

R-Type architecture

Generated by Doxygen 1.13.2

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	7
2.1 Class List	7
3 File Index	13
3.1 File List	13
4 Class Documentation	19
4.1 ecs::AComponent Class Reference	19
4.2 ecs::AComponentArray< T > Class Template Reference	20
4.2.1 Member Function Documentation	20
4.2.1.1 getMaxEntityId()	20
4.2.1.2 removeComponents()	20
4.2.1.3 removeOneComponent()	20
4.3 ActionFactory Class Reference	21
4.4 err::AError Class Reference	21
4.4.1 Member Function Documentation	22
4.4.1.1 getCode()	22
4.4.1.2 getDetails()	22
4.4.1.3 getType()	22
4.4.1.4 what()	22
4.5 ui::AFocusableElement Class Reference	22
4.5.1 Member Function Documentation	24
4.5.1.1 canBeFocused()	24
4.5.1.2 handleInput()	24
4.5.1.3 isFocused()	25
4.5.1.4 onActivated()	25
4.5.1.5 onFocusGained()	25
4.5.1.6 onFocusLost()	25
4.5.1.7 setFocused()	25
4.6 gsm::AGameState Class Reference	26
4.6.1 Member Function Documentation	27
4.6.1.1 addSystem() [1/2]	27
4.6.1.2 addSystem() [2/2]	27
4.6.1.3 enter() [1/2]	27
4.6.1.4 enter() [2/2]	27
4.6.1.5 exit() [1/2]	27
4.6.1.6 exit() [2/2]	27
4.6.1.7 getSystems() [1/2]	28
4.6.1.8 getSystems() [2/2]	28
4.6.1.9 update() [1/2]	28

4.6.1.10 update() [2/2]	28
4.7 gsm::AGameStateMachine Class Reference	28
4.7.1 Member Function Documentation	29
4.7.1.1 changeState() [1/2]	29
4.7.1.2 changeState() [2/2]	29
4.7.1.3 popState() [1/2]	29
4.7.1.4 popState() [2/2]	29
4.7.1.5 pushState() [1/2]	30
4.7.1.6 pushState() [2/2]	30
4.7.1.7 requestStateChange() [1/2]	30
4.7.1.8 requestStateChange() [2/2]	30
4.7.1.9 requestStatePop()	30
4.7.1.10 requestStatePush()	30
4.7.1.11 update() [1/2]	30
4.7.1.12 update() [2/2]	31
4.8 ecs::AnimationClip Struct Reference	31
4.9 ecs::AnimationComponent Class Reference	31
4.10 ecs::AnimationCondition Struct Reference	33
4.11 ecs::AnimationConditionFactory Class Reference	33
4.12 ecs::AnimationRenderingSystem Class Reference	34
4.12.1 Member Function Documentation	34
4.12.1.1 update()	34
4.13 APrefab Class Reference	34
4.13.1 Member Function Documentation	35
4.13.1.1 instantiate() [1/2]	35
4.13.1.2 instantiate() [2/2]	35
4.14 ecs::ASystem Class Reference	35
4.14.1 Member Function Documentation	36
4.14.1.1 updateSystem()	36
4.15 ecs::ASystemManager Class Reference	37
4.15.1 Member Function Documentation	37
4.15.1.1 addSystem()	37
4.15.1.2 removeSystem()	37
4.15.1.3 updateAllSystems()	37
4.16 ui::Background Class Reference	38
4.16.1 Member Function Documentation	39
4.16.1.1 render()	39
4.16.1.2 update()	39
4.17 ecs::BackGroundMusicTag Class Reference	40
4.18 gsm::BootState Class Reference	40
4.18.1 Member Function Documentation	41
4.18.1.1 enter() [1/2]	41

4.18.1.2 enter() [2/2]	41
4.18.1.3 exit()	41
4.18.1.4 update()	41
4.19 ui::Button Class Reference	42
4.19.1 Member Function Documentation	44
4.19.1.1 render()	44
4.20 math::Chrono Class Reference	45
4.21 ecs::ClientEffectCleanupSystem Class Reference	45
4.21.1 Member Function Documentation	46
4.21.1.1 update()	46
4.22 ecs::ClientEffectTag Class Reference	46
4.23 err::ClientError Class Reference	46
4.23.1 Member Function Documentation	47
4.23.1.1 getType()	47
4.24 ClientNetwork Class Reference	48
4.25 err::ClientNetworkError Class Reference	50
4.25.1 Member Function Documentation	51
4.25.1.1 getType()	51
4.26 ecs::ColliderComponent Class Reference	51
4.27 ecs::CollisionRule Struct Reference	52
4.28 ecs::CollisionRules Class Reference	52
4.29 ecs::CollisionRulesData Struct Reference	53
4.30 ecs::CollisionRulesParser Class Reference	53
4.31 gfx::color_t Struct Reference	54
4.32 rserv::ComponentDeltaTracker Class Reference	54
4.33 rserv::ComponentSerializer Class Reference	55
4.34 ComposantParser Class Reference	55
4.35 ecs::CompositeEntityComponent Class Reference	56
4.36 ecs::ControllableTag Class Reference	57
4.37 Core Class Reference	57
4.38 ecs::DamageComponent Class Reference	58
4.39 ecs::DamageIntentComponent Class Reference	58
4.40 DeathAnimationSpawner Class Reference	59
4.41 ecs::DeathIntentComponent Class Reference	59
4.42 ecs::DeathSystem Class Reference	60
4.42.1 Member Function Documentation	60
4.42.1.1 update()	60
4.43 debug::Debug Class Reference	61
4.44 DLLoader< T > Class Template Reference	61
4.44.1 Member Function Documentation	61
4.44.1.1 Close()	61
4.44.1.2 Error()	62

4.44.1.3 getHandler()	62
4.44.1.4 Open()	62
4.44.1.5 Symbol()	62
4.45 ecs::EndOfMapDetectionSystem Class Reference	62
4.45.1 Member Function Documentation	63
4.45.1.1 update()	63
4.46 ecs::EnemyProjectileTag Class Reference	63
4.47 ecs::EntityCreationContext Struct Reference	63
4.48 ecs::EntityFactory Class Reference	64
4.48.1 Member Function Documentation	64
4.48.1.1 createEntity()	64
4.49 EntityParser Class Reference	64
4.50 ecs::EntityPartsComponent Class Reference	65
4.51 EntityPrefabManager Class Reference	65
4.52 rserv::EntitySnapshot Struct Reference	66
4.53 Field Struct Reference	66
4.54 FieldValue Struct Reference	67
4.55 math::FRect Class Reference	67
4.56 gsm::GameEndState Class Reference	68
4.56.1 Member Function Documentation	68
4.56.1.1 enter()	68
4.56.1.2 update()	69
4.57 ecs::GameEndTag Class Reference	69
4.58 gsm::GameStateMachine Class Reference	69
4.58.1 Member Function Documentation	70
4.58.1.1 requestStateChange()	70
4.58.1.2 requestStatePop()	70
4.58.1.3 requestStatePush()	70
4.59 ecs::GameZoneColliderTag Class Reference	71
4.60 ecs::GameZoneComponent Class Reference	71
4.61 ecs::GameZoneRenderingSystem Class Reference	72
4.61.1 Member Function Documentation	72
4.61.1.1 update()	72
4.62 ecs::GameZoneViewSystem Class Reference	72
4.62.1 Member Function Documentation	73
4.62.1.1 update()	73
4.63 ecs::GraphicalInputProvider Class Reference	73
4.63.1 Member Function Documentation	74
4.63.1.1 getActionAxis()	74
4.63.1.2 getAxisValue()	74
4.63.1.3 getInputMapping()	74
4.63.1.4 isActionPressed()	75

4.64	ecs::Group< Components > Class Template Reference	75
4.65	ecs::HealthBarComponent Class Reference	75
4.66	ecs::HealthBarRenderingSystem Class Reference	75
4.66.1	Member Function Documentation	76
4.66.1.1	update()	76
4.67	ecs::HealthComponent Class Reference	76
4.68	ecs::HealthSystem Class Reference	77
4.68.1	Member Function Documentation	77
4.68.1.1	update()	77
4.69	ecs::HitboxRenderComponent Class Reference	78
4.70	ecs::HitboxRenderingSystem Class Reference	78
4.70.1	Member Function Documentation	79
4.70.1.1	update()	79
4.71	gfx::IAudio Class Reference	79
4.72	IBuffer Class Reference	80
4.73	ecs::IComponent Class Reference	80
4.74	ecs::IComponentArray Class Reference	81
4.75	ecs::IEntityFactory Class Reference	81
4.76	err::IError Class Reference	81
4.77	gfx::IEvent Class Reference	82
4.78	net::IEventLoop Class Reference	82
4.79	ui::IFocusable Class Reference	83
4.80	gsm::IGameState Class Reference	83
4.81	gsm::IGameStateMachine Class Reference	84
4.82	ecs::IInputProvider Class Reference	85
4.83	ILoader Class Reference	85
4.84	net::INetwork Class Reference	86
4.85	net::INetworkAddress Class Reference	86
4.86	net::INetworkEndpoint Class Reference	87
4.87	net::INetworkErrorCode Class Reference	87
4.88	net::INetworkFactory Class Reference	87
4.89	net::INetworkResolver Class Reference	88
4.90	net::INetworkSocket Class Reference	88
4.91	gsm::InfiniteState Class Reference	88
4.91.1	Member Function Documentation	89
4.91.1.1	enter()	89
4.91.1.2	exit()	89
4.91.1.3	update()	90
4.92	gsm::InGameState Class Reference	90
4.92.1	Member Function Documentation	91
4.92.1.1	enter() [1/2]	91
4.92.1.2	enter() [2/2]	91

4.92.1.3 exit()	91
4.92.1.4 update() [1/2]	91
4.92.1.5 update() [2/2]	92
4.93 ecs::InputIntentComponent Class Reference	92
4.94 ecs::InputMapping Struct Reference	92
4.95 ecs::InputMappingManager Class Reference	93
4.96 ecs::InputNormalizer Class Reference	93
4.97 ecs::InputToVelocitySystem Class Reference	94
4.97.1 Member Function Documentation	94
4.97.1.1 update()	94
4.98 ecs::IntentToVelocitySystem Class Reference	94
4.98.1 Member Function Documentation	95
4.98.1.1 update()	95
4.99 ecs::InteractionConfigComponent Class Reference	95
4.100 ecs::InteractionMapping Struct Reference	96
4.101 ecs::InteractionSystem Class Reference	96
4.101.1 Member Function Documentation	97
4.101.1.1 update()	97
4.102 pm::IPacketManager Class Reference	97
4.103 IPrefab Class Reference	98
4.104 ecs::ISystem Class Reference	98
4.105 ecs::ISystemManager Class Reference	99
4.106 ecs::View< Components >::Iterator Class Reference	99
4.107 gfx::IWindow Class Reference	100
4.108 ui::Background::Layer Struct Reference	101
4.109 ui::LayoutConfig Struct Reference	101
4.110 err::LibrariesLoadError Class Reference	101
4.110.1 Member Function Documentation	102
4.110.1.1 getType()	102
4.111 ecs::LifetimeComponent Class Reference	102
4.112 ecs::LifetimeSystem Class Reference	103
4.112.1 Member Function Documentation	103
4.112.1.1 update()	103
4.113 gsm::LoadingState Class Reference	104
4.113.1 Member Function Documentation	105
4.113.1.1 enter() [1/2]	105
4.113.1.2 enter() [2/2]	105
4.113.1.3 exit()	105
4.113.1.4 update()	105
4.114 rserv::Lobby Class Reference	105
4.115 gsm::LobbyBrowserState Class Reference	108
4.115.1 Member Function Documentation	108

4.115.1.1 enter()	108
4.115.1.2 exit()	109
4.115.1.3 update()	109
4.116 gsm::LobbyState Class Reference	109
4.116.1 Member Function Documentation	110
4.116.1.1 enter() [1/2]	110
4.116.1.2 enter() [2/2]	110
4.116.1.3 exit()	110
4.116.1.4 update() [1/2]	110
4.116.1.5 update() [2/2]	111
4.117 rserv::LobbyStruct Struct Reference	111
4.118 gsm::LobbyWaitingState Class Reference	111
4.118.1 Member Function Documentation	112
4.118.1.1 enter()	112
4.118.1.2 exit()	113
4.118.1.3 update()	113
4.119 ecs::LocalPlayerTag Class Reference	113
4.120 gsm::MainMenuState Class Reference	113
4.120.1 Member Function Documentation	115
4.120.1.1 enter()	115
4.120.1.2 exit()	115
4.120.1.3 update()	115
4.121 ecs::MapGeneratorSystem Class Reference	115
4.121.1 Member Function Documentation	116
4.121.1.1 update()	116
4.122 MapParser Class Reference	116
4.123 ecs::MobTag Class Reference	117
4.124 MouseClickInfo Struct Reference	117
4.125 MouseInputHandler Class Reference	118
4.126 ecs::MovementInputSystem Class Reference	118
4.126.1 Member Function Documentation	119
4.126.1.1 update()	119
4.127 ecs::MovementIntentComponent Class Reference	119
4.128 ecs::MovementSystem Class Reference	120
4.128.1 Member Function Documentation	121
4.128.1.1 update()	121
4.129 ecs::MultiShotPattern Struct Reference	121
4.130 ecs::MusicComponent Class Reference	121
4.131 ecs::MusicIntentComponent Class Reference	122
4.132 ecs::MusicSystem Class Reference	123
4.132.1 Member Function Documentation	123
4.132.1.1 update()	123

4.133 NetworkEvent Struct Reference	124
4.134 ecs::NetworkHealthState Struct Reference	124
4.135 ecs::NetworkInterpolationSystem Class Reference	124
4.135.1 Member Function Documentation	125
4.135.1.1 update()	125
4.136 ecs::NetworkStateComponent Class Reference	125
4.137 ecs::NetworkTransformState Struct Reference	126
4.138 ecs::ObstacleTag Class Reference	126
4.139 math::OrientedRect Class Reference	127
4.140 ecs::OutOfBoundsSystem Class Reference	127
4.140.1 Member Function Documentation	128
4.140.1.1 update()	128
4.141 ecs::OwnerComponent Class Reference	128
4.142 err::PacketError Class Reference	129
4.142.1 Member Function Documentation	130
4.142.1.1 getType()	130
4.143 ecs::ParallaxComponent Class Reference	130
4.144 ecs::ParallaxLayer Struct Reference	131
4.145 ecs::ParallaxRenderingSystem Class Reference	131
4.145.1 Member Function Documentation	132
4.145.1.1 update()	132
4.146 ParsedEntityPrefab Class Reference	132
4.146.1 Member Function Documentation	133
4.146.1.1 instantiate() [1/2]	133
4.146.1.2 instantiate() [2/2]	133
4.147 Parser Class Reference	133
4.148 err::ParserError Class Reference	134
4.148.1 Member Function Documentation	135
4.148.1.1 getType()	135
4.149 gsm::PauseState Class Reference	135
4.149.1 Member Function Documentation	136
4.149.1.1 enter()	136
4.149.1.2 exit()	136
4.149.1.3 update()	136
4.150 ecs::PlayerProjectileTag Class Reference	137
4.151 ecs::PlayerTag Class Reference	137
4.152 ecs::ProjectilePassThroughTag Class Reference	137
4.153 ecs::ProjectilePrefabComponent Class Reference	138
4.154 ecs::RectangleRenderComponent Class Reference	138
4.155 ecs::RectangleRenderingSystem Class Reference	139
4.155.1 Member Function Documentation	139
4.155.1.1 update()	139

4.156	ecs::Registry Class Reference	140
4.157	ecs::RemappableKeyBinding Struct Reference	140
4.158	ResourceManager Class Reference	141
4.159	gsm::ResultsState Class Reference	141
4.159.1	Member Function Documentation	142
4.159.1.1	enter()	142
4.159.1.2	exit()	142
4.159.1.3	update()	143
4.160	ecs::ScoreComponent Class Reference	143
4.161	gsm::ScoreFeedback Struct Reference	143
4.162	ScoreIntentComponent Class Reference	144
4.163	ecs::ScoreSystem Class Reference	144
4.163.1	Member Function Documentation	144
4.163.1.1	update()	144
4.164	ecs::ScoreValueComponent Class Reference	145
4.165	ecs::ScriptingComponent Class Reference	145
4.166	err::ScriptingError Class Reference	146
4.166.1	Member Function Documentation	147
4.166.1.1	getType()	147
4.167	ecs::ScriptingSystem Class Reference	147
4.167.1	Member Function Documentation	148
4.167.1.1	update()	148
4.168	rserv::Server Class Reference	148
4.169	rserv::ServerConfig Class Reference	149
4.170	err::ServerError Class Reference	150
4.170.1	Member Function Documentation	151
4.170.1.1	getType()	151
4.171	ecs::ServerInputProvider Class Reference	151
4.171.1	Member Function Documentation	152
4.171.1.1	getActionAxis()	152
4.171.1.2	getAxisValue()	152
4.171.1.3	getInputMapping()	153
4.171.1.4	isActionPressed()	153
4.172	ecs::ServerMovementInputSystem Class Reference	153
4.172.1	Member Function Documentation	154
4.172.1.1	update()	154
4.173	ecs::ServerShootInputSystem Class Reference	154
4.173.1	Member Function Documentation	155
4.173.1.1	update()	155
4.174	SettingsConfig Class Reference	155
4.175	SettingsManager Class Reference	156
4.176	gsm::SettingsState Class Reference	157

4.176.1 Member Function Documentation	159
4.176.1.1 enter()	159
4.176.1.2 exit()	159
4.176.1.3 update()	159
4.177 ecs::ShooterTag Class Reference	160
4.178 ecs::ShootingStatsComponent Class Reference	160
4.179 ecs::ShootingSystem Class Reference	161
4.179.1 Member Function Documentation	161
4.179.1.1 update()	161
4.180 ecs::ShootInputSystem Class Reference	162
4.180.1 Member Function Documentation	162
4.180.1.1 update()	162
4.181 ecs::ShootIntentComponent Class Reference	163
4.182 gsm::ShutdownState Class Reference	163
4.182.1 Member Function Documentation	164
4.182.1.1 enter()	164
4.183 Signal Class Reference	164
4.184 ui::Slider Class Reference	165
4.184.1 Member Function Documentation	168
4.184.1.1 handleInput()	168
4.184.1.2 onActivated()	168
4.184.1.3 onNavigateLeft()	168
4.184.1.4 onNavigateRight()	168
4.184.1.5 render()	169
4.185 ecs::SoundIntentComponent Class Reference	169
4.186 ecs::SoundSystem Class Reference	169
4.186.1 Member Function Documentation	170
4.186.1.1 update()	170
4.187 ecs::SpatialGrid Class Reference	170
4.188 ecs::SpawnIntentComponent Class Reference	171
4.189 ecs::SpawnSystem Class Reference	172
4.189.1 Member Function Documentation	172
4.189.1.1 update()	172
4.190 ecs::SpeedComponent Class Reference	172
4.191 ecs::SpriteComponent Class Reference	173
4.192 ecs::SpriteRenderingSystem Class Reference	174
4.192.1 Member Function Documentation	174
4.192.1.1 update()	174
4.193 ecs::SystemManager Class Reference	174
4.194 TagRegistry Class Reference	175
4.195 ui::Text Class Reference	176
4.195.1 Member Function Documentation	177

4.195.1.1 render()	177
4.195.1.2 setScale()	177
4.195.1.3 update()	178
4.196 ecs::TextComponent Class Reference	178
4.197 ui::TextInput Class Reference	179
4.197.1 Member Function Documentation	181
4.197.1.1 handleInput()	181
4.197.1.2 render()	182
4.197.1.3 update()	182
4.198 ecs::TextRenderingSystem Class Reference	182
4.198.1 Member Function Documentation	183
4.198.1.1 update()	183
4.199 ui::ToggleSwitch Class Reference	183
4.199.1 Member Function Documentation	185
4.199.1.1 containsPoint()	185
4.199.1.2 handleInput()	186
4.199.1.3 render()	186
4.200 ecs::TransformComponent Class Reference	186
4.201 ecs::Transition Struct Reference	187
4.202 ecs::TriggerIntentComponent Class Reference	187
4.203 ecs::TriggerSystem Class Reference	188
4.203.1 Member Function Documentation	188
4.203.1.1 update()	188
4.204 ui::UIElement Class Reference	189
4.205 ui::UILayout Class Reference	190
4.205.1 Member Function Documentation	192
4.205.1.1 render()	192
4.205.1.2 setScale()	192
4.205.1.3 update()	192
4.206 ui::UIManager Class Reference	193
4.207 ui::UINavigationController Class Reference	194
4.208 Utils Class Reference	194
4.209 math::Vector2f Class Reference	195
4.210 ecs::VelocityComponent Class Reference	195
4.211 ecs::View< Components > Class Template Reference	196
5 File Documentation	197
5.1 ClientNetwork.hpp	197
5.2 colors.hpp	199
5.3 AnimationComponent.hpp	200
5.4 HealthBarComponent.hpp	202
5.5 HitboxRenderComponent.hpp	202

5.6 MusicComponent.hpp	203
5.7 ParallaxComponent.hpp	204
5.8 RectangleRenderComponent.hpp	205
5.9 SpriteComponent.hpp	205
5.10 TextComponent.hpp	206
5.11 BackGroundMusicTag.hpp	206
5.12 MusicIntentComponent.hpp	207
5.13 SoundIntentComponent.hpp	207
5.14 constants.hpp	208
5.15 constants.hpp	209
5.16 Core.hpp	212
5.17 DeathAnimationSpawner.hpp	213
5.18 AGameStateMachine.hpp	213
5.19 AGameStateMachine.hpp	214
5.20 GameStateMachine.hpp	214
5.21 GameStateMachine.hpp	214
5.22 AGameState.hpp	215
5.23 AGameState.hpp	215
5.24 BootState.hpp	216
5.25 BootState.hpp	216
5.26 InfiniteState.hpp	216
5.27 InGameState.hpp	217
5.28 InGameState.hpp	218
5.29 LoadingState.hpp	218
5.30 LoadingState.hpp	219
5.31 LobbyState.hpp	219
5.32 LobbyState.hpp	219
5.33 LobbyBrowserState.hpp	220
5.34 LobbyWaitingState.hpp	220
5.35 MainMenuState.hpp	221
5.36 PauseState.hpp	222
5.37 ResultsState.hpp	222
5.38 SettingsState.hpp	223
5.39 GraphicalInputProvider.hpp	224
5.40 initResourcesManager.hpp	225
5.41 initResourcesManager.hpp	225
5.42 MouseInputHandler.hpp	226
5.43 NetworkStateComponent.hpp	226
5.44 DefaultPacketHandlers.hpp	227
5.45 DefaultPacketHandlers.hpp	228
5.46 DefaultPacketHandlers.hpp	228
5.47 SettingsConfig.hpp	228

5.48 SettingsManager.hpp	229
5.49 MusicSystem.hpp	230
5.50 SoundSystem.hpp	230
5.51 ClientEffectCleanupSystem.hpp	231
5.52 MovementInputSystem.hpp	231
5.53 ShootInputSystem.hpp	232
5.54 MapGeneratorSystem.hpp	232
5.55 NetworkInterpolationSystem.hpp	233
5.56 AnimationRenderingSystem.hpp	233
5.57 GameZoneRenderingSystem.hpp	233
5.58 GameZoneViewSystem.hpp	234
5.59 HealthBarRenderingSystem.hpp	234
5.60 HitboxRenderingSystem.hpp	235
5.61 ParallaxRenderingSystem.hpp	235
5.62 RectangleRenderingSystem.hpp	236
5.63 SpriteRenderingSystem.hpp	236
5.64 TextRenderingSystem.hpp	236
5.65 AFocusableElement.hpp	237
5.66 IFocusable.hpp	238
5.67 UILayout.hpp	238
5.68 Background.hpp	239
5.69 UIElement.hpp	240
5.70 Button.hpp	241
5.71 Slider.hpp	242
5.72 TextInput.hpp	243
5.73 ToggleSwitch.hpp	244
5.74 Text.hpp	245
5.75 UIManager.hpp	245
5.76 UINavigationManager.hpp	246
5.77 Utils.hpp	247
5.78 Utils.hpp	248
5.79 CollisionRules.hpp	248
5.80 CollisionRulesData.hpp	249
5.81 AComponent.hpp	249
5.82 IComponent.hpp	249
5.83 ColliderComponent.hpp	250
5.84 CompositeEntityComponent.hpp	251
5.85 DamageComponent.hpp	251
5.86 EntityPartsComponent.hpp	251
5.87 GameZoneComponent.hpp	252
5.88 HealthComponent.hpp	252
5.89 InteractionConfigComponent.hpp	253

5.90 LifetimeComponent.hpp	253
5.91 OwnerComponent.hpp	254
5.92 ProjectilePrefabComponent.hpp	254
5.93 ScoreComponent.hpp	255
5.94 ScoreValueComponent.hpp	255
5.95 ScriptingComponent.hpp	256
5.96 ShootingStatsComponent.hpp	257
5.97 SpeedComponent.hpp	257
5.98 TransformComponent.hpp	258
5.99 VelocityComponent.hpp	258
5.100 ClientEffectTag.hpp	259
5.101 ControllableTag.hpp	259
5.102 EnemyProjectileTag.hpp	259
5.103 GameEndTag.hpp	260
5.104 GameZoneColliderTag.hpp	260
5.105 LocalPlayerTag.hpp	261
5.106 MobTag.hpp	261
5.107 ObstacleTag.hpp	261
5.108 PlayerProjectileTag.hpp	262
5.109 PlayerTag.hpp	262
5.110 ProjectilePassThroughTag.hpp	262
5.111 ShooterTag.hpp	263
5.112 DamageIntentComponent.hpp	263
5.113 DeathIntentComponent.hpp	263
5.114 InputIntentComponent.hpp	264
5.115 MovementIntentComponent.hpp	264
5.116 ScoreIntentComponent.hpp	265
5.117 ShootIntentComponent.hpp	265
5.118 SpawnIntentComponent.hpp	266
5.119 TriggerIntentComponent.hpp	266
5.120 debug.hpp	267
5.121 DLLoader.hpp	268
5.122 ILoader.hpp	269
5.123 LoaderType.hpp	269
5.124 AComponentArray.hpp	270
5.125 IComponentArray.hpp	270
5.126 Entity.hpp	271
5.127 EntityCreationContext.hpp	271
5.128 EntityFactory.hpp	272
5.129 IEntityFactory.hpp	272
5.130 Registry.hpp	273
5.131 View.hpp	273

5.132 AError.hpp	274
5.133 ClientError.hpp	275
5.134 ClientNetworkError.hpp	275
5.135 IError.hpp	276
5.136 LibrariesLoadError.hpp	276
5.137 PacketError.hpp	276
5.138 ParserError.hpp	277
5.139 ScriptingError.hpp	277
5.140 ServerError.hpp	278
5.141 IGameState.hpp	278
5.142 IGameStateMachine.hpp	279
5.143 IInputProvider.hpp	279
5.144 InputAction.hpp	280
5.145 InputMapping.hpp	280
5.146 InputMappingManager.hpp	281
5.147 IAudio.hpp	282
5.148 IBuffer.hpp	282
5.149 IEvent.hpp	283
5.150 IEventLoop.hpp	283
5.151 INetwork.hpp	283
5.152 INetworkAddress.hpp	284
5.153 INetworkEndpoint.hpp	285
5.154 INetworkErrorCode.hpp	285
5.155 INetworkFactory.hpp	286
5.156 INetworkResolver.hpp	286
5.157 INetworkSocket.hpp	286
5.158 IPacketManager.hpp	287
5.159 IWindow.hpp	288
5.160 GameStateHandlers.hpp	289
5.161 AnimationConditionFactory.hpp	289
5.162 CollisionRulesParser.hpp	290
5.163 ComposantParser.hpp	290
5.164 EntityParser.hpp	291
5.165 MapParser.hpp	292
5.166 Parser.hpp	292
5.167 ParserParam.hpp	293
5.168 APrefab.hpp	294
5.169 EntityPrefabManager.hpp	295
5.170 IPrefab.hpp	295
5.171 ParsedEntityPrefab.hpp	296
5.172 ResourceManager.hpp	296
5.173 Signal.hpp	297

5.174 SpatialGrid.hpp	297
5.175 ASystem.hpp	298
5.176 ISystem.hpp	299
5.177 OutOfBoundsSystem.hpp	299
5.178 DeathSystem.hpp	300
5.179 HealthSystem.hpp	300
5.180 InputNormalizer.hpp	300
5.181 ActionFactory.hpp	301
5.182 InteractionSystem.hpp	301
5.183 TagRegistry.hpp	302
5.184 TriggerSystem.hpp	303
5.185 LifetimeSystem.hpp	303
5.186 InputToVelocitySystem.hpp	304
5.187 IntentToVelocitySystem.hpp	304
5.188 MovementSystem.hpp	304
5.189 ScoreSystem.hpp	306
5.190 ScriptingSystem.hpp	306
5.191 ShootingSystem.hpp	307
5.192 SpawnSystem.hpp	307
5.193 ASystemManager.hpp	308
5.194 ISystemManager.hpp	308
5.195 SystemManager.hpp	309
5.196 translationToECS.hpp	309
5.197 Chrono.hpp	309
5.198 FRect.hpp	310
5.199 OrientedRect.hpp	311
5.200 Vector2f.hpp	311
5.201 Constants.hpp	312
5.202 ComponentDeltaTracker.hpp	312
5.203 ComponentSerializer.hpp	313
5.204 gsmStates.hpp	313
5.205 GameEndState.hpp	314
5.206 ShutdownState.hpp	314
5.207 ServerInputProvider.hpp	315
5.208 Lobby.hpp	315
5.209 LobbyStruct.hpp	318
5.210 Server.hpp	318
5.211 ServerConfig.hpp	320
5.212 EndOfMapDetectionSystem.hpp	320
5.213 ServerMovementInputSystem.hpp	321
5.214 ServerShootInputSystem.hpp	321

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

ActionFactory	21
ecs::AnimationClip	31
ecs::AnimationCondition	33
ecs::AnimationConditionFactory	33
math::Chrono	45
ClientNetwork	48
ecs::CollisionRule	52
ecs::CollisionRules	52
ecs::CollisionRulesData	53
ecs::CollisionRulesParser	53
gfx::color_t	54
rserve::ComponentDeltaTracker	54
rserve::ComponentSerializer	55
ComposantParser	55
Core	57
DeathAnimationSpawner	59
debug::Debug	61
std::enable_shared_from_this	
ecs::Registry	140
ui::UIElement	189
ui::AFocusableElement	22
ui::Button	42
ui::Slider	165
ui::TextInput	179
ui::ToggleSwitch	183
ui::Background	38
ui::Text	176
ui::UILayout	190
ecs::EntityCreationContext	63
EntityParser	64
EntityPrefabManager	65
rserve::EntitySnapshot	66
std::exception	
err::IError	81
err::AError	21

err::ClientError	46
err::ClientNetworkError	50
err::LibrairiesLoadError	101
err::PacketError	129
err::ParserError	134
err::ScriptingError	146
err::ServerError	150
Field	66
FieldValueVariant	
FieldValue	67
math::FRect	67
ecs::Group< Components >	75
gfx::IAudio	79
IBuffer	80
ecs::IComponent	80
ecs::AComponent	19
ecs::AnimationComponent	31
ecs::BackGroundMusicTag	40
ecs::ClientEffectTag	46
ecs::ColliderComponent	51
ecs::CompositeEntityComponent	56
ecs::ControllableTag	57
ecs::DamageComponent	58
ecs::DamageIntentComponent	58
ecs::DeathIntentComponent	59
ecs::EnemyProjectileTag	63
ecs::EntityPartsComponent	65
ecs::GameEndTag	69
ecs::GameZoneColliderTag	71
ecs::GameZoneComponent	71
ecs::HealthBarComponent	75
ecs::HealthComponent	76
ecs::HitboxRenderComponent	78
ecs::InputIntentComponent	92
ecs::InteractionConfigComponent	95
ecs::LifetimeComponent	102
ecs::LocalPlayerTag	113
ecs::MobTag	117
ecs::MovementIntentComponent	119
ecs::MusicComponent	121
ecs::MusicIntentComponent	122
ecs::NetworkStateComponent	125
ecs::ObstacleTag	126
ecs::OwnerComponent	128
ecs::ParallaxComponent	130
ecs::PlayerProjectileTag	137
ecs::PlayerTag	137
ecs::ProjectilePassThroughTag	137
ecs::ProjectilePrefabComponent	138
ecs::RectangleRenderComponent	138
ecs::ScoreComponent	143
ecs::ScoreValueComponent	145
ecs::ScriptingComponent	145
ecs::ShootIntentComponent	163
ecs::ShooterTag	160
ecs::ShootingStatsComponent	160
ecs::SoundIntentComponent	169
ecs::SpawnIntentComponent	171

ecs::SpeedComponent	172
ecs::SpriteComponent	173
ecs::TextComponent	178
ecs::TransformComponent	186
ecs::TriggerIntentComponent	187
ecs::VelocityComponent	195
ecs::IComponentArray	81
ecs::AComponentArray< T >	20
ecs::IEntityFactory	81
ecs::EntityFactory	64
gfx::IEvent	82
net::IEventLoop	82
ui::IFocusable	83
ui::AFocusableElement	22
gsm::IGameState	83
gsm::AGameState	26
gsm::BootState	40
gsm::BootState	40
gsm::GameEndState	68
gsm::InGameState	90
gsm::InGameState	90
gsm::InfiniteState	88
gsm::LoadingState	104
gsm::LoadingState	104
gsm::LobbyBrowserState	108
gsm::LobbyState	109
gsm::LobbyState	109
gsm::LobbyWaitingState	111
gsm::MainMenuState	113
gsm::PauseState	135
gsm::ResultsState	141
gsm::SettingsState	157
gsm::ShutdownState	163
gsm::AGameState	26
gsm::IGameStateMachine	84
gsm::AGameStateMachine	28
gsm::GameStateMachine	69
gsm::GameStateMachine	69
gsm::AGameStateMachine	28
ecs::IInputProvider	85
ecs::GraphicalInputProvider	73
ecs::ServerInputProvider	151
ILoader	85
DLLoader< createNetworkLib_t >	61
DLLoader< createBuffer_t >	61
DLLoader< createPacket_t >	61
DLLoader< T >	61
net::INetwork	86
net::INetworkAddress	86
net::INetworkEndpoint	87
net::INetworkErrorCode	87
net::INetworkFactory	87
net::INetworkResolver	88
net::INetworkSocket	88
ecs::InputMapping	92
ecs::InputMappingManager	93

ecs::InputNormalizer	93
ecs::InteractionMapping	96
pm::IPacketManager	97
IPrefab	98
APrefab	34
ParsedEntityPrefab	132
ecs::ISystem	98
ecs::ASystem	35
ecs::AnimationRenderingSystem	34
ecs::ClientEffectCleanupSystem	45
ecs::DeathSystem	60
ecs::EndOfMapDetectionSystem	62
ecs::GameZoneRenderingSystem	72
ecs::GameZoneViewSystem	72
ecs::HealthBarRenderingSystem	75
ecs::HealthSystem	77
ecs::HitboxRenderingSystem	78
ecs::InputToVelocitySystem	94
ecs::IntentToVelocitySystem	94
ecs::InteractionSystem	96
ecs::LifetimeSystem	103
ecs::MapGeneratorSystem	115
ecs::MovementInputSystem	118
ecs::MovementSystem	120
ecs::MusicSystem	123
ecs::NetworkInterpolationSystem	124
ecs::OutOfBoundsSystem	127
ecs::ParallaxRenderingSystem	131
ecs::RectangleRenderingSystem	139
ecs::ScoreSystem	144
ecs::ScriptingSystem	147
ecs::ServerMovementInputSystem	153
ecs::ServerShootInputSystem	154
ecs::ShootInputSystem	162
ecs::ShootingSystem	161
ecs::SoundSystem	169
ecs::SpawnSystem	172
ecs::SpriteRenderingSystem	174
ecs::TextRenderingSystem	182
ecs::TriggerSystem	188
ecs::ISystemManager	99
ecs::ASystemManager	37
ecs::SystemManager	174
ecs::View< Components >::Iterator	99
gfx::IWindow	100
ui::Background::Layer	101
ui::LayoutConfig	101
rserv::Lobby	105
rserv::LobbyStruct	111
MapParser	116
MouseClickedInfo	117
MouseInputHandler	118
ecs::MultiShotPattern	121
NetworkEvent	124
ecs::NetworkHealthState	124
ecs::NetworkTransformState	126
math::OrientedRect	127

ecs::ParallaxLayer	131
Parser	133
ecs::RemappableKeyBinding	140
ResourceManager	141
gsm::ScoreFeedback	143
ScoreIntentComponent	144
rserve::Server	148
rserve::ServerConfig	149
SettingsConfig	155
SettingsManager	156
Signal	164
ecs::SpatialGrid	170
TagRegistry	175
ecs::Transition	187
ui::UIManager	193
ui::UINavigationController	194
Utils	194
math::Vector2f	195
ecs::View< Components >	196

Chapter 2

Class Index

2.1 Class List

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

ecs::AComponent	19
ecs::AComponentArray< T >	20
ActionFactory	21
err::AError	21
ui::AFocusableElement	22
gsm::AGameState	26
gsm::AGameStateMachine	28
ecs::AnimationClip	31
ecs::AnimationComponent	31
ecs::AnimationCondition	33
ecs::AnimationConditionFactory	33
ecs::AnimationRenderingSystem	34
APrefab	34
ecs::ASystem	35
ecs::ASystemManager	37
ui::Background	38
ecs::BackGroundMusicTag	40
gsm::BootState	40
ui::Button	42
math::Chrono	45
ecs::ClientEffectCleanupSystem	45
ecs::ClientEffectTag	46
err::ClientError	46
ClientNetwork	48
err::ClientNetworkError	50
ecs::ColliderComponent	51
ecs::CollisionRule	52
ecs::CollisionRules	52
ecs::CollisionRulesData	53
ecs::CollisionRulesParser	53
gfx::color_t	54
rserv::ComponentDeltaTracker	54
rserv::ComponentSerializer	55
ComposantParser	55
ecs::CompositeEntityComponent	56

ecs::ControllableTag	57
Core	57
ecs::DamageComponent	58
ecs::DamageIntentComponent	58
DeathAnimationSpawner	59
ecs::DeathIntentComponent	59
ecs::DeathSystem	60
debug::Debug	61
DLLoader< T >	61
ecs::EndOfMapDetectionSystem	62
ecs::EnemyProjectileTag	63
ecs::EntityCreationContext	63
ecs::EntityFactory	64
EntityParser	64
ecs::EntityPartsComponent	65
EntityPrefabManager	65
rserv::EntitySnapshot	66
Field	66
FieldValue	67
math::FRect	67
gsm::GameEndState	68
ecs::GameEndTag	69
gsm::GameStateMachine	69
ecs::GameZoneColliderTag	71
ecs::GameZoneComponent	71
ecs::GameZoneRenderingSystem	72
ecs::GameZoneViewSystem	72
ecs::GraphicalInputProvider	73
ecs::Group< Components >	75
ecs::HealthBarComponent	75
ecs::HealthBarRenderingSystem	75
ecs::HealthComponent	76
ecs::HealthSystem	77
ecs::HitboxRenderComponent	78
ecs::HitboxRenderingSystem	78
gfx::IAudio	79
IBuffer	80
ecs::IComponent	80
ecs::IComponentArray	81
ecs::IEntityFactory	81
err::IError	81
gfx::IEvent	82
net::IEventLoop	82
ui::IFocusable	83
gsm::IGameState	83
gsm::IGameStateMachine	84
ecs::IInputProvider	85
ILoader	85
net::INetwork	86
net::INetworkAddress	86
net::INetworkEndpoint	87
net::INetworkErrorCode	87
net::INetworkFactory	87
net::INetworkResolver	88
net::INetworkSocket	88
gsm::InfiniteState	88
gsm::InGameState	90
ecs::InputIntentComponent	92

ecs::InputMapping	92
ecs::InputMappingManager	93
ecs::InputNormalizer	93
ecs::InputToVelocitySystem	94
ecs::IntentToVelocitySystem	94
ecs::InteractionConfigComponent	95
ecs::InteractionMapping	96
ecs::InteractionSystem	96
pm::IPacketManager	97
IPrefab	98
ecs::ISystem	98
ecs::ISystemManager	99
ecs::View< Components >::Iterator	99
gfx::IWindow	100
ui::Background::Layer	101
ui::LayoutConfig	101
err::LibrariesLoadError	101
ecs::LifetimeComponent	102
ecs::LifetimeSystem	103
gsm::LoadingState	104
rserv::Lobby	105
gsm::LobbyBrowserState	108
gsm::LobbyState	109
rserv::LobbyStruct	111
gsm::LobbyWaitingState	111
ecs::LocalPlayerTag	113
gsm::MainMenuState	113
ecs::MapGeneratorSystem	115
MapParser	116
ecs::MobTag	117
MouseClickedInfo	117
MouseInputHandler	118
ecs::MovementInputSystem	118
ecs::MovementIntentComponent	119
ecs::MovementSystem	120
ecs::MultiShotPattern	121
ecs::MusicComponent	121
ecs::MusicIntentComponent	122
ecs::MusicSystem	123
NetworkEvent	124
ecs::NetworkHealthState	124
ecs::NetworkInterpolationSystem	124
ecs::NetworkStateComponent	125
ecs::NetworkTransformState	126
ecs::ObstacleTag	126
math::OrientedRect	127
ecs::OutOfBoundsSystem	127
ecs::OwnerComponent	128
err::PacketError	129
ecs::ParallaxComponent	130
ecs::ParallaxLayer	131
ecs::ParallaxRenderingSystem	131
ParsedEntityPrefab	132
Parser	133
err::ParserError	134
gsm::PauseState	135
ecs::PlayerProjectileTag	137
ecs::PlayerTag	137

ecs::ProjectilePassThroughTag	137
ecs::ProjectilePrefabComponent	138
ecs::RectangleRenderComponent	138
ecs::RectangleRenderingSystem	139
ecs::Registry	140
ecs::RemappableKeyBinding	140
ResourceManager	141
gsm::ResultsState	141
ecs::ScoreComponent	143
gsm::ScoreFeedback	143
ScoreIntentComponent	144
ecs::ScoreSystem	144
ecs::ScoreValueComponent	145
ecs::ScriptingComponent	145
err::ScriptingError	146
ecs::ScriptingSystem	147
rserve::Server	148
rserve::ServerConfig	149
err::ServerError	150
ecs::ServerInputProvider	151
ecs::ServerMovementInputSystem	153
ecs::ServerShootInputSystem	154
SettingsConfig	155
SettingsManager	156
gsm::SettingsState	157
ecs::ShooterTag	160
ecs::ShootingStatsComponent	160
ecs::ShootingSystem	161
ecs::ShootInputSystem	162
ecs::ShootIntentComponent	163
gsm::ShutdownState	163
Signal	164
ui::Slider	165
ecs::SoundIntentComponent	169
ecs::SoundSystem	169
ecs::SpatialGrid	170
ecs::SpawnIntentComponent	171
ecs::SpawnSystem	172
ecs::SpeedComponent	172
ecs::SpriteComponent	173
ecs::SpriteRenderingSystem	174
ecs::SystemManager	174
TagRegistry	175
ui::Text	176
ecs::TextComponent	178
ui::TextInput	179
ecs::TextRenderingSystem	182
ui::ToggleSwitch	183
ecs::TransformComponent	186
ecs::Transition	187
ecs::TriggerIntentComponent	187
ecs::TriggerSystem	188
ui::UIElement	189
ui::UILayout	190
ui::UIManager	193
ui::UINavigationManager	194
Utils	194
math::Vector2f	195

ecs::VelocityComponent	195
ecs::View< Components >	196

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/ryanR-type/client/ClientNetwork.hpp	197
/home/albane/epitech/tech3/r-type/ryanR-type/client/colors.hpp	199
/home/albane/epitech/tech3/r-type/ryanR-type/client/constants.hpp	208
/home/albane/epitech/tech3/r-type/ryanR-type/client/Core.hpp	212
/home/albane/epitech/tech3/r-type/ryanR-type/client/DeathAnimationSpawner.hpp	213
/home/albane/epitech/tech3/r-type/ryanR-type/client/SettingsConfig.hpp	228
/home/albane/epitech/tech3/r-type/ryanR-type/client/SettingsManager.hpp	229
/home/albane/epitech/tech3/r-type/ryanR-type/client/Utils.hpp	247
/home/albane/epitech/tech3/r-type/ryanR-type/client/components/rendering/AnimationComponent.hpp .	200
/home/albane/epitech/tech3/r-type/ryanR-type/client/components/rendering/HealthBarComponent.hpp .	202
/home/albane/epitech/tech3/r-type/ryanR-type/client/components/rendering/HitboxRenderComponent.hpp	
202	
/home/albane/epitech/tech3/r-type/ryanR-type/client/components/rendering/MusicComponent.hpp . . .	203
/home/albane/epitech/tech3/r-type/ryanR-type/client/components/rendering/ParallaxComponent.hpp . .	204
/home/albane/epitech/tech3/r-type/ryanR-type/client/components/rendering/RectangleRenderComponent.hpp	
205	
/home/albane/epitech/tech3/r-type/ryanR-type/client/components/rendering/SpriteComponent.hpp . . .	205
/home/albane/epitech/tech3/r-type/ryanR-type/client/components/rendering/TextComponent.hpp	206
/home/albane/epitech/tech3/r-type/ryanR-type/client/components/tags/BackGroundMusicTag.hpp	206
/home/albane/epitech/tech3/r-type/ryanR-type/client/components/temporary/MusicIntentComponent.hpp	207
/home/albane/epitech/tech3/r-type/ryanR-type/client/components/temporary/SoundIntentComponent.hpp	207
/home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/machine/AGameStateMachine.hpp	213
/home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/machine/GameStateMachine.hpp	214
/home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/base/AGameState.hpp	215
/home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/Boot/BootState.hpp	216
/home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/Infinite/InfiniteState.hpp	216
/home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/InGame/InGameState.hpp . . .	217
/home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/Loading/LoadingState.hpp . . .	218
/home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/Lobby/LobbyState.hpp	219
/home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/LobbyBrowser/LobbyBrowserState.hpp	
220	
/home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/LobbyWaiting/LobbyWaitingState.hpp	
220	
/home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/MainMenu/MainMenuState.hpp	221
/home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/Pause/PauseState.hpp	222

/home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/Results/ResultsState.hpp	222
/home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/Settings/SettingsState.hpp	223
/home/albane/epitech/tech3/r-type/ryanR-type/client/initResourcesManager/GraphicalInputProvider.hpp	224
/home/albane/epitech/tech3/r-type/ryanR-type/client/initResourcesManager/initResourcesManager.hpp	225
/home/albane/epitech/tech3/r-type/ryanR-type/client/input/MouseInputHandler.hpp	226
/home/albane/epitech/tech3/r-type/ryanR-type/client/interpolation/NetworkStateComponent.hpp	226
/home/albane/epitech/tech3/r-type/ryanR-type/client/packet/DefaultPacketHandlers.hpp	227
/home/albane/epitech/tech3/r-type/ryanR-type/client/systems/audio/MusicSystem.hpp	230
/home/albane/epitech/tech3/r-type/ryanR-type/client/systems/audio/SoundSystem.hpp	230
/home/albane/epitech/tech3/r-type/ryanR-type/client/systems/effects/ClientEffectCleanupSystem.hpp	231
/home/albane/epitech/tech3/r-type/ryanR-type/client/systems/input/MovementInputSystem.hpp	231
/home/albane/epitech/tech3/r-type/ryanR-type/client/systems/input/ShootInputSystem.hpp	232
/home/albane/epitech/tech3/r-type/ryanR-type/client/systems/map/MapGeneratorSystem.hpp	232
/home/albane/epitech/tech3/r-type/ryanR-type/client/systems/network/NetworkInterpolationSystem.hpp	233
/home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/AnimationRenderingSystem.hpp	233
/home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/GameZoneRenderingSystem.hpp	233
233	
/home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/GameZoneViewSystem.hpp	234
/home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/HealthBarRenderingSystem.hpp	234
/home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/HitboxRenderingSystem.hpp	235
/home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/ParallaxRenderingSystem.hpp	235
/home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/RectangleRenderingSystem.hpp	236
/home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/SpriteRenderingSystem.hpp	236
/home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/TextRenderingSystem.hpp	236
/home/albane/epitech/tech3/r-type/ryanR-type/client/ui/core/AFocusableElement.hpp	237
/home/albane/epitech/tech3/r-type/ryanR-type/client/ui/core/IFocusable.hpp	238
/home/albane/epitech/tech3/r-type/ryanR-type/client/ui/core/UILayout.hpp	238
/home/albane/epitech/tech3/r-type/ryanR-type/client/ui/elements/Background.hpp	239
/home/albane/epitech/tech3/r-type/ryanR-type/client/ui/elements/Text.hpp	245
/home/albane/epitech/tech3/r-type/ryanR-type/client/ui/elements/base/UIElement.hpp	240
/home/albane/epitech/tech3/r-type/ryanR-type/client/ui/elements/focusable/Button.hpp	241
/home/albane/epitech/tech3/r-type/ryanR-type/client/ui/elements/focusable/Slider.hpp	242
/home/albane/epitech/tech3/r-type/ryanR-type/client/ui/elements/focusable/TextInput.hpp	243
/home/albane/epitech/tech3/r-type/ryanR-type/client/ui/elements/focusable/ToggleSwitch.hpp	244
/home/albane/epitech/tech3/r-type/ryanR-type/client/ui/manager/UIManager.hpp	245
/home/albane/epitech/tech3/r-type/ryanR-type/client/ui/navigation/UINavigationManager.hpp	246
/home/albane/epitech/tech3/r-type/ryanR-type/common/constants.hpp	209
/home/albane/epitech/tech3/r-type/ryanR-type/common/debug.hpp	267
/home/albane/epitech/tech3/r-type/ryanR-type/common/translationToECS.hpp	309
/home/albane/epitech/tech3/r-type/ryanR-type/common/CollisionRules/CollisionRules.hpp	248
/home/albane/epitech/tech3/r-type/ryanR-type/common/CollisionRules/CollisionRulesData.hpp	249
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/base/AComponent.hpp	249
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/base/IComponent.hpp	249
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/ColliderComponent.hpp	250
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/CompositeEntityComponent.hpp	251
251	
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/DamageComponent.hpp	251
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/EntityPartsComponent.hpp	251
251	
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/GameZoneComponent.hpp	252
252	
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/HealthComponent.hpp	252
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/InteractionConfigComponent.hpp	253
253	
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/LifetimeComponent.hpp	253
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/OwnerComponent.hpp	254
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/ProjectilePrefabComponent.hpp	254
254	

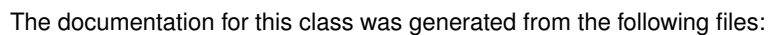
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/ScoreComponent.hpp	255
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/ScoreValueComponent.hpp	255
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/ScriptingComponent.hpp	256
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/ShootingStatsComponent.hpp	257
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/SpeedComponent.hpp	257
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/TransformComponent.hpp	258
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/VelocityComponent.hpp	258
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/tags/ClientEffectTag.hpp	259
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/tags/ControllableTag.hpp	259
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/tags/EnemyProjectileTag.hpp	259
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/tags/GameEndTag.hpp	260
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/tags/GameZoneColliderTag.hpp	260
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/tags/LocalPlayerTag.hpp	261
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/tags/MobTag.hpp	261
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/tags/ObstacleTag.hpp	261
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/tags/PlayerProjectileTag.hpp	262
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/tags/PlayerTag.hpp	262
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/tags/ProjectilePassThroughTag.hpp	262
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/tags/ShooterTag.hpp	263
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/temporary/DamageIntentComponent.hpp	263
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/temporary/DeathIntentComponent.hpp	263
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/temporary/InputIntentComponent.hpp	264
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/temporary/MovementIntentComponent.hpp	264
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/temporary/ScoreIntentComponent.hpp	265
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/temporary/ShootIntentComponent.hpp	265
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/temporary/SpawnIntentComponent.hpp	266
/home/albane/epitech/tech3/r-type/ryanR-type/common/components/temporary/TriggerIntentComponent.hpp	266
/home/albane/epitech/tech3/r-type/ryanR-type/common/DLLoader/DLLoader.hpp	268
/home/albane/epitech/tech3/r-type/ryanR-type/common/DLLoader/ILoader.hpp	269
/home/albane/epitech/tech3/r-type/ryanR-type/common/DLLoader/LoaderType.hpp	269
/home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/entity/Entity.hpp	271
/home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/entity/EntityCreationContext.hpp	271
/home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/entity/componentArray/AComponentArray.hpp	270
/home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/entity/componentArray/IComponentArray.hpp	270
/home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/entity/factory/EntityFactory.hpp	272
/home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/entity/factory/IEntityFactory.hpp	272
/home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/entity/registry/Registry.hpp	273
/home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/view/View.hpp	273
/home/albane/epitech/tech3/r-type/ryanR-type/common/Error/AError.hpp	274
/home/albane/epitech/tech3/r-type/ryanR-type/common/Error/ClientError.hpp	275
/home/albane/epitech/tech3/r-type/ryanR-type/common/Error/ClientNetworkError.hpp	275
/home/albane/epitech/tech3/r-type/ryanR-type/common/Error/IError.hpp	276
/home/albane/epitech/tech3/r-type/ryanR-type/common/Error/LibrariesLoadError.hpp	276
/home/albane/epitech/tech3/r-type/ryanR-type/common/Error/PacketError.hpp	276

/home/albane/epitech/tech3/r-type/ryanR-type/common/Error/ParserError.hpp	277
/home/albane/epitech/tech3/r-type/ryanR-type/common/Error/ScriptingError.hpp	277
/home/albane/epitech/tech3/r-type/ryanR-type/common/Error/ServerError.hpp	278
/home/albane/epitech/tech3/r-type/ryanR-type/common/gsm/IGameState.hpp	278
/home/albane/epitech/tech3/r-type/ryanR-type/common/gsm/IGameStateMachine.hpp	279
/home/albane/epitech/tech3/r-type/ryanR-type/common/InputMapping/IInputProvider.hpp	279
/home/albane/epitech/tech3/r-type/ryanR-type/common/InputMapping/InputAction.hpp	280
/home/albane/epitech/tech3/r-type/ryanR-type/common/InputMapping/InputMapping.hpp	280
/home/albane/epitech/tech3/r-type/ryanR-type/common/InputMapping/InputMappingManager.hpp	281
/home/albane/epitech/tech3/r-type/ryanR-type/common/interfaces/IAudio.hpp	282
/home/albane/epitech/tech3/r-type/ryanR-type/common/interfaces/IBuffer.hpp	282
/home/albane/epitech/tech3/r-type/ryanR-type/common/interfaces/IEvent.hpp	283
/home/albane/epitech/tech3/r-type/ryanR-type/common/interfaces/IEventLoop.hpp	283
/home/albane/epitech/tech3/r-type/ryanR-type/common/interfaces/INetwork.hpp	283
/home/albane/epitech/tech3/r-type/ryanR-type/common/interfaces/INetworkAddress.hpp	284
/home/albane/epitech/tech3/r-type/ryanR-type/common/interfaces/INetworkEndpoint.hpp	285
/home/albane/epitech/tech3/r-type/ryanR-type/common/interfaces/INetworkErrorCode.hpp	285
/home/albane/epitech/tech3/r-type/ryanR-type/common/interfaces/INetworkFactory.hpp	286
/home/albane/epitech/tech3/r-type/ryanR-type/common/interfaces/INetworkResolver.hpp	286
/home/albane/epitech/tech3/r-type/ryanR-type/common/interfaces/INetworkSocket.hpp	286
/home/albane/epitech/tech3/r-type/ryanR-type/common/interfaces/IPacketManager.hpp	287
/home/albane/epitech/tech3/r-type/ryanR-type/common/interfaces/IWindow.hpp	288
/home/albane/epitech/tech3/r-type/ryanR-type/common/packet/DefaultPacketHandlers.hpp	228
/home/albane/epitech/tech3/r-type/ryanR-type/common/packet/GameStateHandlers.hpp	289
/home/albane/epitech/tech3/r-type/ryanR-type/common/Parser/CollisionRulesParser.hpp	290
/home/albane/epitech/tech3/r-type/ryanR-type/common/Parser/Parser.hpp	292
/home/albane/epitech/tech3/r-type/ryanR-type/common/Parser/ParserParam.hpp	293
/home/albane/epitech/tech3/r-type/ryanR-type/common/Parser/Animation/AnimationConditionFactory.hpp	289
/home/albane/epitech/tech3/r-type/ryanR-type/common/Parser/ComposantParser/ComposantParser.hpp	290
/home/albane/epitech/tech3/r-type/ryanR-type/common/Parser/EntityParser/EntityParser.hpp	291
/home/albane/epitech/tech3/r-type/ryanR-type/common/Parser/MapParser/MapParser.hpp	292
/home/albane/epitech/tech3/r-type/ryanR-type/common/Prefab/APrefab.hpp	294
/home/albane/epitech/tech3/r-type/ryanR-type/common/Prefab/IPrefab.hpp	295
/home/albane/epitech/tech3/r-type/ryanR-type/common/Prefab/ParsedEntityPrefab.hpp	296
/home/albane/epitech/tech3/r-type/ryanR-type/common/Prefab/entityPrefabManager/EntityPrefabManager.hpp	295
/home/albane/epitech/tech3/r-type/ryanR-type/common/resourceManager/ResourceManager.hpp	296
/home/albane/epitech/tech3/r-type/ryanR-type/common/Signal/Signal.hpp	297
/home/albane/epitech/tech3/r-type/ryanR-type/common/SpatialGrid/SpatialGrid.hpp	297
/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/base/ASystem.hpp	298
/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/base/ISystem.hpp	299
/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/bounds/OutOfBoundsSystem.hpp	299
/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/death/DeathSystem.hpp	300
/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/health/HealthSystem.hpp	300
/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/input/InputNormalizer.hpp	300
/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/interactions/ActionFactory.hpp	301
/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/interactions/InteractionSystem.hpp	301
/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/interactions/TagRegistry.hpp	302
/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/interactions/TriggerSystem.hpp	303
/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/lifetime/LifetimeSystem.hpp	303
/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/movement/InputToVelocitySystem.hpp	304
/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/movement/IntentToVelocitySystem.hpp	304
/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/movement/MovementSystem.hpp	304
/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/score/ScoreSystem.hpp	306
/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/scripting/ScriptingSystem.hpp	306
/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/shooting/ShootingSystem.hpp	307
/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/spawn/SpawnSystem.hpp	307
/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/systemManager/ASystemManager.hpp	308

/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/systemManager/ ISystemManager.hpp	308
/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/systemManager/ SystemManager.hpp	309
/home/albane/epitech/tech3/r-type/ryanR-type/common/types/ Chrono.hpp	309
/home/albane/epitech/tech3/r-type/ryanR-type/common/types/ FRect.hpp	310
/home/albane/epitech/tech3/r-type/ryanR-type/common/types/ OrientedRect.hpp	311
/home/albane/epitech/tech3/r-type/ryanR-type/common/types/ Vector2f.hpp	311
/home/albane/epitech/tech3/r-type/ryanR-type/server/ Constants.hpp	312
/home/albane/epitech/tech3/r-type/ryanR-type/server/ Lobby.hpp	315
/home/albane/epitech/tech3/r-type/ryanR-type/server/ LobbyStruct.hpp	318
/home/albane/epitech/tech3/r-type/ryanR-type/server/ Server.hpp	318
/home/albane/epitech/tech3/r-type/ryanR-type/server/ ServerConfig.hpp	320
/home/albane/epitech/tech3/r-type/ryanR-type/server/ Utils.hpp	248
/home/albane/epitech/tech3/r-type/ryanR-type/server/deltaTracker/ ComponentDeltaTracker.hpp	312
/home/albane/epitech/tech3/r-type/ryanR-type/server/deltaTracker/ ComponentSerializer.hpp	313
/home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/ gsmStates.hpp	313
/home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/machine/ AGameStateMachine.hpp	214
/home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/machine/ GameStateMachine.hpp	214
/home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/ AGameState.hpp	215
/home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/scenes/Boot/ BootState.hpp	216
/home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/scenes/GameEnd/ GameEndState.hpp	314
/home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/scenes/InGame/ InGameState.hpp	218
/home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/scenes/Loading/ LoadingState.hpp	219
/home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/scenes/Lobby/ LobbyState.hpp	219
/home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/scenes/Shutdown/ ShutdownState.hpp	314
/home/albane/epitech/tech3/r-type/ryanR-type/server/initResourcesManager/ initResourcesManager.hpp	225
/home/albane/epitech/tech3/r-type/ryanR-type/server/initResourcesManager/ ServerInputProvider.hpp	315
/home/albane/epitech/tech3/r-type/ryanR-type/server/packet/ DefaultPacketHandlers.hpp	228
/home/albane/epitech/tech3/r-type/ryanR-type/server/systems/gameEnd/ EndOfMapDetectionSystem.hpp	320
/home/albane/epitech/tech3/r-type/ryanR-type/server/systems/input/ ServerMovementInputSystem.hpp	321
/home/albane/epitech/tech3/r-type/ryanR-type/server/systems/input/ ServerShootInputSystem.hpp	321

Class Documentation

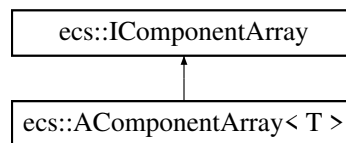
Inheritance diagram for `ecs::AComponent`:



- 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/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

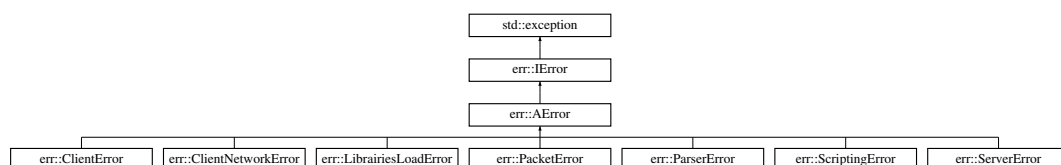
- std::unordered_map< std::string, ActionFunction > **_actions**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/systems/interactions/ActionFactory.hpp
- /home/albane/epitech/tech3/r-type/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]
```

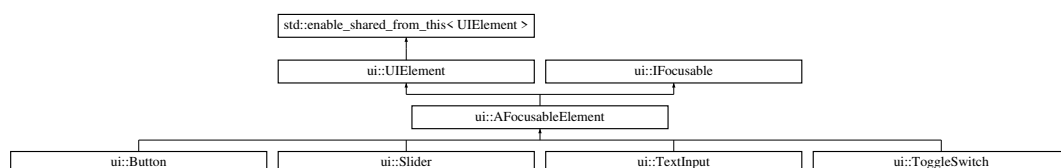
Implements [err::IError](#).

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

- `/home/albane/epitech/tech3/r-type/ryanR-type/common/Error/AError.hpp`
- `/home/albane/epitech/tech3/r-type/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]
```

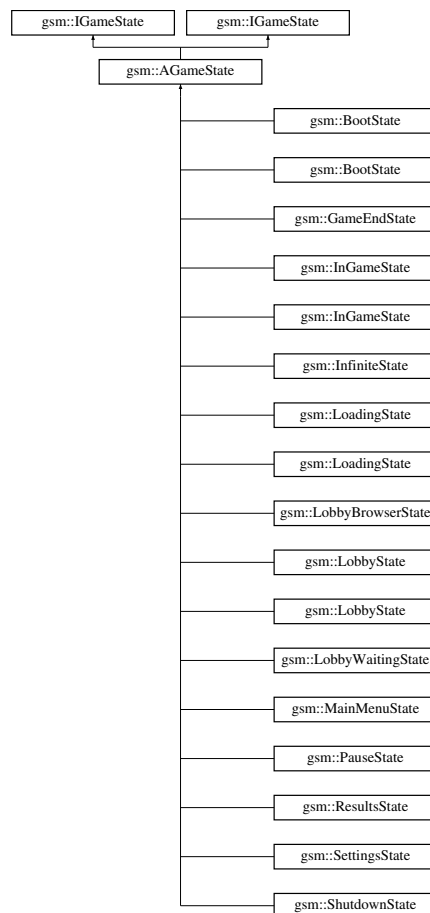
Implements [ui::IFocusable](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/ui/core/AFocusableElement.hpp
- /home/albane/epitech/tech3/r-type/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]
```

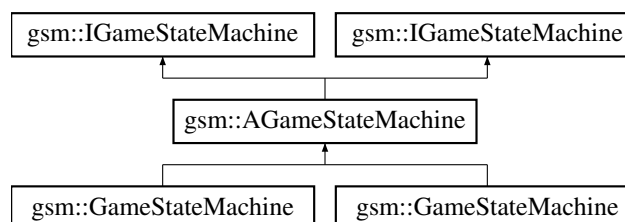
Implements [gsm::IGameState](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/base/AGameState.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/AGameState.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/base/AGameState.cpp
- /home/albane/epitech/tech3/r-type/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
- void [requestStateChange](#) (std::shared_ptr< [IGameState](#) > newState) 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() [1/2]

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

Implements [gsm::IGameStateMachine](#).

4.7.1.8 requestStateChange() [2/2]

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

Implements [gsm::IGameStateMachine](#).

4.7.1.9 requestStatePop()

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

Implements [gsm::IGameStateMachine](#).

4.7.1.10 requestStatePush()

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

Implements [gsm::IGameStateMachine](#).

4.7.1.11 update() [1/2]

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

Implements [gsm::IGameStateMachine](#).

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

4.8 ecs::AnimationClip Struct Reference

Public Attributes

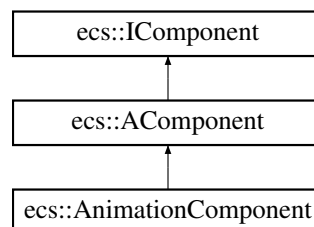
- std::string **texturePath**
- float **frameWidth**
- float **frameHeight**
- int **frameCount**
- float **startWidth**
- float **startHeight**
- float **speed**
- bool **loop**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/components/rendering/AnimationComponent.hpp

4.9 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
- std::unordered_map< std::string, std::shared_ptr< [AnimationClip](#) > > **getStates** () 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/ryanR-type/client/components/rendering/AnimationComponent.hpp

4.10 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/ryanR-type/client/components/rendering/AnimationComponent.hpp

4.11 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 ¶m, std::shared_ptr< [Registry](#) > registry, Entity entity)

Private Member Functions

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

Private Attributes

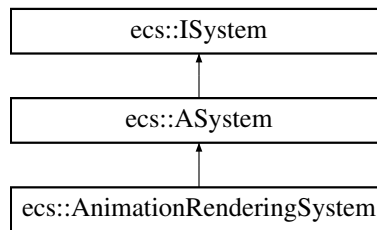
- std::unordered_map< std::string, ConditionFunction > **_conditions**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/Parser/Animation/AnimationConditionFactory.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/Parser/Animation/AnimationConditionFactory.cpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/Parser/Animation/AnimationConditionsRegistry.cpp

4.12 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

4.12.1 Member Function Documentation

4.12.1.1 update()

```

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

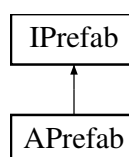
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/AnimationRenderingSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/AnimationRenderingSystem.cpp

4.13 APrefab Class Reference

Inheritance diagram for APrefab:



Public Member Functions

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

4.13.1 Member Function Documentation**4.13.1.1 instantiate() [1/2]**

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

Implements [IPrefab](#).

4.13.1.2 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]
```

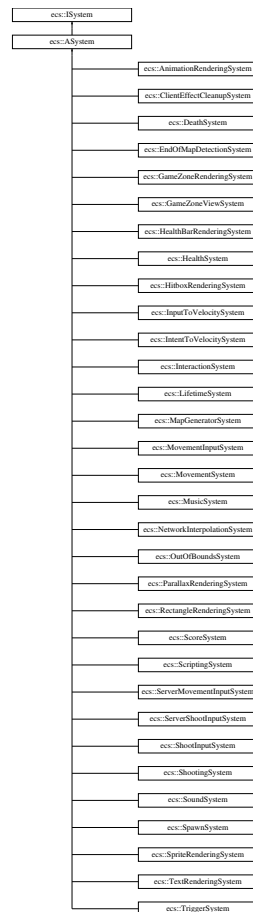
Implements [IPrefab](#).

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

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

4.14 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

4.14.1 Member Function Documentation

4.14.1.1 updateSystem()

```

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

```

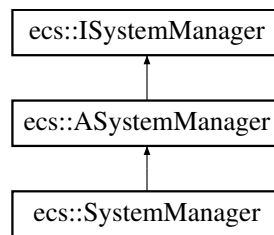
Implements [ecs::ISystem](#).

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

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

4.15 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](#)

4.15.1 Member Function Documentation

4.15.1.1 addSystem()

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

Implements [ecs::ISystemManager](#).

4.15.1.2 removeSystem()

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

Implements [ecs::ISystemManager](#).

4.15.1.3 updateAllSystems()

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

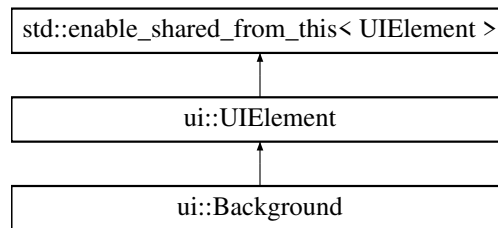
Implements [ecs::ISystemManager](#).

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

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

4.16 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**

4.16.1 Member Function Documentation

4.16.1.1 render()

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

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

4.16.1.2 update()

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

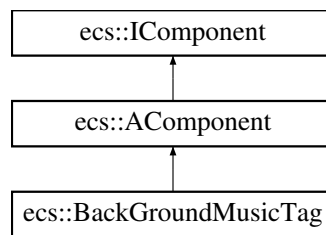
Reimplemented from [ui::UIElement](#).

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

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

4.17 ecs::BackGroundMusicTag Class Reference

Inheritance diagram for ecs::BackGroundMusicTag:

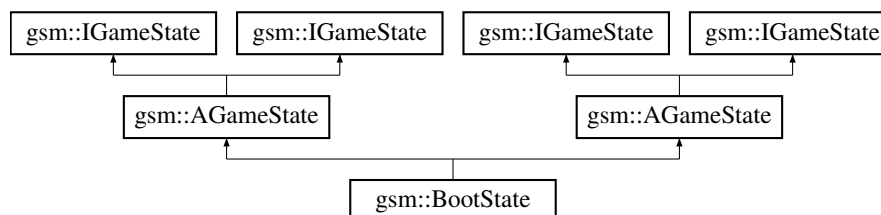


The documentation for this class was generated from the following file:

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

4.18 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](#)

4.18.1 Member Function Documentation

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

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

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

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

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

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

4.18.1.3 [exit\(\)](#)

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

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

4.18.1.4 [update\(\)](#)

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

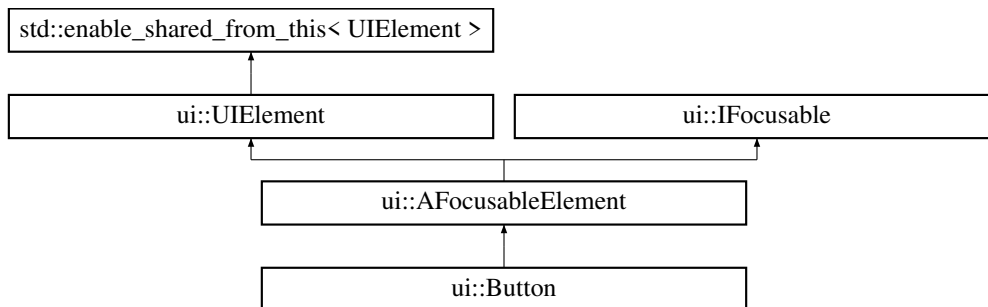
Reimplemented from [gsm::AGameState](#).

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

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

4.19 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/abduction2002.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`

4.19.1 Member Function Documentation

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

4.20 math::Chrono Class Reference

Public Member Functions

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

Private Attributes

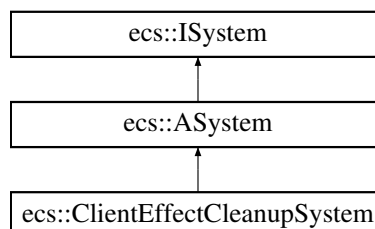
- std::chrono::high_resolution_clock::time_point **_startTime**
- std::chrono::high_resolution_clock::time_point **_stopTime**
- bool **_isRunning**

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

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

4.21 ecs::ClientEffectCleanupSystem Class Reference

Inheritance diagram for ecs::ClientEffectCleanupSystem:



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

4.21.1 Member Function Documentation

4.21.1.1 update()

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

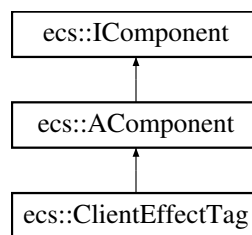
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/effects/ClientEffectCleanupSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/effects/ClientEffectCleanupSystem.cpp

4.22 ecs::ClientEffectTag Class Reference

Inheritance diagram for `ecs::ClientEffectTag`:

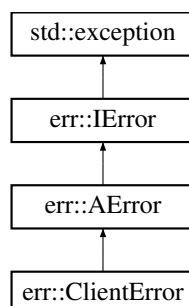


The documentation for this class was generated from the following file:

- /home/albane/epitech/tech3/r-type/ryanR-type/common/components/tags/ClientEffectTag.hpp

4.23 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**

4.23.1 Member Function Documentation

4.23.1.1 [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/ryanR-type/common/Error/ClientError.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/Error/ClientError.cpp

4.24 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)
- std::string **getLobbyCode** () const
- void **setLobbyCode** (std::string lobbyCode)
- net::ConnectionState **getConnectionState** () const
- void **eventPacket** (const constants::EventType &eventType, double depth)
- void **disconnectionPacket** ()
- void **connectionPacket** ()
- void **sendWhoAml** ()
- void **requestCode** ()
- void **sendLobbyConnection** (std::string lobbyCode)
- void **sendMasterStartGame** ()
- void **addToEventQueue** (const [NetworkEvent](#) &event)
- bool **isConnected** () const
- bool **isReady** () const
- size_t **getConnectedClients** () const
- size_t **getReadyClients** () const
- uint8_t **getClientId** () const
- bool **getClientReadyStatus** () const
- bool **isConnectedToLobby** () const
- bool **isLobbyMaster** () 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**
- std::atomic< bool > **_isConnectedToLobby**
- std::atomic< bool > **_isLobbyMaster**
- std::atomic< size_t > **_connectedClients**
- std::atomic< size_t > **_readyClients**
- std::atomic< uint8_t > **_clientId**
- std::atomic< bool > **_clientReadyStatus**

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 **handleEndGame** ()
- void **handleCanStart** ()
- void **handleEntitySpawn** ()
- void **handleEntityDeath** ()
- void **handleWhoAmI** ()
- void **handleServerStatus** ()
- void **handleCode** ()
- void **handleLobbyConnectValue** ()
- 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 **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 **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 **parseGameZoneComponent** (const std::vector< uint64_t > &payload, size_t index, ecs::Entity entityId)

Private Attributes

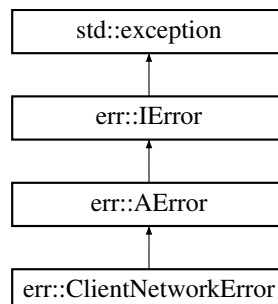
- PacketHandler **_packetHandlers** [constants::MAX_INDEX_PACKET_TYPE]
- std::map< uint64_t, ComponentParser > **_componentParsers**
- [DLLoader](#)< createNetworkLib_t > **_networloader**
- [DLLoader](#)< createBuffer_t > **_bufferloader**
- [DLLoader](#)< createPacket_t > **_packetloader**
- std::shared_ptr< [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**
- std::shared_ptr< [net::INetworkEndpoint](#) > **_serverEndpoint**
- std::queue< [NetworkEvent](#) > **_eventQueue**
- std::mutex **_queueMutex**
- std::condition_variable **_queueCond**
- std::unordered_map< size_t, ecs::Entity > **_serverToLocalEntityMap**
- std::string **_lobbyCode**
- bool **_shouldConnect**

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

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

4.25 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**

4.25.1 Member Function Documentation

4.25.1.1 [getType\(\)](#)

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

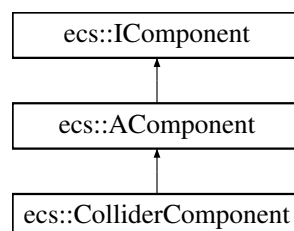
Implements [err::AError](#).

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

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

4.26 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
- [math::OrientedRect](#) **getOrientedHitbox** ([math::Vector2f](#) entityPosition, [math::Vector2f](#) scale, float rotation) const
- [math::FRect](#) **getHitbox** ([math::Vector2f](#) entityPosition, [math::Vector2f](#) scale, float rotation) const

Private Attributes

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

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/ColliderComponent.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/ColliderComponent.cpp

4.27 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/ryanR-type/common/CollisionRules/CollisionRulesData.hpp

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

4.29 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/ryanR-type/common/CollisionRules/CollisionRulesData.hpp

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

4.31 gfx::color_t Struct Reference

Public Attributes

- uint8_t **r**
- uint8_t **g**
- uint8_t **b**
- uint8_t **a** = 255

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/interfaces/IWindow.hpp

4.32 rserv::ComponentDeltaTracker Class Reference

Public Member Functions

- std::vector< uint64_t > **createEntityDelta** (uint8_t clientId, uint32_t entityId, const [EntitySnapshot](#) ¤tSnapshot)
- 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** ()
- void **clearDeadEntities** (const std::set< uint32_t > &aliveEntityIds)

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/ryanR-type/server/deltaTracker/ComponentDeltaTracker.hpp`
- `/home/albane/epitech/tech3/r-type/ryanR-type/server/deltaTracker/ComponentDeltaTracker.cpp`

4.33 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 (uint32_t vx, uint32_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 ¤t, 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 > 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/ryanR-type/server/deltaTracker/ComponentSerializer.hpp`
- `/home/albane/epitech/tech3/r-type/ryanR-type/server/deltaTracker/ComponentSerializer.cpp`

4.34 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

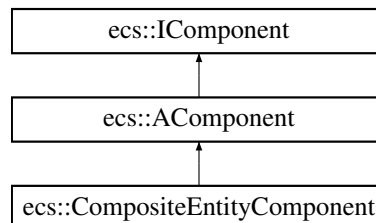
- std::shared_ptr< const std::map< std::string, std::pair< std::type_index, std::vector< [Field](#) > > > > **_componentDefinitions**
- const std::map< std::type_index, ComponentCreator > & **_componentCreators**
- ShouldParseComponentCallback **_shouldParseCallback**

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

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

4.35 ecs::CompositeEntityComponent Class Reference

Inheritance diagram for ecs::CompositeEntityComponent:



Public Member Functions

- **CompositeEntityComponent** (size_t parent_id)
- size_t **getParentId** () const
- void **setParentId** (size_t id)

Private Attributes

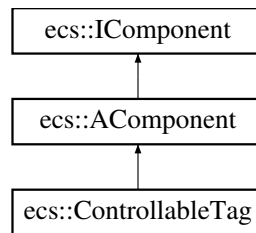
- size_t **parentId**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/CompositeEntityComponent.[↔](#)hpp

4.36 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/ryanR-type/common/components/tags/ControllableTag.hpp

4.37 Core Class Reference

Public Member Functions

- void **initFirstScene** ()
- void **run** ()
- void **startNetwork** ()
- std::shared_ptr< [ClientNetwork](#) > **getNetwork** ()

Private Member Functions

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

Private Attributes

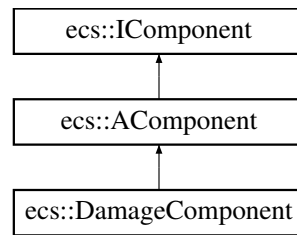
- std::shared_ptr< [DLLoader](#) < gfx::createWindow_t > > **_windowLoader**
- std::shared_ptr< [DLLoader](#) < gfx::createEvent_t > > **_eventLoader**
- std::shared_ptr< [DLLoader](#) < gfx::createAudio_t > > **_audioLoader**
- std::shared_ptr< [ResourceManager](#) > **_resourceManager**
- std::shared_ptr< [gsm::GameStateMachine](#) > **_gsm**
- std::shared_ptr< [ecs::Registry](#) > **_registry**
- std::shared_ptr< [ClientNetwork](#) > **_clientNetwork**
- std::shared_ptr< [Parser](#) > **_parser**
- std::thread **_networkThread**

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

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

4.38 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

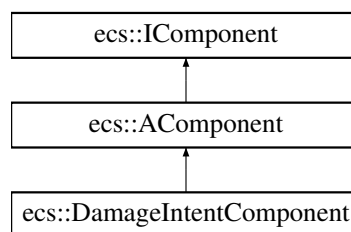
- float **_damage**

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

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

4.39 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/ryanR-type/common/components/temporary/DamageIntentComponent.↔
hpp

4.40 DeathAnimationSpawner Class Reference

Static Public Member Functions

- static void **spawnDeathAnimation** (std::shared_ptr< [ResourceManager](#) > resourceManager, std::shared_ptr< [ecs::Registry](#) > registry, ecs::Entity entity)

Static Private Member Functions

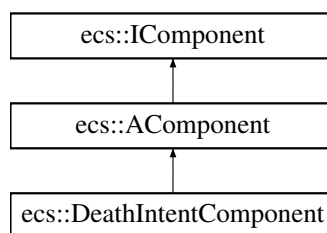
- static [math::Vector2f](#) **getFirstHitboxCenter** (std::shared_ptr< [ecs::Registry](#) > registry, ecs::Entity entity)

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/DeathAnimationSpawner.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/DeathAnimationSpawner.cpp

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

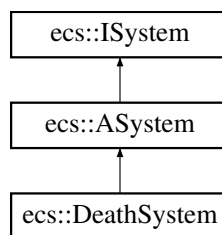
- `ecs::Entity_source`

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

- `/home/albane/epitech/tech3/r-type/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`

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

4.43 debug::Debug Class Reference

Static Public Member Functions

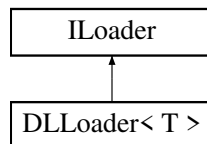
- static void **printDebug** (const bool isDebug, const std::string &message, debugType type, debugLevel level)

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

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

4.44 DLoader< T > Class Template Reference

Inheritance diagram for DLoader< 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

4.44.1 Member Function Documentation

4.44.1.1 Close()

```

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

Implements [ILoader](#).

4.44.1.2 Error()

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

Implements [ILoader](#).

4.44.1.3 getHandler()

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

Implements [ILoader](#).

4.44.1.4 Open()

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

Implements [ILoader](#).

4.44.1.5 Symbol()

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

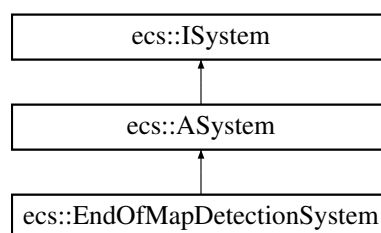
Implements [ILoader](#).

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

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

4.45 ecs::EndOfMapDetectionSystem Class Reference

Inheritance diagram for `ecs::EndOfMapDetectionSystem`:



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

4.45.1 Member Function Documentation**4.45.1.1 update()**

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

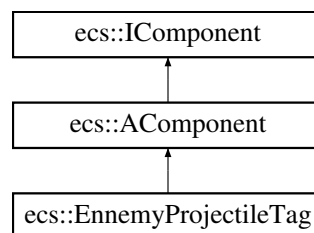
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/server/systems/gameEnd/EndOfMapDetectionSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/systems/gameEnd/EndOfMapDetectionSystem.cpp

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/ryanR-type/common/components/tags/EnemyProjectileTag.hpp

4.47 ecs::EntityCreationContext Struct Reference**Static Public Member Functions**

- static [EntityCreationContext](#) [forServer](#) ()
- static [EntityCreationContext](#) [forLocalClient](#) ()

Public Attributes

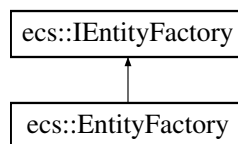
- EntityCreationOrigin **origin** = EntityCreationOrigin::CLIENT_LOCAL

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

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

4.48 ecs::EntityFactory Class Reference

Inheritance diagram for ecs::EntityFactory:



Public Member Functions

- Entity [createEntity](#) (const std::shared_ptr< [Registry](#) > ®istry, const [EntityCreationContext](#) &context=EntityCreationContext::forLocalClient()) override

Private Attributes

- 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](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/entity/factory/EntityFactory.hpp
- /home/albane/epitech/tech3/r-type/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 ShouldParseComponentCallback &shouldParseCallback=nullptr)
- std::shared_ptr< [IPrefab](#) > **parseEntity** (const std::string &filePath)

Private Attributes

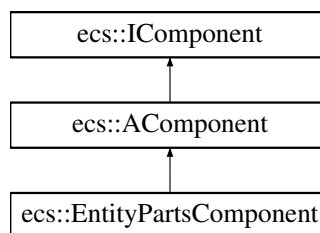
- [ComposantParser](#) **_composantParser**
- const std::map< std::type_index, ComponentAdder > & **_componentAdders**
- ShouldParseComponentCallback **_shouldParseCallback**

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

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

4.50 ecs::EntityPartsComponent Class Reference

Inheritance diagram for ecs::EntityPartsComponent:



Public Attributes

- std::vector< size_t > **partIds**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/EntityPartsComponent.hpp

4.51 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](#) > ®istry, const [ecs::EntityCreationContext](#) &context)
- ecs::Entity **createEntityFromPrefab** (const std::string &prefabName, const std::shared_ptr< [ecs::Registry](#) > ®istry)
- 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/ryanR-type/common/Prefab/entityPrefabManager/EntityPrefabManager.↔hpp`
- `/home/albane/epitech/tech3/r-type/ryanR-type/common/Prefab/entityPrefabManager/EntityPrefabManager.↔cpp`

4.52 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/ryanR-type/server/deltaTracker/ComponentDeltaTracker.hpp`

4.53 Field Struct Reference

Public Member Functions

- **Field** (`std::string n`, `FieldType t`, `bool opt=false`, `std::shared_ptr< FieldValue > def=nullptr`)

Public Attributes

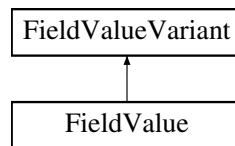
- `std::string name = ""`
- `FieldType type`
- `bool optional = false`
- `std::shared_ptr< FieldValue > defaultValue = nullptr`

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

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

4.54 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/ryanR-type/common/Parser/ParserParam.hpp`

4.55 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

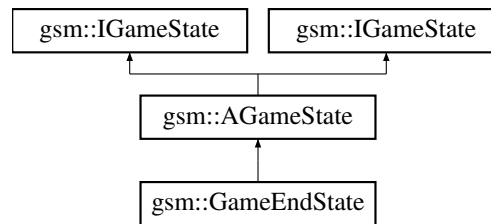
- float `left`
- float `top`
- float `width`
- float `height`

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

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

4.56 `gsm::GameState` Class Reference

Inheritance diagram for `gsm::GameState`:



Public Member Functions

- **GameState** (`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)
- 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 [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](#)

4.56.1 Member Function Documentation

4.56.1.1 `enter()`

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

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

4.56.1.2 update()

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

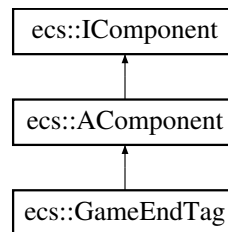
Reimplemented from [gsm::AGameState](#).

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

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

4.57 ecs::GameEndTag Class Reference

Inheritance diagram for ecs::GameEndTag:

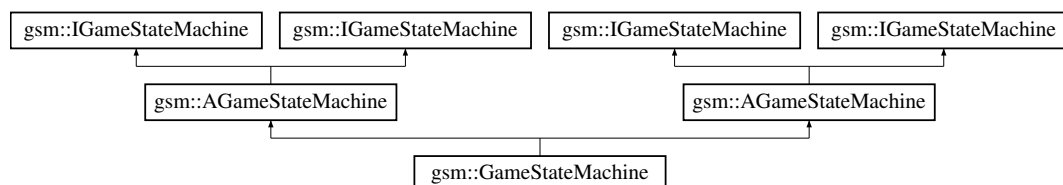


The documentation for this class was generated from the following file:

- /home/albane/epitech/tech3/r-type/ryanR-type/common/components/tags/GameEndTag.hpp

4.58 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

4.58.1 Member Function Documentation

4.58.1.1 requestStateChange()

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

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

4.58.1.2 requestStatePop()

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

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

4.58.1.3 requestStatePush()

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

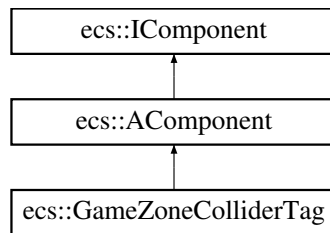
Reimplemented from [gsm::AGameStateMachine](#).

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

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

4.59 ecs::GameZoneColliderTag Class Reference

Inheritance diagram for ecs::GameZoneColliderTag:

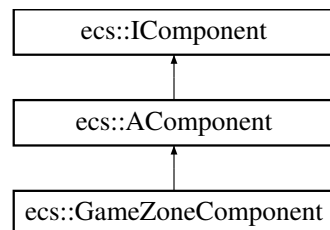


The documentation for this class was generated from the following file:

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

4.60 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

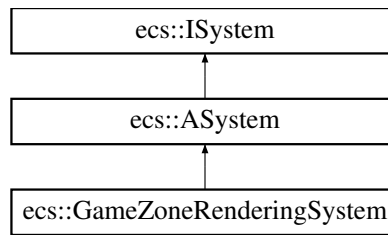
- [math::FRect](#) **_zone**

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

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

4.61 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

4.61.1 Member Function Documentation

4.61.1.1 update()

```

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

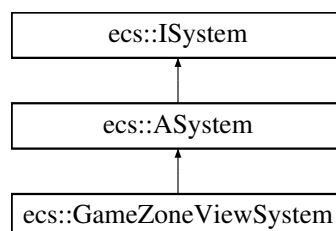
Implements [ecs::ASystem](#).

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

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

4.62 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

4.62.1 Member Function Documentation**4.62.1.1 update()**

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

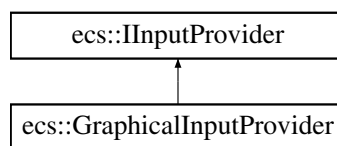
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/GameZoneViewSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/GameZoneViewSystem.cpp

4.63 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`

4.63.1 Member Function Documentation

4.63.1.1 `getActionAxis()`

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

Implements [ecs::IInputProvider](#).

4.63.1.2 `getAxisValue()`

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

Implements [ecs::IInputProvider](#).

4.63.1.3 `getInputMapping()`

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

Implements [ecs::IInputProvider](#).

4.63.1.4 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/ryanR-type/client/initResourceManager/GraphicalInputProvider.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/initResourceManager/GraphicalInputProvider.cpp

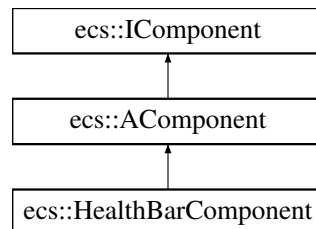
4.64 ecs::Group< Components > Class Template Reference

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

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

4.65 ecs::HealthBarComponent Class Reference

Inheritance diagram for ecs::HealthBarComponent:

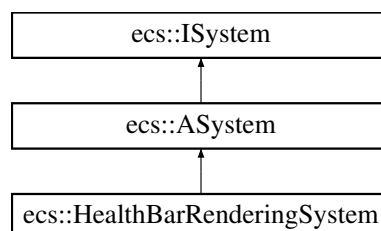


The documentation for this class was generated from the following file:

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

4.66 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

4.66.1 Member Function Documentation

4.66.1.1 update()

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

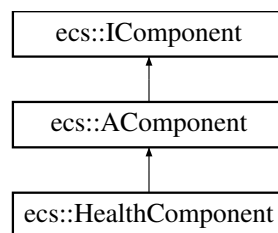
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/HealthBarRenderingSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/HealthBarRenderingSystem.cpp

4.67 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

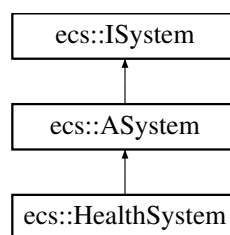
- float **_health**
- float **_baseHealth**
- ecs::Entity **_lastDamageSource**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/HealthComponent.hpp

4.68 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)

4.68.1 Member Function Documentation

4.68.1.1 update()

```

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

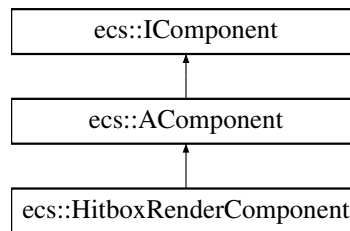
Implements [ecs::ASystem](#).

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

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

4.69 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

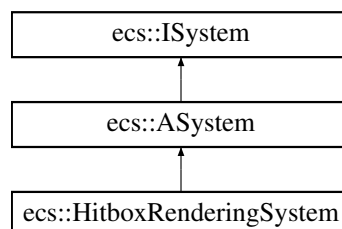
- gfx::color_t _color
- float _outlineThickness

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

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

4.70 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

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

4.70.1 Member Function Documentation

4.70.1.1 update()

```
void ecs::HitboxRenderingSystem::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/ryanR-type/client/systems/rendering/HitboxRenderingSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/HitboxRenderingSystem.cpp

4.71 gfx::IAudio Class Reference

Public Member Functions

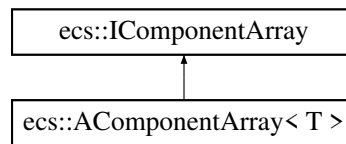
- virtual void **playMusic** (const std::string &musicPath, bool loop=true)=0
- virtual void **stopMusic** ()=0
- virtual void **pauseMusic** ()=0
- virtual void **resumeMusic** ()=0
- virtual void **setMusicVolume** (float volume)=0
- virtual float **getMusicVolume** () const =0
- virtual bool **isMusicPlaying** () const =0
- virtual void **playSound** (const std::string &soundPath, float volume=100.0f)=0
- virtual void **setSoundVolume** (float volume)=0
- virtual float **getSoundVolume** () const =0
- virtual void **stopAllSounds** ()=0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/interfaces/IAudio.hpp

4.74 ecs::IComponentArray Class Reference

Inheritance diagram for ecs::IComponentArray:



Public Member Functions

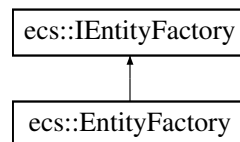
- virtual Entity **getMaxEntityId** () const =0
- virtual void **removeComponents** (Entity entityId)=0
- virtual void **removeOneComponent** (Entity entityId)=0

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

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

4.75 ecs::IEntityFactory Class Reference

Inheritance diagram for ecs::IEntityFactory:



Public Member Functions

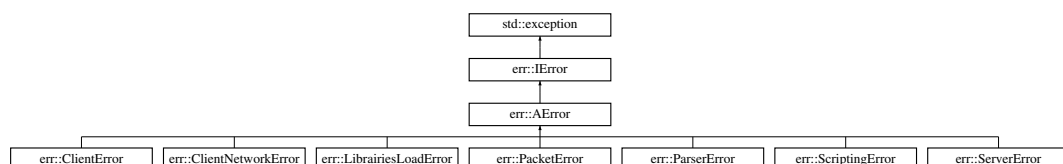
- virtual Entity **createEntity** (const std::shared_ptr< [Registry](#) > ®istry, const [EntityCreationContext](#) &context=EntityCreationContext::forLocalClient())=0

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

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

4.76 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/ryanR-type/common/Error/LError.hpp

4.77 gfx::IEvent Class Reference

Public Types

- using **event_t** = EventType

Public Member Functions

- virtual void **init** ()=0
- virtual event_t **pollEvents** ()=0
- virtual std::string **getLastTextInput** ()=0
- virtual void **cleanup** ()=0
- virtual std::pair< int, int > **getMousePos** ()=0
- virtual bool **isKeyPressed** (event_t key)=0
- virtual bool **isMouseButtonPressed** (int button)=0
- virtual float **getAxisValue** (event_t axis)=0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/interfaces/IEvent.hpp

4.78 net::IEventLoop Class Reference

Public Member Functions

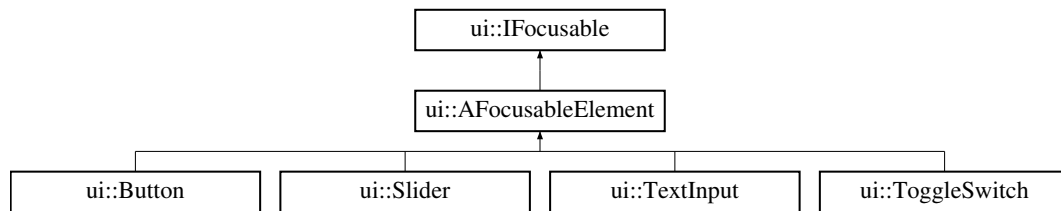
- virtual void **run** ()=0
- virtual void **runOne** ()=0
- virtual void **stop** ()=0
- virtual bool **stopped** () const =0
- virtual void **post** (std::function< void()> task)=0
- virtual void **restart** ()=0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/interfaces/IEventLoop.hpp

4.79 ui::IFocusable Class Reference

Inheritance diagram for ui::IFocusable:



Public Member Functions

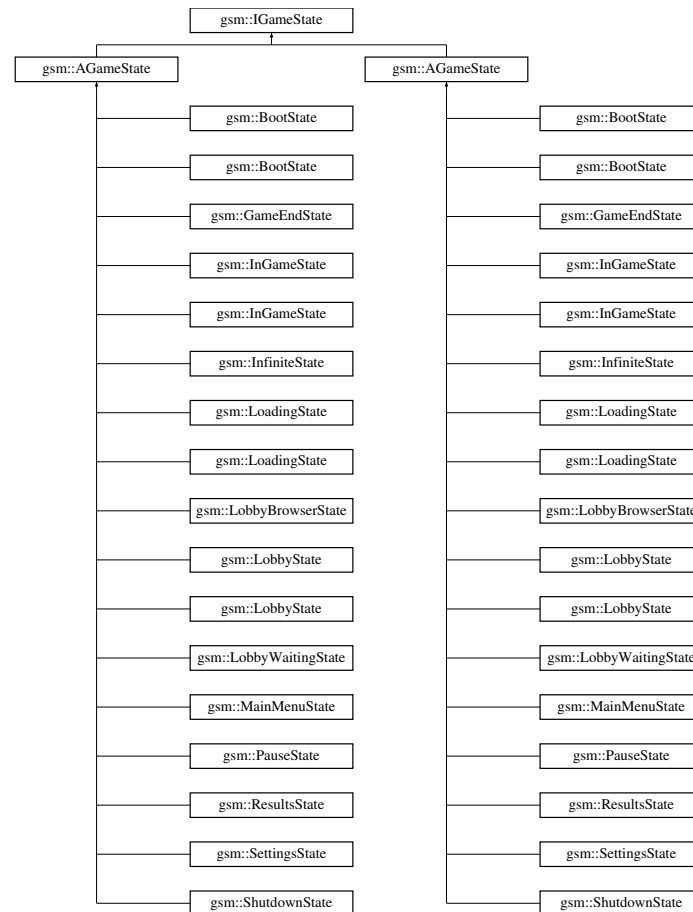
- virtual void **setFocused** (bool focused)=0
- virtual bool **isFocused** () const =0
- virtual bool **canBeFocused** () const =0
- virtual void **onFocusGained** ()=0
- virtual void **onFocusLost** ()=0
- virtual void **onActivated** ()=0
- virtual bool **onNavigateLeft** ()
- virtual bool **onNavigateRight**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

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

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

4.80 gsm::IGameState Class Reference

Inheritance diagram for gsm::IGameState:



Public Member Functions

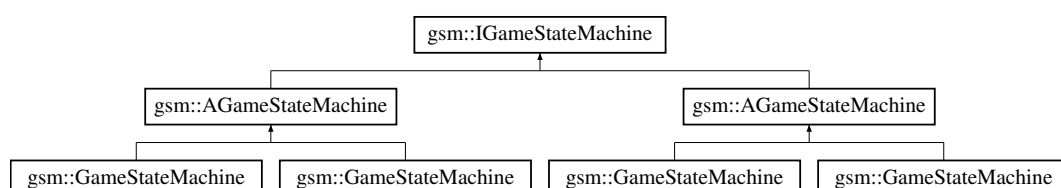
- virtual void **enter** ()=0
- virtual void **update** (float deltaTime)=0
- virtual void **exit** ()=0
- virtual void **addSystem** (std::shared_ptr< [ecs::ISystem](#) > system)=0
- virtual std::vector< std::shared_ptr< [ecs::ISystem](#) > > **getSystems** () const =0

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

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

4.81 gsm::IGameStateMachine Class Reference

Inheritance diagram for gsm::IGameStateMachine:



Public Member Functions

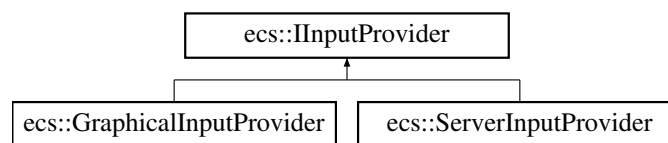
- virtual void **changeState** (std::shared_ptr< [IGameState](#) > newState)=0
- virtual void **pushState** (std::shared_ptr< [IGameState](#) > newState)=0
- virtual void **popState** ()=0
- virtual void **requestStateChange** (std::shared_ptr< [IGameState](#) > newState)=0
- virtual void **requestStatePush** (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/ryanR-type/common/gsm/IGameStateMachine.hpp

4.82 ecs::IInputProvider Class Reference

Inheritance diagram for ecs::IInputProvider:

**Public Types**

- using **event_t** = gfx::EventType

Public Member Functions

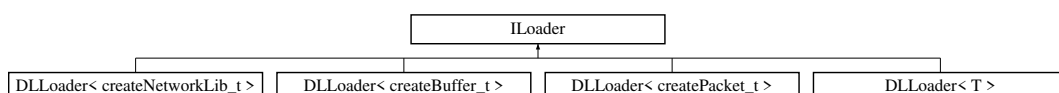
- virtual float **getAxisValue** (event_t axis, size_t clientID=0)=0
- virtual bool **isActionPressed** (InputAction action, size_t clientID=0)=0
- virtual float **getActionAxis** (InputAction action, size_t clientID=0)=0
- virtual [InputMapping](#) **getInputMapping** (size_t clientID=0) const =0

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

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

4.83 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/ryanR-type/common/DLLoader/ILoader.hpp

4.84 net::INetwork Class Reference

Public Member Functions

- virtual void **init** (uint16_t port, const std::string host)=0
- virtual void **stop** ()=0
- virtual bool **sendTo** (const [INetworkEndpoint](#) &endpoint, std::vector< uint8_t > packet)=0
- virtual bool **broadcast** (const std::vector< std::shared_ptr< [INetworkEndpoint](#) > > &endpoints, const std::vector< uint8_t > &data)=0
- virtual bool **hasIncomingData** () const =0
- virtual std::vector< uint8_t > **receiveFrom** (const uint8_t &connectionId)=0
- virtual std::pair< std::shared_ptr< [INetworkEndpoint](#) >, std::vector< uint8_t > > **receiveAny** ()=0
- virtual void **setConnectionCallback** (std::function< void(int)> onConnect)=0
- virtual void **setDisconnectionCallback** (std::function< void(int)> onDisconnect)=0
- virtual ConnectionState **getConnectionState** () const =0
- virtual void **setConnectionState** (ConnectionState state)=0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/interfaces/INetwork.hpp

4.85 net::INetworkAddress Class Reference

Public Member Functions

- virtual bool **isV4** () const =0
- virtual bool **isV6** () const =0
- virtual std::string **toString** () const =0
- virtual std::shared_ptr< [INetworkAddress](#) > **operator=** (const [INetworkAddress](#) &other)=0
- virtual std::shared_ptr< void > **getInternalAddress** ()=0
- virtual std::shared_ptr< const void > **getInternalAddress** () const =0
- virtual void **setFromInternal** (std::shared_ptr< void > internalAddr)=0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/interfaces/INetworkAddress.hpp

4.86 net::INetworkEndpoint Class Reference

Public Member Functions

- virtual const std::string & **getAddress** () const =0
- virtual uint16_t **getPort** () const =0
- virtual void **setAddress** (const std::string &address)=0
- virtual void **setPort** (uint16_t port)=0
- virtual bool **operator==** (const [INetworkEndpoint](#) &other) const =0
- virtual bool **operator!=** (const [INetworkEndpoint](#) &other) const =0
- virtual bool **operator<** (const [INetworkEndpoint](#) &other) const =0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/interfaces/INetworkEndpoint.hpp

4.87 net::INetworkErrorCode Class Reference

Public Member Functions

- virtual void **clear** ()=0
- virtual bool **hasError** () const =0
- virtual **operator bool** () const =0
- virtual std::string **message** () const =0
- virtual NetworkError **getError** () const =0
- virtual void **setError** (NetworkError error, const std::string &msg="")=0
- virtual bool **operator==** (NetworkError error) const =0
- virtual bool **operator!=** (NetworkError error) const =0
- virtual std::shared_ptr< void > **getInternalErrorCode** ()=0
- virtual std::shared_ptr< const void > **getInternalErrorCode** () const =0
- virtual void **setFromInternal** (std::shared_ptr< void > internalEc)=0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/interfaces/INetworkErrorCode.hpp

4.88 net::INetworkFactory Class Reference

Public Member Functions

- virtual std::shared_ptr< [IEventLoop](#) > **createEventLoop** ()=0
- virtual std::shared_ptr< [INetworkSocket](#) > **createSocket** (std::shared_ptr< [IEventLoop](#) > eventLoop)=0
- virtual std::shared_ptr< [INetworkResolver](#) > **createResolver** (std::shared_ptr< [IEventLoop](#) > eventLoop)=0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/interfaces/INetworkFactory.hpp

4.89 net::INetworkResolver Class Reference

Public Member Functions

- virtual std::vector< std::shared_ptr< [INetworkEndpoint](#) > > **resolve** (const std::string &host, const std::string &port, std::shared_ptr< [INetworkErrorCode](#) > ec)=0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/interfaces/INetworkResolver.hpp

4.90 net::INetworkSocket Class Reference

Public Member Functions

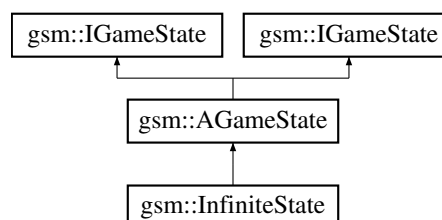
- virtual bool **open** (std::shared_ptr< [INetworkErrorCode](#) > ec)=0
- virtual bool **bind** (const [INetworkEndpoint](#) &endpoint, std::shared_ptr< [INetworkErrorCode](#) > ec)=0
- virtual std::size_t **sendTo** (const std::vector< uint8_t > &data, const [INetworkEndpoint](#) &endpoint, int flags, std::shared_ptr< [INetworkErrorCode](#) > ec)=0
- virtual std::size_t **receiveFrom** (std::shared_ptr< std::vector< uint8_t > > buffer, std::shared_ptr< [INetworkEndpoint](#) > sender, int flags, std::shared_ptr< [INetworkErrorCode](#) > ec)=0
- virtual bool **setNonBlocking** (bool nonBlocking, std::shared_ptr< [INetworkErrorCode](#) > ec)=0
- virtual bool **close** (std::shared_ptr< [INetworkErrorCode](#) > ec)=0
- virtual bool **isOpen** () const =0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/interfaces/INetworkSocket.hpp

4.91 gsm::InfiniteState Class Reference

Inheritance diagram for gsm::InfiniteState:



Public Member Functions

- **InfiniteState** (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 Attributes

- std::shared_ptr< [ecs::Registry](#) > [_registry](#)
- std::shared_ptr< [EntityPrefabManager](#) > [_prefabManager](#)
- std::shared_ptr< [Parser](#) > [_parser](#)

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](#)

4.91.1 Member Function Documentation

4.91.1.1 enter()

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

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

4.91.1.2 exit()

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

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

4.91.1.3 update()

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

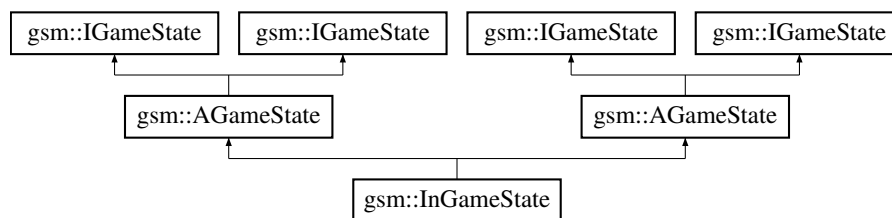
Reimplemented from [gsm::AGameState](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/Infinite/InfiniteState.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/Infinite/InfiniteState.cpp

4.92 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
- 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

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**

4.92.1 Member Function Documentation**4.92.1.1 enter() [1/2]**

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

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

4.92.1.2 enter() [2/2]

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

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

4.92.1.3 exit()

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

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

4.92.1.4 update() [1/2]

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

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

4.92.1.5 update() [2/2]

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

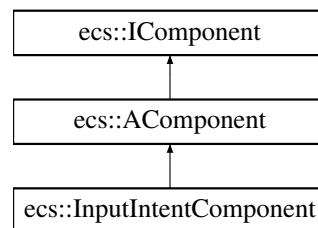
Reimplemented from [gsm::AGameState](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/InGame/InGameState.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/scenes/InGame/InGameState.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/InGame/InGameState.cpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/scenes/InGame/InGameState.cpp

4.93 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/ryanR-type/common/components/temporary/InputIntentComponent.hpp

4.94 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/ryanR-type/common/InputMapping/InputMapping.hpp

4.95 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/ryanR-type/common/InputMapping/InputMappingManager.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/InputMapping/InputMappingManager.cpp

4.96 ecs::InputNormalizer Class Reference

Static Public Member Functions

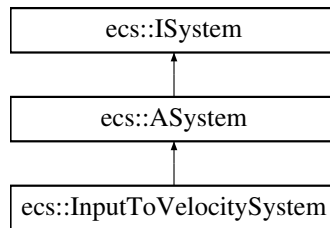
- static [math::Vector2f](#) **normalizeDirection** (const [math::Vector2f](#) &direction)
- static [math::Vector2f](#) **normalizeAnalogInput** (float rawX, float rawY)

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

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

4.97 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

4.97.1 Member Function Documentation

4.97.1.1 update()

```

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

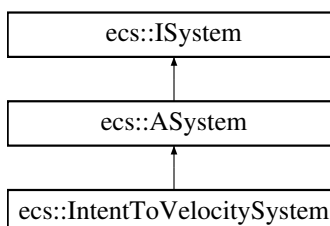
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/systems/movement/InputToVelocitySystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/systems/movement/InputToVelocitySystem.cpp

4.98 ecs::IntentToVelocitySystem Class Reference

Inheritance diagram for ecs::IntentToVelocitySystem:



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

4.98.1 Member Function Documentation

4.98.1.1 update()

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

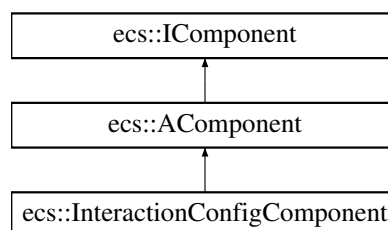
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/systems/movement/IntentToVelocitySystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/systems/movement/IntentToVelocitySystem.cpp

4.99 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/ryanR-type/common/components/permanent/InteractionConfigComponent.↔hpp`

4.100 `ecs::InteractionMapping` Struct Reference

Public Attributes

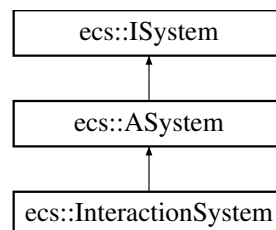
- `std::vector< std::string > targetTags`
- `std::vector< std::string > actionsToOther`
- `std::vector< std::string > actionsToSelf`

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

- `/home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/InteractionConfigComponent.↔hpp`

4.101 `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`

4.101.1 Member Function Documentation

4.101.1.1 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/ryanR-type/common/systems/interactions/InteractionSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/systems/interactions/InteractionSystem.cpp

4.102 pm::IPacketManager Class Reference

Public Member Functions

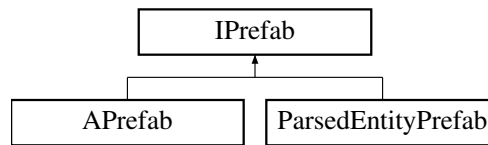
- virtual uint32_t **getLength** () const =0
- virtual uint32_t **getSequenceNumber** () const =0
- virtual uint8_t **getType** () const =0
- virtual std::vector< uint64_t > **getPayload** () const =0
- virtual uint8_t **getIdClient** () const =0
- virtual void **setType** (uint8_t type)=0
- virtual void **setLength** (uint32_t length)=0
- virtual void **setSequenceNumber** (uint32_t sequenceNumber)=0
- virtual void **setPayload** (std::vector< uint64_t > payload)=0
- virtual void **setIdClient** (uint8_t idClient)=0
- virtual std::vector< uint64_t > **formatString** (const std::string str)=0
- virtual std::vector< uint8_t > **pack** (uint8_t idClient, uint32_t sequenceNumber, uint8_t type, std::vector< uint64_t > payload)=0
- virtual bool **unpack** (std::vector< uint8_t > data)=0
- virtual void **reset** ()=0
- virtual void **registerBuilder** (uint8_t type, std::function< std::vector< uint8_t > (std::vector< uint64_t >) >> builder)=0
- virtual void **registerParser** (uint8_t type, std::function< bool(const std::vector< uint8_t >) >> parser)=0
- virtual void **registerLength** (uint8_t type, uint32_t length)=0
- virtual void **registerGameStatePackFunction** (std::function< std::vector< uint8_t > (std::vector< uint64_t >, std::shared_ptr< unsigned int >) >> func)=0
- virtual void **registerGameStateUnpackFunction** (std::function< unsigned int(const std::vector< uint8_t >, unsigned int) >> func)=0
- virtual void **registerLengthCombEntry** (uint8_t compType, uint32_t compLength, uint64_t compSize)=0
- virtual void **clearAllHandlers** ()=0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/interfaces/IPacketManager.hpp

4.103 IPrefab Class Reference

Inheritance diagram for IPrefab:



Public Member Functions

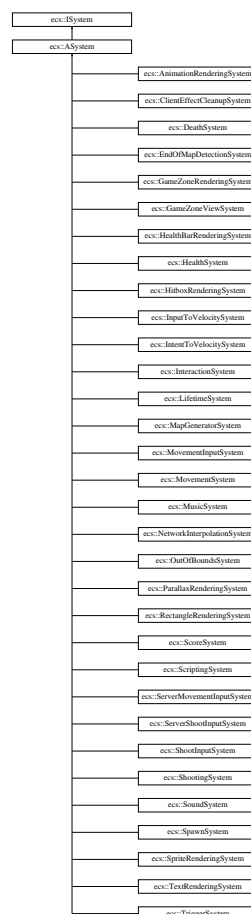
- virtual `ecs::Entity` **instantiate** (const std::shared_ptr< [ecs::Registry](#) > ®istry, const std::shared_ptr< [ecs::IEntityFactory](#) > &factory, const [ecs::EntityCreationContext](#) &context=ecs::EntityCreationContext::for↵ LocalClient())=0
- virtual `ecs::Entity` **instantiate** (const std::shared_ptr< [ecs::Registry](#) > ®istry)=0

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

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

4.104 ecs::ISystem Class Reference

Inheritance diagram for ecs::ISystem:



Public Member Functions

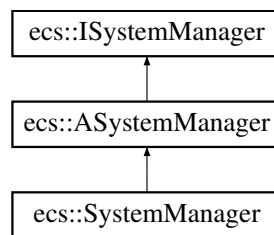
- virtual void **updateSystem** (std::shared_ptr< [ResourceManager](#) > resourceManager, std::shared_ptr< [Registry](#) > registry, float deltaTime)=0

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

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

4.105 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/ryanR-type/common/systems/systemManager/ISystemManager.hpp

4.106 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/ryanR-type/common/ECS/view/View.hpp`

4.107 gfx::IWindow Class Reference

Public Member Functions

- virtual void **init** ()=0
- virtual void **display** ()=0
- virtual void **closeWindow** ()=0
- virtual bool **isOpen** ()=0
- virtual void **clear** ()=0
- virtual void **resizeWindow** (size_t x, size_t y)=0
- virtual void **drawSprite** (std::string asset, color_t color, std::pair< size_t, size_t > position)=0
- virtual void **drawText** (std::string text, color_t color, std::pair< size_t, size_t > position, const std::string &fontPath, size_t fontSize=24, color_t outlineColor={0, 0, 0}, float outlineThickness=0.0f)=0
- virtual std::pair< size_t, size_t > **getTextSize** (const std::string &text, const std::string &fontPath, size_t fontSize=24)=0
- virtual void **drawRectangleOutline** (color_t color, std::pair< size_t, size_t > position, std::pair< size_t, size_t > size)=0
- virtual void **drawFilledRectangle** (color_t color, std::pair< size_t, size_t > position, std::pair< size_t, size_t > size)=0
- virtual void **drawRoundedRectangleFilled** (color_t color, std::pair< size_t, size_t > position, std::pair< size_t, size_t > size, float radius)=0
- virtual void **drawRoundedRectangleOutline** (color_t color, std::pair< size_t, size_t > position, std::pair< size_t, size_t > size, float radius)=0
- virtual bool **isMouseOver** (std::pair< size_t, size_t > position, std::pair< size_t, size_t > size)=0
- virtual std::pair< int, int > **getWindowSize** ()=0
- virtual void **drawSprite** (const std::string &texturePath, float x, float y, float scaleX=1.0f, float scaleY=1.0f, float rotation=0.0f)=0
- virtual void **drawSprite** (const std::string &texturePath, float x, float y, const math::FRect frameRect, float scaleX=1.0f, float scaleY=1.0f, float rotation=0.0f)=0
- virtual void **updateView** ()=0
- virtual void **setViewCenter** (float x, float y)=0
- virtual math::Vector2f **getViewCenter** ()=0
- virtual math::Vector2f **mapPixelToCoords** (int x, int y)=0
- virtual std::pair< int, int > **getLogicalSize** () const =0
- virtual float **getScaleFactor** () const =0
- virtual void **addShaderFilter** (const std::string &path)=0
- virtual void **removeShaderFilter** (const std::string &path)=0
- virtual void **setShaderUniform** (const std::string &filterPath, const std::string &name, float value)=0
- virtual void **setFramerateLimit** (unsigned int fps)=0
- virtual void **setFullscreen** (bool fullscreen)=0
- virtual void **setRenderQuality** (float quality)=0
- virtual void **setCursor** (bool isHand)=0

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

- `/home/albane/epitech/tech3/r-type/ryanR-type/common/interfaces/IWindow.hpp`

4.108 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/ryanR-type/client/ui/elements/Background.hpp

4.109 ui::LayoutConfig Struct Reference

Public Attributes

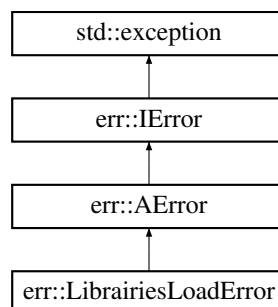
- LayoutDirection **direction** = LayoutDirection::Vertical
- LayoutAlignment **alignment** = LayoutAlignment::Start
- float **spacing** = 0.0f
- [math::Vector2f](#) **padding** = [math::Vector2f](#)(0.0f, 0.0f)
- [math::Vector2f](#) **offset** = [math::Vector2f](#)(0.0f, 0.0f)
- bool **autoResize** = 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/ryanR-type/client/ui/core/UILayout.hpp

4.110 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**

4.110.1 Member Function Documentation

4.110.1.1 [getType\(\)](#)

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

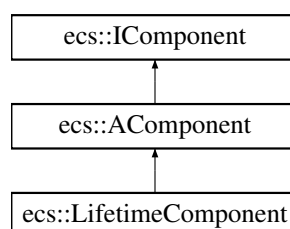
Implements [err::AError](#).

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

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

4.111 [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

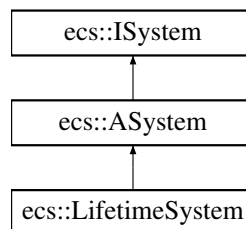
- float **_lifetime**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/LifetimeComponent.hpp

4.112 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

4.112.1 Member Function Documentation

4.112.1.1 update()

```

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

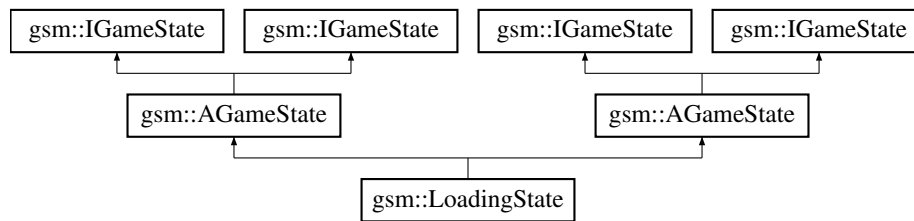
Implements [ecs::ASystem](#).

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

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

4.113 `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`

4.113.1 Member Function Documentation

4.113.1.1 enter() [1/2]

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

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

4.113.1.2 enter() [2/2]

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

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

4.113.1.3 exit()

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

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

4.113.1.4 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/ryanR-type/client/gsm/states/scenes/Loading/LoadingState.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/scenes/Loading/LoadingState.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/Loading/LoadingState.cpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/scenes/Loading/LoadingState.cpp

4.114 rserv::Lobby Class Reference

Public Member Functions

- **Lobby** (std::shared_ptr< [net::INetwork](#) > network, std::vector< std::tuple< uint8_t, std::shared_ptr< [net::INetworkEndpoint](#) >, std::string > > lobbyPlayerInfo, std::string lobbyCode, bool debug)
- void **stop** ()
- void **startNetworkThread** ()
- void **startGameThread** ()
- void **networkLoop** ()
- void **gameLoop** ()
- void **setIsDebug** (bool debug)
- bool **getIsDebug** () const
- std::vector< uint8_t > **getConnectedClients** () const

- `std::vector< std::shared_ptr< net::INetworkEndpoint > > getConnectedClientEndpoints () const`
- `size_t getClientCount () const`
- `std::string getLobbyCode () const`
- `std::shared_ptr< net::INetwork > getNetwork () const`
- `std::shared_ptr< std::queue< std::tuple< uint8_t, constants::EventType, double > > > getEventQueue ()`
- `bool hasEvents () const`
- `void enqueuePacket (std::pair< std::shared_ptr< net::INetworkEndpoint >, std::vector< uint8_t > > packet)`
- `void processIncomingPackets ()`
- `bool processDisconnections (uint8_t idClient)`
- `bool processEvents (uint8_t idClient)`
- `bool processEndOfGame (uint8_t idClient)`
- `bool processWhoAmI (uint8_t idClient)`
- `bool gameStatePacket ()`
- `bool endGamePacket (bool isWin)`
- `std::vector< uint64_t > spawnPacket (size_t entity, const std::string prefabName)`
- `std::vector< uint64_t > deathPacket (size_t entity)`
- `bool serverStatusPacket ()`
- `bool isGameStarted () const`
- `bool allClientsReady () const`
- `uint32_t getSequenceNumber () const`
- `void setPacketManager (std::shared_ptr< pm::IPacketManager > packet)`
- `std::shared_ptr< pm::IPacketManager > getPacketManager () const`
- `void incrementSequenceNumber ()`
- `void setResourceManager (std::shared_ptr< ResourceManager > resourceManager)`
- `void clearEntityDeltaCache (uint8_t clientId, uint32_t entityId)`
- `void createPlayerEntities ()`
- `void processLobbyEvents ()`

Private Member Functions

- `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 > 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 > 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 > convertGameZoneComponent` (`std::shared_ptr< ecs::Registry > registry`, `ecs::Entity i`)

Private Attributes

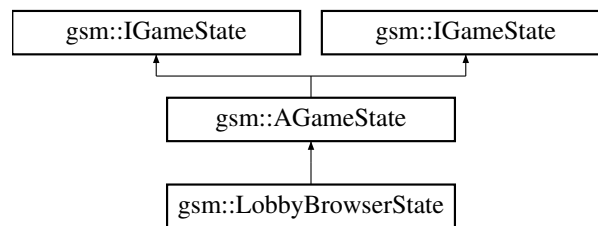
- `bool _isDebug`
- `std::shared_ptr< net::INetwork > _network`
- `std::vector< std::tuple< uint8_t, std::shared_ptr< net::INetworkEndpoint >, std::string > > _clients`
- `std::string _lobbyCode`
- `std::map< uint8_t, bool > _clientsReady`
- `std::shared_ptr< pm::IPacketManager > _packet`
- `uint32_t _sequenceNumber`
- `std::shared_ptr< std::queue< std::tuple< uint8_t, constants::EventType, double > > > _eventQueue`
- `std::queue< std::pair< std::shared_ptr< net::INetworkEndpoint >, std::vector< uint8_t > > > _incomingPackets`
- `std::mutex _packetMutex`
- `bool _gameStarted`
- `std::shared_ptr< ResourceManager > _resourceManager`
- `std::shared_ptr< gsm::GameStateMachine > _gsm`
- `std::chrono::steady_clock::time_point _lastGameStateTime`
- `float _statusUpdateTimer`
- `std::atomic_bool _running`
- `std::thread _networkThread`
- `std::thread _gameThread`
- `std::mutex _eventMutex`
- `ComponentDeltaTracker _deltaTracker`
- `std::vector< std::function< std::vector< uint64_t > (std::shared_ptr< ecs::Registry >, ecs::Entity) > > _convertFunctions`

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

- `/home/albane/epitech/tech3/r-type/ryanR-type/server/Lobby.hpp`
- `/home/albane/epitech/tech3/r-type/ryanR-type/server/ECSConversions.cpp`
- `/home/albane/epitech/tech3/r-type/ryanR-type/server/Lobby.cpp`

4.115 `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`

4.115.1 Member Function Documentation

4.115.1.1 `enter()`

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

Reimplemented from `gsm::AGameState`.

4.115.1.2 exit()

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

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

4.115.1.3 update()

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

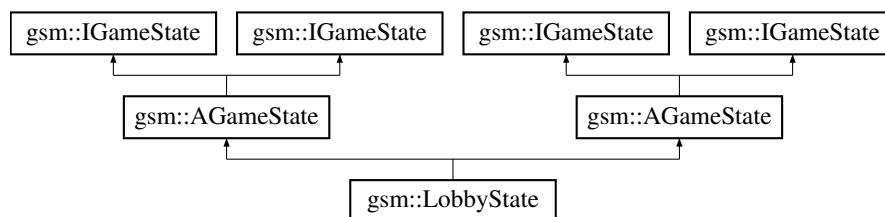
Reimplemented from [gsm::AGameState](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/LobbyBrowser/LobbyBrowser↔State.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/LobbyBrowser/LobbyBrowser↔State.cpp

4.116 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](#)

4.116.1 Member Function Documentation

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

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

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

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

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

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

4.116.1.3 [exit\(\)](#)

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

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

4.116.1.4 [update\(\)](#) [1/2]

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

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

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

4.117 rserv::LobbyStruct Struct Reference

Public Attributes

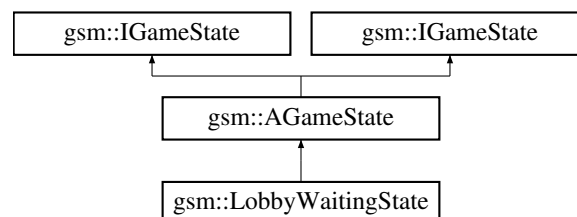
- std::string **_lobbyCode**
- std::vector< std::tuple< uint8_t, std::shared_ptr< [net::INetworkEndpoint](#) >, std::string > > **_clients**
- std::shared_ptr< [Lobby](#) > **_lobby**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/server/LobbyStruct.hpp

4.118 gsm::LobbyWaitingState Class Reference

Inheritance diagram for gsm::LobbyWaitingState:



Public Member Functions

- **LobbyWaitingState** (std::shared_ptr< [IGameStateMachine](#) > gsm, std::shared_ptr< [ResourceManager](#) > resourceManager, bool isLobbyMaster)
- 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 **updateUIStatus** ()
- void **setupLobbyMasterUI** ()
- void **setupPlayerUI** ()

Private Attributes

- std::unique_ptr< [MouseInputHandler](#) > **_mouseHandler**
- std::unique_ptr< [ui::UIManager](#) > **_uiManager**
- std::shared_ptr< [ui::UILayout](#) > **_centerLayout**
- std::shared_ptr< [ui::Text](#) > **_lobbyCodeText**
- std::shared_ptr< [ui::Text](#) > **_statusText**
- std::shared_ptr< [ui::Button](#) > **_startGameButton**
- bool **_isLobbyMaster**

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**

4.118.1 Member Function Documentation

4.118.1.1 enter()

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

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

4.118.1.2 exit()

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

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

4.118.1.3 update()

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

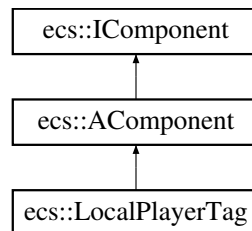
Reimplemented from [gsm::AGameState](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/LobbyWaiting/LobbyWaitingState.↔hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/LobbyWaiting/LobbyWaitingState.↔cpp

4.119 ecs::LocalPlayerTag Class Reference

Inheritance diagram for ecs::LocalPlayerTag:

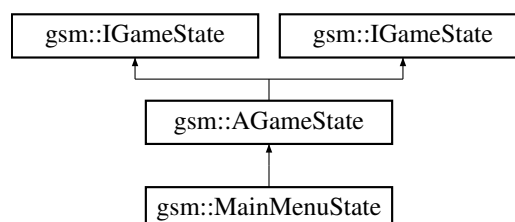


The documentation for this class was generated from the following file:

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

4.120 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 [updateUIStatus](#) ()
- void [checkLobbyConnectionTransition](#) ()

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::shared_ptr< [ui::Button](#) > [_requestCodeButton](#)
- std::shared_ptr< [ui::Button](#) > [_lobbyConnectButton](#)
- 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::Button](#) > [_infiniteButton](#)
- std::shared_ptr< [ui::TextInput](#) > [_ipInput](#)
- std::shared_ptr< [ui::TextInput](#) > [_portInput](#)
- std::shared_ptr< [ui::TextInput](#) > [_lobbyCodeInput](#)
- std::shared_ptr< [ui::Text](#) > [_connectionStatusText](#)
- std::shared_ptr< [ui::Text](#) > [_serverStatusText](#)
- std::shared_ptr< [ui::Background](#) > [_background](#)
- bool [_previousLobbyConnectedState](#)
- bool [_previousLobbyMasterState](#)

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`

4.120.1 Member Function Documentation

4.120.1.1 `enter()`

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

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

4.120.1.2 `exit()`

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

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

4.120.1.3 `update()`

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

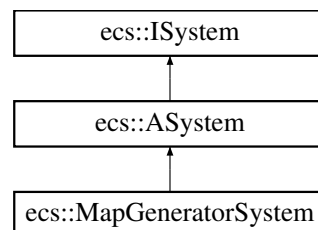
Reimplemented from [gsm::AGameState](#).

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

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

4.121 [ecs::MapGeneratorSystem](#) Class Reference

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



Public Member Functions

- **MapGeneratorSystem** (unsigned int seed=42)
- 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 **generateObstaclesAt** (float x, std::shared_ptr< [ResourceManager](#) > resourceManager, std::shared_ptr< [Registry](#) > registry)
- float **noise** (float x)

Private Attributes

- unsigned int **_seed**
- std::mt19937 **_rng**
- float **_lastGeneratedX**
- const float **_generationStep**
- const float **_startGenerationX**

4.121.1 Member Function Documentation

4.121.1.1 update()

```
void ecs::MapGeneratorSystem::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/ryanR-type/client/systems/map/MapGeneratorSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/map/MapGeneratorSystem.cpp

4.122 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 **createGameEndEntity** (float mapLength)
- 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

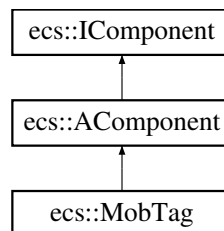
- std::shared_ptr< [EntityPrefabManager](#) > **_prefabManager**
- std::shared_ptr< [ecs::Registry](#) > **_registry**
- [ecs::EntityCreationContext](#) **_creationContext**
- nlohmann::json **_mapJson**

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

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

4.123 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/ryanR-type/common/components/tags/MobTag.hpp

4.124 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/ryanR-type/client/input/MouseInputHandler.hpp

4.125 MouseButtonHandler Class Reference

Public Member Functions

- **MouseButtonHandler** (std::shared_ptr< [ResourceManager](#) > resourceManager)
- std::optional< [MouseClickedInfo](#) > **pollMouseClicked** ()
- [math::Vector2f](#) **getMousePosition** () const
- [math::Vector2f](#) **getWorldMousePosition** () const
- [math::Vector2f](#) **getNormalizedMousePosition** () const
- bool **isMouseButtonPressed** (int button) const

Private Attributes

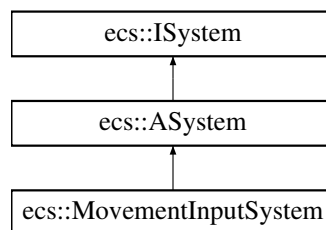
- std::weak_ptr< [ResourceManager](#) > **_resourceManager**

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

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

4.126 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)
- bool **isPlayerAlive** (std::shared_ptr< [Registry](#) > registry, Entity entityId) const

Private Attributes

- `bool _wasMovingLastFrame = false`

4.126.1 Member Function Documentation

4.126.1.1 update()

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

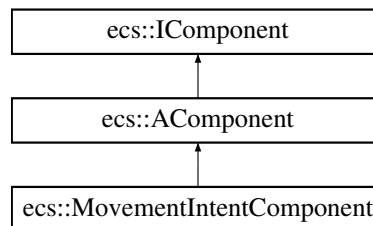
Implements [ecs::ASystem](#).

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

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

4.127 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

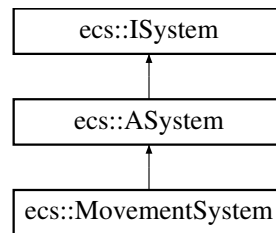
- [math::Vector2f](#) **_direction**
- bool **_active**

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

- `/home/albane/epitech/tech3/r-type/ryanR-type/common/components/temporary/MovementIntentComponent.↔.hpp`

4.128 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**

4.128.1 Member Function Documentation

4.128.1.1 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/ryanR-type/common/systems/movement/MovementSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/systems/movement/MovementSystem.cpp

4.129 ecs::MultiShotPattern Struct Reference

Public Member Functions

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

Public Attributes

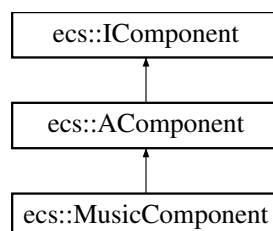
- int **shotCount** = 1
- float **angleSpread** = 0.0f
- float **offsetDistance** = 0.0f

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

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

4.130 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

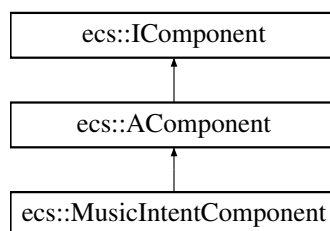
- std::string **_currentMusic**
- MusicState **_state**
- float **_volume**
- bool **_loop**

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

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

4.131 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

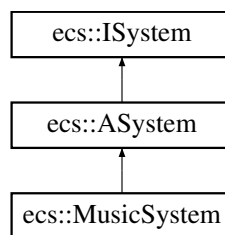
- MusicAction **_action**
- std::string **_musicPath**
- float **_volume**

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

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

4.132 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

4.132.1 Member Function Documentation**4.132.1.1 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/ryanR-type/client/systems/audio/MusicSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/audio/MusicSystem.cpp

4.133 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/ryanR-type/client/ClientNetwork.hpp

4.134 ecs::NetworkHealthState Struct Reference

Public Attributes

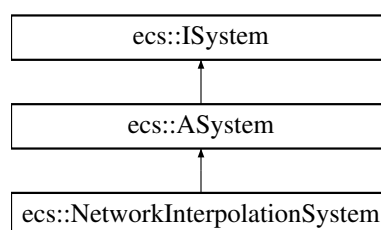
- uint32_t **health**
- uint32_t **baseHealth**
- std::chrono::steady_clock::time_point **timestamp**

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

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

4.135 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)

4.135.1 Member Function Documentation

4.135.1.1 update()

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

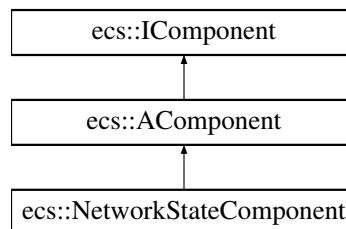
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/network/NetworkInterpolationSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/network/NetworkInterpolationSystem.cpp

4.136 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/ryanR-type/client/interpolation/NetworkStateComponent.hpp

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

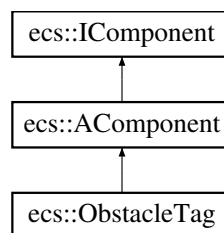
- [math::Vector2f](#) **position**
- float **rotation**
- [math::Vector2f](#) **scale**
- std::chrono::steady_clock::time_point **timestamp**

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

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

4.138 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/ryanR-type/common/components/tags/ObstacleTag.hpp

4.139 math::OrientedRect Class Reference

Public Member Functions

- **OrientedRect** ([Vector2f](#) center, [Vector2f](#) size, float rotation)
- **OrientedRect** ([OrientedRect](#) const &other)
- [Vector2f](#) **getCenter** () const
- void **setCenter** ([Vector2f](#) center)
- [Vector2f](#) **getSize** () const
- void **setSize** ([Vector2f](#) size)
- float **getRotation** () const
- void **setRotation** (float rotation)
- std::vector< [Vector2f](#) > **getCorners** () const
- [Vector2f](#) **getAxisX** () const
- [Vector2f](#) **getAxisY** () const
- bool **intersects** ([OrientedRect](#) const &other) const
- [OrientedRect](#) & **operator=** ([OrientedRect](#) const &other)

Private Member Functions

- float **projectPoint** ([Vector2f](#) point, [Vector2f](#) axis) const
- bool **overlapOnAxis** ([OrientedRect](#) const &other, [Vector2f](#) axis) const

Private Attributes

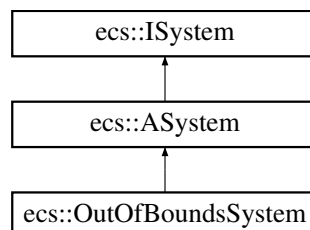
- [Vector2f](#) **_center**
- [Vector2f](#) **_size**
- float **_rotation**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/types/OrientedRect.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/types/OrientedRect.cpp

4.140 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](#)

4.140.1 Member Function Documentation

4.140.1.1 [update\(\)](#)

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

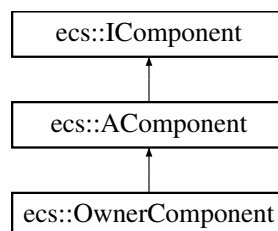
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/systems/bounds/OutOfBoundsSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/systems/bounds/OutOfBoundsSystem.cpp

4.141 [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

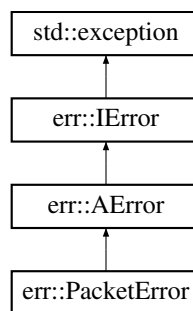
- ecs::Entity **_owner**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/OwnerComponent.hpp

4.142 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**

4.142.1 Member Function Documentation

4.142.1.1 getType()

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

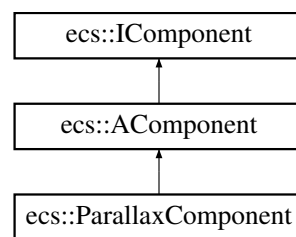
Implements [err::AError](#).

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

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

4.143 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/ryanR-type/client/components/rendering/ParallaxComponent.hpp

4.144 ecs::ParallaxLayer Struct Reference

Public Attributes

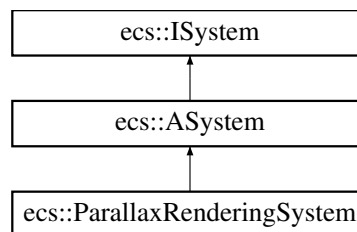
- std::string **name**
- std::string **filePath**
- float **speedMultiplier**
- [math::Vector2f](#) **scale**
- ParallaxScaleMode **scaleMode**
- [math::Vector2f](#) **sourceSize**
- bool **repeat**
- int **zIndex**
- [math::Vector2f](#) **currentOffset**

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

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

4.145 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

4.145.1 Member Function Documentation

4.145.1.1 update()

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

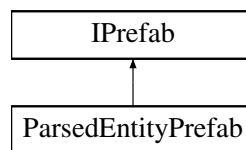
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/ParallaxRenderingSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/ParallaxRenderingSystem.cpp

4.146 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](#) > ®istry, const std::shared_ptr< [ecs::IEntityFactory](#) > &factory, const [ecs::EntityCreationContext](#) &context=ecs::EntityCreationContext↔::forLocalClient()) override
- ecs::Entity **instantiate** (const std::shared_ptr< [ecs::Registry](#) > ®istry) override

Private Member Functions

- void **addParsedComponents** (const std::shared_ptr< [ecs::Registry](#) > ®istry, 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**

4.146.1 Member Function Documentation

4.146.1.1 instantiate() [1/2]

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

Implements [IPrefab](#).

4.146.1.2 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/ryanR-type/common/Prefab/ParsedEntityPrefab.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/Prefab/ParsedEntityPrefab.cpp

4.147 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

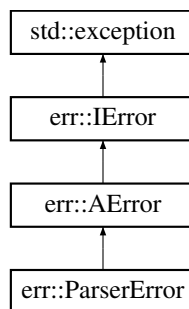
- `std::shared_ptr< EntityParser > _entityParser`
- `std::shared_ptr< MapParser > _mapParser`
- `std::shared_ptr< EntityPrefabManager > _prefabManager`
- `std::shared_ptr< std::map< std::string, std::pair< std::type_index, std::vector< Field > > > > _componentDefinitions`
- `std::map< std::type_index, 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/ryanR-type/common/Parser/Parser.hpp`
- `/home/albane/epitech/tech3/r-type/ryanR-type/common/Parser/ComponentParserCreators.cpp`
- `/home/albane/epitech/tech3/r-type/ryanR-type/common/Parser/Parser.cpp`

4.148 `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, ErrorCodes 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`

4.148.1 Member Function Documentation

4.148.1.1 `getType()`

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

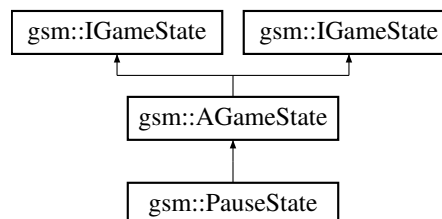
Implements `err::AError`.

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

- `/home/albane/epitech/tech3/r-type/ryanR-type/common/Error/ParserError.hpp`
- `/home/albane/epitech/tech3/r-type/ryanR-type/common/Error/ParserError.cpp`

4.149 `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](#)

4.149.1 Member Function Documentation

4.149.1.1 [enter\(\)](#)

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

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

4.149.1.2 [exit\(\)](#)

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

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

4.149.1.3 [update\(\)](#)

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

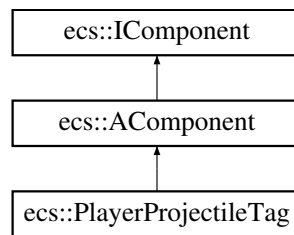
Reimplemented from [gsm::AGameState](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/Pause/PauseState.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/Pause/PauseState.cpp

4.150 ecs::PlayerProjectileTag Class Reference

Inheritance diagram for ecs::PlayerProjectileTag:

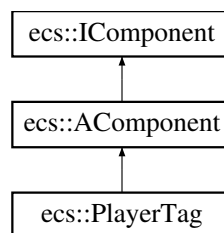


The documentation for this class was generated from the following file:

- /home/albane/epitech/tech3/r-type/ryanR-type/common/components/tags/PlayerProjectileTag.hpp

4.151 ecs::PlayerTag Class Reference

Inheritance diagram for ecs::PlayerTag:

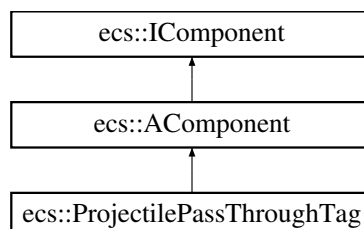


The documentation for this class was generated from the following file:

- /home/albane/epitech/tech3/r-type/ryanR-type/common/components/tags/PlayerTag.hpp

4.152 ecs::ProjectilePassThroughTag Class Reference

Inheritance diagram for ecs::ProjectilePassThroughTag:

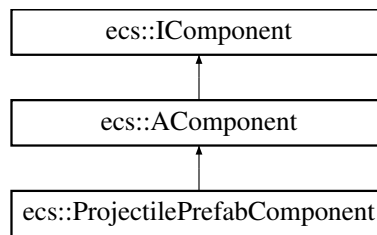


The documentation for this class was generated from the following file:

- /home/albane/epitech/tech3/r-type/ryanR-type/common/components/tags/ProjectilePassThroughTag.hpp

4.153 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

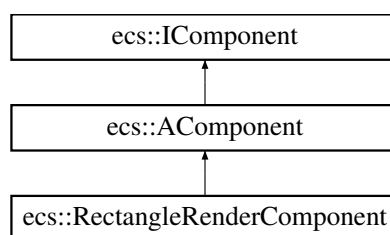
- std::string **_prefabName**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/ProjectilePrefabComponent.↵
hpp

4.154 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

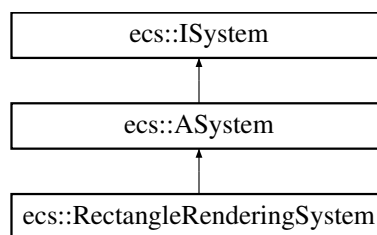
- [gfx::color_t](#) _color
- std::pair< float, float > _size

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/components/rendering/RectangleRenderComponent.↔
hpp

4.155 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

4.155.1 Member Function Documentation**4.155.1.1 update()**

```

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

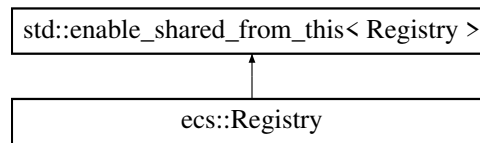
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/RectangleRenderingSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/RectangleRenderingSystem.cpp

4.156 ecs::Registry Class Reference

Inheritance diagram for ecs::Registry:



Public Member Functions

- **Registry** (Entity nextEntityId)
- 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)
- 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**
- std::recursive_mutex **_mutex**

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

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

4.157 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/ryanR-type/common/InputMapping/InputMapping.hpp

4.158 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

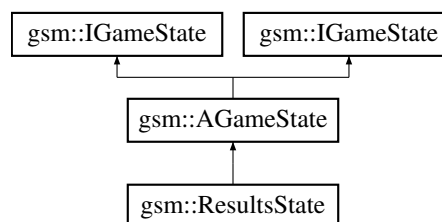
- std::unordered_map< size_t, std::shared_ptr< void > > **resources**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/resourceManager/ResourceManager.hpp

4.159 gsm::ResultsState Class Reference

Inheritance diagram for gsm::ResultsState:



Public Member Functions

- **ResultsState** (std::shared_ptr< [IGameStateMachine](#) > gsm, std::shared_ptr< [ResourceManager](#) > resourceManager, bool isWin)
- 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 Attributes

- bool [_isWin](#)
- std::unique_ptr< [ui::UIManager](#) > [_uiManager](#)
- std::shared_ptr< [ui::Text](#) > [_resultText](#)

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::ResultsState::enter () [override], [virtual]
```

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

4.159.1.2 [exit](#)()

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

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

4.159.1.3 update()

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

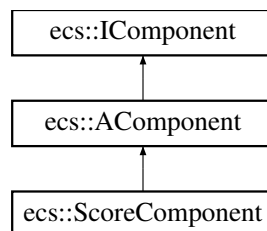
Reimplemented from [gsm::AGameState](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/Results/ResultsState.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/Results/ResultsState.cpp

4.160 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/ryanR-type/common/components/permanent/ScoreComponent.hpp

4.161 gsm::ScoreFeedback Struct Reference

Public Attributes

- std::string **text**
- float **lifetime**
- float **maxLifetime**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/InGame/InGameState.hpp

4.162 ScoreIntentComponent Class Reference

Public Member Functions

- **ScoreIntentComponent** (int score=0)
- int **getScore** () const
- void **setScore** (int newScore)

Private Attributes

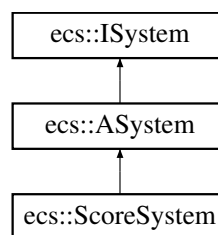
- int **_score**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/components/temporary/ScoreIntentComponent.hpp

4.163 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

4.163.1 Member Function Documentation

4.163.1.1 update()

```

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

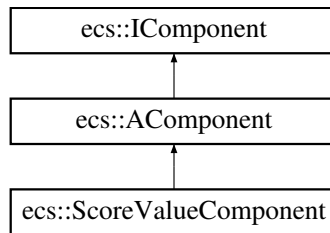
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/systems/score/ScoreSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/systems/score/ScoreSystem.cpp

4.164 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

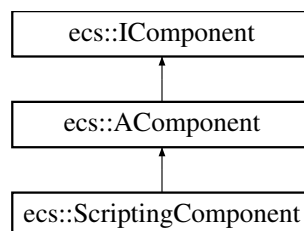
- int **_scoreValue**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/ScoreValueComponent.hpp

4.165 ecs::ScriptingComponent Class Reference

Inheritance diagram for ecs::ScriptingComponent:



Public Member Functions

- **ScriptingComponent** (std::string script_name="", std::vector< std::string > additionalFunctions=std::vector< std::string >(), std::shared_ptr< sol::state > lua=nullptr, size_t entityId=0)
- void **init** (sol::state &lua, size_t entityId)
- const std::string & **getScriptName** () const
- void **setEnvironment** (const sol::table &table)
- sol::table **getEnvironment** () const
- bool **hasFunction** (const std::string &name) const
- void **addFunction** (const std::string &name, const sol::function &function)
- sol::function **getFunction** (const std::string &name) const
- std::vector< std::string > **getFunctionNames** () const
- void **removeFunction** (const std::string &name)
- bool **isInitialized** () const
- void **setInitialized** (bool value)

Private Attributes

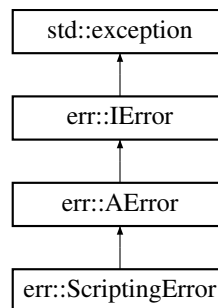
- `std::string _scriptName`
- `std::vector< std::string > _additionalFunctions`
- `sol::table _env`
- `std::map< std::string, sol::function > _functions`
- `bool _initialized = false`

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

- `/home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/ScriptingComponent.hpp`
- `/home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/ScriptingComponent.cpp`

4.166 `err::ScriptingError` Class Reference

Inheritance diagram for `err::ScriptingError`:



Public Types

- `enum ErrorCode { UNKNOWN = 1000 , LOAD_FAILED = 1001 , RUN_FAILED = 1002 }`

Public Member Functions

- **`ScriptingError`** (`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`

4.166.1 Member Function Documentation

4.166.1.1 getType()

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

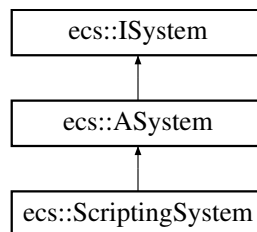
Implements [err::AError](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/Error/ScriptingError.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/Error/ScriptingError.cpp

4.167 ecs::ScriptingSystem Class Reference

Inheritance diagram for ecs::ScriptingSystem:



Public Member Functions

- void [update](#) (std::shared_ptr< [ResourceManager](#) > resourceManager, std::shared_ptr< [Registry](#) > reg, 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 [bindAPI](#) ()

Private Attributes

- sol::state [lua](#)
- std::shared_ptr< [Registry](#) > [registry](#)
- std::shared_ptr< [ResourceManager](#) > [resourceManager](#)

4.167.1 Member Function Documentation

4.167.1.1 update()

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

Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/systems/scripting/ScriptingSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/systems/scripting/ScriptingApiFunctions.cpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/systems/scripting/ScriptingSystem.cpp

4.168 rserv::Server Class Reference

Public Member Functions

- void **init** ()
- void **start** ()
- void **stop** ()
- void **setConfig** (std::shared_ptr< [ServerConfig](#) > config)
- std::shared_ptr< [ServerConfig](#) > **getConfig** () const
- uint16_t **getPort** () const
- void **setPort** (uint16_t port)
- int **getState** () const
- void **setState** (int state)
- void **initRessourceManager** (std::shared_ptr< [Lobby](#) > lobby)
- **operator int** () const noexcept
- std::shared_ptr< [net::INetwork](#) > **getNetwork** () const
- void **setNetwork** (std::shared_ptr< [net::INetwork](#) > network)
- void **onClientConnected** (uint8_t idClient)
- void **onClientDisconnected** (uint8_t idClient)
- void **onPacketReceived** (uint8_t idClient, const [pm::IPacketManager](#) &packet)
- std::vector< uint8_t > **getConnectedClients** () const
- std::vector< std::shared_ptr< [net::INetworkEndpoint](#) > > **getConnectedClientEndpoints** () const
- size_t **getClientCount** () const
- void **processIncomingPackets** ()
- bool **processConnections** (std::pair< std::shared_ptr< [net::INetworkEndpoint](#) >, std::vector< uint8_t > > client)
- bool **processDisconnections** (uint8_t idClient)
- bool **requestCode** (const [net::INetworkEndpoint](#) &endpoint)
- bool **processConnectToLobby** (std::pair< std::shared_ptr< [net::INetworkEndpoint](#) >, std::vector< uint8_t > > payload)
- bool **processMasterStart** (std::pair< std::shared_ptr< [net::INetworkEndpoint](#) >, std::vector< uint8_t > > payload)
- bool **connectionPacket** (const [net::INetworkEndpoint](#) &endpoint)
- bool **canStartPacket** (std::vector< std::shared_ptr< [net::INetworkEndpoint](#) > > endpoints)
- bool **serverStatusPacket** ()
- bool **sendCodeLobbyPacket** (const [net::INetworkEndpoint](#) &endpoint)
- bool **lobbyConnectValuePacket** (const [net::INetworkEndpoint](#) &endpoint, bool canConnect)
- uint32_t **getSequenceNumber** () const
- std::shared_ptr< [pm::IPacketManager](#) > **getPacketManager** () const
- std::shared_ptr< [pm::IPacketManager](#) > **createNewPacketManager** ()
- uint32_t **getNextEntityId** ()
- void **incrementSequenceNumber** ()

Private Member Functions

- void **loadNetworkLibrary** ()
- void **loadBufferLibrary** ()
- void **loadPacketLibrary** ()

Private Attributes

- [DLLoader](#)< createNetworkLib_t > **_networloader**
- [DLLoader](#)< createBuffer_t > **_bufferloader**
- [DLLoader](#)< createPacket_t > **_packetloader**
- std::shared_ptr< [ServerConfig](#) > **_config**
- std::shared_ptr< [net::INetwork](#) > **_network**
- std::shared_ptr< [IBuffer](#) > **_buffer**
- std::shared_ptr< [pm::IPacketManager](#) > **_packet**
- uint8_t **_nextClientId**
- uint32_t **_sequenceNumber**
- uint32_t **_nextEntityId**
- std::vector< std::tuple< uint8_t, std::shared_ptr< [net::INetworkEndpoint](#) >, std::string > > **_clients**
- std::map< uint8_t, bool > **_clientsReady**
- std::vector< std::shared_ptr< [LobbyStruct](#) > > **_lobbyThreads**
- std::vector< std::shared_ptr< [Lobby](#) > > **_lobbies**
- std::map< uint8_t, std::shared_ptr< [Lobby](#) > > **_clientToLobby**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/server/Server.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/Server.cpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/ServerLibsLoading.cpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/ServerReceivePacket.cpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/ServerSentPacket.cpp

4.169 rserv::ServerConfig Class Reference

Public Member Functions

- int **getState** () const
- void **setPort** (uint16_t port)
- uint16_t **getPort** () const
- void **setState** (int state)
- std::string **getIp** () const
- void **setIp** (std::string ip)
- void **setIsDebug** (bool isDebug)
- bool **getIsDebug** () const

Private Attributes

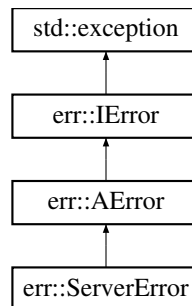
- `int _state`
- `uint16_t _port`
- `std::string _ip`
- `bool _isDebug`

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

- `/home/albane/epitech/tech3/r-type/ryanR-type/server/ServerConfig.hpp`
- `/home/albane/epitech/tech3/r-type/ryanR-type/server/ServerConfig.cpp`

4.170 `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`

4.170.1 Member Function Documentation

4.170.1.1 `getType()`

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

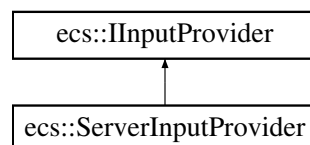
Implements [err::AError](#).

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

- `/home/albane/epitech/tech3/r-type/ryanR-type/common/Error/ServerError.hpp`
- `/home/albane/epitech/tech3/r-type/ryanR-type/common/Error/ServerError.cpp`

4.171 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 registerClient (size_t clientID)`
- `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::map< size_t, std::map< ecs::InputAction, std::chrono::steady_clock::time_point > > **_clientInputTimestamps**
- std::set< size_t > **_registeredClients**
- std::vector< InputHandler > **_inputHandlers**

Static Private Attributes

- static constexpr std::chrono::milliseconds **INPUT_TIMEOUT** = std::chrono::milliseconds(200)

Additional Inherited Members

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

- using **event_t** = gfx::EventType

4.171.1 Member Function Documentation

4.171.1.1 getActionAxis()

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

Implements [ecs::IInputProvider](#).

4.171.1.2 getAxisValue()

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

Implements [ecs::IInputProvider](#).

4.171.1.3 getInputMapping()

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

Implements [ecs::IInputProvider](#).

4.171.1.4 isActionPressed()

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

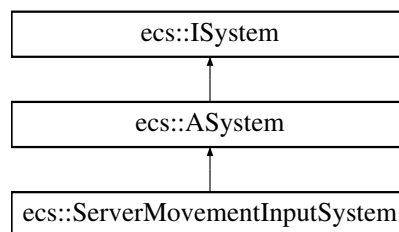
Implements [ecs::IInputProvider](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/server/initResourceManager/ServerInputProvider.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/initResourceManager/ServerInputProvider.cpp

4.172 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

4.172.1 Member Function Documentation

4.172.1.1 update()

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

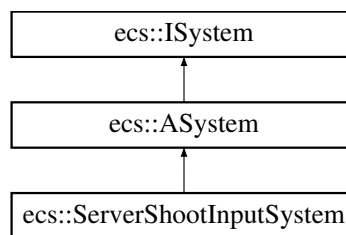
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/server/systems/input/ServerMovementInputSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/systems/input/ServerMovementInputSystem.cpp

4.173 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)

4.173.1 Member Function Documentation

4.173.1.1 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/ryanR-type/server/systems/input/ServerShootInputSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/systems/input/ServerShootInputSystem.cpp

4.174 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/ryanR-type/client/SettingsConfig.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/SettingsConfig.cpp

4.175 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

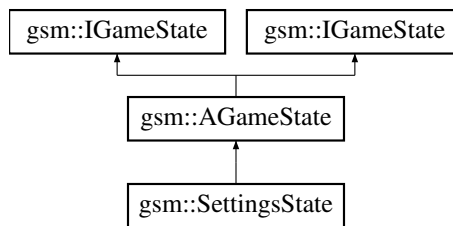
- std::shared_ptr< [ecs::InputMappingManager](#) > **_mappingManager**
- std::shared_ptr< [ecs::IInputProvider](#) > **_inputProvider**
- std::shared_ptr< [SettingsConfig](#) > **_settingsConfig**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/SettingsManager.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/SettingsManager.cpp

4.176 `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](#)

4.176.1 Member Function Documentation

4.176.1.1 enter()

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

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

4.176.1.2 exit()

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

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

4.176.1.3 update()

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

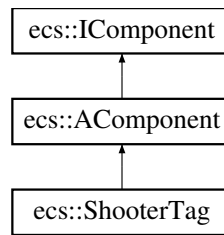
Reimplemented from [gsm::AGameState](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/Settings/SettingsState.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/Settings/SettingsState.cpp

4.177 ecs::ShooterTag Class Reference

Inheritance diagram for ecs::ShooterTag:

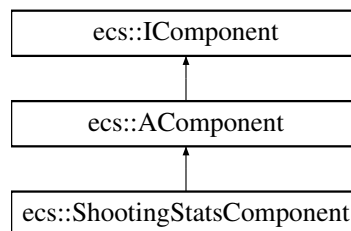


The documentation for this class was generated from the following file:

- /home/albane/epitech/tech3/r-type/ryanR-type/common/components/tags/ShooterTag.hpp

4.178 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

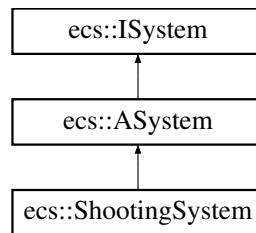
- float **_fireRate**
- [MultiShotPattern](#) **_multiShotPattern**
- float **_cooldownTimer**

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

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

4.179 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)

4.179.1 Member Function Documentation

4.179.1.1 update()

```

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

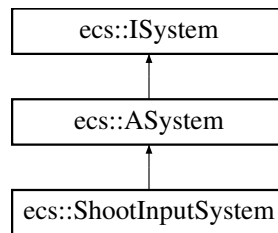
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/systems/shooting/ShootingSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/systems/shooting/ShootingSystem.cpp

4.180 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

Private Member Functions

- bool [isPlayerAlive](#) (std::shared_ptr< [Registry](#) > registry, Entity entityId) const

4.180.1 Member Function Documentation

4.180.1.1 update()

```

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

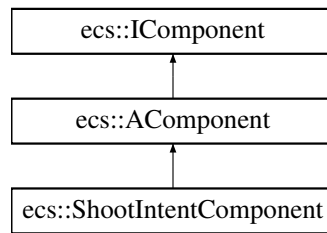
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/input/ShootInputSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/input/ShootInputSystem.cpp

4.181 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

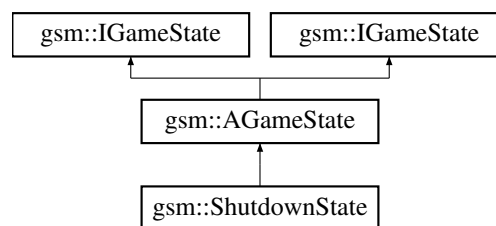
- float **_angle**
- [math::Vector2f](#) **_position**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/components/temporary/ShootIntentComponent.hpp

4.182 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**

4.182.1 Member Function Documentation

4.182.1.1 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/ryanR-type/server/gsm/states/scenes/Shutdown/ShutdownState.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/scenes/Shutdown/ShutdownState.cpp

4.183 Signal Class Reference

Static Public Member Functions

- static void **signalHandler** (int signum)
- static void **setupSignalHandlers** ()

Static Public Attributes

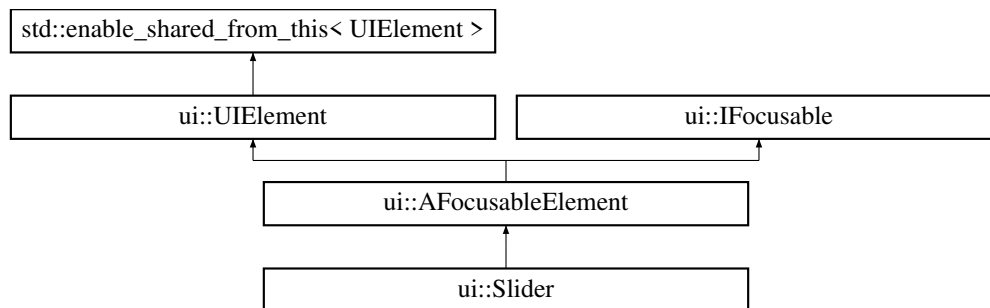
- static volatile sig_atomic_t **stopFlag** = 0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/Signal/Signal.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/Signal/Signal.cpp

4.184 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/abduction2002.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`

4.184.1 Member Function Documentation

4.184.1.1 `handleInput()`

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

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

4.184.1.2 `onActivated()`

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

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

4.184.1.3 `onNavigateLeft()`

```
bool ui::Slider::onNavigateLeft () [override], [virtual]
```

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

4.184.1.4 `onNavigateRight()`

```
bool ui::Slider::onNavigateRight () [override], [virtual]
```

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

4.184.1.5 render()

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

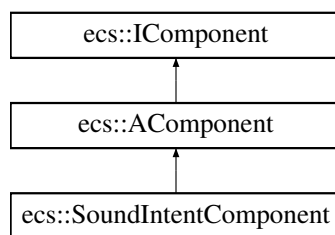
Reimplemented from [ui::UIElement](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/ui/elements/focusable/Slider.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/ui/elements/focusable/Slider.cpp

4.185 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

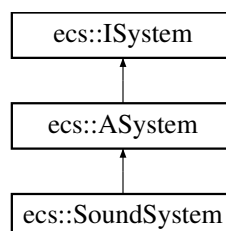
- std::string **_soundPath**
- float **_volume**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/components/temporary/SoundIntentComponent.hpp

4.186 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

4.186.1 Member Function Documentation

4.186.1.1 [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/ryanR-type/client/systems/audio/SoundSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/audio/SoundSystem.cpp

4.187 [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

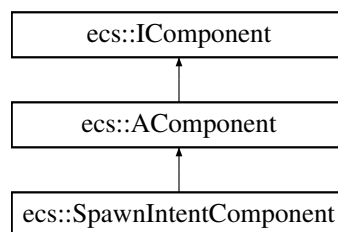
- float **_worldWidth**
- float **_worldHeight**
- float **_cellSize**
- float **_padding**
- float **_offsetX**
- float **_offsetY**
- size_t **_numCols**
- size_t **_numRows**
- std::vector< std::vector< Entity > > **_cells**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/SpatialGrid/SpatialGrid.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/SpatialGrid/SpatialGrid.cpp

4.188 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

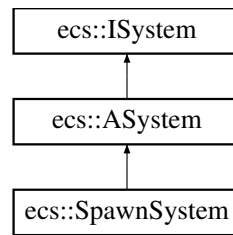
- std::string **_prefabName**
- [math::Vector2f](#) **_position**
- [EntityCreationContext](#) **_creationContext**
- float **_gameViewXTrigger**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/components/temporary/SpawnIntentComponent.↔.hpp

4.189 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

4.189.1 Member Function Documentation

4.189.1.1 update()

```

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

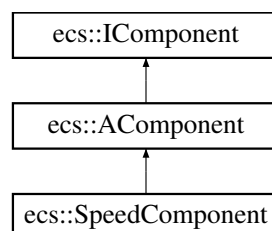
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/systems/spawn/SpawnSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/systems/spawn/SpawnSystem.cpp

4.190 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

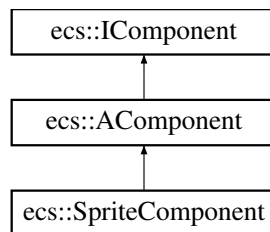
- float **_speed**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/components/permanent/SpeedComponent.hpp

4.191 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

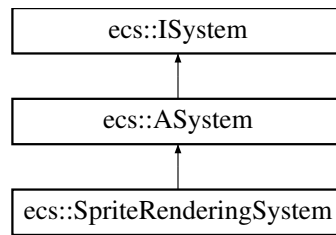
- std::string **_texturePath**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/components/rendering/SpriteComponent.hpp

4.192 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

4.192.1 Member Function Documentation

4.192.1.1 update()

```

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

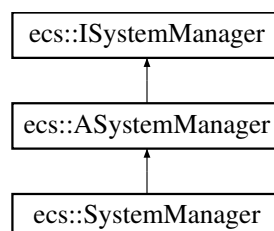
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/SpriteRenderingSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/SpriteRenderingSystem.cpp

4.193 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/ryanR-type/common/systems/systemManager/SystemManager.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/systems/systemManager/SystemManager.cpp

4.194 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

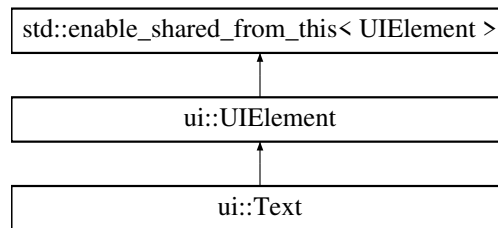
- std::unordered_map< std::string, std::function< bool(std::shared_ptr< [ecs::Registry](#) >, ecs::Entity)> > > **tagCheckers**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/systems/interactions/TagRegistry.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/systems/interactions/TagRegistry.cpp

4.195 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`

4.195.1 Member Function Documentation

4.195.1.1 `render()`

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

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

4.195.1.2 `setScale()`

```
void ui::Text::setScale (
    UIScale scale) [override], [virtual]
```

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

4.195.1.3 update()

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

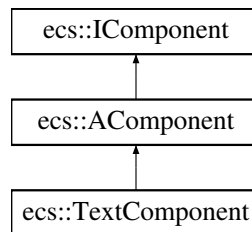
Reimplemented from [ui::UIElement](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/ui/elements/Text.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/ui/elements/Text.cpp

4.196 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

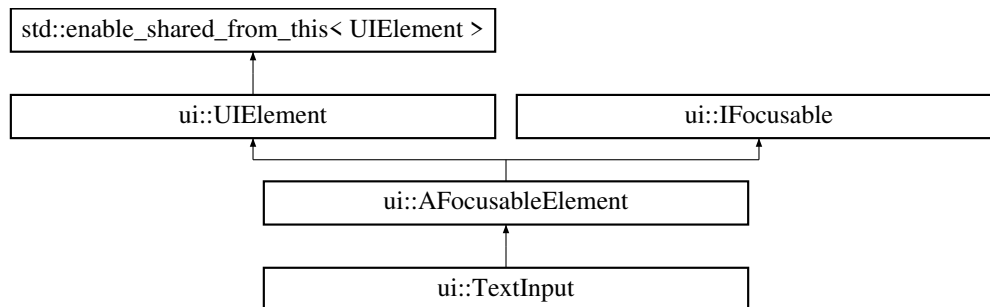
- std::string **_text**
- std::string **_fontPath**
- [gfx::color_t](#) **_color**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/components/rendering/TextComponent.hpp

4.197 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/abduction2002.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**

4.197.1 Member Function Documentation

4.197.1.1 handleInput()

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

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

4.197.1.2 render()

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

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

4.197.1.3 update()

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

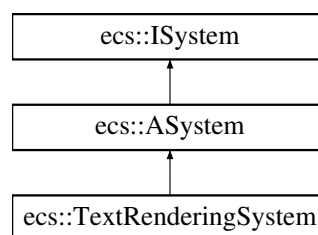
Reimplemented from [ui::UIElement](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/ui/elements/focusable/TextInput.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/ui/elements/focusable/TextInput.cpp

4.198 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

4.198.1 Member Function Documentation

4.198.1.1 update()

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

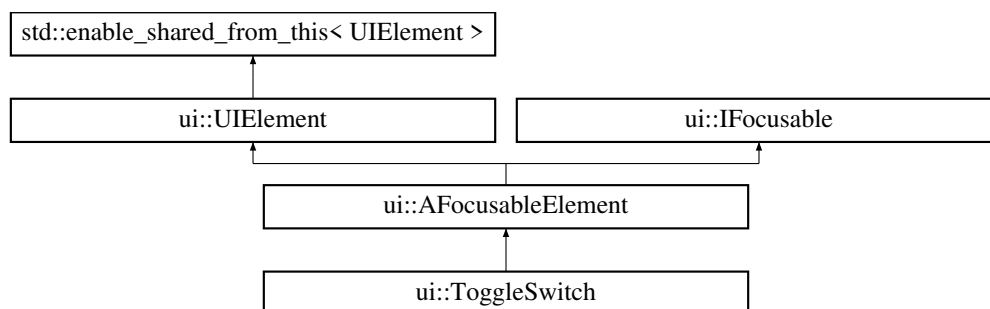
Implements [ecs::ASystem](#).

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/TextRenderingSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/TextRenderingSystem.cpp

4.199 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/abduction2002.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**

4.199.1 Member Function Documentation

4.199.1.1 containsPoint()

```
bool ui::ToggleSwitch::containsPoint (
    const math::Vector2f & point) const    [override], [virtual]
```

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

4.199.1.2 `handleInput()`

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

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

4.199.1.3 `render()`

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

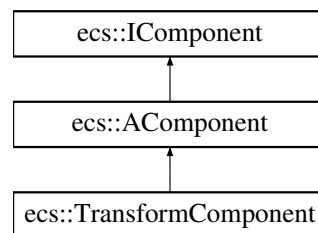
Reimplemented from [ui::UIElement](#).

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

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

4.200 `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/ryanR-type/common/components/permanent/TransformComponent.hpp`

4.201 ecs::Transition Struct Reference

Public Attributes

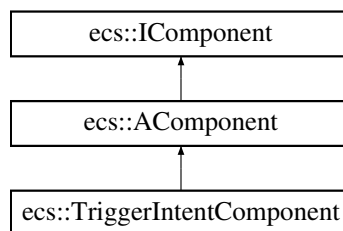
- std::string **from**
- std::string **to**
- std::vector< [AnimationCondition](#) > **conditions**
- bool **playRewind** = false

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/components/rendering/AnimationComponent.hpp

4.202 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

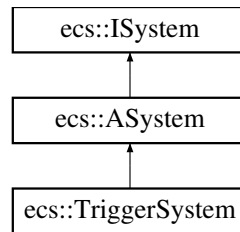
- Entity **_self**
- Entity **_other**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/components/temporary/TriggerIntentComponent.↵
hpp

4.203 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

4.203.1 Member Function Documentation

4.203.1.1 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/ryanR-type/common/systems/interactions/TriggerSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/systems/interactions/TriggerSystem.cpp

4.204 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

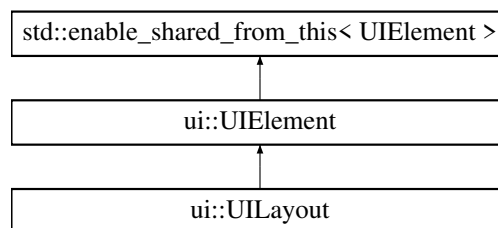
- `std::weak_ptr< ResourceManager > _resourceManager`
- `math::Vector2f _position`
- `math::Vector2f _size`
- `bool _visible = true`
- `UIState _state = UIState::Normal`
- `UIScale _scale = UIScale::Normal`
- `std::weak_ptr< UIElement > _parent`
- `std::vector< std::shared_ptr< UIElement > > _children`
- `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/ryanR-type/client/ui/elements/base/UIElement.hpp`
- `/home/albane/epitech/tech3/r-type/ryanR-type/client/ui/elements/base/UIElement.cpp`

4.205 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`

4.205.1 Member Function Documentation

4.205.1.1 `render()`

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

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

4.205.1.2 `setScale()`

```
void ui::UILayout::setScale (
    UIScale scale) [override], [virtual]
```

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

4.205.1.3 `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/ryanR-type/client/ui/core/UILayout.hpp`
- `/home/albane/epitech/tech3/r-type/ryanR-type/client/ui/core/UILayout.cpp`

4.206 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/ryanR-type/client/ui/manager/UIManager.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/ui/manager/UIManager.cpp

4.207 ui::UINavigationController 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/ryanR-type/client/ui/navigation/UINavigationController.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/ui/navigation/UINavigationController.cpp

4.208 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)

Static Public Member Functions

- static std::string **createAlphaNumericCode** ()

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/Utils.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/Utils.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/Utils.cpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/Utils.cpp

4.209 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

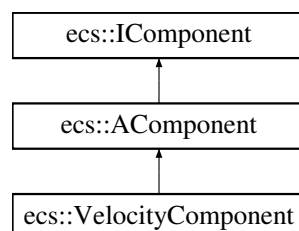
- float **_x**
- float **_y**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/types/Vector2f.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/types/Vector2f.cpp

4.210 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/ryanR-type/common/components/permanent/VelocityComponent.hpp

4.211 `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/ryanR-type/common/ECS/entity/registry/Registry.hpp
- /home/albane/epitech/tech3/r-type/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 "../common/interfaces/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     std::string getLobbyCode() const;
00070     void setLobbyCode(std::string lobbyCode);
00071
00072     net::ConnectionState getConnectionState() const;
00073
00074     /* Packet Handling */
00075     void eventPacket(const constants::EventType &eventType, double depth);
00076     void disconnectionPacket();
00077     void connectionPacket();
00078     void sendWhoAmI();
00079     void requestCode();
00080     void sendLobbyConnection(std::string lobbyCode);
00081     void sendMasterStartGame();
00082
00083     void addToEventQueue(const NetworkEvent &event);
00084
00085     bool isConnected() const;
00086     bool isReady() const;
00087
00088     size_t getConnectedClients() const;
00089     size_t getReadyClients() const;
00090     uint8_t getClientId() const;
00091     bool getClientReadyStatus() const;
00092
00093     bool isConnectedToLobby() const;
00094     bool isLobbyMaster() const;
00095
00096     std::atomic<bool> _isConnected;
00097     std::atomic<bool> _ready;
00098     std::atomic<bool> _isConnectedToLobby;
00099     std::atomic<bool> _isLobbyMaster;
00100
00101     std::atomic<size_t> _connectedClients;
00102     std::atomic<size_t> _readyClients;
00103     std::atomic<uint8_t> _clientId;
00104     std::atomic<bool> _clientReadyStatus;
00105
00106     void setResourceManager(std::shared_ptr<ResourceManager> resourceManager);
00107     void setGameStateMachine(std::shared_ptr<gsm::IGameStateMachine> gsm);
00108     std::shared_ptr<gsm::IGameStateMachine> getGameStateMachine() const;
00109
00110     void redoServerEndpoint();
00111
00112     protected:
00113         std::pair<int, std::chrono::steady_clock::time_point> tryConnection(const int maxRetries, int
retryCount, std::chrono::steady_clock::time_point lastRetryTime);
00114         void handlePacketType(uint8_t type);
00115
00116     private:
00117         typedef void (ClientNetwork::*PacketHandler)();
00118         PacketHandler _packetHandlers[constants::MAX_INDEX_PACKET_TYPE];
00119
00120         void handleNoOp();
00121         void handleConnectionAcceptation();
00122         void handleGameState();
00123         void handleEndGame();
00124         void handleCanStart();
00125         void handleEntitySpawn();
00126         void handleEntityDeath();
00127         void handleWhoAmI();
00128         void handleServerStatus();
00129         void handleCode();
00130         void handleLobbyConnectValue();
00131
00132         typedef size_t (ClientNetwork::*ComponentParser)(const std::vector<uint64_t> &, size_t,
ecs::Entity);
00133         std::map<uint64_t, ComponentParser> _componentParsers;
00134
00135         size_t parsePlayerTagComponent(const std::vector<uint64_t> &payload, size_t index, ecs::Entity
entityId);
00136         size_t parseTransformComponent(const std::vector<uint64_t> &payload, size_t index, ecs::Entity
entityId);
00137         size_t parseSpeedComponent(const std::vector<uint64_t> &payload, size_t index, ecs::Entity
entityId);
00138         size_t parseHealthComponent(const std::vector<uint64_t> &payload, size_t index, ecs::Entity
entityId);

```

```

00139         size_t parseColliderComponent(const std::vector<uint64_t> &payload, size_t index, ecs::Entity
entityId);
00140         size_t parseShootingStatsComponent(const std::vector<uint64_t> &payload, size_t index,
ecs::Entity entityId);
00141         size_t parseScoreComponent(const std::vector<uint64_t> &payload, size_t index, ecs::Entity
entityId);
00142         size_t parseDamageComponent(const std::vector<uint64_t> &payload, size_t index, ecs::Entity
entityId);
00143         size_t parseLifetimeComponent(const std::vector<uint64_t> &payload, size_t index, ecs::Entity
entityId);
00144         size_t parseVelocityComponent(const std::vector<uint64_t> &payload, size_t index, ecs::Entity
entityId);
00145         size_t parseControllableTagComponent(const std::vector<uint64_t> &payload, size_t index,
ecs::Entity entityId);
00146         size_t parseEnemyProjectileTagComponent(const std::vector<uint64_t> &payload, size_t index,
ecs::Entity entityId);
00147         size_t parseGameZoneColliderTagComponent(const std::vector<uint64_t> &payload, size_t index,
ecs::Entity entityId);
00148         size_t parseMobTagComponent(const std::vector<uint64_t> &payload, size_t index, ecs::Entity
entityId);
00149         size_t parseObstacleTagComponent(const std::vector<uint64_t> &payload, size_t index,
ecs::Entity entityId);
00150         size_t parsePlayerProjectileTagComponent(const std::vector<uint64_t> &payload, size_t index,
ecs::Entity entityId);
00151         size_t parseScoreTagComponent(const std::vector<uint64_t> &payload, size_t index, ecs::Entity
entityId);
00152         size_t parseShooterTagComponent(const std::vector<uint64_t> &payload, size_t index,
ecs::Entity entityId);
00153         size_t parseProjectilePassThroughTagComponent(const std::vector<uint64_t> &payload, size_t
index, ecs::Entity entityId);
00154         size_t parseProjectilePrefabComponent(const std::vector<uint64_t> &payload, size_t index,
ecs::Entity entityId);
00155         size_t parseGameZoneComponent(const std::vector<uint64_t> &payload, size_t index, ecs::Entity
entityId);
00156
00157         DLoader<createNetworkLib_t> _networkloader;
00158         DLoader<createBuffer_t> _bufferloader;
00159         DLoader<createPacket_t> _packetloader;
00160
00161         std::shared_ptr<net::INetwork> _network;
00162         std::shared_ptr<IBuffer> _receptionBuffer;
00163         std::shared_ptr<IBuffer> _sendBuffer;
00164         std::shared_ptr<pm::IPacketManager> _packet;
00165
00166         std::shared_ptr<ResourceManager> _resourceManager;
00167         std::shared_ptr<gsm::IGameStateMachine> _gsm;
00168
00169         uint32_t _sequenceNumber;
00170         uint16_t _port;
00171         std::string _ip;
00172         std::string _name;
00173         std::vector<std::string> _clientNames;
00174         bool _isDebug;
00175
00176
00177         uint8_t _idClient;
00178         std::shared_ptr<net::INetworkEndpoint> _serverEndpoint;
00179
00180         std::queue<NetworkEvent> _eventQueue;
00181         std::mutex _queueMutex;
00182         std::condition_variable _queueCond;
00183
00184         std::unordered_map<size_t, ecs::Entity> _serverToLocalEntityMap;
00185
00186         std::string _lobbyCode;
00187         bool _shouldConnect;
00188     };
00189
00190 #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 "../common/interfaces/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     const gfx::color_t PANEL_BORDER = {0, 200, 255}; // Cyan border
00048
00049     // Slider Colors
00050     const gfx::color_t SLIDER_TRACK = {50, 50, 100}; // Dark blue track
00051     const gfx::color_t SLIDER_FILL = {0, 150, 255}; // Bright blue fill
00052     const gfx::color_t SLIDER_HANDLE = {0, 100, 200}; // Blue handle
00053     const gfx::color_t SLIDER_HANDLE_HOVER = {0, 150, 255}; // Bright blue hover
00054     const gfx::color_t SLIDER_HANDLE_FOCUSED = {0, 200, 255}; // Cyan focused
00055     const gfx::color_t SLIDER_LABEL = {255, 255, 255}; // White label
00056
00057     // Toggle Switch Colors
00058     const gfx::color_t TOGGLE_TRACK = {100, 50, 150}; // Purple track
00059     const gfx::color_t TOGGLE_HANDLE = {150, 100, 200}; // Light purple handle
00060     const gfx::color_t TOGGLE_HANDLE_HOVER = {200, 150, 255}; // Bright purple hover
00061     const gfx::color_t TOGGLE_HANDLE_FOCUSED = {255, 200, 255}; // Pinkish purple
00062     const gfx::color_t TOGGLE_ON = {150, 0, 200}; // Purple on
00063     const gfx::color_t TOGGLE_OFF = {50, 0, 100}; // Dark purple off
00064
00065     // General UI States
00066     const gfx::color_t UI_HOVER = {0, 150, 255}; // Bright blue hover
00067     const gfx::color_t UI_FOCUSED = {0, 200, 255}; // Cyan focused
00068     const gfx::color_t UI_DISABLED = {100, 100, 100}; // Gray disabled
00069 }
00070
00071 #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; }

```

```

00104
00105     bool isValid() const { return !_states.empty(); }
00106
00107     bool isAnimationFinished() const {
00108         auto clip = getCurrentClip();
00109         if (!clip) return true;
00110         if (clip->loop) return false;
00111         int currentFrame = static_cast<int>(_timer / clip->speed);
00112         if (_playRewind) {
00113             return currentFrame >= clip->frameCount;
00114         } else {
00115             return currentFrame >= clip->frameCount - 1;
00116         }
00117     }
00118
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     std::unordered_map<std::string, std::shared_ptr<AnimationClip> getStates() const {
00126         return _states;
00127     }
00128     private:
00129         std::unordered_map<std::string, std::shared_ptr<AnimationClip> _states;
00130         std::vector<Transition> _transitions;
00131         std::string _currentState;
00132         float _timer;
00133         bool _isPlaying;
00134         bool _playRewind;
00135         int _currentFrame;
00136         int _rewindStartFrame;
00137         math::FRect _frameRect;
00138         bool _stateJustChanged = false;
00139         float _minAnimationTime = 0.0f;
00140 };
00141
00142 } // namespace ecs
00143
00144 #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 "../common/interfaces/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,

```



```

00071         direction.getY() * speed * deltaTime
00072     );
00073     layer.currentOffset = math::Vector2f(
00074         layer.currentOffset.getX() + movement.getX(),
00075         layer.currentOffset.getY() + movement.getY()
00076     );
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/interfaces/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 "../common/interfaces/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 BACKGROUND MUSIC TAG_HPP_
00009  #define BACKGROUND MUSIC TAG_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 /* !BACKGROUND MUSIC TAG_HPP_ */

```

5.12 MusicIntentComponent.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** MusicIntentComponent
00006  */
00007
00008  #ifndef MUSIC INTENT COMPONENT_HPP_
00009  #define MUSIC INTENT COMPONENT_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 /* !MUSIC INTENT COMPONENT_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/deutanopia.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  const std::string WIN_TEXT = "YOU WIN!";
00074  const std::string LOSE_TEXT = "YOU LOSE!";
00075
00076  /* Home page input place holders */
00077  const std::string IP_PLACEHOLDER = "Enter an IP address";
00078  const std::string PORT_PLACEHOLDER = "Enter a port";
00079 }
00080
00081 #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";
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 HITBOXRENDERCOMPONENT = "HitboxRenderComponent";
00071      const std::string INTERACTIONCONFIGCOMPONENT = "InteractionConfigComponent";
00072      const std::string SCRIPTINGCOMPONENT = "ScriptingComponent";
00073      const std::string SCRIPT_PATH_FIELD = "scriptPath";
00074      const std::string ADDITIONAL_FUNCTIONS_FIELD = "additionalFunctions";
00075      const std::string SCORE_FIELD = "score";
00076      const std::string SCOREVALUE_FIELD = "scoreValue";
00077      const std::string DAMAGE_FIELD = "damage";
00078      const std::string HEALTH_FIELD = "health";
00079      const std::string TARGET_FIELD = "target";
00080      const std::string POSITION_FIELD = "position";
00081      const std::string OFFSET_FIELD = "offset";
00082      const std::string SCALE_FIELD = "scale";
00083      const std::string ROTATION_FIELD = "rotation";
00084      const std::string SPEED_FIELD = "speed";
00085      const std::string FILEPATH_FIELD = "filePath";
00086      const std::string ANIMATIONPATH_FIELD = "animationPath";
00087      const std::string FRAMEWIDTH_FIELD = "frameWidth";
00088      const std::string FRAMEHEIGHT_FIELD = "frameHeight";
00089      const std::string FRAMECOUNT_FIELD = "frameCount";
00090      const std::string STARTWIDTH_FIELD = "startWidth";
00091      const std::string STARHEIGHT_FIELD = "startHeight";
00092      const std::string SIZE_FIELD = "size";
00093      const std::string FIRERATE_FIELD = "fireRate";
00094      const std::string SHOTCOUNT_FIELD = "shotCount";
00095      const std::string ANGLEOFFSET_FIELD = "angleOffset";
00096      const std::string SPREADANGLE_FIELD = "spreadAngle";
00097      const std::string DEFAULTBEHAVIOR_FIELD = "defaultBehavior";
00098      const std::string ZIGZAGAMPLITUDE_FIELD = "zigzagAmplitude";
00099      const std::string ZIGZAGFREQUENCY_FIELD = "zigzagFrequency";
00100      const std::string DETECTIONRANGE_FIELD = "detectionRange";
00101      const std::string VERTICALDEADZONE_FIELD = "verticalDeadzone";
00102      const std::string STRAIGHT_LINE_VALUE = "StraightLine";
00103      const std::string ZIGZAG_VALUE = "Zigzag";
00104      const std::string VERTICAL_MIRROR_VALUE = "VerticalMirror";
00105      const std::string FOLLOW_RIGHT_VALUE = "FollowRight";
00106      const std::string WIDTH_FIELD = "width";
00107      const std::string HEIGHT_FIELD = "height";
00108      const std::string COLOR_FIELD = "color";
00109      const std::string R_FIELD = "r";
00110      const std::string G_FIELD = "g";
00111      const std::string B_FIELD = "b";
00112
00113      const std::string STATES_FIELD = "states";
00114      const std::string INITIALSTATE_FIELD = "initialState";
00115      const std::string TRANSITIONS_FIELD = "transitions";
00116      const std::string CONDITIONS_FIELD = "conditions";
00117      const std::string CONDITION_FIELD = "condition";
00118      const std::string PARAM_FIELD = "param";
00119      const std::string EQUALS_FIELD = "equals";
00120      const std::string FROM_FIELD = "from";
00121      const std::string BASESCROLLSPEED_FIELD = "baseScrollSpeed";
00122      const std::string DIRECTION_FIELD = "direction";
00123      const std::string LAYERS_FIELD = "layers";
00124      const std::string ZONERECT_FIELD = "zoneRect";
00125      const std::string NAME_FIELD = "name";
00126      const std::string SPEEDMULTIPLIER_FIELD = "speedMultiplier";
00127      const std::string SCALEMODE_FIELD = "scaleMode";
00128      const std::string SOURCESIZE_FIELD = "sourceSize";
00129      const std::string REPEAT_FIELD = "repeat";
00130      const std::string ZINDEX_FIELD = "zIndex";
00131      const std::string TO_FIELD = "to";
00132      const std::string REWIND_FIELD = "rewind";
00133      const std::string TEXTUREPATH_FIELD = "texturePath";
00134      const std::string LOOP_FIELD = "loop";
00135      const std::string SCALEMODE_FITSCREEN = "FIT_SCREEN";
00136      const std::string SCALEMODE_STRETCH = "STRETCH";

```

```

00137     const std::string SCALEMODE_MANUAL = "MANUAL";
00138     const std::string ENTITYPARTSCOMPONENT = "EntityPartsComponent";
00139     const std::string COLLISION_TYPE_SOLID = "Solid";
00140     const std::string COLLISION_TYPE_TRIGGER = "Trigger";
00141     const std::string COLLISION_TYPE_PUSH = "Push";
00142     const std::string COLLISION_TYPE_NONE = "None";
00143     const std::string X_FIELD = "x";
00144     const std::string Y_FIELD = "y";
00145
00146     const std::string PREFABNAME_FIELD = "prefabName";
00147     const std::string LIFETIMECOMPONENT = "LifetimeComponent";
00148     const std::string LIFETIME_FIELD = "lifetime";
00149     const std::string LIFESPANCOMPONENT = "LifeSpanComponent";
00150     const std::string LIFESPAN_FIELD = "lifespan";
00151     const std::string BACKGROUNDMUSICTAG = "BackGroundMusicTag";
00152     const std::string TEXT_FIELD = "text";
00153     const std::string FONTPATH_FIELD = "fontPath";
00154     const std::string SOUNDINTENTCOMPONENT = "SoundIntentComponent";
00155     const std::string SOUND_FILE_FIELD = "soundFile";
00156
00157     const std::string MAPPINGS_FIELD = "mappings";
00158     const std::string TAGS_FIELD = "tags";
00159     const std::string TOENTITY_FIELD = "toEntity";
00160     const std::string TOSELF_FIELD = "toSelf";
00161
00162     const std::string MUSICCOMPONENT = "MusicComponent";
00163     const std::string MUSICFILE_FIELD = "musicFile";
00164     const std::string VOLUME_FIELD = "volume";
00165     const std::string INITIALSTATEMUSIC_FIELD = "initialState";
00166     const std::string PLAYING_FIELD = "PLAYING";
00167     const std::string PAUSED_FIELD = "PAUSED";
00168     const std::string STOPPED_FIELD = "STOPPED";
00169     const std::string CHANGING_FIELD = "CHANGING";
00170
00171     const float MAX_HEIGHT = 1080.0f;
00172     const float MAX_WIDTH = 1920.0f;
00173     const float GAME_ZONE_BOUNDARY_THICKNESS = 100.0f;
00174
00175     const float SPATIAL_GRID_CELL_SIZE = 128.0f;
00176     const float SPATIAL_GRID_PADDING = 200.0f;
00177     const float OUT_OF_BOUNDS_MARGIN = 200.0f;
00178
00179     /* Map parsing constants */
00180     const std::string BACKGROUND_FIELD = "background";
00181     const std::string BACKGROUND_SCROLL_SPEED_FIELD = "scrollSpeed";
00182     const std::string MUSIC_FIELD = "music";
00183     const std::string POWERUPS_FIELD = "powerUps";
00184     const std::string OBSTACLES_FIELD = "obstacles";
00185     const std::string WAVES_FIELD = "waves";
00186     const std::string MAP_LENGTH_FIELD = "mapLength";
00187     const std::string POSITIONS_FIELD = "positions";
00188     const std::string TYPE_FIELD = "type";
00189     const std::string FROMX_FIELD = "fromX";
00190     const std::string FROMY_FIELD = "fromY";
00191     const std::string POSX_FIELD = "posX";
00192     const std::string POSY_FIELD = "posY";
00193     const std::string COUNT_FIELD = "count";
00194     const std::string GAMEXTRIGGER_FIELD = "gameXTrigger";
00195     const std::string DISTRIBUTIONX_FIELD = "distributionX";
00196     const std::string DISTRIBUTIONY_FIELD = "distributionY";
00197     const std::string ENEMIES_FIELD = "enemies";
00198     const std::string MIN_FIELD = "min";
00199     const std::string MAX_FIELD = "max";
00200     const math::Vector2f BACKGROUND_POSITION = math::Vector2f(0.0f, 0.0f);
00201     const float BACKGROUND_PARALLAX_DIRECTION_X = -1.0f;
00202     const float BACKGROUND_PARALLAX_DIRECTION_Y = 0.0f;
00203     const std::string EMPTY_PREFAB = "empty";
00204
00205     const std::string HORIZONTAL_LINE_TYPE = "horizontalLine";
00206     const std::string VERTICAL_LINE_TYPE = "verticalLine";
00207     const std::string UNIQUE_TYPE = "unique";
00208     const std::string RANDOM_TYPE = "random";
00209     const std::string UNIFORM_TYPE = "uniform";
00210
00211     /* Animation conditions */
00212     const std::string VELOCITY_UP_CONDITION = "isVelocityUp";
00213     const std::string VELOCITY_DOWN_CONDITION = "isVelocityDown";
00214
00215     /* Tags */
00216     const std::string CONTROLLABLETAG = "ControllableTag";
00217     const std::string PLAYERTAG = "PlayerTag";
00218     const std::string COLLIDERCOMPONENT = "ColliderComponent";
00219     const std::string MOBTAG = "MobTag";
00220     const std::string SHOOTERTAG = "ShooterTag";
00221     const std::string PLAYERPROJECTILETAG = "PlayerProjectileTag";
00222     const std::string ENEMYPROJECTILETAG = "EnemyProjectileTag";
00223     const std::string PROJECTILEPASSTHROUGHTAG = "ProjectilePassThroughTag";

```

```

00224     const std::string PARALLAXCOMPONENT = "ParallaxComponent";
00225     const std::string GAMEZONECOMPONENT = "GameZoneComponent";
00226     const std::string GAMEZONECOLLIDERTAG = "GameZoneColliderTag";
00227     const std::string OBSTACLETAG = "ObstacleTag";
00228     const std::string CLIENTEFFECTTAG = "ClientEffectTag";
00229
00230     /* Action constants */
00231     const std::string DEALDEATH_ACTION = "DealDeath";
00232     const std::string TAKEDEATH_ACTION = "TakeDeath";
00233     const std::string DEALDAMAGE_ACTION = "DealDamage";
00234     const std::string TAKEDAMAGE_ACTION = "TakeDamage";
00235
00236     /* Prefabs */
00237     const std::string GAME_ZONE_PREFAB = "gamezone";
00238     const std::string SMALL_EXPLOSION = "small_explosion";
00239     const std::string BIG_EXPLOSION = "big_explosion";
00240
00241
00242     constexpr float DEFAULT_TIMER = 0.0f;
00243
00244     /* Packet constants */
00245     constexpr std::uint8_t PACKET_NO_OP = 0x00;
00246     constexpr std::uint8_t PACKET_CONNECTION = 0x01;
00247     constexpr std::uint8_t PACKET_ACCEPT = 0x02;
00248     constexpr std::uint8_t PACKET_DISC = 0x03;
00249     constexpr std::uint8_t PACKET_EVENT = 0x04;
00250     constexpr std::uint8_t PACKET_GAME_STATE = 0x05;
00251     constexpr std::uint8_t PACKET_END_GAME = 0x06;
00252     constexpr std::uint8_t PACKET_CAN_START = 0x07;
00253     constexpr std::uint8_t PACKET_CLIENT_READY = 0x08;
00254     constexpr std::uint8_t PACKET_SPAWN = 0x09;
00255     constexpr std::uint8_t PACKET_DEATH = 0x0A;
00256     constexpr std::uint8_t PACKET_WHOAMI = 0x0B;
00257     constexpr std::uint8_t PACKET_SERVER_STATUS = 0x0C;
00258     constexpr std::uint8_t PACKET_REQUEST_LOBBY = 0x0D;
00259     constexpr std::uint8_t PACKET_SEND_LOBBY_CODE = 0x0E;
00260     constexpr std::uint8_t PACKET_CONNECT_TO_LOBBY = 0x0F;
00261     constexpr std::uint8_t PACKET_LOBBY_MASTER_REQUEST_START = 0x10;
00262     constexpr std::uint8_t PACKET_LOBBY_CONNECT_VALUE = 0x11;
00263
00264     const int MAX_INDEX_PACKET_TYPE = 18;
00265     const int MAX_CLIENT_PER_LOBBY = 4;
00266
00267     /* Scripting constant */
00268     const std::string INIT_FUNCTION = "init";
00269     const std::string UPDATE_FUNCTION = "update";
00270     const std::string DEATH_FUNCTION = "death";
00271
00272     /* Constants for Scripting API */
00273     const std::string PRINT_FUNCTION = "print";
00274     const std::string CREATE_MOVE_INTENT_FUNCTION = "createMoveIntent";
00275     const std::string GET_ENTITY_POSITION_FUNCTION = "getEntityPosition";
00276     const std::string GET_NEAREST_PLAYER_POSITION_FUNCTION = "getNearestPlayerPosition";
00277     const std::string GET_ENTITY_SPEED_FUNCTION = "getEntitySpeed";
00278     const std::string CREATE_SHOOT_INTENT_FUNCTION = "createShootIntent";
00279     const std::string SPAWN_ENTITY_FUNCTION = "spawnEntity";
00280     const std::string GET_ENTITY_ID_FUNCTION = "getEntityId";
00281     const std::string ADD_PART_ID_FUNCTION = "addPartId";
00282     const std::string SET_PARENT_ID_FUNCTION = "setParentId";
00283     const std::string SET_ENTITY_ROTATION_FUNCTION = "setEntityRotation";
00284     const std::string GET_ENTITY_PARTS_FUNCTION = "getEntityParts";
00285     const std::string CREATE_DEATH_INTENT_FUNCTION = "createDeathIntent";
00286     const std::string IS_ENTITY_ALIVE_FUNCTION = "isEntityAlive";
00287     const std::string GET_PARENT_ID_FUNCTION = "getParentId";
00288     const std::string REMOVE_PART_ID_FUNCTION = "removePartId";
00289 }
00290
00291 #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 "../common/interfaces/IWindow.hpp"
00016 #include "../common/interfaces/IEvent.hpp"
00017 #include "../common/interfaces/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 DeathAnimationSpawner.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** DeathAnimationSpawner
00006 */
00007
00008 #ifndef DEATHANIMATIONSRAWNER_HPP_
00009 #define DEATHANIMATIONSRAWNER_HPP_
00010
00011 #include <memory>
00012 #include <string>
00013 #include "../common/resourceManager/ResourceManager.hpp"
00014 #include "../common/ECS/entity/registry/Registry.hpp"
00015 #include "../common/ECS/entity/Entity.hpp"
00016 #include "../common/types/Vector2f.hpp"
00017
00018 class DeathAnimationSpawner {
00019     public:
00020         static void spawnDeathAnimation(
00021             std::shared_ptr<ResourceManager> resourceManager,
00022             std::shared_ptr<ecs::Registry> registry,
00023             ecs::Entity entity
00024         );
00025
00026     private:
00027         static math::Vector2f getFirstHitboxCenter(
00028             std::shared_ptr<ecs::Registry> registry,
00029             ecs::Entity entity
00030         );
00031 };
00032
00033 #endif /* !DEATHANIMATIONSRAWNER_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     void requestStateChange(std::shared_ptr<IGameState> newState) override;
00020
00021 protected:
00022     std::stack<std::shared_ptr<IGameState>> _states;
00023 };
00024
00025 } // 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 "../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 };
00023
00024 } // namespace gsm
00025
00026 #endif // SERVER_BOOTSTATE_HPP_

```

5.26 InfiniteState.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** InfiniteState

```

```

00006 */
00007
00008 #ifndef INFINITESTATE_HPP_
00009 #define INFINITESTATE_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 "../common/interfaces/IWindow.hpp"
00020
00021 namespace gsm {
00022
00023 class InfiniteState : public AGameState {
00024     public:
00025         InfiniteState(std::shared_ptr<IGameStateMachine> gsm, std::shared_ptr<ResourceManager>
resourceManager);
00026         ~InfiniteState() override = default;
00027
00028         void enter() override;
00029         void update(float deltaTime) override;
00030         void exit() override;
00031
00032     private:
00033         std::shared_ptr<ecs::Registry> _registry;
00034         std::shared_ptr<EntityPrefabManager> _prefabManager;
00035         std::shared_ptr<Parser> _parser;
00036 };
00037
00038 } // namespace gsm
00039
00040 #endif // INFINITESTATE_HPP_

```

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 "../common/interfaces/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
00023     void enter() override;
00024
00025     void update(float deltaTime) override;
00026 private:
00027 };
00028
00029 } // namespace gsm
00030
00031 #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/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>
00019 resourceManager);
00020     ~LoadingState() override = default;
00021
00022     void enter() override;
00023     void update(float deltaTime) override;
00024     void exit() override;
00025 };
00026
00027 } // namespace gsm
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>
00019 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>
00019 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>
00019         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);
00019     ~LobbyBrowserState() 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 // LOBBYBROWSERSTATE_HPP_

```

5.34 LobbyWaitingState.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** LobbyWaitingState
00006 */
00007
00008 #ifndef LOBBYWAITINGSTATE_HPP_
00009 #define LOBBYWAITINGSTATE_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/Text.hpp"
00018
00019 namespace gsm {
00020
00021 class LobbyWaitingState : public AGameState {
00022 public:
00023     LobbyWaitingState(std::shared_ptr<IGameStateMachine> gsm, std::shared_ptr<ResourceManager>
00024         resourceManager, bool isLobbyMaster);
00024     ~LobbyWaitingState() override = default;
00025
00026     void enter() override;
00027     void update(float deltaTime) override;

```



```

00028     void exit() override;
00029
00030 private:
00031     void renderUI();
00032     void updateUIStatus();
00033     void setupLobbyMasterUI();
00034     void setupPlayerUI();
00035
00036 private:
00037     std::unique_ptr<MouseInputHandler> _mouseHandler;
00038     std::unique_ptr<ui::UIManager> _uiManager;
00039     std::shared_ptr<ui::UILayout> _centerLayout;
00040     std::shared_ptr<ui::Text> _lobbyCodeText;
00041     std::shared_ptr<ui::Text> _statusText;
00042     std::shared_ptr<ui::Button> _startGameButton;
00043
00044     bool _isLobbyMaster;
00045 };
00046
00047 } // namespace gsm
00048
00049 #endif // LOBBYWAITINGSTATE_HPP_

```

5.35 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 #include "../ui/elements/Text.hpp"
00020
00021
00022 namespace gsm {
00023
00024 class MainMenuState : public AGameState {
00025 public:
00026     MainMenuState(std::shared_ptr<IGameStateMachine> gsm, std::shared_ptr<ResourceManager>
resourceManager);
00027     ~MainMenuState() override = default;
00028
00029     void enter() override;
00030     void update(float deltaTime) override;
00031     void exit() override;
00032
00033 private:
00034     void renderUI();
00035     void updateUIStatus();
00036     void checkLobbyConnectionTransition();
00037
00038 private:
00039     std::unique_ptr<MouseInputHandler> _mouseHandler;
00040     std::shared_ptr<ui::Button> _playButton;
00041     std::shared_ptr<ui::Button> _settingsButton;
00042     std::shared_ptr<ui::Button> _quitButton;
00043     std::shared_ptr<ui::Button> _connectButton;
00044     std::shared_ptr<ui::Button> _requestCodeButton;
00045     std::shared_ptr<ui::Button> _lobbyConnectButton;
00046     std::unique_ptr<ui::UIManager> _uiManager;
00047     std::shared_ptr<ui::UILayout> _leftLayout;
00048     std::shared_ptr<ui::UILayout> _mainMenuLayout;
00049
00050     std::shared_ptr<ui::UILayout> _rightLayout;
00051     std::shared_ptr<ui::Button> _devButton;
00052     std::shared_ptr<ui::Button> _infiniteButton;
00053
00054     std::shared_ptr<ui::TextInput> _ipInput;
00055     std::shared_ptr<ui::TextInput> _portInput;
00056     std::shared_ptr<ui::TextInput> _lobbyCodeInput;
00057

```

```

00058     std::shared_ptr<ui::Text> _connectionStatusText;
00059     std::shared_ptr<ui::Text> _serverStatusText;
00060
00061     std::shared_ptr<ui::Background> _background;
00062
00063     bool _previousLobbyConnectedState;
00064     bool _previousLobbyMasterState;
00065 };
00066
00067 } // namespace gsm
00068
00069 #endif // MAINMENUSTATE_HPP_

```

5.36 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>
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_

```

5.37 ResultsState.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ResultsState
00006 */
00007
00008 #ifndef RESULTSSTATE_HPP_
00009 #define RESULTSSTATE_HPP_
00010
00011 #include "../base/AGameState.hpp"
00012 #include "resourceManager/ResourceManager.hpp"
00013 #include "../common/interfaces/IWindow.hpp"
00014 #include "ui/manager/UIManager.hpp"
00015 #include "ui/elements/Text.hpp"
00016 #include "../common/colors.hpp"
00017
00018 namespace gsm {
00019
00020     class ResultsState : public AGameState {
00021     public:
00022         ResultsState(std::shared_ptr<IGameStateMachine> gsm, std::shared_ptr<ResourceManager>
resourceManager, bool isWin);
00023         ~ResultsState() override = default;
00024
00025         void enter() override;
00026         void update(float deltaTime) override;
00027         void exit() override;
00028
00029     private:
00030         bool _isWin;

```

```

00031         std::unique_ptr<ui::UIManager> _uiManager;
00032         std::shared_ptr<ui::Text> _resultText;
00033     };
00034
00035 } // namespace gsm
00036
00037 #endif // RESULTSSTATE_HPP_

```

5.38 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/UIManager.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_

```

5.39 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 "../common/interfaces/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

```

5.40 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 <memory>
00013 #include "../common/Parser/Parser.hpp"
00014 #include "../common/DLLoader/DLLoader.hpp"
00015 #include "../common/interfaces/IWindow.hpp"
00016 #include "../common/interfaces/IEvent.hpp"
00017 #include "../common/interfaces/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_ */

```

5.41 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 "../Lobby.hpp"
00014 #include "../common/ECS/entity/registry/Registry.hpp"
00015 #include "../common/Parser/Parser.hpp"
00016 #include "../common/systems/systemManager/ISystemManager.hpp"
00017 #include "../common/InputMapping/IInputProvider.hpp"
00018 #include "../gsm/machine/GameStateMachine.hpp"
00019 #include <memory>
00020
00021 std::shared_ptr<ResourceManager> initResourcesManager(
00022     std::shared_ptr<rserve::Server> server,
00023     std::shared_ptr<rserve::Lobby> lobby
00024 );
00025
00026 #endif /* !initResourcesManager_HPP_ */

```

5.42 MouseButtonHandler.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** MouseButtonHandler
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 MouseButtonInfo {
00018     math::Vector2f position;
00019     constants::MouseButton button;
00020 };
00021
00022 class MouseButtonHandler {
00023 public:
00024     MouseButtonHandler(std::shared_ptr<ResourceManager> resourceManager);
00025     ~MouseButtonHandler() = default;
00026
00027     std::optional<MouseButtonInfo> pollMouseButton();
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_ */

```

5.43 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_ */

```

5.44 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 "../common/interfaces/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_

```

5.45 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 "../interfaces/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_

```

5.46 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 "../../common/interfaces/IPacketManager.hpp"
00013
00014 namespace rserv::packet {
00015     bool registerDefaultPacketHandlers(std::shared_ptr<pm::IPacketManager> packet);
00016 }
00017
00018 #endif // DEFAULT_PACKET_HANDLERS_HPP_

```

5.47 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_

```

5.48 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 "../common/interfaces/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_ */

```

5.49 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_ */

```

5.50 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_ */

```

5.51 ClientEffectCleanupSystem.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ClientEffectCleanupSystem
00006 */
00007
00008 #ifndef CLIENTEFFECTCLEANUPSYSTEM_HPP_
00009 #define CLIENTEFFECTCLEANUPSYSTEM_HPP_
00010
00011 #include "../common/systems/base/ASystem.hpp"
00012
00013 namespace ecs {
00014
00015 class ClientEffectCleanupSystem : public ASystem {
00016     public:
00017         ClientEffectCleanupSystem();
00018         ~ClientEffectCleanupSystem() = 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 /* !CLIENTEFFECTCLEANUPSYSTEM_HPP_ */

```

5.52 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>
00029 registry, float deltaTime) override;
00030
00031     private:
00032         math::Vector2f getMovementDirection(std::shared_ptr<ResourceManager> resourceManager) const;
00033         void updateInputIntent(std::shared_ptr<Registry> registry, Entity entityId, const
00034 math::Vector2f &direction);
00035         math::Vector2f getAnalogStickInput(std::shared_ptr<IInputProvider> inputProvider) const;
00036         void sendAxisEvents(std::shared_ptr<ResourceManager> resourceManager, const math::Vector2f
00037 &direction);
00038         bool isPlayerAlive(std::shared_ptr<Registry> registry, Entity entityId) const;
00039         bool _wasMovingLastFrame = false;
00040 };
00041
00042 } // namespace ecs
00043
00044 #endif /* !MOVEMENTINPUTSYSTEM_HPP_ */

```

5.53 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         bool isPlayerAlive(std::shared_ptr<Registry> registry, Entity entityId) const;
00029 };
00030
00031 }
00032
00033 #endif /* !SHOOTINPUTSYSTEM_HPP_ */

```

5.54 MapGeneratorSystem.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** MapGeneratorSystem
00006 */
00007
00008 #ifndef MAPGENERATORSYSTEM_HPP_
00009 #define MAPGENERATORSYSTEM_HPP_
00010
00011 #include "../common/systems/base/ASystem.hpp"
00012 #include <random>
00013
00014 namespace ecs {
00015
00016 class MapGeneratorSystem : public ASystem {
00017     public:
00018         MapGeneratorSystem(unsigned int seed = 42);
00019         ~MapGeneratorSystem() = 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 generateObstaclesAt(
00029             float x,
00030             std::shared_ptr<ResourceManager> resourceManager,
00031             std::shared_ptr<Registry> registry
00032         );
00033         float noise(float x);
00034         unsigned int _seed;
00035         std::mt19937 _rng;
00036         float _lastGeneratedX;
00037         const float _generationStep;
00038         const float _startGenerationX;
00039 };
00040
00041 }
00042
00043 #endif /* !MAPGENERATORSYSTEM_HPP_ */

```

5.55 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_ */

```

5.56 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_ */

```

5.57 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_ */

```

5.58 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_ */

```

5.59 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_ */

```

5.60 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_ */

```

5.61 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_ */

```

5.62 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_ */

```

5.63 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_ */

```

5.64 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_ */

```

5.65 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_ */

```

5.66 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_ */

```

5.67 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_ */

```

5.68 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));
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_

```

5.69 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_ */

```

5.70 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 "../common/interfaces/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/abduction2002.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_ */

```

5.71 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 "../common/interfaces/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/abduction2002.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_ */

```

5.72 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 "../common/interfaces/IWindow.hpp"
00016 #include "../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/abduction2002.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_ */

```

5.73 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 "../common/interfaces/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/abduction2002.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_

```

5.74 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 "../common/interfaces/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_

```

5.75 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_ */

```

5.76 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_ */

```

5.77 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_ */

```

5.78 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<rserve::ServerConfig> config);
00023         static std::string createAlphaNumericCode();
00024     protected:
00025     private:
00026 };
00027
00028 #endif /* !UTILS_HPP_ */

```

5.79 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_

```

5.80 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_

```

5.81 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_ */

```

5.82 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_ */

```

5.83 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  #include "../types/OrientedRect.hpp"
00015  #include <cmath>
00016  #include <algorithm>
00017  #include <limits>
00018  #include <vector>
00019
00020  namespace ecs {
00021
00022  enum class CollisionType {
00023      None = 0,
00024      Solid = 1,
00025      Trigger = 2,
00026      Push = 3
00027  };
00028
00029  class ColliderComponent : public AComponent {
00030      public:
00031          ColliderComponent(math::Vector2f offset = math::Vector2f(0.0f, 0.0f), math::Vector2f size =
math::Vector2f(0.0f, 0.0f), CollisionType type = CollisionType::Solid)
: _offset(offset), _size(size), _type(type) {};
00032
00033          ~ColliderComponent() = default;
00034
00035          math::Vector2f getOffset() const;
00036          void setOffset(math::Vector2f offset);
00037
00038          math::Vector2f getSize() const;
00039          void setSize(math::Vector2f size);
00040
00041          CollisionType getType() const;
00042          void setType(CollisionType type);
00043
00044          math::FRect getHitbox(math::Vector2f entityPosition, math::Vector2f scale =
math::Vector2f(1.0f, 1.0f)) const;
00045
00046          math::FRect getScaledHitbox(math::Vector2f entityPosition, math::Vector2f scale) const;
00047
00048          math::OrientedRect getOrientedHitbox(math::Vector2f entityPosition, math::Vector2f scale,
float rotation) const;
00049
00050          math::FRect getHitbox(math::Vector2f entityPosition, math::Vector2f scale, float rotation)
const;
00051
00052      private:
00053          math::Vector2f _offset;
00054          math::Vector2f _size;

```

```

00055         CollisionType _type;
00056     };
00057
00058 } // namespace ecs
00059
00060 #endif /* !COLLIDERCOMPONENT_HPP_ */

```

5.84 CompositeEntityComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** CompositeEntityComponent
00006 */
00007
00008 #ifndef COMPOSITEENTITYCOMPONENT_HPP_
00009 #define COMPOSITEENTITYCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include <cstdint>
00013
00014 namespace ecs {
00015
00016 class CompositeEntityComponent : public AComponent {
00017     public:
00018         CompositeEntityComponent(size_t parent_id) : parentId(parent_id) {};
00019         ~CompositeEntityComponent() override = default;
00020
00021         size_t getParentId() const { return parentId; }
00022         void setParentId(size_t id) { parentId = id; }
00023     private:
00024         size_t parentId;
00025 };
00026
00027 } // namespace ecs
00028
00029 #endif /* !COMPOSITEENTITYCOMPONENT_HPP_ */

```

5.85 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     private:
00023         float _damage;
00024 };
00025
00026
00027 } // namespace ecs
00028
00029 #endif /* !DAMAGECOMPONENT_HPP_ */

```

5.86 EntityPartsComponent.hpp

```

00001 /*

```

```

00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** EntityPartsComponent
00006 */
00007
00008 #ifndef ENTITYPARTSCOMPONENT_HPP_
00009 #define ENTITYPARTSCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include <vector>
00013
00014 namespace ecs {
00015
00016 class EntityPartsComponent : public AComponent {
00017     public:
00018         EntityPartsComponent() = default;
00019         ~EntityPartsComponent() override = default;
00020
00021         std::vector<size_t> partIds;
00022 };
00023
00024 } // namespace ecs
00025
00026 #endif /* !ENTITYPARTSCOMPONENT_HPP_ */

```

5.87 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_ */

```

5.88 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) {};
00019         ~HealthComponent() override = default;
00020
00021         float getHealth() const { return _health; }
00022         void setHealth(float health) { _health = health; };
00023
00024         void decreaseHealth(float quantity) { _health -= quantity; };
00025
00026         float getBaseHealth() const { return _baseHealth; };
00027         void setBaseHealth(float health) { _baseHealth = health; };
00028
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_ */

```

5.89 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_ */

```

5.90 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_ */

```

5.91 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_ */

```

5.92 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_ */

```

5.93 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_ */

```

5.94 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_ */

```

5.95 ScriptingComponent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2026
00003 ** ryanR-type
00004 ** File description:
00005 ** ScriptingComponent
00006 */
00007
00008 #ifndef SCRIPTINGCOMPONENT_HPP_
00009 #define SCRIPTINGCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include <string>
00013 #include <map>
00014 #include "../Error/ScriptingError.hpp"
00015 #include "../constants.hpp"
00016
00017 // To suppress warnings from sol2 includes
00018 #ifdef __GNUC__
00019 #pragma GCC diagnostic push
00020 #pragma GCC diagnostic ignored "-Wsign-conversion"
00021 #endif
00022 #ifdef _MSC_VER
00023 #pragma warning(push)
00024 #pragma warning(disable: 5321)
00025 #endif
00026 #include <sol/sol.hpp>
00027 #ifdef _MSC_VER
00028 #pragma warning(pop)
00029 #endif
00030 #ifdef __GNUC__
00031 #pragma GCC diagnostic pop
00032 #endif
00033
00034 namespace ecs {
00035
00036 class ScriptingComponent : public AComponent {
00037 public:
00038     ScriptingComponent(std::string script_name = "", std::vector<std::string> additionalFunctions
= std::vector<std::string>(), std::shared_ptr<sol::state> lua = nullptr, size_t entityId = 0)
00039         : _scriptName(script_name), _additionalFunctions(additionalFunctions), _initialized(false)
00040     {
00041         if (lua != nullptr) {
00042             init(*lua, entityId);
00043         }
00044         ~ScriptingComponent() = default;
00045
00046         void init(sol::state& lua, size_t entityId);
00047
00048         const std::string& getScriptName() const;
00049
00050         void setEnvironment(const sol::table& table) { _env = table; };
00051         sol::table getEnvironment() const { return _env; };
00052
00053         bool hasFunction(const std::string& name) const { return _functions.find(name) !=
_functions.end(); };
00054         void addFunction(const std::string& name, const sol::function& function) { _functions[name] =
function; };
00055         sol::function getFunction(const std::string& name) const { return _functions.at(name); };
00056         std::vector<std::string> getFunctionNames() const {
00057             std::vector<std::string> names;
00058             for (const auto& pair : _functions) {
00059                 names.push_back(pair.first);
00060             }
00061             return names;
00062         }
00063         void removeFunction(const std::string& name) { _functions.erase(name); };
00064         bool isInitialized() const { return _initialized; };
00065         void setInitialized(bool value) { _initialized = value; };
00066     protected:
00067     private:
00068         std::string _scriptName;
00069         std::vector<std::string> _additionalFunctions;
00070         sol::table _env;
00071         std::map<std::string, sol::function> _functions;
00072         bool _initialized = false;
00073     };
00074
00075 } // namespace ecs
00076
00077 #endif /* !SCRIPTINGCOMPONENT_HPP_ */

```

5.96 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_ */

```

5.97 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_ */

```

5.98 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_ */

```

5.99 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         {};
00019         ~VelocityComponent() = default;
00020
00021         math::Vector2f getVelocity() const { return _velocity; };
00022         void setVelocity(math::Vector2f velocity) { _velocity = velocity; };
00023     private:
00024         math::Vector2f _velocity;
00025 };
00026
00027 } // namespace ecs
00028
00029 #endif /* !VELOCITYCOMPONENT_HPP_ */

```

5.100 ClientEffectTag.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ClientEffectTag
00006 */
00007
00008 #ifndef CLIENTEFFECTTAG_HPP_
00009 #define CLIENTEFFECTTAG_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012
00013 namespace ecs {
00014
00015 class ClientEffectTag : public AComponent {
00016     public:
00017         ClientEffectTag() = default;
00018         ~ClientEffectTag() = default;
00019 };
00020
00021 }
00022
00023 #endif /* !CLIENTEFFECTTAG_HPP_ */

```

5.101 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_ */

```

5.102 EnemyProjectileTag.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_ */

```

5.103 GameEndTag.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** GameEndTag
00006  */
00007
00008  #ifndef GAMEENDTAG_HPP_
00009  #define GAMEENDTAG_HPP_
00010
00011  #include "../base/AComponent.hpp"
00012
00013  namespace ecs {
00014
00015  class GameEndTag : public AComponent {
00016      public:
00017          GameEndTag() = default;
00018          ~GameEndTag() = default;
00019
00020      protected:
00021      private:
00022  };
00023
00024  } // namespace ecs
00025
00026  #endif /* !GAMEENDTAG_HPP_ */

```

5.104 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_ */

```


5.105 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_ */

```

5.106 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_ */

```

5.107 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_ */

```

5.108 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_ */

```

5.109 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_ */

```

5.110 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_ */

```

5.111 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_ */

```

5.112 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_ */

```

5.113 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_ */

```

5.114 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_ */

```

5.115 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
00019             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_ */

```

5.116 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_ */

```

5.117 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_ */

```

5.118 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         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_ */

```

5.119 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_ */

```

5.120 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     #ifdef ERROR
00017         #undef ERROR
00018     #endif
00019     #ifdef INFO
00020         #undef INFO
00021     #endif
00022     #ifdef WARNING
00023         #undef WARNING
00024     #endif
00025 #endif
00026
00027 #include <string>
00028
00029 namespace debug {
00030
00031     enum debugType {
00032         NETWORK = 0,
00033         ECS = 1,
00034         CORE = 2
00035     };
00036
00037     enum debugLevel {
00038         INFO = 0,
00039         WARNING = 1,
00040         ERROR = 2
00041     };
00042
00043     class Debug {
00044     public:
00045         ~Debug() = default;
00046         static void printDebug(const bool isDebug, const std::string &message, debugType type,
00047                               debugLevel level);
00048     };
00049
00050 } // namespace debug
00051
00052 #endif /* !DEBUG_HPP_ */

```

5.121 DLoader.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2025
00003  ** r-type
00004  ** File description:
00005  ** DLoader
00006  */
00007
00008  #ifndef DLOADER_HPP_
00009  #define DLOADER_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 DLoader : 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      ~DLoader() 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_ */

```

5.122 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 /* !ILoader_HPP_ */

```

5.123 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_ */

```

5.124 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_ */

```

5.125 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_ */

```

5.126 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_ */

```

5.127 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 <cstdint>
00012  #include <optional>
00013
00014  namespace ecs {
00015
00016  enum class EntityCreationOrigin {
00017      SERVER,
00018      CLIENT_LOCAL
00019  };
00020
00021  struct EntityCreationContext {
00022      EntityCreationOrigin origin = EntityCreationOrigin::CLIENT_LOCAL;
00023
00024      static EntityCreationContext forServer() {
00025          return {EntityCreationOrigin::SERVER};
00026      }
00027
00028      static EntityCreationContext forLocalClient() {
00029          return {EntityCreationOrigin::CLIENT_LOCAL};
00030      }
00031  };

```

```

00031 };
00032
00033 } // namespace ecs
00034
00035 #endif /* !ENTITYCREATIONCONTEXT_HPP_ */

```

5.128 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();
00019         ~EntityFactory() override;
00020
00021         Entity createEntity(
00022             const std::shared_ptr<Registry>& registry,
00023             const EntityCreationContext& context = EntityCreationContext::forLocalClient()
00024         ) override;
00025
00026     private:
00027         std::atomic<size_t> _nextLocalId;
00028 };
00029
00030 } // namespace ecs
00031
00032 #endif /* !ENTITYFACTORY_HPP_ */

```

5.129 IEntityFactory.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** IEntityFactory
00006 */
00007
00008 #ifndef IENTITYFACTORY_HPP_
00009 #define IENTITYFACTORY_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
00029 } // namespace ecs
00030
00031 #endif /* !ENTITYFACTORY_HPP_ */

```

5.130 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 #include <mutex>
00019
00020 namespace ecs {
00021
00022 template <typename... Components> class View;
00023 template <typename... Components> class Group;
00024
00025 class Registry : public std::enable_shared_from_this<Registry> {
00026     public:
00027         Registry();
00028         explicit Registry(Entity nextEntityId);
00029         ~Registry();
00030
00031         template <typename T>
00032         void registerComponent();
00033
00034         template <typename T>
00035         void addComponent(Entity entityId, std::shared_ptr<T> component);
00036
00037         template <typename T>
00038         std::shared_ptr<T> getComponent(Entity entityId) const;
00039
00040         template <typename T>
00041         std::vector<std::shared_ptr<T>> getComponents(Entity entityId) const;
00042
00043         template <typename T>
00044         void removeAllComponents(Entity entityId);
00045
00046         template <typename T>
00047         void removeOneComponent(Entity entityId);
00048
00049         template <typename T>
00050         bool hasComponent(Entity entityId) const;
00051
00052         template <typename... Components>
00053         View<Components...> view();
00054
00055         Entity getMaxEntityId() const;
00056
00057         Entity createEntity();
00058         void destroyEntity(Entity entityId);
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         mutable std::recursive_mutex _mutex;
00067 };
00068
00069 } // namespace ecs
00070
00071 #include "Registry.hpp"
00072
00073 #endif /* !REGISTRY_HPP_ */

```

5.131 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  } // namespace ecs
00046
00047  #include "View.hpp"
00048
00049  #endif /* !VIEW_HPP_ */

```

5.132 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_ */

```

5.133 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_ */

```

5.134 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_ */

```

5.135 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 /* !IERROR_HPP_ */
00032

```

5.136 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_ */

```

5.137 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 */

```

5.138 ParserError.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** ParserError
00006  */
00007
00008  #ifndef PARSEERROR_HPP_
00009  #define PARSEERROR_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 /* !PARSEERROR_HPP_ */

```

5.139 ScriptingError.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2026
00003  ** ryanR-type
00004  ** File description:
00005  ** ScriptingError
00006  */
00007
00008  #ifndef SCRIPTINGERROR_HPP_
00009  #define SCRIPTINGERROR_HPP_
00010
00011
00012  #include "AError.hpp"

```

```

00013
00014 namespace err {
00015
00016 class ScriptingError : public AError {
00017     public:
00018         enum ErrorCode {
00019             UNKNOWN = 1000,
00020             LOAD_FAILED = 1001,
00021             RUN_FAILED = 1002
00022         };
00023
00024         ScriptingError(const std::string &message, ErrorCode code = UNKNOWN);
00025         ~ScriptingError() noexcept = default;
00026         std::string getType() const noexcept override;
00027
00028     protected:
00029     private:
00030 };
00031
00032 } // namespace err
00033
00034 #endif /* !SCRIPTINGERROR_HPP_ */

```

5.140 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 */

```

5.141 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

```

5.142 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

```

5.143 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_ */

```

5.144 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_ */

```

5.145 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_ */

```

5.146 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_ */

```

5.147 IAudio.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** IAudio
00006 */
00007
00008 #ifndef IAUDIO_HPP_
00009     #define IAUDIO_HPP_
00010
00011     #include <string>
00012
00013     namespace gfx
00014     {
00015
00016         class IAudio
00017         {
00018             public:
00019                 virtual ~IAudio() = default;
00020
00021                 virtual void playMusic(const std::string& musicPath, bool loop = true) = 0;
00022                 virtual void stopMusic() = 0;
00023                 virtual void pauseMusic() = 0;
00024                 virtual void resumeMusic() = 0;
00025                 virtual void setMusicVolume(float volume) = 0;
00026                 virtual float getMusicVolume() const = 0;
00027                 virtual bool isMusicPlaying() const = 0;
00028
00029                 virtual void playSound(const std::string& soundPath, float volume = 100.0f) = 0;
00030                 virtual void setSoundVolume(float volume) = 0;
00031                 virtual float getSoundVolume() const = 0;
00032                 virtual void stopAllSounds() = 0;
00033         };
00034
00035         typedef IAudio *(*createAudio_t)();
00036
00037     }
00038
00039 #endif /* !IAUDIO_HPP_ */

```

5.148 IBuffer.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** IBuffer
00006 */
00007
00008 #ifndef IBUFFER_HPP_
00009     #define IBUFFER_HPP_
00010
00011     #include <memory>
00012     #include <vector>
00013     #include <cstdint>
00014
00015
00016     class IBuffer {
00017     public:
00018         virtual ~IBuffer() = default;
00019
00020
00021         virtual void createBuffer(size_t size) = 0;
00022         virtual void deleteBuffer() = 0;
00023         virtual void clear() = 0;
00024
00025         virtual bool writeBuffer(const std::vector<uint64_t> &data, size_t size) = 0;
00026         virtual std::shared_ptr<std::vector<uint64_t> > readBuffer(size_t size) = 0;
00027
00028         virtual size_t getCapacity() const = 0;
00029         virtual size_t getUsedSize() const = 0;
00030         virtual size_t getAvailableSize() const = 0;
00031         virtual bool isEmpty() const = 0;
00032         virtual bool isFull() const = 0;
00033
00034         virtual std::vector<uint64_t> getBuffer() const = 0;
00035     protected:
00036     private:
00037     };
00038
00039 #endif /* !IBUFFER_HPP_ */

```

5.149 IEvent.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** r-type
00004 ** File description:
00005 ** IEvent
00006 */
00007
00008 #ifndef IEVENT_HPP_
00009 #define IEVENT_HPP_
00010
00011 #include <utility>
00012 #include <memory>
00013 #include <string>
00014 #include "EventTypes.hpp"
00015
00016 namespace gfx {
00017
00018     class IEvent {
00019     public:
00020         using event_t = EventType;
00021         virtual ~IEvent() = default;
00022         virtual void init() = 0;
00023         virtual event_t pollEvents() = 0;
00024         virtual std::string getLastTextInput() = 0;
00025         virtual void cleanup() = 0;
00026         virtual std::pair<int, int> getMousePos() = 0;
00027         virtual bool isKeyPressed(event_t key) = 0;
00028         virtual bool isMouseButtonPressed(int button) = 0;
00029         virtual float getAxisValue(event_t axis) = 0;
00030
00031     };
00032
00033     typedef IEvent *(*createEvent_t)(void*, void*);
00034
00035 } // namespace gfx
00036
00037 #endif /* !IEVENT_HPP_ */

```

5.150 IEventLoop.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** IEventLoop
00006 */
00007
00008 #ifndef IEVENTLOOP_HPP_
00009 #define IEVENTLOOP_HPP_
00010
00011 #include <functional>
00012 #include <memory>
00013
00014 namespace net {
00015
00016     class IEventLoop {
00017     public:
00018         virtual ~IEventLoop() = default;
00019         virtual void run() = 0;
00020         virtual void runOne() = 0;
00021         virtual void stop() = 0;
00022         virtual bool stopped() const = 0;
00023         virtual void post(std::function<void()> task) = 0;
00024         virtual void restart() = 0;
00025
00026     };
00027
00028 } // namespace net
00029
00029 #endif /* !IEVENTLOOP_HPP_ */

```

5.151 INetwork.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type

```

```

00004  ** File description:
00005  ** INetwork
00006  */
00007
00008  #ifndef INetwork_HPP_
00009  #define INetwork_HPP_
00010
00011  #include <vector>
00012  #include <functional>
00013  #include <memory>
00014  #include "INetworkEndpoint.hpp"
00015  #include "IPacketManager.hpp"
00016  #include "IBuffer.hpp"
00017
00018  namespace net {
00019
00020  enum class ConnectionState {
00021      DISCONNECTED,
00022      CONNECTING,
00023      CONNECTED,
00024      RECONNECTING,
00025      ERROR_STATE
00026  };
00027
00028
00029  class INetwork {
00030  public:
00031
00032      virtual ~INetwork() = default;
00033
00034      virtual void init(uint16_t port, const std::string host) = 0;
00035      virtual void stop() = 0;
00036
00037      virtual bool sendTo(const INetworkEndpoint& endpoint, std::vector<uint8_t> packet) = 0;
00038      virtual bool broadcast(const std::vector<std::shared_ptr<INetworkEndpoint>>& endpoints, const
std::vector<uint8_t>& data) = 0;
00039      virtual bool hasIncomingData() const = 0;
00040      virtual std::vector<uint8_t> receiveFrom(const uint8_t &connectionId) = 0;
00041      virtual std::pair<std::shared_ptr<INetworkEndpoint>, std::vector<uint8_t>> receiveAny() = 0;
00042
00043      virtual void setConnectionCallback(std::function<void(int)> onConnect) = 0;
00044      virtual void setDisconnectionCallback(std::function<void(int)> onDisconnect) = 0;
00045      virtual ConnectionState getConnectionState() const = 0;
00046      virtual void setConnectionState(ConnectionState state) = 0;
00047  };
00048
00049  } // namespace net
00050
00051  #endif /* !INetwork_HPP_ */

```

5.152 INetworkAddress.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2025
00003  ** R-Type
00004  ** File description:
00005  ** INetworkAddress - Interface for IP address representation
00006  */
00007
00008  #ifndef INetworkAddress_HPP
00009  #define INetworkAddress_HPP
00010
00011  #include <string>
00012  #include <memory>
00013
00014  namespace net {
00015
00016  class INetworkAddress {
00017  public:
00018      virtual bool isV4() const = 0;
00019      virtual bool isV6() const = 0;
00020      virtual std::string toString() const = 0;
00021
00022      virtual ~INetworkAddress() = default;
00023      virtual std::shared_ptr<INetworkAddress> operator=(const INetworkAddress& other) = 0;
00024      virtual std::shared_ptr<void> getInternalAddress() = 0;
00025      virtual std::shared_ptr<const void> getInternalAddress() const = 0;
00026      virtual void setFromInternal(std::shared_ptr<void> internalAddr) = 0;
00027  };
00028
00029  } // namespace net
00030
00031  #endif // INetworkAddress_HPP

```


5.153 INetworkEndpoint.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** NetworkEndpoint
00006 */
00007
00008 #ifndef INetworkEndpoint_HPP_
00009 #define INetworkEndpoint_HPP_
00010
00011 #include <string>
00012 #include <cstdint>
00013
00014 namespace net {
00015
00016 class INetworkEndpoint {
00017     public:
00018         virtual ~INetworkEndpoint() noexcept = default;
00019
00020         virtual const std::string& getAddress() const = 0;
00021         virtual uint16_t getPort() const = 0;
00022
00023         virtual void setAddress(const std::string& address) = 0;
00024         virtual void setPort(uint16_t port) = 0;
00025         virtual bool operator==(const INetworkEndpoint& other) const = 0;
00026         virtual bool operator!=(const INetworkEndpoint& other) const = 0;
00027         virtual bool operator<(const INetworkEndpoint& other) const = 0;
00028 };
00029
00030 } // namespace net
00031
00032 #endif /* !INetworkEndpoint_HPP_ */

```

5.154 INetworkErrorCode.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** R-Type
00004 ** File description:
00005 ** NetworkErrorCode
00006 */
00007
00008 #ifndef INetworkErrorCode_HPP_
00009 #define INetworkErrorCode_HPP_
00010
00011 #include <string>
00012 #include <memory>
00013
00014 namespace net {
00015
00016 enum class NetworkError {
00017     SUCCESS = 0,
00018     WOULD_BLOCK = 1,
00019     AGAIN = 2,
00020     CONNECTION_REFUSED = 3,
00021     NETWORK_UNREACHABLE = 4,
00022     TIMED_OUT = 5,
00023     OTHER = 6
00024 };
00025
00026 class INetworkErrorCode {
00027     public:
00028         virtual ~INetworkErrorCode() noexcept = default;
00029
00030         virtual void clear() = 0;
00031         virtual bool hasError() const = 0;
00032         virtual explicit operator bool() const = 0;
00033         virtual std::string message() const = 0;
00034         virtual NetworkError getError() const = 0;
00035         virtual void setError(NetworkError error, const std::string& msg = "") = 0;
00036         virtual bool operator==(NetworkError error) const = 0;
00037         virtual bool operator!=(NetworkError error) const = 0;
00038
00039         virtual std::shared_ptr<void> getInternalErrorCode() = 0;
00040         virtual std::shared_ptr<const void> getInternalErrorCode() const = 0;
00041         virtual void setFromInternal(std::shared_ptr<void> internalEc) = 0;
00042 };
00043
00044 } // namespace net
00045
00046 #endif // INetworkErrorCode_HPP_

```

5.155 INetworkFactory.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2026
00003 ** R-Type
00004 ** File description:
00005 ** INetworkFactory - Factory interface for creating network objects
00006 */
00007
00008 #ifndef INETWORK_FACTORY_HPP_
00009 #define INETWORK_FACTORY_HPP_
00010
00011 #include <memory>
00012 #include "INetworkSocket.hpp"
00013 #include "INetworkResolver.hpp"
00014 #include "IEventLoop.hpp"
00015
00016 namespace net {
00017
00018 class INetworkFactory {
00019     public:
00020         virtual ~INetworkFactory() = default;
00021         virtual std::shared_ptr<IEventLoop> createEventLoop() = 0;
00022         virtual std::shared_ptr<INetworkSocket> createSocket(std::shared_ptr<IEventLoop> eventLoop) =
00023             0;
00024         virtual std::shared_ptr<INetworkResolver> createResolver(std::shared_ptr<IEventLoop>
00025             eventLoop) = 0;
00026 };
00027
00028 } // namespace net
00029
00030 #endif /* !INETWORK_FACTORY_HPP_ */

```

5.156 INetworkResolver.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** R-Type
00004 ** File description:
00005 ** NetworkResolver
00006 */
00007
00008 #ifndef INETWORK_RESOLVER_HPP
00009 #define INETWORK_RESOLVER_HPP
00010
00011 #include <memory>
00012 #include <string>
00013 #include <vector>
00014
00015 namespace net {
00016
00017 class INetworkEndpoint;
00018 class INetworkErrorCode;
00019
00020 class INetworkResolver {
00021     public:
00022         virtual ~INetworkResolver() = default;
00023         virtual std::vector<std::shared_ptr<INetworkEndpoint>> resolve(const std::string& host,
00024             const std::string& port, std::shared_ptr<INetworkErrorCode> ec) = 0;
00025 };
00026
00027 } // namespace net
00028
00029 #endif // INETWORK_RESOLVER_HPP

```

5.157 INetworkSocket.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** R-Type
00004 ** File description:
00005 ** Header
00006 */
00007
00008 #ifndef INETWORK_SOCKET_HPP
00009 #define INETWORK_SOCKET_HPP
00010

```

```

00011 #include <memory>
00012 #include <string>
00013 #include <vector>
00014 #include <cstdint>
00015
00016 namespace net {
00017
00018 class INetworkEndpoint;
00019 class INetworkErrorCode;
00020
00021 class INetworkSocket {
00022     public:
00023         virtual ~INetworkSocket() = default;
00024         virtual bool open(std::shared_ptr<INetworkErrorCode> ec) = 0;
00025         virtual bool bind(const INetworkEndpoint& endpoint, std::shared_ptr<INetworkErrorCode> ec) =
0;
00026         virtual std::size_t sendTo(const std::vector<uint8_t>& data, const INetworkEndpoint& endpoint,
int flags, std::shared_ptr<INetworkErrorCode> ec) = 0;
00027         virtual std::size_t receiveFrom(std::shared_ptr<std::vector<uint8_t>> buffer,
std::shared_ptr<INetworkEndpoint> sender, int flags, std::shared_ptr<INetworkErrorCode> ec) = 0;
00028         virtual bool setNonBlocking(bool nonBlocking, std::shared_ptr<INetworkErrorCode> ec) = 0;
00029         virtual bool close(std::shared_ptr<INetworkErrorCode> ec) = 0;
00030         virtual bool isOpen() const = 0;
00031 };
00032
00033 } // namespace net
00034
00035 #endif // INETWORK_SOCKET_HPP

```

5.158 IPacketManager.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** IPacketManager
00006 */
00007
00008 #ifndef IPacketManager_HPP_
00009 #define IPacketManager_HPP_
00010
00011 #include <vector>
00012 #include <cstdint>
00013 #include <functional>
00014 #include <string>
00015 #include "IBuffer.hpp"
00016
00017 #define MAGIC_NUMBER 0x93
00018 #define HEADER_SIZE 11
00019
00020 #define LENGTH_CONNECTION_PACKET 8
00021 #define LENGTH_ACCEPATION_PACKET 1
00022 #define LENGTH_DISCONNECTION_PACKET 1
00023 #define LENGTH_EVENT_PACKET 9
00024 #define LENGTH_END_GAME_PACKET 0
00025 #define LENGTH_DEATH_PACKET 8
00026 #define LENGTH_WHOAMI_PACKET 0
00027 #define LENGTH_SERVER_STATUS_PACKET 32
00028 #define LENGTH_LOBBY_CODE_PACKET 8
00029 #define LENGTH_REQUEST_LOBBY_PACKET 0
00030 #define LENGTH_CONNECT_TO_LOBBY_PACKET 1
00031
00032 #define NO_OP_PACKET 0x00
00033 #define CONNECTION_CLIENT_PACKET 0x01
00034 #define ACCEPATION_PACKET 0x02
00035 #define DISCONNECTION_PACKET 0x03
00036 #define EVENT_PACKET 0x04
00037 #define GAME_STATE_PACKET 0x05
00038 #define END_GAME_PACKET 0x06
00039 #define CAN_START_PACKET 0x07
00040 #define CLIENT_READY_PACKET 0x08
00041 #define SPAWN_PLAYER_PACKET 0x09
00042 #define DEATH_PLAYER_PACKET 0x0A
00043 #define WHOAMI_PACKET 0x0B
00044 #define SERVER_STATUS_PACKET 0x0C
00045 #define REQUEST_LOBBY_PACKET 0x0D
00046 #define SEND_LOBBY_CODE_PACKET 0x0E
00047 #define CONNECT_TO_LOBBY 0x0F
00048 #define LOBBY_MASTER_REQUEST_START 0x10
00049 #define LOBBY_CONNECT_VALUE 0x11
00050
00051 namespace pm {
00052

```

```

00053     class IPacketManager {
00054     public:
00055         virtual ~IPacketManager() = default;
00056
00057         virtual uint32_t getLength() const = 0;
00058         virtual uint32_t getSequenceNumber() const = 0;
00059         virtual uint8_t getType() const = 0;
00060         virtual std::vector<uint64_t> getPayload() const = 0;
00061         virtual uint8_t getIdClient() const = 0;
00062
00063         virtual void setType(uint8_t type) = 0;
00064         virtual void setLength(uint32_t length) = 0;
00065         virtual void setSequenceNumber(uint32_t sequenceNumber) = 0;
00066         virtual void setPayload(std::vector<uint64_t> payload) = 0;
00067         virtual void setIdClient(uint8_t idClient) = 0;
00068
00069         virtual std::vector<uint64_t> formatString(const std::string str) = 0;
00070         virtual std::vector<uint8_t> pack(uint8_t idClient, uint32_t sequenceNumber, uint8_t type,
std::vector<uint64_t> payload) = 0;
00071         virtual bool unpack(std::vector<uint8_t> data) = 0;
00072
00073         virtual void reset() = 0;
00074
00075         virtual void registerBuilder(uint8_t type,
std::function<std::vector<uint8_t>(std::vector<uint64_t>)> builder) = 0;
00076         virtual void registerParser(uint8_t type, std::function<bool(const std::vector<uint8_t>)>
parser) = 0;
00077         virtual void registerLength(uint8_t type, uint32_t length) = 0;
00078         virtual void
registerGameStatePackFunction(std::function<std::vector<uint8_t>(std::vector<uint64_t>,
std::shared_ptr<unsigned int>)> func) = 0;
00079         virtual void registerGameStateUnpackFunction(std::function<unsigned int(const
std::vector<uint8_t>, unsigned int)> func) = 0;
00080         virtual void registerLengthCombEntry(uint8_t compType, uint32_t compLength, uint64_t compSize)
= 0;
00081         virtual void clearAllHandlers() = 0;
00082     };
00083     } // namespace pm
00084
00085 #endif /* !IPacketManager_HPP_ */

```

5.159 IWindow.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** IWindow
00006 */
00007
00008 #ifndef IWINDOW_HPP_
00009 #define IWINDOW_HPP_
00010
00011 #include <string>
00012 #include <utility>
00013 #include <stdint>
00014 #include "../common/types/FRect.hpp"
00015 #include "../common/types/Vector2f.hpp"
00016
00017 namespace gfx {
00018
00019     struct color_t {
00020         uint8_t r;
00021         uint8_t g;
00022         uint8_t b;
00023         uint8_t a = 255;
00024     };
00025
00026     class IWindow {
00027     public:
00028         virtual ~IWindow() = default;
00029
00030         virtual void init() = 0;
00031         virtual void display() = 0;
00032         virtual void closeWindow() = 0;
00033         virtual bool isOpen() = 0;
00034         virtual void clear() = 0;
00035         virtual void resizeWindow(size_t x, size_t y) = 0;
00036
00037         virtual void drawSprite(std::string asset, color_t color, std::pair<size_t, size_t> position)
= 0;
00038         virtual void drawText(std::string text, color_t color, std::pair<size_t, size_t> position,
const std::string& fontPath, size_t fontSize = 24, color_t outlineColor = {0, 0, 0}, float
outlineThickness = 0.0f) = 0;

```

```

00039     virtual std::pair<size_t, size_t> getTextSize(const std::string& text, const std::string&
fontPath, size_t fontSize = 24) = 0;
00040     virtual void drawRectangleOutline(color_t color, std::pair<size_t, size_t> position,
std::pair<size_t, size_t> size) = 0;
00041     virtual void drawFilledRectangle(color_t color, std::pair<size_t, size_t> position,
std::pair<size_t, size_t> size) = 0;
00042     virtual void drawRoundedRectangleFilled(color_t color, std::pair<size_t, size_t> position,
std::pair<size_t, size_t> size, float radius) = 0;
00043     virtual void drawRoundedRectangleOutline(color_t color, std::pair<size_t, size_t> position,
std::pair<size_t, size_t> size, float radius) = 0;
00044
00045     virtual bool isMouseOver(std::pair<size_t, size_t> position, std::pair<size_t, size_t> size) =
0;
00046     virtual std::pair<int, int> getWindowSize() = 0;
00047
00048     virtual void drawSprite(const std::string& texturePath, float x, float y, float scaleX = 1.0f,
float scaleY = 1.0f, float rotation = 0.0f) = 0;
00049     virtual void drawSprite(const std::string& texturePath, float x, float y, const math::FRect
frameRect, float scaleX = 1.0f, float scaleY = 1.0f, float rotation = 0.0f) = 0;
00050
00051     virtual void updateView() = 0;
00052     virtual void setViewCenter(float x, float y) = 0;
00053     virtual math::Vector2f getViewCenter() = 0;
00054     virtual math::Vector2f mapPixelToCoords(int x, int y) = 0;
00055
00056     virtual std::pair<int, int> getLogicalSize() const = 0;
00057     virtual float getScaleFactor() const = 0;
00058
00059     virtual void addShaderFilter(const std::string& path) = 0;
00060     virtual void removeShaderFilter(const std::string& path) = 0;
00061     virtual void setShaderUniform(const std::string& filterPath, const std::string& name, float
value) = 0;
00062     virtual void setFrameRateLimit(unsigned int fps) = 0;
00063     virtual void setFullscreen(bool fullscreen) = 0;
00064     virtual void setRenderQuality(float quality) = 0;
00065     virtual void setCursor(bool isHand) = 0;
00066 };
00067
00068 typedef IWindow *(*createWindow_t)();
00069
00070 } // namespace gfx
00071
00072 #endif /* !IWINDOW_HPP_ */

```

5.160 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_

```

5.161 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_ */

```

5.162 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_

```

5.163 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,
std::shared_ptr<FieldValue>&)>>;
00023
00024         ComposantParser(std::shared_ptr<const std::map<std::string, std::pair<std::type_index,
std::vector<Field>>> componentDefinitions,
00025             const std::map<std::type_index, ComponentCreator> &componentCreators,
00026             const ShouldParseComponentCallback &shouldParseCallback = nullptr);
00027         ~ComposantParser();
00028
00029         std::pair<std::shared_ptr<ecs::IComponent>, std::type_index> parseComponent(const std::string
&componentName, const nlohmann::json &componentData);
00030
00031     protected:
00032     private:
00033         std::shared_ptr<FieldValue> parseFieldValue(const nlohmann::json &jsonValue, FieldType type);
00034         std::shared_ptr<const std::map<std::string, std::pair<std::type_index, std::vector<Field>>>
_componentDefinitions;
00035         const std::map<std::type_index, ComponentCreator> &_componentCreators;
00036         ShouldParseComponentCallback _shouldParseCallback;
00037 };
00038
00039 #endif /* !COMPOSANTPARSER_HPP_ */

```

5.164 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>>>
componentDefinitions,
00027             const std::map<std::type_index, ComponentCreator> &componentCreators,
00028             const std::map<std::type_index, ComponentAdder> &componentAdders,
00029             const ShouldParseComponentCallback &shouldParseCallback = nullptr
00030         );
00031         ~EntityParser();
00032
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_ */

```

5.165 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     private:
00040         std::shared_ptr<EntityPrefabManager> _prefabManager;
00041         std::shared_ptr<ecs::Registry> _registry;
00042         ecs::EntityCreationContext _creationContext;
00043         nlohmann::json _mapJson;
00044
00045         void createBackgroundEntity(const std::string &entityName);
00046         void createMusicEntity(const std::string &prefabName);
00047         void createGameZoneEntity(float scrollSpeed);
00048         void createGameEndEntity(float mapLength);
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
00060         ecs::Entity createEntityFromPrefab(
00061             const std::string &prefabName,
00062             float x, float y
00063         );
00064 };
00065
00066 #endif /* !MAPPARSER_HPP_ */

```

5.166 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>>>
00058     _componentDefinitions;
00059     std::map<std::type_index, ComponentCreator> _componentCreators;
00060     std::map<std::type_index, ComponentAdder> _componentAdders;
00061     ParsingType _parsingType;
00062 };
00063 #endif /* !PARSER_HPP_ */

```

5.167 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/VelosityComponent.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_ */

```

5.168 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_ */

```

5.169 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
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_ */

```

5.170 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_ */

```

5.171 ParsedEntityPrefab.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ParsedEntityPrefab
00006 */
00007
00008 #ifndef PARSEDENTITYPREFAB_HPP_
00009 #define PARSEDENTITYPREFAB_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>&
00027             adders);
00028         ~ParsedEntityPrefab();
00029
00030         void addComponent(std::shared_ptr<ecs::IComponent> component, std::type_index typeIndex);
00031         const std::vector<std::shared_ptr<ecs::IComponent>>& getComponents() const;
00032         std::string getName() const;
00033
00034         ecs::Entity instantiate(
00035             const std::shared_ptr<ecs::Registry>& registry,
00036             const std::shared_ptr<ecs::IEntityFactory>& factory,
00037             const ecs::EntityCreationContext& context = ecs::EntityCreationContext::forLocalClient()
00038         ) override;
00039
00040         ecs::Entity instantiate(const std::shared_ptr<ecs::Registry>& registry) override;
00041
00042     private:
00043         std::string _name;
00044         std::vector<std::pair<std::shared_ptr<ecs::IComponent>, std::type_index>> _components;
00045         const std::map<std::type_index, ComponentAdder>& _componentAdders;
00046
00047         void addParsedComponents(
00048             const std::shared_ptr<ecs::Registry>& registry,
00049             ecs::Entity entity
00050         );
00051 };
00052 #endif /* !PARSEDENTITYPREFAB_HPP_ */

```

5.172 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_ */

```

5.173 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_ */

```

5.174 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_ */

```

5.175 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_ */

```

5.176 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,
std::shared_ptr<Registry> registry, float deltaTime) = 0;
00021     };
00022
00023 } // namespace ecs
00024
00025 #endif /* !ISystem_HPP_ */

```

5.177 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_ */

```

5.178 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
00029 }
00030
00031 #endif /* !DEATHSYSTEM_HPP_ */

```

5.179 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_ */

```

5.180 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_ */

```

5.181 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
00037  private:
00038      ActionFactory();
00039      ~ActionFactory() = default;
00040      ActionFactory(const ActionFactory&) = delete;
00041      ActionFactory& operator=(const ActionFactory&) = delete;
00042
00043      void initializeConditions();
00044
00045      std::unordered_map<std::string, ActionFunction> _actions;
00046  };
00047  #endif /* !ACTIONFACTORY_HPP_ */

```

5.182 InteractionSystem.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** InteractionSystem

```

```

00006 */
00007
00008 #ifndef INTERACTIONSYSTEM_HPP_
00009 #define INTERACTIONSYSTEM_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_ */

```

5.183 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&
00030 tagName) const;
00031         std::vector<std::string> getTags(std::shared_ptr<ecs::Registry> registry, ecs::Entity entity)
00032 const;
00033
00034     private:
00035         TagRegistry();
00036         ~TagRegistry() = default;
00037         TagRegistry(const TagRegistry&) = delete;
00038         TagRegistry& operator=(const TagRegistry&) = delete;
00039
00040         void initializeTags();
00041
00042         std::unordered_map<std::string,
00043             std::function<bool(std::shared_ptr<ecs::Registry>, ecs::Entity)>> _tagCheckers;
00044 };
00045
00046 #endif /* !TAGREGISTRY_HPP_ */

```

5.184 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_ */

```

5.185 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_ */

```

5.186 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_ */

```

5.187 IntentToVelocitySystem.hpp

```

00001 /*
00002  ** EPITECH PROJECT, 2025
00003  ** ryanR-type
00004  ** File description:
00005  ** IntentToVelocitySystem
00006  */
00007
00008 #ifndef INTENTTOVELOCITYSYSTEM_HPP_
00009 #define INTENTTOVELOCITYSYSTEM_HPP_
00010
00011 #include "../base/ASystem.hpp"
00012
00013 namespace ecs {
00014
00015     class IntentToVelocitySystem : public ASystem {
00016     public:
00017         IntentToVelocitySystem();
00018         ~IntentToVelocitySystem() = 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 /* !INTENTTOVELOCITYSYSTEM_HPP_ */

```

5.188 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/VelocityEngine.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_ */

```

5.189 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_ */

```

5.190 ScriptingSystem.hpp

```

00001  /*
00002  ** EPITECH PROJECT, 2026
00003  ** ryanR-type
00004  ** File description:
00005  ** ScriptingSystem
00006  */
00007
00008  #ifndef SCRIPTINGSYSTEM_HPP_
00009  #define SCRIPTINGSYSTEM_HPP_
00010
00011  #include "../base/ASystem.hpp"
00012  #include "../components/permanent/ScriptingComponent.hpp"
00013  #include "../components/permanent/TransformComponent.hpp"
00014  #include "../components/permanent/SpeedComponent.hpp"
00015  #include "../components/permanent/EntityPartsComponent.hpp"
00016  #include "../components/permanent/CompositeEntityComponent.hpp"
00017  #include "../components/tags/LocalPlayerTag.hpp"
00018
00019  #include <sol/sol.hpp>
00020
00021  namespace ecs {
00022
00023  class ScriptingSystem : public ASystem {
00024  public:
00025      ScriptingSystem();
00026      ~ScriptingSystem() = default;
00027
00028      void update(
00029          std::shared_ptr<ResourceManager> resourceManager,
00030          std::shared_ptr<Registry> reg,
00031          float deltaTime
00032      ) override;
00033
00034  protected:
00035  private:
00036      void bindAPI();
00037
00038      sol::state lua;
00039      std::shared_ptr<Registry> registry;
00040      std::shared_ptr<ResourceManager> resourceManager;
00041  };
00042
00043  } // namespace ecs
00044
00045  #endif /* !SCRIPTINGSYSTEM_HPP_ */

```

5.191 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/VelosityComponent.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_ */

```

5.192 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_ */

```

5.193 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_ */

```

5.194 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_ */

```


5.195 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_ */

```

5.196 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     GAME_ZONE = 0x17
00036 };
00037
00038 #endif /* !TRASLATION_TO_ECS_HPP_ */

```

5.197 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_ */

```

5.198 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_ */

```

5.199 OrientedRect.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** OrientedRect
00006 */
00007
00008 #ifndef ORIENTEDRECT_HPP_
00009 #define ORIENTEDRECT_HPP_
00010
00011 #include "Vector2f.hpp"
00012 #include <vector>
00013 #include <cmath>
00014
00015 namespace math {
00016
00017 class OrientedRect {
00018     public:
00019         OrientedRect();
00020         OrientedRect(Vector2f center, Vector2f size, float rotation);
00021         OrientedRect(OrientedRect const &other);
00022         ~OrientedRect() = default;
00023
00024         Vector2f getCenter() const;
00025         void setCenter(Vector2f center);
00026         Vector2f getSize() const;
00027         void setSize(Vector2f size);
00028         float getRotation() const;
00029         void setRotation(float rotation);
00030
00031         std::vector<Vector2f> getCorners() const;
00032         Vector2f getAxisX() const;
00033         Vector2f getAxisY() const;
00034
00035         bool intersects(OrientedRect const &other) const;
00036
00037         OrientedRect &operator=(OrientedRect const &other);    private:
00038         Vector2f _center;
00039         Vector2f _size;
00040         float _rotation;
00041
00042         float projectPoint(Vector2f point, Vector2f axis) const;
00043         bool overlapOnAxis(OrientedRect const &other, Vector2f axis) const;
00044 };
00045 } // namespace math
00046
00047 #endif /* !ORIENTEDRECT_HPP_ */

```

5.200 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_ */

```

5.201 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     constexpr int MAX_CLIENT = 4;
00026
00027     /* Packets */
00028     constexpr char END_OFSTRING_ST = '\r';
00029     constexpr char END_OFSTRING_ND = '\n';
00030     constexpr char END_OFSTRING_TRD = '\0';
00031 }
00032
00033 #endif /* !SERVER_CONSTANTS */

```

5.202 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 <set>
00016 #include "../common/translationToECS.hpp"
00017
00018 namespace rserv {
00019
00020 struct EntitySnapshot {
00021     uint32_t entityId;
00022     uint32_t componentMask;
00023     std::map<uint8_t, std::vector<uint64_t> components;
00024
00025     EntitySnapshot() : entityId(0), componentMask(0) {}
00026 };
00027
00028 class ComponentDeltaTracker {

```

```

00029     public:
00030         std::vector<uint64_t> createEntityDelta(uint8_t clientId, uint32_t entityId, const
EntitySnapshot& currentSnapshot);
00031         std::vector<uint64_t> createMultiEntityDelta(uint8_t clientId, const
std::vector<EntitySnapshot>& entities);
00032         EntitySnapshot applyDelta(uint8_t clientId, const std::vector<uint64_t>& deltaPayload);
00033         void clearClientCache(uint8_t clientId);
00034         void clearEntityCache(uint8_t clientId, uint32_t entityId);
00035         void clearAllCaches();
00036         void clearDeadEntities(const std::set<uint32_t>& aliveEntityIds);
00037     private:
00038         std::unordered_map<uint8_t, std::unordered_map<uint32_t, EntitySnapshot> _clientEntityCache;
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 } // namespace rserv
00043 #endif // COMPONENT_DELTA_TRACKER_HPP

```

5.203 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 #ifndef COMPONENT_SERIALIZER_HPP
00008 #define COMPONENT_SERIALIZER_HPP
00009 #include "ComponentDeltaTracker.hpp"
00010 #include <cstring>
00011 #include <string>
00012 namespace rserv {
00013     class ComponentSerializer {
00014     public:
00015         static std::vector<uint64_t> serializePosition(uint32_t x, uint32_t y);
00016         static void deserializePosition(const std::vector<uint64_t>& data, uint32_t& x, uint32_t& y);
00017         static std::vector<uint64_t> serializeVelocity(int32_t vx, int32_t vy);
00018         static void deserializeVelocity(const std::vector<uint64_t>& data, int32_t& vx, int32_t& vy);
00019         static std::vector<uint64_t> serializeHealth(uint32_t current, uint32_t max);
00020         static void deserializeHealth(const std::vector<uint64_t>& data, uint32_t& current, uint32_t&
max);
00021         static std::vector<uint64_t> serializeCollider(uint32_t x, uint32_t y, uint32_t width,
uint32_t height, uint32_t rotation);
00022         static std::vector<uint64_t> serializeShootingStats(uint32_t fireRate, uint32_t damage,
uint32_t lastShot);
00023         static std::vector<uint64_t> serializeScore(uint64_t score);
00024         static std::vector<uint64_t> serializeDamage(uint32_t damage);
00025         static std::vector<uint64_t> serializeLifetime(uint64_t lifetime);
00026         static std::vector<uint64_t> serializeSpeed(uint64_t speed);
00027         static std::vector<uint64_t> serializeAIMovementPattern(uint32_t patternId);
00028         static std::vector<uint64_t> serializeGameZone(uint32_t x, uint32_t y, uint32_t width,
uint32_t height);
00029         static EntitySnapshot createSnapshotFromComponents(uint32_t entityId, const
std::vector<uint64_t>& componentData);
00030         static std::vector<uint64_t> snapshotToComponentData(const EntitySnapshot& snapshot);
00031         static bool isTagComponent(uint8_t component);
00032         static bool isOneParamComponent(uint8_t component);
00033     };
00034 } // namespace rserv
00035 #endif // COMPONENT_SERIALIZER_HPP

```

5.204 gsmStates.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** gsmStates

```

```

00006 */
00007
00008 #ifndef GSMMACHINEENUM_HPP_
00009 #define GSMMACHINEENUM_HPP_
00010
00011 namespace gsm {
00012
00013 enum GameStateType {
00014     BOOT,
00015     LOBBY,
00016     LOADING,
00017     IN_GAME,
00018     GAME_END,
00019     SHUTDOWN
00020 };
00021
00022 } // namespace gsm
00023
00024 #endif /* !GSMMACHINEENUM_HPP_ */

```

5.205 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     void update(float deltaTime) override;
00023 };
00024
00025 } // namespace gsm
00026
00027 #endif // SERVER_GAMEENDSTATE_HPP_

```

5.206 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_

```

5.207 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 #include <chrono>
00018 #include <map>
00019 #include <set>
00020
00021 namespace ecs {
00022
00023 class ServerInputProvider : public IInputProvider {
00024     public:
00025         ServerInputProvider();
00026         ~ServerInputProvider() override = default;
00027
00028         float getAxisValue(event_t axis, size_t clientID = 0) override;
00029
00030         bool isActionPressed(InputAction action, size_t clientID = 0) override;
00031         float getActionAxis(InputAction action, size_t clientID = 0) override;
00032         InputMapping getInputMapping(size_t clientID = 0) const override;
00033
00034         void setAxisValue(ecs::InputAction action, float value, size_t clientID = 0);
00035
00036         void addClientInputMapping(size_t clientID, size_t identity, const InputMapping& mapping);
00037         void registerClient(size_t clientID);
00038         void updateInputFromEvent(size_t clientID, constants::EventType eventType, float value);
00039         std::vector<size_t> getConnectedClients() const;
00040
00041     private:
00042         std::vector<std::tuple<size_t, size_t, InputMapping>> _inputMapping;
00043         std::map<size_t, std::map<ecs::InputAction, float>> _clientAxisValues;
00044         std::map<size_t, std::map<ecs::InputAction, std::chrono::steady_clock::time_point>>
00045         _clientInputTimestamps;
00046         std::set<size_t> _registeredClients;
00047
00048         static constexpr std::chrono::milliseconds INPUT_TIMEOUT = std::chrono::milliseconds(200);
00049
00050         using InputHandler = void (ServerInputProvider::*)(size_t, float);
00051         std::vector<InputHandler> _inputHandlers;
00052
00053         void handleUp(size_t clientID, float value);
00054         void handleDown(size_t clientID, float value);
00055         void handleLeft(size_t clientID, float value);
00056         void handleRight(size_t clientID, float value);
00057         void handleStop(size_t clientID, float value);
00058         void handleShoot(size_t clientID, float value);
00059 };
00060
00061 } // namespace ecs
00062
00063 #endif /* !SERVERINPUTPROVIDER_HPP_ */

```

5.208 Lobby.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2026
00003 ** ryanR-type
00004 ** File description:
00005 ** Lobby
00006 */
00007
00008 #ifndef LOBBY_HPP_
00009 #define LOBBY_HPP_
00010 #ifdef _WIN32
00011     #ifndef _WIN32_WINNT
00012         #define _WIN32_WINNT 0x0A00
00013     #endif

```

```

00014
00015     #ifndef WIN32_LEAN_AND_MEAN
00016         #define WIN32_LEAN_AND_MEAN
00017     #endif
00018 #endif
00019
00020 #include <queue>
00021 #include <map>
00022 #include <memory>
00023 #include <mutex>
00024
00025 #include "LobbyStruct.hpp"
00026 #include "ServerConfig.hpp"
00027 #include "deltaTracker/ComponentDeltaTracker.hpp"
00028 #include "deltaTracker/ComponentSerializer.hpp"
00029 #include "../common/interfaces/INetwork.hpp"
00030 #include "../common/interfaces/IBuffer.hpp"
00031 #include "../common/DLLoader/DLLoader.hpp"
00032 #include "../common/DLLoader/LoaderType.hpp"
00033 #include "../common/constants.hpp"
00034 #include "../common/InputMapping/InputAction.hpp"
00035 #include "../common/resourceManager/ResourceManager.hpp"
00036 #include "../common/ECS/entity/registry/Registry.hpp"
00037 #include "gsm/machine/GameStateMachine.hpp"
00038 #include "Signal.hpp"
00039
00040 namespace rserv {
00041
00042     class Lobby {
00043     public:
00044         Lobby(std::shared_ptr<net::INetwork> network,
00045             std::vector<std::tuple<uint8_t, std::shared_ptr<net::INetworkEndpoint>, std::string>
00046             lobbyPlayerInfo,
00047                 std::string lobbyCode, bool debug);
00048         ~Lobby();
00049         void stop();
00050
00051         void startNetworkThread();
00052         void startGameThread();
00053         void networkLoop();
00054         void gameLoop();
00055
00056         void setIsDebug(bool debug);
00057         bool getIsDebug() const;
00058
00059         std::vector<uint8_t> getConnectedClients() const;
00060         std::vector<std::shared_ptr<net::INetworkEndpoint> > getConnectedClientEndpoints() const;
00061         size_t getClientCount() const;
00062         std::string getLobbyCode() const;
00063         std::shared_ptr<net::INetwork> getNetwork() const;
00064
00065         std::shared_ptr<std::queue<std::tuple<uint8_t, constants::EventType, double>>
00066         getEventQueue();
00067         bool hasEvents() const;
00068
00069         void enqueuePacket(std::pair<std::shared_ptr<net::INetworkEndpoint>, std::vector<uint8_t>
00070         packet);
00071
00072         /* Received Packet Handling */
00073         void processIncomingPackets();
00074         bool processDisconnections(uint8_t idClient);
00075         bool processEvents(uint8_t idClient);
00076         bool processEndOfGame(uint8_t idClient);
00077         bool processWhoAmI(uint8_t idClient);
00078
00079         /* Sent Packet Handling */
00080         bool gameStatePacket();
00081         bool endGamePacket(bool isWin);
00082         std::vector<uint64_t> spawnPacket(size_t entity, const std::string prefabName);
00083         std::vector<uint64_t> deathPacket(size_t entity);
00084         bool serverStatusPacket();
00085
00086         bool isGameStarted() const;
00087         bool allClientsReady() const;
00088
00089         uint32_t getSequenceNumber() const;
00090
00091         void setPacketManager(std::shared_ptr<pm::IPacketManager> packet);
00092         std::shared_ptr<pm::IPacketManager> getPacketManager() const;
00093         void incrementSequenceNumber();
00094         void setResourceManager(std::shared_ptr<ResourceManager> resourceManager);
00095         void clearEntityDeltaCache(uint8_t clientId, uint32_t entityId);
00096         void createPlayerEntities();
00097         void processLobbyEvents();
00098
00099     private:
00100         bool _isDebug;

```



```

00098
00099     /* Network handling variable */
00100     std::shared_ptr<net::INetwork> _network;
00101     std::vector<std::tuple<uint8_t, std::shared_ptr<net::INetworkEndpoint>, std::string>>
    _clients;
00102     std::string _lobbyCode;
00103     std::map<uint8_t, bool> _clientsReady;
00104     // std::map<uint8_t, bool> _clientsAlive;
00105     std::shared_ptr<pm::IPacketManager> _packet;
00106     uint32_t _sequenceNumber;
00107     std::shared_ptr<std::queue<std::tuple<uint8_t, constants::EventType, double>> _eventQueue;
00108
00109     /* Packet queue for incoming packets */
00110     std::queue<std::pair<std::shared_ptr<net::INetworkEndpoint>, std::vector<uint8_t>>
    _incomingPackets;
00111     std::mutex _packetMutex;
00112
00113     /* ECS/Game handling variable */
00114     bool _gameStarted;
00115     std::shared_ptr<ResourceManager> _resourceManager;
00116     std::shared_ptr<gsm::GameStateMachine> _gsm;
00117     std::chrono::steady_clock::time_point _lastGameStateTime;
00118     float _statusUpdateTimer;
00119
00120     /* Threading */
00121     std::atomic_bool _running;
00122     std::thread _networkThread;
00123     std::thread _gameThread;
00124     mutable std::mutex _eventMutex;
00125
00126
00127     ComponentDeltaTracker _deltaTracker;
00128     /* Functions to build game state packets */
00129     std::vector<std::function<std::vector<uint64_t>(std::shared_ptr<ecs::Registry>,
    ecs::Entity)>> _convertFunctions;
00130     std::vector<uint64_t> convertTagComponent (std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00131     std::vector<uint64_t> convertTransformComponent (std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00132     std::vector<uint64_t> convertSpeedComponent (std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00133     std::vector<uint64_t> convertHealthComponent (std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00134     std::vector<uint64_t> convertColliderComponent (std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00135     std::vector<uint64_t> convertShootStatComponent (std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00136     std::vector<uint64_t> convertScoreComponent (std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00137     std::vector<uint64_t> convertDamageComponent (std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00138     std::vector<uint64_t> convertLifetimeComponent (std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00139     std::vector<uint64_t> convertVelocityComponent (std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00140     // std::vector<uint64_t> convertAIMoverTagComponent (std::shared_ptr<ecs::Registry>
    registry, ecs::Entity i);
00141     // std::vector<uint64_t> convertAIShooterTagComponent (std::shared_ptr<ecs::Registry>
    registry, ecs::Entity i);
00142     std::vector<uint64_t> convertControllableTagComponent (std::shared_ptr<ecs::Registry>
    registry, ecs::Entity i);
00143     std::vector<uint64_t> convertEnemyProjectileTagComponent (std::shared_ptr<ecs::Registry>
    registry, ecs::Entity i);
00144     std::vector<uint64_t> convertGameZoneColliderTagComponent (std::shared_ptr<ecs::Registry>
    registry, ecs::Entity i);
00145     std::vector<uint64_t> convertMobTagComponent (std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00146     std::vector<uint64_t> convertObstacleTagComponent (std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00147     std::vector<uint64_t> convertPlayerProjectileTagComponent (std::shared_ptr<ecs::Registry>
    registry, ecs::Entity i);
00148     std::vector<uint64_t> convertShooterTagComponent (std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00149     std::vector<uint64_t>
    convertProjectilePassThroughTagComponent (std::shared_ptr<ecs::Registry> registry, ecs::Entity i);
00150     std::vector<uint64_t> convertProjectilePrefabComponent (std::shared_ptr<ecs::Registry>
    registry, ecs::Entity i);
00151     std::vector<uint64_t> convertGameZoneComponent (std::shared_ptr<ecs::Registry> registry,
    ecs::Entity i);
00152
00153     };
00154 } // namespace rserv = r-type server
00155
00156 #endif /* !LOBBY_HPP_ */

```

5.209 LobbyStruct.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2026
00003 ** ryanR-type
00004 ** File description:
00005 ** Lobby
00006 */
00007
00008 #include <string>
00009 #include <vector>
00010 #include <thread>
00011 #include "../common/interfaces/INetwork.hpp"
00012 #include <asio/ip/udp.hpp>
00013
00014 #ifndef LOBBYSTRUCT_HPP_
00015 #define LOBBYSTRUCT_HPP_
00016
00017 namespace rserv {
00018
00019 class Lobby;
00020
00021 struct LobbyStruct {
00022
00023     std::string _lobbyCode;
00024     std::vector<std::tuple<uint8_t, std::shared_ptr<net::INetworkEndpoint>, std::string>> _clients;
00025     std::shared_ptr<Lobby> _lobby;
00026 };
00027
00028 }
00029
00030 #endif /* !LOBBYSTRUCT_HPP_ */

```

5.210 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
00025 #include "LobbyStruct.hpp"
00026 #include "Lobby.hpp"
00027 #include "ServerConfig.hpp"
00028 #include "deltaTracker/ComponentDeltaTracker.hpp"
00029 #include "deltaTracker/ComponentSerializer.hpp"
00030 #include "../common/interfaces/INetwork.hpp"
00031 #include "../common/interfaces/IBuffer.hpp"
00032 #include "../libs/Network/common.hpp"
00033 #include "../common/DLLoader/DLLoader.hpp"
00034 #include "../common/DLLoader/LoaderType.hpp"
00035 #include "../common/constants.hpp"
00036 #include "../common/InputMapping/InputAction.hpp"
00037 #include "../common/resourceManager/ResourceManager.hpp"
00038 #include "../common/ECS/entity/registry/Registry.hpp"
00039 #include "../common/resourceManager/ResourceManager.hpp"
00040 #include <thread>
00041 #include "../common/Parser/Parser.hpp"
00042 #include "../common/systems/systemManager/ISystemManager.hpp"
00043 #include "gsm/machine/GameStateMachine.hpp"
00044 #include "initResourcesManager/ServerInputProvider.hpp"
00045 #include "Signal.hpp"
00046
00047 namespace rserv {

```

```

00048     class Server {
00049     public:
00050         Server();
00051         ~Server();
00052
00053         void init();
00054         void start();
00055         void stop();
00056
00057         void setConfig(std::shared_ptr<ServerConfig> config);
00058         std::shared_ptr<ServerConfig> getConfig() const;
00059         uint16_t getPort() const;
00060         void setPort(uint16_t port);
00061
00062         int getState() const;
00063         void setState(int state);
00064
00065         void initResourceManager(std::shared_ptr<Lobby> lobby);
00066         operator int() const noexcept;
00067
00068         std::shared_ptr<net::INetwork> getNetwork() const;
00069         void setNetwork(std::shared_ptr<net::INetwork> network);
00070
00071         void onClientConnected(uint8_t idClient);
00072         void onClientDisconnected(uint8_t idClient);
00073         void onPacketReceived(uint8_t idClient, const pm::IPacketManager &packet);
00074
00075         std::vector<uint8_t> getConnectedClients() const;
00076         std::vector<std::shared_ptr<net::INetworkEndpoint>> getConnectedClientEndpoints() const;
00077         size_t getClientCount() const;
00078
00079         /* Received Packet Handling */
00080         void processIncomingPackets();
00081         bool processConnections(std::pair<std::shared_ptr<net::INetworkEndpoint>,
00082 std::vector<uint8_t> client);
00083         bool processDisconnections(uint8_t idClient);
00084         bool requestCode(const net::INetworkEndpoint &endpoint);
00085         bool processConnectToLobby(std::pair<std::shared_ptr<net::INetworkEndpoint>,
std::vector<uint8_t> payload);
00086         bool processMasterStart(std::pair<std::shared_ptr<net::INetworkEndpoint>,
std::vector<uint8_t> payload);
00087
00088         /* Sent Packet Handling */
00089         bool connectionPacket(const net::INetworkEndpoint & endpoint);
00090         bool canStartPacket(std::vector<std::shared_ptr<net::INetworkEndpoint>> endpoints);
00091         bool serverStatusPacket();
00092         bool sendCodeLobbyPacket(const net::INetworkEndpoint &endpoint);
00093         bool lobbyConnectValuePacket(const net::INetworkEndpoint &endpoint, bool canConnect);
00094
00095         uint32_t getSequenceNumber() const;
00096         std::shared_ptr<pm::IPacketManager> getPacketManager() const;
00097         std::shared_ptr<pm::IPacketManager> createNewPacketManager();
00098         uint32_t getNextEntityId();
00099         void incrementSequenceNumber();
00100
00101     private:
00102         void loadNetworkLibrary();
00103         void loadBufferLibrary();
00104         void loadPacketLibrary();
00105         DLLoader<createNetworkLib_t> _networkloader;
00106         DLLoader<createBuffer_t> _bufferloader;
00107         DLLoader<createPacket_t> _packetloader;
00108
00109         std::shared_ptr<ServerConfig> _config;
00110         std::shared_ptr<net::INetwork> _network;
00111         std::shared_ptr<IBuffer> _buffer;
00112         std::shared_ptr<pm::IPacketManager> _packet;
00113
00114         /* Network handling variables */
00115         uint8_t _nextClientId;
00116         uint32_t _sequenceNumber;
00117         uint32_t _nextEntityId;
00118
00119         /* Lobby handling variables */
00120         std::vector<std::tuple<uint8_t, std::shared_ptr<net::INetworkEndpoint>, std::string>
_clients;
00121         std::map<uint8_t, bool> _clientsReady;
00122         std::vector<std::shared_ptr<LobbyStruct>> _lobbyThreads;
00123         std::vector<std::shared_ptr<Lobby>> _lobbies;
00124         std::map<uint8_t, std::shared_ptr<Lobby>> _clientToLobby;
00125
00126     };
00127 } // namespace rserv = r-type server
00128
00129 #endif /* !SERVER_HPP_ */

```

5.211 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 <stdint>
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                 std::string getIp() const;
00028                 void setIp(std::string ip);
00029
00030                 void setIsDebug(bool isDebug);
00031                 bool getIsDebug() const;
00032             private:
00033                 int _state;
00034                 uint16_t _port;
00035                 std::string _ip;
00036                 bool _isDebug;
00037         };
00038     } // namespace rserv = r-type server
00039
00040 #endif /* !SERVER_CONFIG_HPP_ */

```

5.212 EndOfMapDetectionSystem.hpp

```

00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** EndOfMapDetectionSystem
00006 */
00007
00008 #ifndef ENDOFMAPDETECTIONSYSTEM_HPP_
00009     #define ENDOFMAPDETECTIONSYSTEM_HPP_
00010
00011     #include "../common/systems/base/ASystem.hpp"
00012
00013     namespace ecs {
00014
00015         class EndOfMapDetectionSystem : public ASystem{
00016             public:
00017                 EndOfMapDetectionSystem();
00018                 ~EndOfMapDetectionSystem();
00019
00020                 void update(std::shared_ptr<ResourceManager> resourceManager, std::shared_ptr<Registry>
00021 registry, float deltaTime) override;
00022
00023             protected:
00024             private:
00025         };
00026     };
00027
00028 } // namespace ecs
00029
00030 #endif /* !ENDOFMAPDETECTIONSYSTEM_HPP_ */

```

5.213 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_ */

```

5.214 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>
00026 registry, float deltaTime) override;
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/ryanR-type/client/ClientNetwork.hpp, 196
197 /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/L
/home/albane/epitech/tech3/r-type/ryanR-type/client/Core.hpp, 218
212 /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/L
/home/albane/epitech/tech3/r-type/ryanR-type/client/DeathAnimationSpawner.hpp,
213 /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/L
/home/albane/epitech/tech3/r-type/ryanR-type/client/SettingsConfig.hpp,
228 /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/L
/home/albane/epitech/tech3/r-type/ryanR-type/client/SettingsManager.hpp,
229 /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/M
/home/albane/epitech/tech3/r-type/ryanR-type/client/Utils.hpp, 221
247 /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/Pa
/home/albane/epitech/tech3/r-type/ryanR-type/client/colors.hpp, 222
199 /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/Re
/home/albane/epitech/tech3/r-type/ryanR-type/client/components/rendering/AnimationComponent.hpp,
200 /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/Se
/home/albane/epitech/tech3/r-type/ryanR-type/client/components/rendering/HealthBarComponent.hpp,
202 /home/albane/epitech/tech3/r-type/ryanR-type/client/initResourcesManager
/home/albane/epitech/tech3/r-type/ryanR-type/client/components/rendering/HitboxRenderComponent.hpp,
202 /home/albane/epitech/tech3/r-type/ryanR-type/client/initResourcesManager
/home/albane/epitech/tech3/r-type/ryanR-type/client/components/rendering/MusicComponent.hpp,
203 /home/albane/epitech/tech3/r-type/ryanR-type/client/input/MouseInputHan
/home/albane/epitech/tech3/r-type/ryanR-type/client/components/rendering/ParallaxComponent.hpp,
204 /home/albane/epitech/tech3/r-type/ryanR-type/client/interpolation/Network
/home/albane/epitech/tech3/r-type/ryanR-type/client/components/rendering/RectangleRenderComponent.hpp,
205 /home/albane/epitech/tech3/r-type/ryanR-type/client/packet/DefaultPacket
/home/albane/epitech/tech3/r-type/ryanR-type/client/components/rendering/SpriteComponent.hpp,
205 /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/audio/Music
/home/albane/epitech/tech3/r-type/ryanR-type/client/components/rendering/TextComponent.hpp,
206 /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/audio/Sound
/home/albane/epitech/tech3/r-type/ryanR-type/client/components/tag/BackgroundMusicTag.hpp,
206 /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/effects/Client
/home/albane/epitech/tech3/r-type/ryanR-type/client/components/temporary/MusicIntentComponent.hpp,
207 /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/input/Movem
/home/albane/epitech/tech3/r-type/ryanR-type/client/components/temporary/SoundIntentComponent.hpp,
207 /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/input/Shootl
/home/albane/epitech/tech3/r-type/ryanR-type/client/constants.hpp, 232
208 /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/map/MapGe
/home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/machine/AGameStateMachine.hpp,
213 /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/network/Net
/home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/machine/GAMEStateMachine.hpp,
214 /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/Ar
/home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/baseGameState.hpp,
215 /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/Ga
/home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/Boot/BootState.hpp,
216 /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/Ga
/home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/InGame/InGameState.hpp,
217 /home/albane/epitech/tech3/r-type/ryanR-type/client/systems/rendering/He
/home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/Infinite/InfiniteState.hpp,
234

[/home/albane/epitech/tech3/r-type/ryanR-type/client/system/scene/scene.h](#) 235
[/home/albane/epitech/tech3/r-type/ryanR-type/client/system/scene/scene.hpp](#) 235
[/home/albane/epitech/tech3/r-type/ryanR-type/client/system/scene/scene.hpp](#) 236
[/home/albane/epitech/tech3/r-type/ryanR-type/client/system/scene/scene.hpp](#) 236
[/home/albane/epitech/tech3/r-type/ryanR-type/client/system/scene/scene.hpp](#) 236
[/home/albane/epitech/tech3/r-type/ryanR-type/client/ui/core/scene/scene.h](#) 237
[/home/albane/epitech/tech3/r-type/ryanR-type/client/ui/core/scene/scene.h](#) 238
[/home/albane/epitech/tech3/r-type/ryanR-type/client/ui/core/scene/scene.h](#) 238
[/home/albane/epitech/tech3/r-type/ryanR-type/client/ui/element/scene/scene.h](#) 239
[/home/albane/epitech/tech3/r-type/ryanR-type/client/ui/element/scene/scene.h](#) 245
[/home/albane/epitech/tech3/r-type/ryanR-type/client/ui/element/scene/scene.h](#) 240
[/home/albane/epitech/tech3/r-type/ryanR-type/client/ui/element/scene/scene.h](#) 241
[/home/albane/epitech/tech3/r-type/ryanR-type/client/ui/element/scene/scene.h](#) 242
[/home/albane/epitech/tech3/r-type/ryanR-type/client/ui/element/scene/scene.h](#) 243
[/home/albane/epitech/tech3/r-type/ryanR-type/client/ui/element/scene/scene.h](#) 244
[/home/albane/epitech/tech3/r-type/ryanR-type/client/ui/manager/scene/scene.h](#) 245
[/home/albane/epitech/tech3/r-type/ryanR-type/client/ui/navigation/scene/scene.h](#) 246
[/home/albane/epitech/tech3/r-type/ryanR-type/common/Collision/scene/scene.h](#) 248
[/home/albane/epitech/tech3/r-type/ryanR-type/common/Collision/scene/scene.h](#) 249
[/home/albane/epitech/tech3/r-type/ryanR-type/common/DL/scene/scene.h](#) 268
[/home/albane/epitech/tech3/r-type/ryanR-type/common/DL/scene/scene.h](#) 269
[/home/albane/epitech/tech3/r-type/ryanR-type/common/DL/scene/scene.h](#) 269
[/home/albane/epitech/tech3/r-type/ryanR-type/common/Entity/scene/scene.h](#) 271
[/home/albane/epitech/tech3/r-type/ryanR-type/common/Entity/scene/scene.h](#) 271
[/home/albane/epitech/tech3/r-type/ryanR-type/common/Entity/scene/scene.h](#) 270
[/home/albane/epitech/tech3/r-type/ryanR-type/common/Entity/scene/scene.h](#) 270
[/home/albane/epitech/tech3/r-type/ryanR-type/common/Entity/scene/scene.h](#) 272
[/home/albane/epitech/tech3/r-type/ryanR-type/common/Entity/scene/scene.h](#) 272
[/home/albane/epitech/tech3/r-type/ryanR-type/common/Entity/scene/scene.h](#) 273

[/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/albane/epitech/tech3/r-type/ryanR-type/server/ServerConfig.hpp](#),
 299 [320](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/albane/epitech/tech3/r-type/ryanR-type/server/Utils.hpp](#),
 299 [248](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/albane/epitech/tech3/r-type/ryanR-type/server/deltaTracker/Compo](#)
 300 [312](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/albane/epitech/tech3/r-type/ryanR-type/server/deltaTracker/Compo](#)
 300 [313](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/albane/epitech/tech3/r-type/ryanR-type/server/gsm/gsmStates.hpp](#),
 300 [313](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/albane/epitech/tech3/r-type/ryanR-type/server/gsm/machine/AGam](#)
 301 [214](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/albane/epitech/tech3/r-type/ryanR-type/server/gsm/machine/Game](#)
 301 [214](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/AGameS](#)
 302 [215](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/scenes/E](#)
 303 [216](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/scenes/C](#)
 303 [314](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/scenes/L](#)
 304 [218](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/scenes/L](#)
 304 [219](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/scenes/L](#)
 304 [219](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/scenes/S](#)
 306 [314](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/albane/epitech/tech3/r-type/ryanR-type/server/initResourcesManag](#)
 306 [315](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/albane/epitech/tech3/r-type/ryanR-type/server/initResourcesManag](#)
 307 [225](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/albane/epitech/tech3/r-type/ryanR-type/server/packet/DefaultPacke](#)
 307 [228](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/albane/epitech/tech3/r-type/ryanR-type/server/systems/gameEnd/E](#)
 308 [320](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/albane/epitech/tech3/r-type/ryanR-type/server/systems/input/Serve](#)
 308 [321](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/common/systems/albane/epitech/tech3/r-type/ryanR-type/server/systems/input/Serve](#)
 309 [321](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/common/translationToECS.hpp](#),
 309 [ActionFactory](#), [21](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/common/types/Component.hpp](#),
 309 [ecs::ASystemManager](#), [37](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/common/types/GameState.hpp](#),
 310 [gsm::AGameState](#), [27](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/common/types/OrientedRect.hpp](#),
 311 [APrefab](#), [34](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/common/types/OrientedRect.hpp](#),
 311 [instantiate](#), [35](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/common/types/Vector2f.hpp](#),
 311 [canBeFocused](#),
[ui::AFocusableElement](#), [24](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/server/Constants.hpp](#),
 312 [changeState](#),
[gsm::AGameStateMachine](#), [29](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/server/Lobby.hpp](#),
 315 [ClientNetwork](#), [48](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/server/LobbyStruct.hpp](#),
 318 [Close](#),
[DLLoader< T >](#), [61](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/server/Server.hpp](#),
 318 [ComposantParser](#), [55](#)
[/home/albane/epitech/tech3/r-type/ryanR-type/server/Server.hpp](#),
 318 [containsPoint](#),
[ui::ToggleSwitch](#), [185](#)
[Core](#), [57](#)

createEntity
 ecs::EntityFactory, 64

DeathAnimationSpawner, 59

debug::Debug, 61

DLLoader< T >, 61
 Close, 61
 Error, 61
 getHandler, 62
 Open, 62
 Symbol, 62

ecs::AComponent, 19

ecs::AComponentArray< T >, 20
 getMaxEntityId, 20
 removeComponents, 20
 removeOneComponent, 20

ecs::AnimationClip, 31

ecs::AnimationComponent, 31

ecs::AnimationCondition, 33

ecs::AnimationConditionFactory, 33

ecs::AnimationRenderingSystem, 34
 update, 34

ecs::ASystem, 35
 updateSystem, 36

ecs::ASystemManager, 37
 addSystem, 37
 removeSystem, 37
 updateAllSystems, 37

ecs::BackGroundMusicTag, 40

ecs::ClientEffectCleanupSystem, 45
 update, 46

ecs::ClientEffectTag, 46

ecs::ColliderComponent, 51

ecs::CollisionRule, 52

ecs::CollisionRules, 52

ecs::CollisionRulesData, 53

ecs::CollisionRulesParser, 53

ecs::CompositeEntityComponent, 56

ecs::ControllableTag, 57

ecs::DamageComponent, 58

ecs::DamageIntentComponent, 58

ecs::DeathIntentComponent, 59

ecs::DeathSystem, 60
 update, 60

ecs::EndOfMapDetectionSystem, 62
 update, 63

ecs::EnemyProjectileTag, 63

ecs::EntityCreationContext, 63

ecs::EntityFactory, 64
 createEntity, 64

ecs::EntityPartsComponent, 65

ecs::GameEndTag, 69

ecs::GameZoneColliderTag, 71

ecs::GameZoneComponent, 71

ecs::GameZoneRenderingSystem, 72
 update, 72

ecs::GameZoneViewSystem, 72
 update, 73

ecs::GraphicalInputProvider, 73
 getActionAxis, 74
 getAxisValue, 74
 getInputMapping, 74
 isActionPressed, 74

ecs::Group< Components >, 75

ecs::HealthBarComponent, 75

ecs::HealthBarRenderingSystem, 75
 update, 76

ecs::HealthComponent, 76

ecs::HealthSystem, 77
 update, 77

ecs::HitboxRenderComponent, 78

ecs::HitboxRenderingSystem, 78
 update, 79

ecs::IComponent, 80

ecs::IComponentArray, 81

ecs::IEntityFactory, 81

ecs::IInputProvider, 85

ecs::InputIntentComponent, 92

ecs::InputMapping, 92

ecs::InputMappingManager, 93

ecs::InputNormalizer, 93

ecs::InputToVelocitySystem, 94
 update, 94

ecs::IntentToVelocitySystem, 94
 update, 95

ecs::InteractionConfigComponent, 95

ecs::InteractionMapping, 96

ecs::InteractionSystem, 96
 update, 97

ecs::ISystem, 98

ecs::ISystemManager, 99

ecs::LifetimeComponent, 102

ecs::LifetimeSystem, 103
 update, 103

ecs::LocalPlayerTag, 113

ecs::MapGeneratorSystem, 115
 update, 116

ecs::MobTag, 117

ecs::MovementInputSystem, 118
 update, 119

ecs::MovementIntentComponent, 119

ecs::MovementSystem, 120
 update, 121

ecs::MultiShotPattern, 121

ecs::MusicComponent, 121

ecs::MusicIntentComponent, 122

ecs::MusicSystem, 123
 update, 123

ecs::NetworkHealthState, 124

ecs::NetworkInterpolationSystem, 124
 update, 125

ecs::NetworkStateComponent, 125

ecs::NetworkTransformState, 126

ecs::ObstacleTag, 126

ecs::OutOfBoundsSystem, 127
 update, 128

- ecs::OwnerComponent, 128
- ecs::ParallaxComponent, 130
- ecs::ParallaxLayer, 131
- ecs::ParallaxRenderingSystem, 131
 - update, 132
- ecs::PlayerProjectileTag, 137
- ecs::PlayerTag, 137
- ecs::ProjectilePassThroughTag, 137
- ecs::ProjectilePrefabComponent, 138
- ecs::RectangleRenderComponent, 138
- ecs::RectangleRenderingSystem, 139
 - update, 139
- ecs::Registry, 140
- ecs::RemappableKeyBinding, 140
- ecs::ScoreComponent, 143
- ecs::ScoreSystem, 144
 - update, 144
- ecs::ScoreValueComponent, 145
- ecs::ScriptingComponent, 145
- ecs::ScriptingSystem, 147
 - update, 148
- ecs::ServerInputProvider, 151
 - getActionAxis, 152
 - getAxisValue, 152
 - getInputMapping, 152
 - isActionPressed, 153
- ecs::ServerMovementInputSystem, 153
 - update, 154
- ecs::ServerShootInputSystem, 154
 - update, 155
- ecs::ShooterTag, 160
- ecs::ShootingStatsComponent, 160
- ecs::ShootingSystem, 161
 - update, 161
- ecs::ShootInputSystem, 162
 - update, 162
- ecs::ShootIntentComponent, 163
- ecs::SoundIntentComponent, 169
- ecs::SoundSystem, 169
 - update, 170
- ecs::SpatialGrid, 170
- ecs::SpawnIntentComponent, 171
- ecs::SpawnSystem, 172
 - update, 172
- ecs::SpeedComponent, 172
- ecs::SpriteComponent, 173
- ecs::SpriteRenderingSystem, 174
 - update, 174
- ecs::SystemManager, 174
- ecs::TextComponent, 178
- ecs::TextRenderingSystem, 182
 - update, 183
- ecs::TransformComponent, 186
- ecs::Transition, 187
- ecs::TriggerIntentComponent, 187
- ecs::TriggerSystem, 188
 - update, 188
- ecs::VelocityComponent, 195
- ecs::View< Components >, 196
- ecs::View< Components >::Iterator, 99
- enter
 - gsm::AGameState, 27
 - gsm::BootState, 41
 - gsm::GameEndState, 68
 - gsm::InfiniteState, 89
 - gsm::InGameState, 91
 - gsm::LoadingState, 105
 - gsm::LobbyBrowserState, 108
 - gsm::LobbyState, 110
 - gsm::LobbyWaitingState, 112
 - gsm::MainMenuState, 115
 - gsm::PauseState, 136
 - gsm::ResultsState, 142
 - gsm::SettingsState, 159
 - gsm::ShutdownState, 164
- EntityParser, 64
- EntityPrefabManager, 65
- err::AError, 21
 - getCode, 22
 - getDetails, 22
 - getType, 22
 - what, 22
- err::ClientError, 46
 - getType, 47
- err::ClientNetworkError, 50
 - getType, 51
- err::LError, 81
- err::LibrariesLoadError, 101
 - getType, 102
- err::PacketError, 129
 - getType, 130
- err::ParserError, 134
 - getType, 135
- err::ScriptingError, 146
 - getType, 147
- err::ServerError, 150
 - getType, 151
- Error
 - DLLoader< T >, 61
- exit
 - gsm::AGameState, 27
 - gsm::BootState, 41
 - gsm::InfiniteState, 89
 - gsm::InGameState, 91
 - gsm::LoadingState, 105
 - gsm::LobbyBrowserState, 108
 - gsm::LobbyState, 110
 - gsm::LobbyWaitingState, 112
 - gsm::MainMenuState, 115
 - gsm::PauseState, 136
 - gsm::ResultsState, 142
 - gsm::SettingsState, 159
- Field, 66
- FieldValue, 67
- getActionAxis

- ecs::GraphicalInputProvider, 74
- ecs::ServerInputProvider, 152
- getAxisValue
 - ecs::GraphicalInputProvider, 74
 - ecs::ServerInputProvider, 152
- getCode
 - err::AError, 22
- getDetails
 - err::AError, 22
- getHandler
 - DLoader< T >, 62
- getInputMapping
 - ecs::GraphicalInputProvider, 74
 - ecs::ServerInputProvider, 152
- getMaxEntityId
 - ecs::AComponentArray< T >, 20
- getSystems
 - gsm::AGameState, 27, 28
- getType
 - err::AError, 22
 - err::ClientError, 47
 - err::ClientNetworkError, 51
 - err::LibrariesLoadError, 102
 - err::PacketError, 130
 - err::ParserError, 135
 - err::ScriptingError, 147
 - err::ServerError, 151
- gfx::color_t, 54
- gfx::IAudio, 79
- gfx::IEvent, 82
- gfx::IWindow, 100
- gsm::AGameState, 26
 - addSystem, 27
 - enter, 27
 - exit, 27
 - getSystems, 27, 28
 - update, 28
- gsm::AGameStateMachine, 28
 - changeState, 29
 - popState, 29
 - pushState, 29, 30
 - requestStateChange, 30
 - requestStatePop, 30
 - requestStatePush, 30
 - update, 30
- gsm::BootState, 40
 - enter, 41
 - exit, 41
 - update, 41
- gsm::GameEndState, 68
 - enter, 68
 - update, 68
- gsm::GameStateMachine, 69
 - requestStateChange, 70
 - requestStatePop, 70
 - requestStatePush, 70
- gsm::IGameState, 83
- gsm::IGameStateMachine, 84
- gsm::InfiniteState, 88
 - enter, 89
 - exit, 89
 - update, 89
- gsm::InGameState, 90
 - enter, 91
 - exit, 91
 - update, 91
- gsm::LoadingState, 104
 - enter, 105
 - exit, 105
 - update, 105
- gsm::LobbyBrowserState, 108
 - enter, 108
 - exit, 108
 - update, 109
- gsm::LobbyState, 109
 - enter, 110
 - exit, 110
 - update, 110
- gsm::LobbyWaitingState, 111
 - enter, 112
 - exit, 112
 - update, 113
- gsm::MainMenuState, 113
 - enter, 115
 - exit, 115
 - update, 115
- gsm::PauseState, 135
 - enter, 136
 - exit, 136
 - update, 136
- gsm::ResultsState, 141
 - enter, 142
 - exit, 142
 - update, 142
- gsm::ScoreFeedback, 143
- gsm::SettingsState, 157
 - enter, 159
 - exit, 159
 - update, 159
- gsm::ShutdownState, 163
 - enter, 164
- handleInput
 - ui::AFocusableElement, 24
 - ui::Slider, 168
 - ui::TextInput, 181
 - ui::ToggleSwitch, 185
- IBuffer, 80
- ILoader, 85
- instantiate
 - APrefab, 35
 - ParsedEntityPrefab, 133
- IPrefab, 98
- isActionPressed
 - ecs::GraphicalInputProvider, 74
 - ecs::ServerInputProvider, 153

- isFocused
 - ui::AFocusableElement, [24](#)
- MapParser, [116](#)
- math::Chrono, [45](#)
- math::FRect, [67](#)
- math::OrientedRect, [127](#)
- math::Vector2f, [195](#)
- MouseClickedInfo, [117](#)
- MouseInputHandler, [118](#)
- net::IEventLoop, [82](#)
- net::INetwork, [86](#)
- net::INetworkAddress, [86](#)
- net::INetworkEndpoint, [87](#)
- net::INetworkErrorCode, [87](#)
- net::INetworkFactory, [87](#)
- net::INetworkResolver, [88](#)
- net::INetworkSocket, [88](#)
- NetworkEvent, [124](#)
- onActivated
 - ui::AFocusableElement, [25](#)
 - ui::Slider, [168](#)
- onFocusGained
 - ui::AFocusableElement, [25](#)
- onFocusLost
 - ui::AFocusableElement, [25](#)
- onNavigateLeft
 - ui::Slider, [168](#)
- onNavigateRight
 - ui::Slider, [168](#)
- Open
 - DLLoader< T >, [62](#)
- ParsedEntityPrefab, [132](#)
 - instantiate, [133](#)
- Parser, [133](#)
- pm::IPacketManager, [97](#)
- popState
 - gsm::AGameStateMachine, [29](#)
- pushState
 - gsm::AGameStateMachine, [29, 30](#)
- removeComponents
 - ecs::AComponentArray< T >, [20](#)
- removeOneComponent
 - ecs::AComponentArray< T >, [20](#)
- removeSystem
 - ecs::ASystemManager, [37](#)
- render
 - ui::Background, [39](#)
 - ui::Button, [44](#)
 - ui::Slider, [168](#)
 - ui::Text, [177](#)
 - ui::TextInput, [181](#)
 - ui::ToggleSwitch, [186](#)
 - ui::UILayout, [192](#)
- requestStateChange
 - gsm::AGameStateMachine, [30](#)
 - gsm::GameStateMachine, [70](#)
- requestStatePop
 - gsm::AGameStateMachine, [30](#)
 - gsm::GameStateMachine, [70](#)
- requestStatePush
 - gsm::AGameStateMachine, [30](#)
 - gsm::GameStateMachine, [70](#)
- ResourceManager, [141](#)
- rserve::ComponentDeltaTracker, [54](#)
- rserve::ComponentSerializer, [55](#)
- rserve::EntitySnapshot, [66](#)
- rserve::Lobby, [105](#)
- rserve::LobbyStruct, [111](#)
- rserve::Server, [148](#)
- rserve::ServerConfig, [149](#)
- ScoreIntentComponent, [144](#)
- setFocused
 - ui::AFocusableElement, [25](#)
- setScale
 - ui::Text, [177](#)
 - ui::UILayout, [192](#)
- SettingsConfig, [155](#)
- SettingsManager, [156](#)
- Signal, [164](#)
- Symbol
 - DLLoader< T >, [62](#)
- TagRegistry, [175](#)
- ui::AFocusableElement, [22](#)
 - canBeFocused, [24](#)
 - handleInput, [24](#)
 - isFocused, [24](#)
 - onActivated, [25](#)
 - onFocusGained, [25](#)
 - onFocusLost, [25](#)
 - setFocused, [25](#)
- ui::Background, [38](#)
 - render, [39](#)
 - update, [39](#)
- ui::Background::Layer, [101](#)
- ui::Button, [42](#)
 - render, [44](#)
- ui::IFocusable, [83](#)
- ui::LayoutConfig, [101](#)
- ui::Slider, [165](#)
 - handleInput, [168](#)
 - onActivated, [168](#)
 - onNavigateLeft, [168](#)
 - onNavigateRight, [168](#)
 - render, [168](#)
- ui::Text, [176](#)
 - render, [177](#)
 - setScale, [177](#)
 - update, [177](#)
- ui::TextInput, [179](#)
 - handleInput, [181](#)

- render, [181](#)
- update, [182](#)
- ui::ToggleSwitch, [183](#)
 - containsPoint, [185](#)
 - handleInput, [185](#)
 - render, [186](#)
- ui::UIElement, [189](#)
- ui::UILayout, [190](#)
 - render, [192](#)
 - setScale, [192](#)
 - update, [192](#)
- ui::UIManager, [193](#)
- ui::UINavigationManager, [194](#)
- update
 - ecs::AnimationRenderingSystem, [34](#)
 - ecs::ClientEffectCleanupSystem, [46](#)
 - ecs::DeathSystem, [60](#)
 - ecs::EndOfMapDetectionSystem, [63](#)
 - ecs::GameZoneRenderingSystem, [72](#)
 - ecs::GameZoneViewSystem, [73](#)
 - ecs::HealthBarRenderingSystem, [76](#)
 - ecs::HealthSystem, [77](#)
 - ecs::HitboxRenderingSystem, [79](#)
 - ecs::InputToVelocitySystem, [94](#)
 - ecs::IntentToVelocitySystem, [95](#)
 - ecs::InteractionSystem, [97](#)
 - ecs::LifetimeSystem, [103](#)
 - ecs::MapGeneratorSystem, [116](#)
 - ecs::MovementInputSystem, [119](#)
 - ecs::MovementSystem, [121](#)
 - ecs::MusicSystem, [123](#)
 - ecs::NetworkInterpolationSystem, [125](#)
 - ecs::OutOfBoundsSystem, [128](#)
 - ecs::ParallaxRenderingSystem, [132](#)
 - ecs::RectangleRenderingSystem, [139](#)
 - ecs::ScoreSystem, [144](#)
 - ecs::ScriptingSystem, [148](#)
 - ecs::ServerMovementInputSystem, [154](#)
 - ecs::ServerShootInputSystem, [155](#)
 - ecs::ShootingSystem, [161](#)
 - ecs::ShootInputSystem, [162](#)
 - ecs::SoundSystem, [170](#)
 - ecs::SpawnSystem, [172](#)
 - ecs::SpriteRenderingSystem, [174](#)
 - ecs::TextRenderingSystem, [183](#)
 - ecs::TriggerSystem, [188](#)
 - gsm::AGameState, [28](#)
 - gsm::AGameStateMachine, [30](#)
 - gsm::BootState, [41](#)
 - gsm::GameEndState, [68](#)
 - gsm::InfiniteState, [89](#)
 - gsm::InGameState, [91](#)
 - gsm::LoadingState, [105](#)
 - gsm::LobbyBrowserState, [109](#)
 - gsm::LobbyState, [110](#)
 - gsm::LobbyWaitingState, [113](#)
 - gsm::MainMenuState, [115](#)
 - gsm::PauseState, [136](#)
 - gsm::ResultsState, [142](#)
 - gsm::SettingsState, [159](#)
 - ui::Background, [39](#)
 - ui::Text, [177](#)
 - ui::TextInput, [182](#)
 - ui::UILayout, [192](#)
- updateAllSystems
 - ecs::ASystemManager, [37](#)
- updateSystem
 - ecs::ASystem, [36](#)
- Utils, [194](#)
- what
 - err::AError, [22](#)