPP_ */

```

## 5.188 IServer.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** R-Type
00004 ** File description:
00005 ** Header
00006 */
00007
00008 #ifndef ISERVER_HPP_
00009 #define ISERVER_HPP_
00010
00011 #include <memory>
00012 #include <vector>
00013 #include <queue>
00014 #include <asio.hpp>
00015
00016 #include "ServerConfig.hpp"
00017 #include "../libs/Packet/IPacketManager.hpp"
00018 #include "../libs/Network/INetwork.hpp"
00019 #include "../common/constants.hpp"
00020
00021 namespace rserv {
00022 class IServer {

```

```

00023 public:
00024     virtual ~IServer() = default;
00025
00026     virtual void init() = 0;
00027     virtual void start() = 0;
00028     virtual void stop() = 0;
00029
00030     virtual void setConfig(std::shared_ptr<ServerConfig> config) = 0;
00031     virtual std::shared_ptr<ServerConfig> getConfig() const = 0;
00032     virtual uint16_t getPort() const = 0;
00033     virtual void setPort(uint16_t port) = 0;
00034
00035     virtual int getState() const = 0;
00036     virtual void setState(int state) = 0;
00037
00038     virtual operator int() const noexcept = 0;
00039
00040     virtual std::shared_ptr<net::INetwork> getNetwork() const = 0;
00041     virtual void setNetwork(std::shared_ptr<net::INetwork> network) = 0;
00042
00043     virtual void onClientConnected(uint8_t idClient) = 0;
00044     virtual void onClientDisconnected(uint8_t idClient) = 0;
00045     virtual void onPacketReceived(uint8_t idClient, const pm::IPacketManager &packet) = 0;
00046
00047     virtual void processIncomingPackets() = 0;
00048     virtual bool processConnections(std::pair<asio::ip::udp::endpoint, std::vector<uint8_t> client) =
0;
00049     virtual bool processDisconnections(uint8_t idClient) = 0;
00050     virtual bool processEvents(uint8_t idClient) = 0;
00051     virtual bool processEndOfGame(uint8_t idClient) = 0;
00052
00053     virtual std::vector<uint8_t> getConnectedClients() const = 0;
00054     virtual std::vector<asio::ip::udp::endpoint> getConnectedClientEndpoints() const = 0;
00055     virtual size_t getClientCount() const = 0;
00056
00057     virtual std::shared_ptr<std::queue<std::tuple<uint8_t, constants::EventType, double>>
getEventQueue() = 0;
00058     virtual bool hasEvents() const = 0;
00059 };
00060 } // namespace rserv = r-type server
00061 #endif /* !ISERVER_HPP_ */

```

## 5.189 Server.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** R-Type
00004 ** File description:
00005 ** Header
00006 */
00007
00008 #ifndef SERVER_HPP_
00009     #define SERVER_HPP_
00010
00011 #ifdef _WIN32
00012     #ifndef _WIN32_WINNT
00013         #define _WIN32_WINNT 0x0A00
00014     #endif
00015
00016     #ifndef WIN32_LEAN_AND_MEAN
00017         #define WIN32_LEAN_AND_MEAN
00018     #endif
00019 #endif
00020
00021 #include <queue>
00022 #include <map>
00023 #include <memory>
00024 #include "IServer.hpp"
00025 #include "ServerConfig.hpp"
00026 #include "deltaTracker/ComponentDeltaTracker.hpp"
00027 #include "deltaTracker/ComponentSerializer.hpp"
00028 #include "../libs/Network/INetwork.hpp"
00029 #include "../libs/Buffer/IBuffer.hpp"
00030 #include "../common/DLLoader/DLLoader.hpp"
00031 #include "../common/DLLoader/LoaderType.hpp"
00032 #include "../common/constants.hpp"
00033 #include "../common/InputMapping/InputAction.hpp"
00034 #include "../common/resourceManager/ResourceManager.hpp"
00035 #include "../common/ECS/entity/registry/Registry.hpp"
00036 #include "Signal.hpp"
00037
00038 namespace rserv {
00039     class Server : public IServer {

```

```

00040     public:
00041         Server(std::shared_ptr<ResourceManager> resourceManager);
00042         ~Server();
00043
00044         void init() override;
00045         void start() override;
00046         void stop() override;
00047
00048         void setConfig(std::shared_ptr<ServerConfig> config) override;
00049         std::shared_ptr<ServerConfig> getConfig() const override;
00050         uint16_t getPort() const override;
00051         void setPort(uint16_t port) override;
00052
00053         int getState() const override;
00054         void setState(int state) override;
00055
00056         operator int() const noexcept override;
00057
00058         std::shared_ptr<net::INetwork> getNetwork() const override;
00059         void setNetwork(std::shared_ptr<net::INetwork> network) override;
00060
00061         void onClientConnected(uint8_t idClient) override;
00062         void onClientDisconnected(uint8_t idClient) override;
00063         void onPacketReceived(uint8_t idClient, const pm::IPacketManager &packet) override;
00064
00065         std::vector<uint8_t> getConnectedClients() const override;
00066         std::vector<asio::ip::udp::endpoint> getConnectedClientEndpoints() const override;
00067         size_t getClientCount() const override;
00068
00069         std::shared_ptr<std::queue<std::tuple<uint8_t, constants::EventType, double>>>
getEventQueue() override;
00070         bool hasEvents() const override;
00071
00072         /* Received Packet Handling */
00073         void processIncomingPackets() override;
00074         bool processConnections(std::pair<asio::ip::udp::endpoint, std::vector<uint8_t> client)
override;
00075         bool processDisconnections(uint8_t idClient) override;
00076         bool processEvents(uint8_t idClient) override;
00077         bool processEndOfGame(uint8_t idClient) override;
00078         bool processWhoAmI(uint8_t idClient);
00079
00080         /* Sent Packet Handling */
00081         bool connectionPacket(asio::ip::udp::endpoint endpoint);
00082         bool gameStatePacket();
00083         bool canStartPacket();
00084         std::vector<uint64_t> spawnPacket(size_t entity, const std::string prefabName);
00085         std::vector<uint64_t> deathPacket(size_t entity);
00086         void setCurrentMap(const std::vector<uint64_t> &map);
00087         std::vector<uint64_t> getCurrentMap() const;
00088         bool isGameStarted() const;
00089         bool allClientsReady() const;
00090         uint32_t getSequenceNumber() const;
00091         std::shared_ptr<pm::IPacketManager> getPacketManager() const;
00092         void incrementSequenceNumber();
00093         void setResourceManager(std::shared_ptr<ResourceManager> resourceManager);
00094         void clearEntityDeltaCache(uint8_t clientId, uint32_t entityId);
00095     private:
00096         void loadNetworkLibrary();
00097         void loadBufferLibrary();
00098         void loadPacketLibrary();
00099         DLLoader<createNetworkLib_t> _networloader;
00100         DLLoader<createBuffer_t> _bufferloader;
00101         DLLoader<createPacket_t> _packetloader;
00102         uint8_t _nextClientId;
00103         uint32_t _sequenceNumber;
00104         std::vector<std::tuple<uint8_t, asio::ip::udp::endpoint, std::string>> _clients;
00105         std::map<uint8_t, bool> _clientsReady;
00106
00107         std::shared_ptr<ServerConfig> _config;
00108         std::shared_ptr<net::INetwork> _network;
00109         std::shared_ptr<IBuffer> _buffer;
00110         std::shared_ptr<pm::IPacketManager> _packet;
00111         std::shared_ptr<std::queue<std::tuple<uint8_t, constants::EventType, double>>> _eventQueue;
00112
00113         bool _gameStarted;
00114         std::shared_ptr<ResourceManager> _resourceManager;
00115         std::chrono::steady_clock::time_point _lastGameStateTime;
00116
00117         ComponentDeltaTracker _deltaTracker;
00118
00119         /* Functions to build game state packets */
00120         std::vector<std::function<std::vector<uint64_t>(std::shared_ptr<ecs::Registry>,
ecs::Entity)>> _convertFunctions;
00121         std::vector<uint64_t> convertTagComponent(std::shared_ptr<ecs::Registry> registry,
ecs::Entity i);
00122         std::vector<uint64_t> convertTransformComponent(std::shared_ptr<ecs::Registry> registry,

```



```

    ecs::Entity i);
00123     std::vector<uint64_t> convertSpeedComponent(std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00124     std::vector<uint64_t> convertHealthComponent(std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00125     std::vector<uint64_t> convertColliderComponent(std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00126     std::vector<uint64_t> convertShootStatComponent(std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00127     std::vector<uint64_t> convertScoreComponent(std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00128     std::vector<uint64_t> convertAIMovementPatternComponent(std::shared_ptr<ecs::Registry>
    registry, ecs::Entity i);
00129     std::vector<uint64_t> convertDamageComponent(std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00130     std::vector<uint64_t> convertLifetimeComponent(std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00131     std::vector<uint64_t> convertVelocityComponent(std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00132     std::vector<uint64_t> convertAIMoverTagComponent(std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00133     std::vector<uint64_t> convertAIShooterTagComponent(std::shared_ptr<ecs::Registry>
    registry, ecs::Entity i);
00134     std::vector<uint64_t> convertControllableTagComponent(std::shared_ptr<ecs::Registry>
    registry, ecs::Entity i);
00135     std::vector<uint64_t> convertEnemyProjectileTagComponent(std::shared_ptr<ecs::Registry>
    registry, ecs::Entity i);
00136     std::vector<uint64_t> convertGameZoneColliderTagComponent(std::shared_ptr<ecs::Registry>
    registry, ecs::Entity i);
00137     std::vector<uint64_t> convertMobTagComponent(std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00138     std::vector<uint64_t> convertObstacleTagComponent(std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00139     std::vector<uint64_t> convertPlayerProjectileTagComponent(std::shared_ptr<ecs::Registry>
    registry, ecs::Entity i);
00140     std::vector<uint64_t> convertShooterTagComponent(std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00141     std::vector<uint64_t>
    convertProjectilePassThroughTagComponent(std::shared_ptr<ecs::Registry> registry, ecs::Entity i);
00142     std::vector<uint64_t> convertProjectilePrefabComponent(std::shared_ptr<ecs::Registry>
    registry, ecs::Entity i);
00143     std::vector<uint64_t> convertNetworkIdComponent(std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00144     std::vector<uint64_t> convertGameZoneComponent(std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00145     };
00146 } // namespace rserv = r-type server
00147
00148 #endif /* !SERVER_HPP_ */

```

## 5.190 ServerConfig.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** Header
00004 ** File description:
00005 ** Header
00006 */
00007
00008 #ifndef SERVER_CONFIG_HPP_
00009     #define SERVER_CONFIG_HPP_
00010
00011     #include <cstdint>
00012     #include <string>
00013
00014     namespace rserv {
00015         class ServerConfig {
00016             public:
00017                 ServerConfig();
00018                 ~ServerConfig();
00019
00020                 int getState() const;
00021
00022                 void setPort(uint16_t port);
00023                 uint16_t getPort() const;
00024
00025                 void setState(int state);
00026
00027                 void setNbClients(int nbClients);
00028                 int getNbClients() const;
00029
00030                 std::string getIp() const;
00031                 void setIp(std::string ip);

```

```

00032
00033         void setIsDebug(bool isDebug);
00034         bool getIsDebug() const;
00035     private:
00036         int _state;
00037         uint16_t _port;
00038         int _nbClients;
00039         std::string _ip;
00040         bool _isDebug;
00041     };
00042 } // namespace rserv = r-type server
00043
00044 #endif /* !SERVER_CONFIG_HPP_ */

```

## 5.191 ServerMovementInputSystem.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ServerMovementInputSystem
00006 */
00007
00008 #ifndef SERVERMOVEMENTINPUTSYSTEM_HPP_
00009 #define SERVERMOVEMENTINPUTSYSTEM_HPP_
00010
00011 #include "../common/systems/base/ASystem.hpp"
00012 #include "../common/components/temporary/InputIntentComponent.hpp"
00013 #include "../common/InputMapping/IInputProvider.hpp"
00014 #include "../common/InputMapping/InputAction.hpp"
00015 #include "../initResourcesManager/ServerInputProvider.hpp"
00016 #include <memory>
00017
00018 namespace ecs {
00019
00020 class ServerMovementInputSystem : public ASystem {
00021     public:
00022         ServerMovementInputSystem();
00023         ~ServerMovementInputSystem() = default;
00024
00025         void update(std::shared_ptr<ResourceManager> resourceManager, std::shared_ptr<Registry>
00026 registry, float deltaTime) override;
00027     private:
00028         math::Vector2f getMovementDirection(std::shared_ptr<IInputProvider> inputProvider, size_t
00029 clientID) const;
00030         void updateInputIntent(std::shared_ptr<Registry> registry, Entity entityId, const
00031 math::Vector2f &direction);
00032         math::Vector2f getAnalogStickInput(std::shared_ptr<IInputProvider> inputProvider, size_t
00033 clientID) const;
00034         math::Vector2f normalizeDirection(const math::Vector2f &direction) const;
00035 };
00036 } // namespace ecs
00037
00038 #endif /* !SERVERMOVEMENTINPUTSYSTEM_HPP_ */

```

## 5.192 ServerShootInputSystem.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ServerShootInputSystem
00006 */
00007
00008 #ifndef SERVERSHOOTINPUTSYSTEM_HPP_
00009 #define SERVERSHOOTINPUTSYSTEM_HPP_
00010
00011 #include "../common/systems/base/ASystem.hpp"
00012 #include "../common/components/temporary/ShootIntentComponent.hpp"
00013 #include "../common/InputMapping/IInputProvider.hpp"
00014 #include "../common/InputMapping/InputAction.hpp"
00015 #include "../initResourcesManager/ServerInputProvider.hpp"
00016 #include <memory>
00017
00018 namespace ecs {
00019

```

```
00020 class ServerShootInputSystem : public ASystem {
00021     public:
00022         ServerShootInputSystem();
00023         ~ServerShootInputSystem() = default;
00024
00025         void update(std::shared_ptr<ResourceManager> resourceManager, std::shared_ptr<Registry>
registry, float deltaTime) override;
00026
00027     private:
00028         void updateShootIntent(std::shared_ptr<Registry> registry, Entity entityId);
00029 };
00030
00031 } // namespace ecs
00032
00033 #endif /* !SERVERSHOOTINPUTSYSTEM_HPP_ */
```



# Index

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ClientNetwork.hpp, 185

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/Core.hpp, 200

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/SettingsConfig.hpp, 215

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/SettingsManager.hpp, 216

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/Utils.hpp, 233

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/colors.hpp, 187

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/AnimationComponent.hpp, 188

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/HealthBarComponent.hpp, 190

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/HitboxRenderComponent.hpp, 190

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/MusicComponent.hpp, 191

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/ParallaxComponent.hpp, 191

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/RectangleRenderComponent.hpp, 193

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/SpriteComponent.hpp, 193

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/rendering/TextComponent.hpp, 194

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/tags/BackGroundMusicTag.hpp, 194

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/temporary/MusicIntentComponent.hpp, 195

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/components/temporary/SoundIntentComponent.hpp, 195

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/constants.hpp, 196

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/machine/AGameStateMachine.hpp, 201

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/machine/GameStateMachine.hpp, 202

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/base/AGameState.hpp, 202

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Boot/BootState.hpp, 203

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Dev/DevState.hpp, 204

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/InGame/InGameState.hpp, 205

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Loading/LoadingState.hpp, 206

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Lobby/LobbyState.hpp, 207

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/LobbyBrowser/LobbyBrowserState.hpp, 208

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/MainMenu/MainMenuState.hpp, 208

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Pause/PauseState.hpp, 209

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Results/ResultsState.hpp, 209

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/gsm/states/scenes/Settings/SettingsState.hpp, 210

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/initResourcesManager/GraphicalInputProvider.hpp, 211

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/initResourcesManager/initResourcesManager.hpp, 212

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/input/MouseInputHandler.hpp, 213

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/interpolation/NetworkStateComponent.hpp, 213

/home/albane/epitech/tech3/r-type/compile-flags/ryanR-

- type/client/packet/DefaultPacketHandlers.hpp, 214
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/audio/MusicSystem.hpp, 217
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/audio/SoundSystem.hpp, 217
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/input/MovementInputSystem.hpp, 218
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/input/ShootInputSystem.hpp, 218
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/network/NetworkInterpolationSystem.hpp, 219
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/AnimationRenderingSystem.hpp, 219
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/GameZoneRenderingSystem.hpp, 219
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/GameZoneViewSystem.hpp, 220
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/HealthBarRenderingSystem.hpp, 220
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/HitboxRenderingSystem.hpp, 221
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/ParallaxRenderingSystem.hpp, 221
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/RectangleRenderingSystem.hpp, 222
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/SpriteRenderingSystem.hpp, 222
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/systems/rendering/TextRenderingSystem.hpp, 222
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/core/AFocusableElement.hpp, 223
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/core/IFocusable.hpp, 224
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/core/UILayout.hpp, 224
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/Background.hpp, 225
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/Text.hpp, 231
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/base/UIElement.hpp, 226
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/focusable/Button.hpp, 227
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/focusable/Slider.hpp, 228
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/focusable/TextInput.hpp, 229
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/elements/focusable/ToggleSwitch.hpp, 230
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/manager/UISystem.hpp, 231
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/client/ui/navigation/UISystem.hpp, 232
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/CollisionRules/CollisionRules.hpp, 234
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/CollisionRules/CollisionRulesData.hpp, 235
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/DLLoader/DLLoader.hpp, 253
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/DLLoader/ILoader.hpp, 254
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/DLLoader/LoaderType.hpp, 255
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/entity/Entity.hpp, 256
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/entity/EntityCreationContext.hpp, 257
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/entity/componentArray/AComponentArray.hpp, 258
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/entity/componentArray/IComponentArray.hpp, 256
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/entity/factory/EntityFactory.hpp, 257
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/entity/factory/IEntityFactory.hpp, 258
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/entity/registry/Registry.hpp, 258
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/ECS/view/View.hpp, 259
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/AError.hpp, 260
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/ClientError.hpp, 260
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/ClientNetworkError.hpp, 261

- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/IError.hpp, [261](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/LibrairiesLoadError.hpp, [262](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/PacketError.hpp, [262](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/ParserError.hpp, [263](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Error/ServerError.hpp, [263](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/InputMapping/IInputProvider.hpp, [264](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/InputMapping/InputAction.hpp, [265](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/InputMapping/InputMapping.hpp, [265](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/InputMapping/InputMappingManager.hpp, [266](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/Animation/AnimationConditionFactory.hpp, [267](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/CollisionRulesParser.hpp, [268](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/ComposantParser/ComposantParser.hpp, [268](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/EntityParser/EntityParser.hpp, [268](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/MapParser/MapParser.hpp, [269](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/Parser.hpp, [270](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Parser/ParserParam.hpp, [271](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Prefab/APrefab.hpp, [272](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Prefab/IPrefab.hpp, [273](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Prefab/ParsedEntityPrefab.hpp, [273](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Prefab/entityPrefabManager/EntityPrefabManager.hpp, [272](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/Signal/Signal.hpp, [274](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/SpatialGrid/SpatialGrid.hpp, [275](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/base/AComponent.hpp, [235](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/base/IComponent.hpp, [235](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/AIMovementPatternComponent.hpp, [236](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/ColliderComponent.hpp, [237](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/DamageComponent.hpp, [237](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/GameZoneComponent.hpp, [238](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/HealthComponent.hpp, [238](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/InteractionConfigComponent.hpp, [239](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/LifetimeComponent.hpp, [239](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/NetworkIdComponent.hpp, [240](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/OwnerComponent.hpp, [240](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/ProjectilePrefabComponent.hpp, [241](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/ScoreComponent.hpp, [241](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/ScoreValueComponent.hpp, [241](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/ShootingStatsComponent.hpp, [242](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/SpeedComponent.hpp, [243](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/TransformComponent.hpp, [243](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/permanent/VelocityComponent.hpp, [244](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/tags/AIMoverTag.hpp, [244](#)
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/components/tags/AIShooterTag.hpp, [244](#)

- 244
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/components/tags/ControllableTag.hpp, 245
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/components/tags/EnemyProjectileTag.hpp, 245
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/components/tags/GameZoneColliderTag.hpp, 245
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/components/tags/LocalPlayerTag.hpp, 246
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/components/tags/MobTag.hpp, 246
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/components/tags/ObstacleTag.hpp, 246
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/components/tags/PlayerProjectileTag.hpp, 247
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/components/tags/PlayerTag.hpp, 247
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/components/tags/ProjectilePassThroughTag.hpp, 248
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/components/tags/ShooterTag.hpp, 248
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/components/temporary/DamageIntentComponent.hpp, 248
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/components/temporary/DeathIntentComponent.hpp, 249
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/components/temporary/InputIntentComponent.hpp, 249
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/components/temporary/MovementIntentComponent.hpp, 250
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/components/temporary/ScoreIntentComponent.hpp, 250
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/components/temporary/ShootIntentComponent.hpp, 250
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/components/temporary/SpawnIntentComponent.hpp, 251
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/components/temporary/TriggerIntentComponent.hpp, 252
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/constants.hpp, 197
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/debug.hpp, 252
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/gsm/IGameState.hpp, 263
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/gsm/IGameStateMachine.hpp, 264
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/packet/DefaultPacketHandlers.hpp, 265
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/packet/GameStateHandlers.hpp, 267
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/resourceManager/ResourceManager.hpp, 274
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/systems/ai/AIMovementSystem.hpp, 276
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/systems/ai/AIShootingSystem.hpp, 277
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/systems/base/ASystem.hpp, 277
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/systems/base/ISystem.hpp, 277
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/systems/bounds/OutOfBoundsSystem.hpp, 278
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/systems/death/DeathSystem.hpp, 278
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/systems/health/HealthSystem.hpp, 279
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/systems/input/InputNormalizer.hpp, 279
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/systems/interactions/ActionFactory.hpp, 280
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/systems/interactions/InteractionSystem.hpp, 280
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/systems/interactions/TagRegistry.hpp, 281
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/systems/interactions/TriggerSystem.hpp, 281
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/systems/lifetime/LifetimeSystem.hpp, 282
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-  
type/common/systems/movement/InputToVelocitySystem.hpp, 283
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-



- type/common/systems/movement/MovementSystem.hpp, 283
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/score/ScoreSystem.hpp, 284
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/shooting/ShootingSystem.hpp, 284
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/spawn/SpawnSystem.hpp, 285
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/systemManager/ASystemManager.hpp, 286
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/systemManager/ISystemManager.hpp, 286
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/systems/systemManager/SystemManager.hpp, 286
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/translationToECS.hpp, 287
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/types/Chrono.hpp, 287
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/types/FRect.hpp, 288
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/common/types/Vector2f.hpp, 288
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/Constants.hpp, 289
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/Core.hpp, 200
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/IServer.hpp, 292
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/Server.hpp, 293
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/ServerConfig.hpp, 295
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/Utils.hpp, 234
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/deltaTracker/ComponentDeltaTracker.hpp, 289
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/deltaTracker/ComponentSerializer.hpp, 290
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/machine/AGameStateMachine.hpp, 202
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/machine/GameStateMachine.hpp, 202
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/AGameState.hpp, 203
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/scenes/Boot/BootState.hpp, 204
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/scenes/GameEnd/GameEndState.hpp, 291
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/scenes/InGame/InGameState.hpp, 206
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/scenes/Loading/LoadingState.hpp, 206
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/scenes/Lobby/LobbyState.hpp, 207
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/gsm/states/scenes/Shutdown/ShutdownState.hpp, 291
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/initResourcesManager/ServerInputProvider.hpp, 292
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/initResourcesManager/initResourcesManager.hpp, 212
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/packet/DefaultPacketHandlers.hpp, 215
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/systems/input/ServerMovementInputSystem.hpp, 296
- /home/albane/epitech/tech3/r-type/compile-flags/ryanR-type/server/systems/input/ServerShootInputSystem.hpp, 296
- ActionFactory, 17
- addSystem
  - ecs::ASystemManager, 36
  - gsm::AGameState, 23
- APrefab, 33
  - instantiate, 34
- canBeFocused
  - ui::AFocusableElement, 20
- changeState
  - gsm::AGameStateMachine, 25
- ClientNetwork, 45
- Close
  - DLLoader< T >, 60
- ComposantParser, 53
- containsPoint
  - ui::ToggleSwitch, 173
- Core, 54
- createEntity
  - ecs::EntityFactory, 63
- debug::Debug, 58
- DLLoader< T >, 60
  - Close, 60
  - Error, 60
  - getHandler, 60
  - Open, 60
  - Symbol, 61

ecs::AComponent, 15  
 ecs::AComponentArray< T >, 16  
     getMaxEntityId, 16  
     removeComponents, 16  
     removeOneComponent, 16  
 ecs::AIMovementPatternComponent, 27  
 ecs::AIMovementSystem, 28  
     update, 28  
 ecs::AIMoverTag, 29  
 ecs::AIShooterTag, 29  
 ecs::AIShootingSystem, 29  
     update, 30  
 ecs::AnimationClip, 30  
 ecs::AnimationComponent, 31  
 ecs::AnimationCondition, 32  
 ecs::AnimationConditionFactory, 32  
 ecs::AnimationRenderingSystem, 33  
     update, 33  
 ecs::ASystem, 34  
     updateSystem, 35  
 ecs::ASystemManager, 36  
     addSystem, 36  
     removeSystem, 36  
     updateAllSystems, 36  
 ecs::BackGroundMusicTag, 39  
 ecs::ColliderComponent, 49  
 ecs::CollisionRule, 50  
 ecs::CollisionRules, 50  
 ecs::CollisionRulesData, 51  
 ecs::CollisionRulesParser, 51  
 ecs::ControllableTag, 54  
 ecs::DamageComponent, 55  
 ecs::DamageIntentComponent, 56  
 ecs::DeathIntentComponent, 56  
 ecs::DeathSystem, 57  
     update, 57  
 ecs::EnemyProjectileTag, 61  
 ecs::EntityCreationContext, 61  
 ecs::EntityFactory, 62  
     createEntity, 63  
     getNextNetworkId, 63  
     setNextNetworkId, 63  
 ecs::GameZoneColliderTag, 68  
 ecs::GameZoneComponent, 69  
 ecs::GameZoneRenderingSystem, 69  
     update, 70  
 ecs::GameZoneViewSystem, 70  
     update, 71  
 ecs::GraphicalInputProvider, 71  
     getActionAxis, 72  
     getAxisValue, 72  
     getInputMapping, 72  
     isActionPressed, 72  
 ecs::Group< Components >, 73  
 ecs::HealthBarComponent, 73  
 ecs::HealthBarRenderingSystem, 73  
     update, 74  
 ecs::HealthComponent, 74  
 ecs::HealthSystem, 75  
     update, 75  
 ecs::HitboxRenderComponent, 76  
 ecs::HitboxRenderingSystem, 76  
     update, 77  
 ecs::IComponent, 77  
 ecs::IComponentArray, 78  
 ecs::IEntityFactory, 78  
 ecs::IInputProvider, 81  
 ecs::InputIntentComponent, 84  
 ecs::InputMapping, 85  
 ecs::InputMappingManager, 85  
 ecs::InputNormalizer, 86  
 ecs::InputToVelocitySystem, 86  
     update, 86  
 ecs::InteractionConfigComponent, 87  
 ecs::InteractionMapping, 87  
 ecs::InteractionSystem, 88  
     update, 88  
 ecs::ISystem, 90  
 ecs::ISystemManager, 90  
 ecs::LifetimeComponent, 93  
 ecs::LifetimeSystem, 94  
     update, 94  
 ecs::LocalPlayerTag, 100  
 ecs::MobTag, 103  
 ecs::MovementInputSystem, 104  
     update, 104  
 ecs::MovementIntentComponent, 105  
 ecs::MovementSystem, 105  
     update, 106  
 ecs::MultiShotPattern, 106  
 ecs::MusicComponent, 107  
 ecs::MusicIntentComponent, 108  
 ecs::MusicSystem, 108  
     update, 109  
 ecs::NetworkHealthState, 109  
 ecs::NetworkIdComponent, 110  
 ecs::NetworkInterpolationSystem, 110  
     update, 111  
 ecs::NetworkStateComponent, 111  
 ecs::NetworkTransformState, 112  
 ecs::ObstacleTag, 112  
 ecs::OutOfBoundsSystem, 113  
     update, 113  
 ecs::OwnerComponent, 114  
 ecs::ParallaxComponent, 115  
 ecs::ParallaxLayer, 116  
 ecs::ParallaxRenderingSystem, 117  
     update, 117  
 ecs::PlayerProjectileTag, 122  
 ecs::PlayerTag, 122  
 ecs::ProjectilePassThroughTag, 123  
 ecs::ProjectilePrefabComponent, 123  
 ecs::RectangleRenderComponent, 124  
 ecs::RectangleRenderingSystem, 124  
     update, 125  
 ecs::Registry, 125

- ecs::RemappableKeyBinding, 126
- ecs::ScoreComponent, 128
- ecs::ScoreSystem, 129
  - update, 130
- ecs::ScoreValueComponent, 130
- ecs::ServerInputProvider, 138
  - getActionAxis, 139
  - getAxisValue, 139
  - getInputMapping, 140
  - isActionPressed, 140
- ecs::ServerMovementInputSystem, 140
  - update, 141
- ecs::ServerShootInputSystem, 141
  - update, 142
- ecs::ShooterTag, 147
- ecs::ShootingStatsComponent, 147
- ecs::ShootingSystem, 148
  - update, 149
- ecs::ShootInputSystem, 149
  - update, 150
- ecs::ShootIntentComponent, 150
- ecs::SoundIntentComponent, 156
- ecs::SoundSystem, 157
  - update, 157
- ecs::SpatialGrid, 158
- ecs::SpawnIntentComponent, 158
- ecs::SpawnSystem, 159
  - update, 160
- ecs::SpeedComponent, 160
- ecs::SpriteComponent, 161
- ecs::SpriteRenderingSystem, 161
  - update, 162
- ecs::SystemManager, 162
- ecs::TextComponent, 166
- ecs::TextRenderingSystem, 170
  - update, 170
- ecs::TransformComponent, 174
- ecs::Transition, 175
- ecs::TriggerIntentComponent, 175
- ecs::TriggerSystem, 176
  - update, 176
- ecs::VelocityComponent, 183
- ecs::View< Components >, 184
- ecs::View< Components >::Iterator, 91
- enter
  - gsm::AGameState, 23
  - gsm::BootState, 40
  - gsm::DevState, 59
  - gsm::GameEndState, 67
  - gsm::InGameState, 83
  - gsm::LoadingState, 95
  - gsm::LobbyBrowserState, 97
  - gsm::LobbyState, 99
  - gsm::MainMenuState, 101
  - gsm::PauseState, 121
  - gsm::ResultsState, 127
  - gsm::SettingsState, 146
  - gsm::ShutdownState, 152
- EntityParser, 63
- EntityPrefabManager, 64
- err::AError, 17
  - getCode, 18
  - getDetails, 18
  - getType, 18
  - what, 18
- err::ClientError, 44
  - getType, 45
- err::ClientNetworkError, 48
  - getType, 49
- err::IError, 79
- err::LibrariesLoadError, 92
  - getType, 93
- err::PacketError, 114
  - getType, 115
- err::ParserError, 119
  - getType, 120
- err::ServerError, 137
  - getType, 138
- Error
  - DLLoader< T >, 60
- exit
  - gsm::AGameState, 23
  - gsm::BootState, 40
  - gsm::DevState, 59
  - gsm::InGameState, 83
  - gsm::LoadingState, 95
  - gsm::LobbyBrowserState, 97
  - gsm::LobbyState, 99
  - gsm::MainMenuState, 101
  - gsm::PauseState, 121
  - gsm::ResultsState, 127
  - gsm::SettingsState, 146
- Field, 65
- FieldValue, 65
- getActionAxis
  - ecs::GraphicalInputProvider, 72
  - ecs::ServerInputProvider, 139
- getAxisValue
  - ecs::GraphicalInputProvider, 72
  - ecs::ServerInputProvider, 139
- getClientCount
  - rserve::Server, 133
- getCode
  - err::AError, 18
- getConfig
  - rserve::Server, 133
- getConnectedClientEndpoints
  - rserve::Server, 133
- getConnectedClients
  - rserve::Server, 133
- getDetails
  - err::AError, 18
- getEventQueue
  - rserve::Server, 133
- getHandler

- DLLoader< T >, 60
- getInputMapping
  - ecs::GraphicalInputProvider, 72
  - ecs::ServerInputProvider, 140
- getMaxEntityId
  - ecs::AComponentArray< T >, 16
- getNetwork
  - rserve::Server, 133
- getNextNetworkId
  - ecs::EntityFactory, 63
- getPort
  - rserve::Server, 134
- getState
  - rserve::Server, 134
- getSystems
  - gsm::AGameState, 23, 24
- getType
  - err::AError, 18
  - err::ClientError, 45
  - err::ClientNetworkError, 49
  - err::LibrariesLoadError, 93
  - err::PacketError, 115
  - err::ParserError, 120
  - err::ServerError, 138
- gsm::AGameState, 22
  - addSystem, 23
  - enter, 23
  - exit, 23
  - getSystems, 23, 24
  - update, 24
- gsm::AGameStateMachine, 24
  - changeState, 25
  - popState, 25
  - pushState, 25, 26
  - requestStateChange, 26
  - requestStatePop, 26
  - requestStatePush, 26
  - update, 26
- gsm::BootState, 39
  - enter, 40
  - exit, 40
  - update, 40
- gsm::DevState, 58
  - enter, 59
  - exit, 59
  - update, 59
- gsm::GameEndState, 66
  - enter, 67
- gsm::GameStateMachine, 67
  - requestStateChange, 68
  - requestStatePop, 68
  - requestStatePush, 68
- gsm::IGameState, 80
- gsm::IGameStateMachine, 80
- gsm::InGameState, 82
  - enter, 83
  - exit, 83
  - update, 83
- gsm::LoadingState, 94
  - enter, 95
  - exit, 95
  - update, 96
- gsm::LobbyBrowserState, 96
  - enter, 97
  - exit, 97
  - update, 97
- gsm::LobbyState, 98
  - enter, 99
  - exit, 99
  - update, 99
- gsm::MainMenuState, 100
  - enter, 101
  - exit, 101
  - update, 101
- gsm::PauseState, 120
  - enter, 121
  - exit, 121
  - update, 121
- gsm::ResultsState, 127
  - enter, 127
  - exit, 127
  - update, 128
- gsm::ScoreFeedback, 129
- gsm::SettingsState, 144
  - enter, 146
  - exit, 146
  - update, 146
- gsm::ShutdownState, 151
  - enter, 152
- handleInput
  - ui::AFocusableElement, 20
  - ui::Slider, 155
  - ui::TextInput, 169
  - ui::ToggleSwitch, 173
- hasEvents
  - rserve::Server, 134
- ILoader, 81
- init
  - rserve::Server, 134
- instantiate
  - APrefab, 34
  - ParsedEntityPrefab, 118
- IPrefab, 88
- isActionPressed
  - ecs::GraphicalInputProvider, 72
  - ecs::ServerInputProvider, 140
- isFocused
  - ui::AFocusableElement, 21
- MapParser, 102
- math::Chrono, 44
- math::FRect, 65
- math::Vector2f, 183
- MouseClickedInfo, 103
- MouseInputHandler, 103

- NetworkEvent, [109](#)
- onActivated
  - ui::AFocusableElement, [21](#)
  - ui::Slider, [155](#)
- onClientConnected
  - rserve::Server, [134](#)
- onClientDisconnected
  - rserve::Server, [134](#)
- onFocusGained
  - ui::AFocusableElement, [21](#)
- onFocusLost
  - ui::AFocusableElement, [21](#)
- onNavigateLeft
  - ui::Slider, [156](#)
- onNavigateRight
  - ui::Slider, [156](#)
- onPacketReceived
  - rserve::Server, [134](#)
- Open
  - DLLoader< T >, [60](#)
- operator int
  - rserve::Server, [135](#)
- ParsedEntityPrefab, [118](#)
  - instantiate, [118](#)
- Parser, [119](#)
- popState
  - gsm::AGameStateMachine, [25](#)
- processConnections
  - rserve::Server, [135](#)
- processDisconnections
  - rserve::Server, [135](#)
- processEndOfGame
  - rserve::Server, [135](#)
- processEvents
  - rserve::Server, [135](#)
- processIncomingPackets
  - rserve::Server, [135](#)
- pushState
  - gsm::AGameStateMachine, [25](#), [26](#)
- removeComponents
  - ecs::AComponentArray< T >, [16](#)
- removeOneComponent
  - ecs::AComponentArray< T >, [16](#)
- removeSystem
  - ecs::ASystemManager, [36](#)
- render
  - ui::Background, [38](#)
  - ui::Button, [43](#)
  - ui::Slider, [156](#)
  - ui::Text, [165](#)
  - ui::TextInput, [169](#)
  - ui::ToggleSwitch, [173](#)
  - ui::UILayout, [180](#)
- requestStateChange
  - gsm::AGameStateMachine, [26](#)
  - gsm::GameStateMachine, [68](#)
- requestStatePop
  - gsm::AGameStateMachine, [26](#)
  - gsm::GameStateMachine, [68](#)
- requestStatePush
  - gsm::AGameStateMachine, [26](#)
  - gsm::GameStateMachine, [68](#)
- ResourceManager, [126](#)
- rserve::ComponentDeltaTracker, [52](#)
- rserve::ComponentSerializer, [52](#)
- rserve::EntitySnapshot, [64](#)
- rserve::IServer, [89](#)
- rserve::Server, [131](#)
  - getClientCount, [133](#)
  - getConfig, [133](#)
  - getConnectedClientEndpoints, [133](#)
  - getConnectedClients, [133](#)
  - getEventQueue, [133](#)
  - getNetwork, [133](#)
  - getPort, [134](#)
  - getState, [134](#)
  - hasEvents, [134](#)
  - init, [134](#)
  - onClientConnected, [134](#)
  - onClientDisconnected, [134](#)
  - onPacketReceived, [134](#)
  - operator int, [135](#)
  - processConnections, [135](#)
  - processDisconnections, [135](#)
  - processEndOfGame, [135](#)
  - processEvents, [135](#)
  - processIncomingPackets, [135](#)
  - setConfig, [135](#)
  - setNetwork, [136](#)
  - setPort, [136](#)
  - setState, [136](#)
  - start, [136](#)
  - stop, [136](#)
- rserve::ServerConfig, [137](#)
- ScoreIntentComponent, [129](#)
- setConfig
  - rserve::Server, [135](#)
- setFocused
  - ui::AFocusableElement, [21](#)
- setNetwork
  - rserve::Server, [136](#)
- setNextNetworkId
  - ecs::EntityFactory, [63](#)
- setPort
  - rserve::Server, [136](#)
- setScale
  - ui::Text, [165](#)
  - ui::UILayout, [180](#)
- setState
  - rserve::Server, [136](#)
- SettingsConfig, [142](#)
- SettingsManager, [143](#)
- Signal, [152](#)
- start

- rserv::Server, 136
- stop
  - rserv::Server, 136
- Symbol
  - DLLoader< T >, 61
- TagRegistry, 163
- ui::AFocusableElement, 19
  - canBeFocused, 20
  - handleInput, 20
  - isFocused, 21
  - onActivated, 21
  - onFocusGained, 21
  - onFocusLost, 21
  - setFocused, 21
- ui::Background, 37
  - render, 38
  - update, 38
- ui::Background::Layer, 91
- ui::Button, 41
  - render, 43
- ui::IFocusable, 79
- ui::LayoutConfig, 92
- ui::Slider, 152
  - handleInput, 155
  - onActivated, 155
  - onNavigateLeft, 156
  - onNavigateRight, 156
  - render, 156
- ui::Text, 163
  - render, 165
  - setScale, 165
  - update, 165
- ui::TextInput, 166
  - handleInput, 169
  - render, 169
  - update, 169
- ui::ToggleSwitch, 171
  - containsPoint, 173
  - handleInput, 173
  - render, 173
- ui::UIElement, 177
- ui::UILayout, 178
  - render, 180
  - setScale, 180
  - update, 180
- ui::UIManager, 181
- ui::UINavigationManager, 182
- update
  - ecs::AIMovementSystem, 28
  - ecs::AIShootingSystem, 30
  - ecs::AnimationRenderingSystem, 33
  - ecs::DeathSystem, 57
  - ecs::GameZoneRenderingSystem, 70
  - ecs::GameZoneViewSystem, 71
  - ecs::HealthBarRenderingSystem, 74
  - ecs::HealthSystem, 75
  - ecs::HitboxRenderingSystem, 77
  - ecs::InputToVelocitySystem, 86
  - ecs::InteractionSystem, 88
  - ecs::LifetimeSystem, 94
  - ecs::MovementInputSystem, 104
  - ecs::MovementSystem, 106
  - ecs::MusicSystem, 109
  - ecs::NetworkInterpolationSystem, 111
  - ecs::OutOfBoundsSystem, 113
  - ecs::ParallaxRenderingSystem, 117
  - ecs::RectangleRenderingSystem, 125
  - ecs::ScoreSystem, 130
  - ecs::ServerMovementInputSystem, 141
  - ecs::ServerShootInputSystem, 142
  - ecs::ShootingSystem, 149
  - ecs::ShootInputSystem, 150
  - ecs::SoundSystem, 157
  - ecs::SpawnSystem, 160
  - ecs::SpriteRenderingSystem, 162
  - ecs::TextRenderingSystem, 170
  - ecs::TriggerSystem, 176
  - gsm::AGameState, 24
  - gsm::AGameStateMachine, 26
  - gsm::BootState, 40
  - gsm::DevState, 59
  - gsm::InGameState, 83
  - gsm::LoadingState, 96
  - gsm::LobbyBrowserState, 97
  - gsm::LobbyState, 99
  - gsm::MainMenuState, 101
  - gsm::PauseState, 121
  - gsm::ResultsState, 128
  - gsm::SettingsState, 146
  - ui::Background, 38
  - ui::Text, 165
  - ui::TextInput, 169
  - ui::UILayout, 180
- updateAllSystems
  - ecs::ASystemManager, 36
- updateSystem
  - ecs::ASystem, 35
- Utils, 182
- what
  - err::AError, 18